

Enfoques de desarrollo de Aplicaciones Móviles

Aplicaciones Nativas con Android
Layouts



Layouts

- El **layout** de una **activity** representa el diseño de la interfaz de usuario determinando la disposición de distintos componentes visuales (vistas o **views**) en la misma.
- Los **layouts** también son vistas pero pertenecen a una categoría específica de vistas (**ViewGroup**) capaz de contener a otras vistas

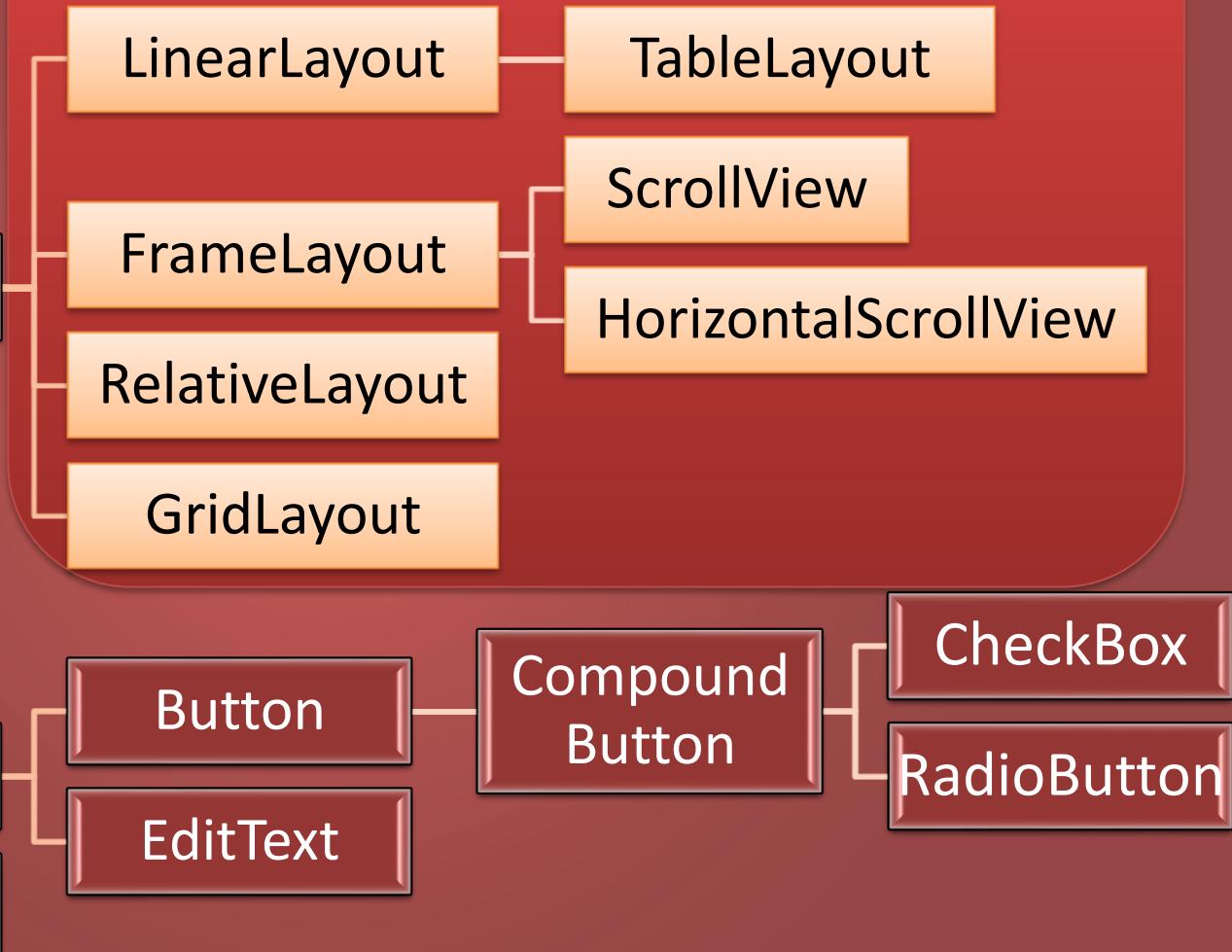
Viewgroups

- Los elementos visuales simples, como el **EditText**, **TextView**, **Button**, etc. son clases particulares de vistas (**View**) que deben ser dispuestos dentro de un contenedor
- El contenedor es un **ViewGroup** que define el modo que se muestran los elementos hijos que aloja.
- Conoceremos algunos de los **viewGroups** más populares

Algunas vistas



Layouts



Layouts

- La interfaz de usuario puede ser definida mediante:
 - Archivos XML
 - En tiempo de ejecución,
programáticamente.

Layouts desde archivos XML

- Las **Activities** que definen su interfaz por medio de **archivos XML** se asocian a estos archivos mediante **setContentView()** en el callback **onCreate()**

The screenshot shows the Android Studio interface with two tabs: 'activity_main.xml' and 'MainActivity.kt'. The code editor displays the following Kotlin code:

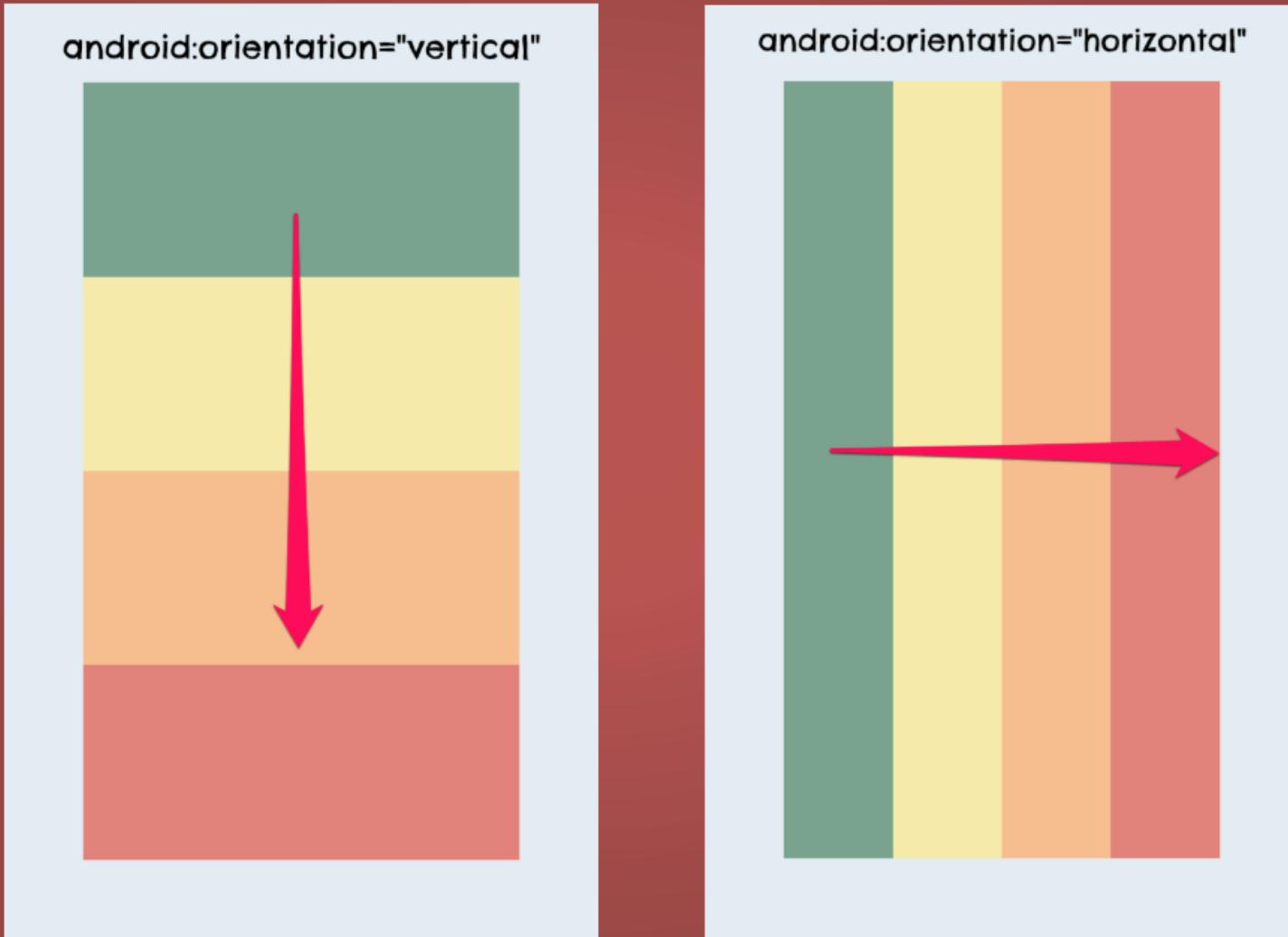
```
1 package com.example.myapplication
2
3 import ...
4
5
6 class MainActivity : AppCompatActivity() {
7     override fun onCreate(savedInstanceState: Bundle?) {
8         super.onCreate(savedInstanceState)
9         setContentView(R.layout.activity_main)
10    }
11 }
```

A red arrow points from the text 'Al crearse la activity' in the callout box to the 'onCreate' method in the code. Another red arrow points from the text 'se asocia el layout' to the 'setContentView' line.

Al crearse la activity **MainActivity** se asocia
el layout **activity_main** definido en el
archivo **activity_main.xml**

Vamos a presentar alguno de
los **layouts** más populares

LinearLayout

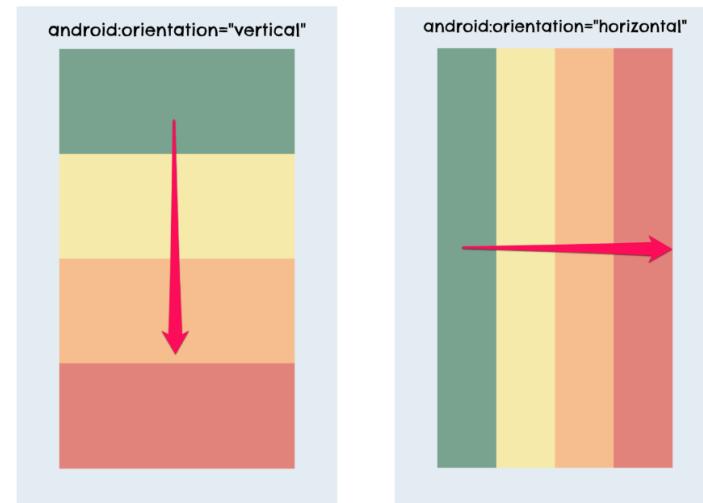


Ya lo hemos utilizado en clases anteriores

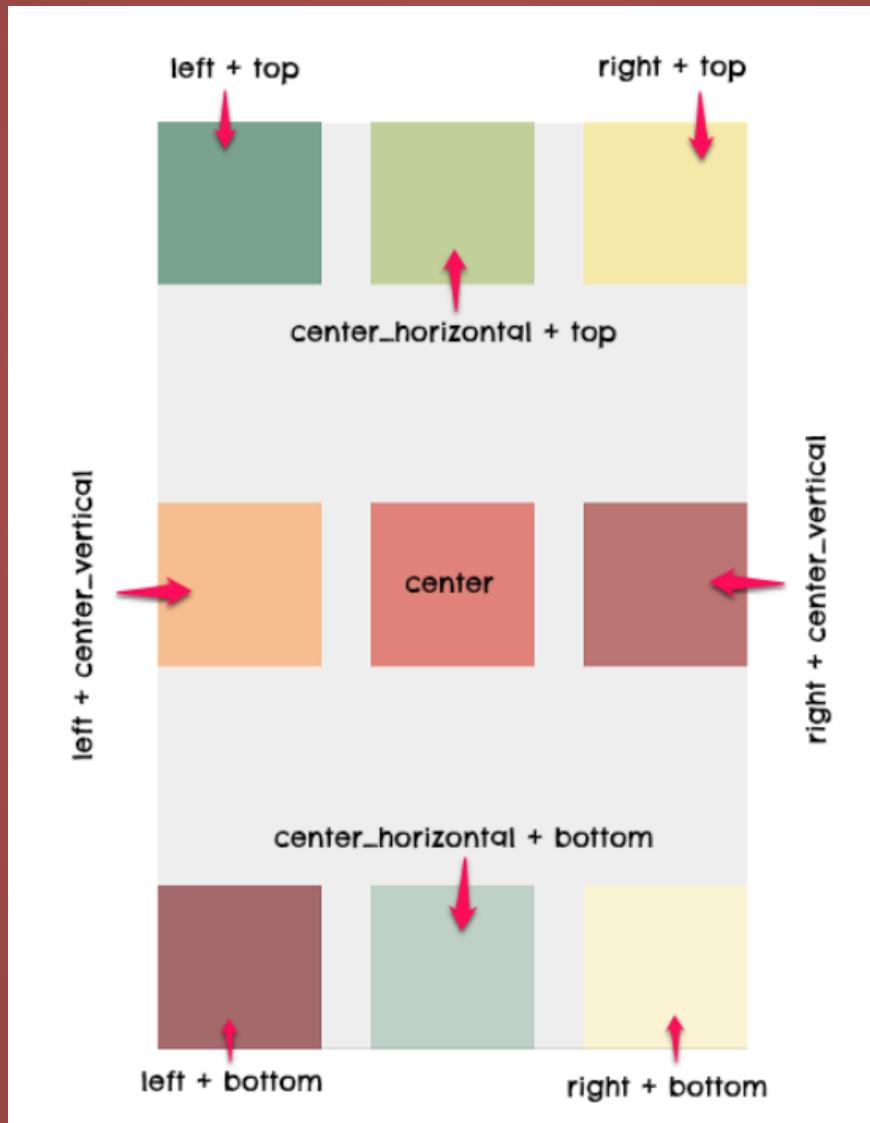
LinearLayout

- Es un **ViewGroup** que alinea a los elementos hijos en una única dirección.
- La dirección puede ser vertical u horizontal.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical" >  
    ---  
</LinearLayout>
```



FrameLayout



FrameLayout

Modificar la vista de la activity principal
de la siguiente manera

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="100dp"
        android:text="Primer texto"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="50dp"
        android:text="Segundo texto"
    />
</FrameLayout>
```

FrameLayout

Modificar la vista de la **activity** principal
de la siguiente manera



Un **FrameLayout** es un **ViewGroup** simple y eficiente.
Pensado para ser usado con un **view** hijo o con **Views** que admitan solapamiento.

Probar y responder



¿ Cuál es el efecto en las vistas de los
siguiente atributos ?

`android:layout_gravity="right"`

`android:layout_gravity="bottom"`

`android:layout_gravity="center"`

`android:layout_gravity="center|right"`

ScrollView y HorizontalScrollView

- **ScrollView** es un **FrameLayout** especializado que puede hacer **scroll vertical** sobre el elemento que contiene (sólo puede alojar un único hijo)
- **HorizontalScrollView** es un **FrameLayout** especializado que puede hacer **scroll horizontal** sobre el elemento que contiene (sólo puede alojar un único hijo)
- El ejemplo a continuación mostrará la necesidad de utilizar un **ScrollView**

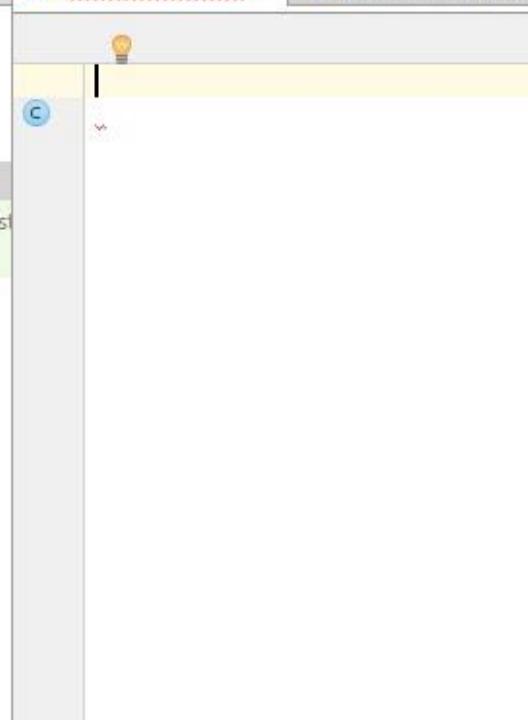


Miaplicacion14 > app > src > main > res > layout > activity_main.xml

Project

app
manifests
java
com.example.leona.miaplicacion1
MainActivity
com.example.leona.miaplicacion1 (androidTest)
com.example.leona.miaplicacion1 (test)
res
Gradle Scripts

activity_main.xml x MainActivity.java x



Borrar la definición de la vista e insertar un
<LinearLayout>

Design Text

Terminal 6: Android Monitor

0: Messages

TODO

Gradle build finished in 8s 109ms

```
activity_main.xml x MainActivity.java x
```

LinearLayout

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

```
<Button
    android:layout_width="match_parent"
    android:layout_height="150dp"
    android:text="Botón 1"
    android:layout_margin="10dp"/>
```

Establecer orientación vertical

Agregar un botón con texto "Botón 1" y establecer sus dimensiones y margen de esta manera

```
</LinearLayout>
```

```
<activity_main.xml x> MainActivity.java <x>
```

```
LinearLayout
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical">
    <Button
        android:layout_width="match_parent"
        android:layout_height="150dp"
        android:text="Botón 1"
        android:layout_margin="10dp"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="150dp"
        android:text="Botón 2"
        android:layout_margin="10dp"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="150dp"
        android:text="Botón 3"
        android:layout_margin="10dp"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="150dp"
        android:text="Botón 4"
        android:layout_margin="10dp"/>
</LinearLayout>
```

Copiar y pegar tres veces para definir así los botones 2, 3 y 4

Ejecutar en un emulador

Ejecutar en un emulador

```
C <LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical">
```

¿ Se visualizan
adecuadamente todos los
botones ?

```
    android:layout_margin="10dp"/>  
<Button  
    android:layout_width="match_parent"  
    android:layout_height="150dp"  
    android:text="Botón 4"  
    android:layout_margin="10dp"/>  
</LinearLayout>
```

Ejecutar en un emulador

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical">  
    <ScrollView>  
        <Button  
            android:layout_width="match_parent"  
            android:layout_height="150dp"  
            android:text="Botón 3"  
            android:layout_margin="10dp"/>  
        <Button  
            android:layout_width="match_parent"  
            android:layout_height="150dp"  
            android:text="Botón 4"  
            android:layout_margin="10dp"/>  
    </LinearLayout>
```

Al rescate

ClasesMiercoles - [C:\Users\leona\AndroidStudioProjects\ClasesMiercoles] - [app] - ...\\app\\src\\r

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

ClasesMiercoles app src main res layout activity_main.xml

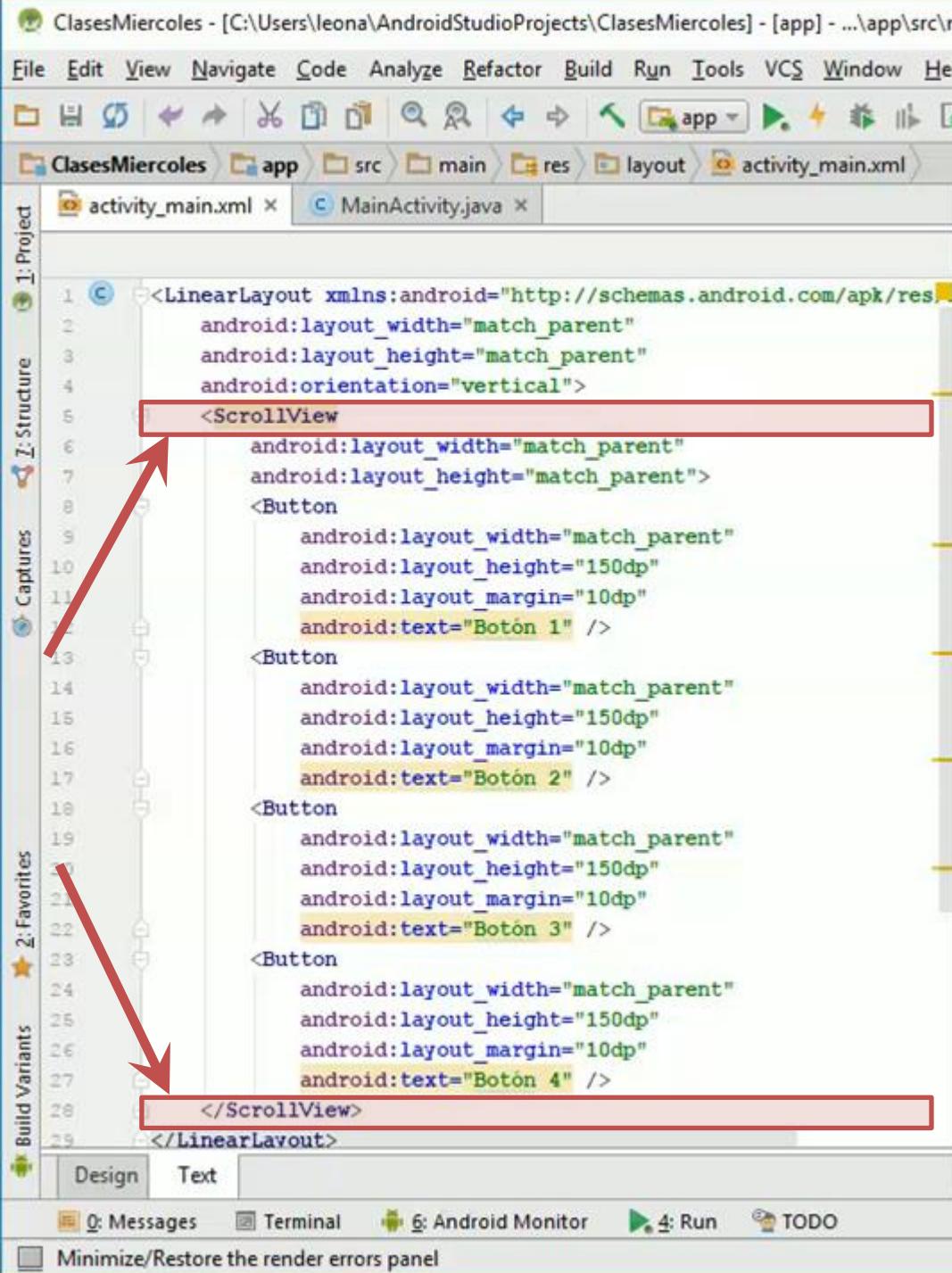
activity_main.xml MainActivity.java

```
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     android:layout_width="match_parent"
3     android:layout_height="match_parent"
4     android:orientation="vertical">
5
6     <ScrollView
7         android:layout_width="match_parent"
8         android:layout_height="match_parent">
9
10        <Button
11            android:layout_width="match_parent"
12            android:layout_height="150dp"
13            android:layout_margin="10dp"
14            android:text="Botón 1" />
15
16        <Button
17            android:layout_width="match_parent"
18            android:layout_height="150dp"
19            android:layout_margin="10dp"
20            android:text="Botón 2" />
21
22        <Button
23            android:layout_width="match_parent"
24            android:layout_height="150dp"
25            android:layout_margin="10dp"
26            android:text="Botón 3" />
27
28        <Button
29            android:layout_width="match_parent"
30            android:layout_height="150dp"
31            android:layout_margin="10dp"
32            android:text="Botón 4" />
33
34    </ScrollView>
35
36 </LinearLayout>
```

Design Text

0: Messages 1: Terminal 2: Android Monitor 3: Run 4: TODO

Minimize/Restore the render errors panel



Pregunta:
¿Por qué esta
solución no podría
ser válida?

Respuesta:
Porque ScrollView es
un FrameLayout
especializado que sólo
puede alojar un hijo
directo

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <Button
            android:layout_width="match_parent"
            android:layout_height="150dp"
            android:layout_margin="10dp"
            android:text="Botón 1" />

        <Button
            android:layout_width="match_parent"
            android:layout_height="150dp"
            android:layout_margin="10dp"
            android:text="Botón 2" />

        <Button
            android:layout_width="match_parent"
            android:layout_height="150dp"
            android:layout_margin="10dp"
            android:text="Botón 3" />

        <Button
            android:layout_width="match_parent"
            android:layout_height="150dp"
            android:layout_margin="10dp"
            android:text="Botón 4" />
    </LinearLayout>
</ScrollView>
```

Solución válida:
El ScrollView aloja
un único hijo directo
(un LinearLayout
que contiene a los
botones)

Modificar la aplicación para disponer los botones horizontalmente.

El scroll ahora debe ser horizontal.

El ancho de los botones debe ser de 150dp



```
<HorizontalScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal">
        <Button
            android:layout_width="150dp"
            android:layout_height="match_parent"
            android:layout_margin="10dp"
            android:text="Botón 1" />
        <Button
            android:layout_width="150dp"
            android:layout_height="match_parent"
            android:layout_margin="10dp"
            android:text="Botón 2" />
        <Button
            android:layout_width="150dp"
            android:layout_height="match_parent"
            android:layout_margin="10dp"
            android:text="Botón 3" />
    

```

Solución:

Se debe utilizar un
<HorizontalScrollView>

23

Vamos a acceder
programáticamente a los
elementos visuales (vistas o
views) de la activity

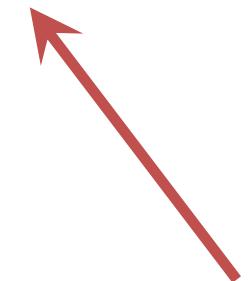
Para identificar los **views** del
layout vamos a usar el atributo **id**

Atributo id

.xml

```
<Button  
    android:id="@+id/boton1"  
    android:layout_width="150dp"  
    android:layout_height="match_parent"  
    android:layout_margin="10dp"  
    android:text="Botón 1" />
```

El **id** es un atributo especial que permite identificar al elemento desde el código kotlin



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

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="horizontal">  
    <Button  
        android:id="@+id/boton1"  
        android:layout_width="150dp"  
        android:layout_height="match_parent"  
        android:layout_margin="10dp"  
        android:text="Botón 1" />  
    <Button  
        android:id="@+id/boton2"  
        android:layout_width="150dp"  
        android:layout_height="match_parent"  
        android:layout_margin="10dp"  
        android:text="Botón 2" />  
    <Button  
        android:id="@+id/boton3"  
        android:layout_width="150dp"
```

Agregar el atributo
android:id:id a
cada uno de los
botones

¿Qué significa "@+id/boton1" ?

- Al tipear **@+id/** el entorno convierte una etiqueta en un recurso con un nombre determinado.
- Así al tipear **"@+id/boton1"** se crea un recurso llamado **boton1** que luego puede referenciarse desde el código kotlin por medio de la clase estática **R.id**
- La clase estática **R** y sus clases miembros anidadas se generan automáticamente
- En versiones más recientes, el plugin de Gradle genera el **archivo de bytecode R.class directamente en lugar del archivo R.java.**

MainActivity.kt

```
1 package com.example.myapplication
2
3 import ...
4
5
6
7 class MainActivity : AppCompatActivity() {
8     override fun onCreate(savedInstanceState: Bundle?) {
9         super.onCreate(savedInstanceState)
10        setContentView(R.layout.activity_main)
11        val b: Button;
12        b = findViewById<Button>(R.id.boton1)
13        b.setText("Este es el botón 1");
14    }
}
```

Agregar estas tres instrucciones
al método **onCreate()**

Ejecutar sobre el emulador y comprobar el resultado

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

- Project Structure:** Shows the package structure: com > cacic2017 > escuela > clasesmiercoles.
- MainActivity.java:** The code is as follows:

```
package com.cacic2017.escuela.clasesmiercoles;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        Button b;
        b = (Button) this.findViewById(R.id.button);
        b.setText("Este es el botón uno");
    }
}
```

- Emulator Preview:** An iPhone-style emulator displays the application's user interface. The title bar says "ClaseMiercoles". The screen contains two buttons:
 - The left button has the text "ESTE ES EL BOTÓN UNO".
 - The right button has the text "BOTÓN 2".
- Bottom Bar:** Includes icons for Messages, Terminal, Android Monitor, Run, TODO, and a progress bar indicating a successful Gradle build.
- Right Sidebar:** Contains tabs for Gradle, Hierarchy, and Android Model.

Explicación del código Java

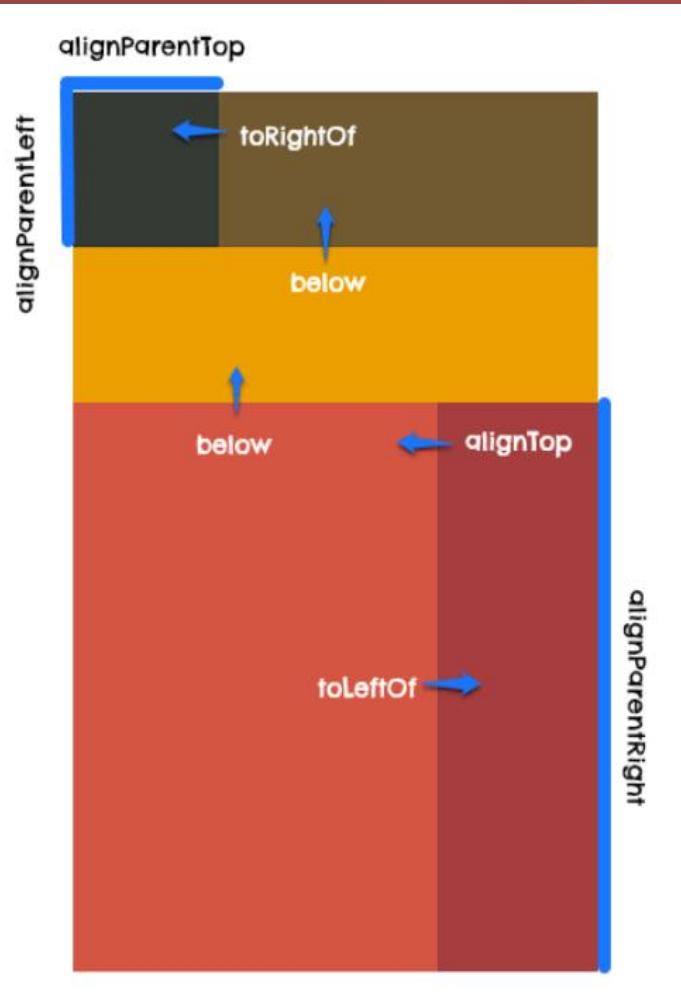
Se define la constante **b** de tipo **Button**

Se asigna a la constante **b** el objeto **View** de la **activity** cuyo **id** es **boton1** (es necesario *castear* a **Button** porque **findViewById** devuelve un objeto **View**)

```
6  
7 class MainActivity  
8     override fun onCreate(savedInstanceState: Bundle?) {  
9         super.onCreate(savedInstanceState)  
10        setContentView(R.layout.activity_main)  
11        val b: Button;  
12        b = findViewById<Button>(R.id.boton1)  
13        b.setText("Este es el botón 1");  
14    }  
15 }
```

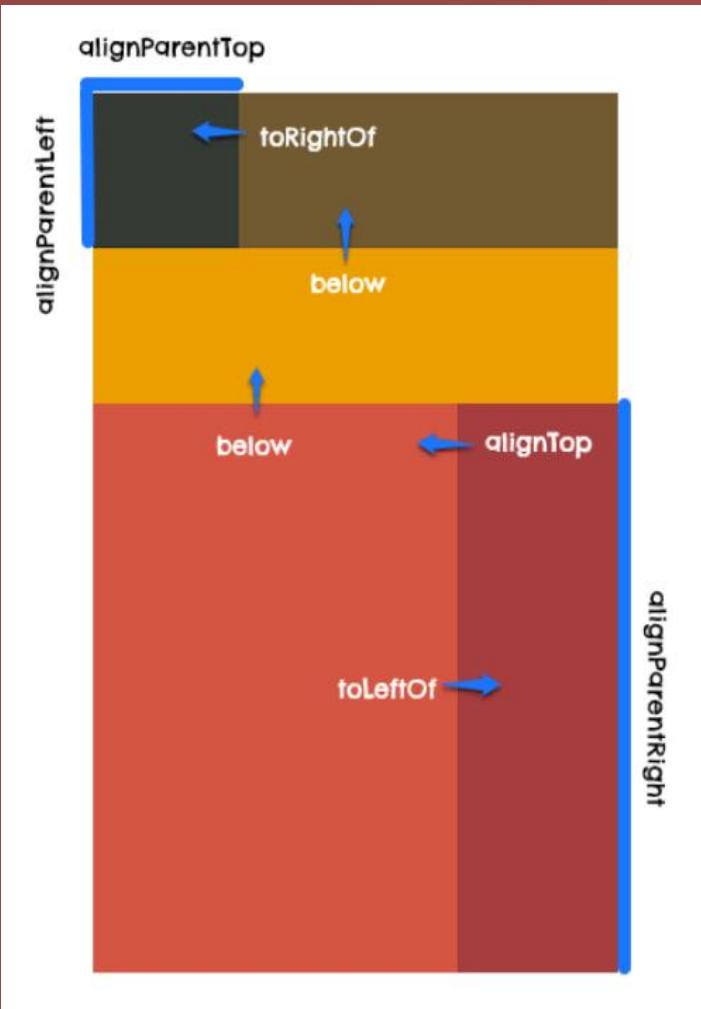
Se establece el texto del **botón 1**

RelativeLayout



Es un **Layout** donde las posiciones de los elementos contenidos (hijos) pueden ser descriptas 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

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

- Toolbar:** Standard Android Studio tools like Open, Save, Find, and Run.
- Project Navigators:** Shows the project structure with Layouts, app, src, main, res, and layout folders, and activity_main.xml selected.
- Code Editor:** Displays the XML code for activity_main.xml. A specific line, `android:layout_centerVertical="true"`, is highlighted with a brown rectangular background.
- Preview Window:** Shows a Nexus 4 device (API 26) with a blue bar at the top labeled "Layouts". A gray box with the number "1" is positioned in the center of the screen.
- Right Panel:** Includes a Palette tab, a ruler, and zoom controls (21%, 25%, 300%, 400%, 500%).

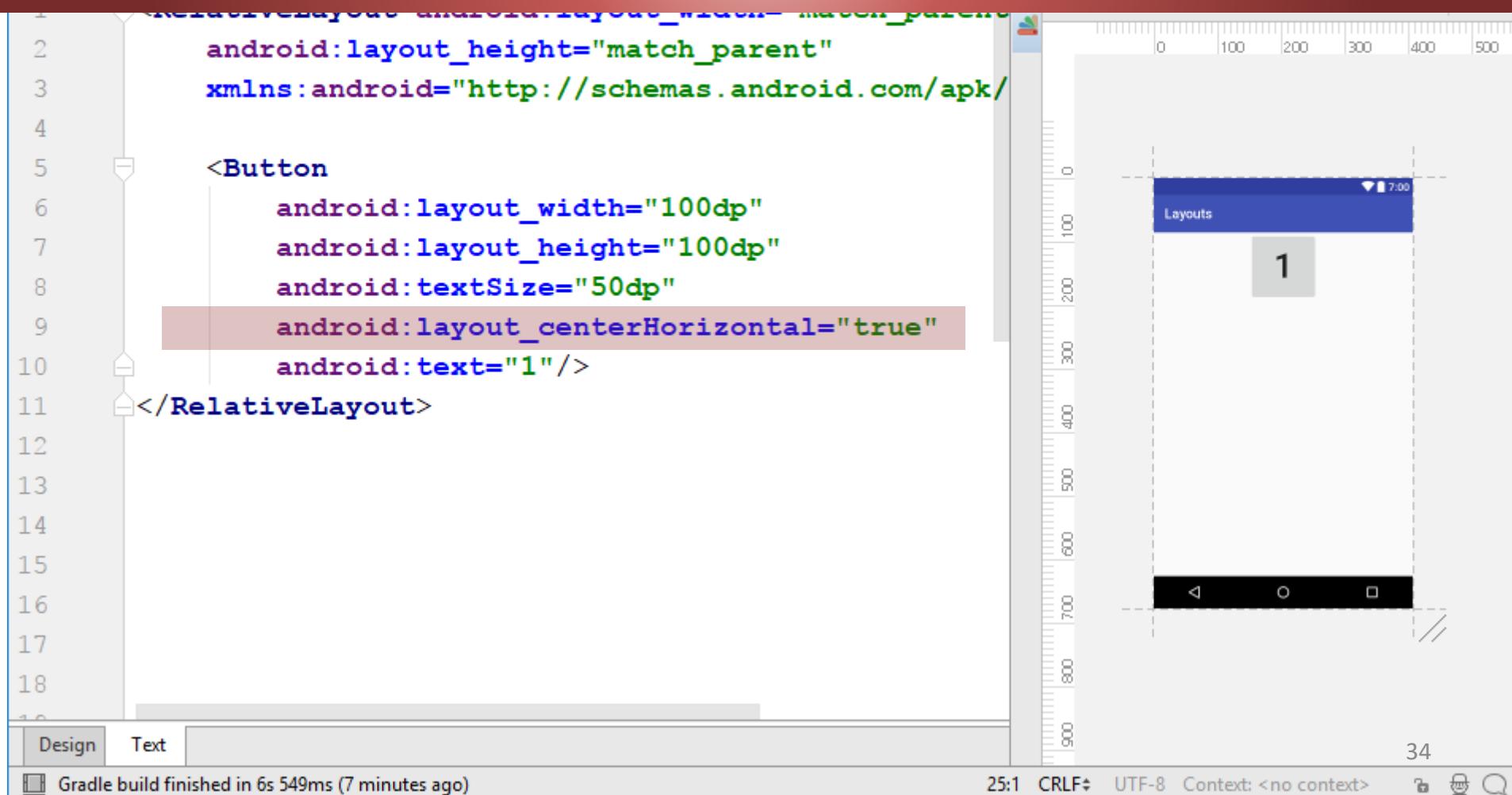
```
<RelativeLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/
```

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

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

RelativeLayout

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



The screenshot shows the Android Studio interface with the Layout Editor on the right and the XML code editor on the left.

XML 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_centerHorizontal="true"
10        android:text="1"/>
11 </RelativeLayout>
```

Layout Editor Preview:

The preview shows a blue rectangular container labeled "Layouts". Inside, there is a gray square containing the number "1". The square is centered horizontally within the container, demonstrating the effect of the `layout_centerHorizontal` attribute.

Bottom Status Bar:

Design Text

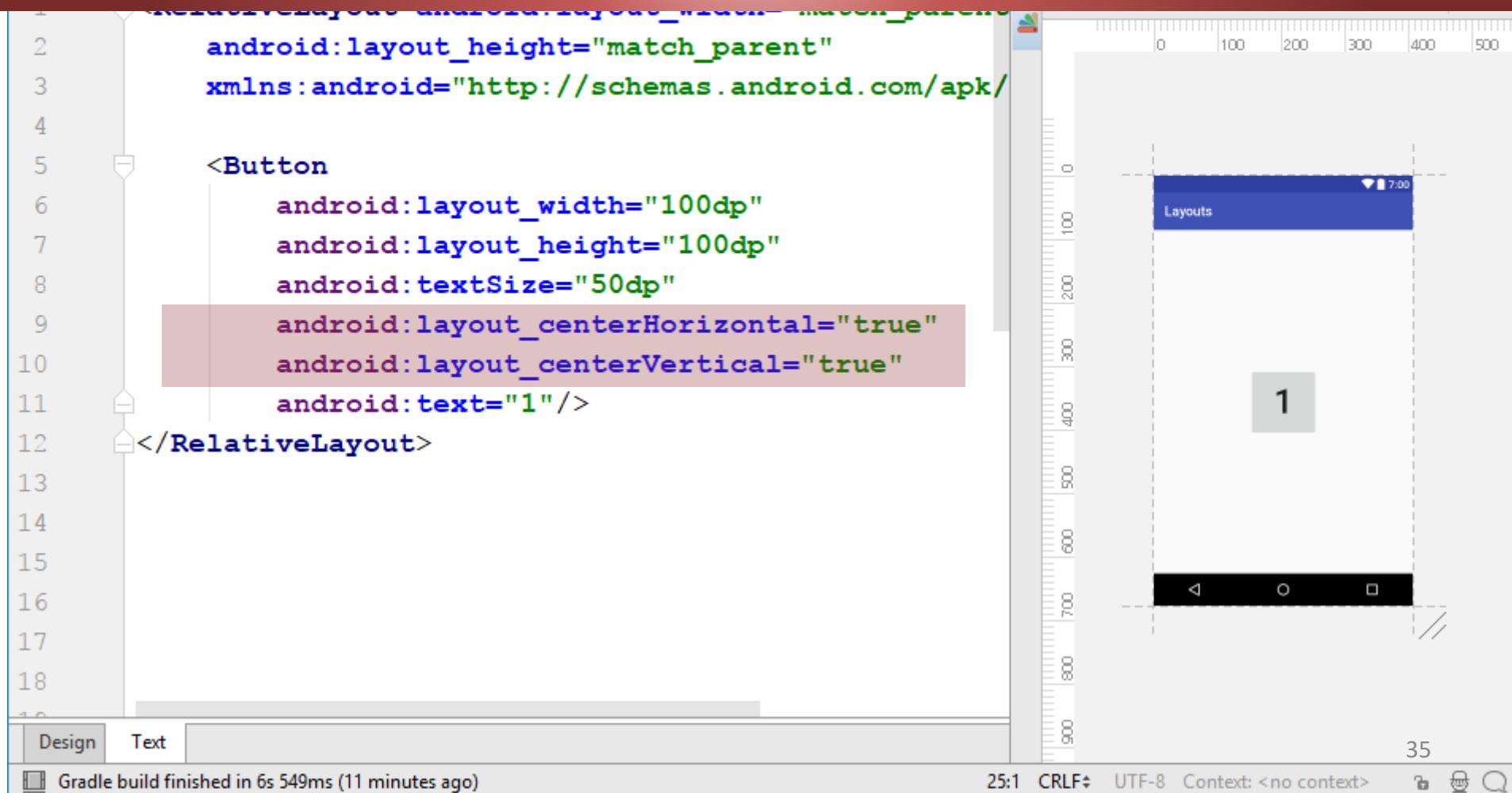
Gradle build finished in 6s 549ms (7 minutes ago)

25:1 CRLF UTF-8 Context: <no context>

34

RelativeLayout

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



The screenshot shows the Android Studio interface with two main panes: the Text pane on the left and the Design pane on the right.

Text Pane: Displays the XML code for a RelativeLayout. A specific section of the code is highlighted with a pink rectangular background:

```
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>
13
14
15
16
17
18
19
```

Design Pane: Shows a preview of the button's position within a container. The button is centered both horizontally and vertically relative to its parent's boundaries. It has a blue background and contains the number '1'.

At the bottom of the screen, there are tabs for "Design" and "Text", and a status bar with build information and file navigation icons.

RelativeLayout

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

The screenshot shows the Android Studio interface with the XML code editor and the design preview. The XML code defines a `RelativeLayout` with a central `Button` that is centered both horizontally and vertically within its parent container.

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

The design preview on the right shows a blue rectangular area representing the parent `RelativeLayout`. Inside it, a smaller gray square represents the centered `Button`, which contains the number "1". The preview includes a ruler at the top and left, and a navigation bar at the bottom.

At the bottom of the screen, there are tabs for "Design" and "Text", and status bars showing "Gradle build finished in 1s 980ms (58 minutes ago)", "12:1 CRLF", "UTF-8", "Context: <no context>", and icons for file, save, and search.

RelativeLayout

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

The screenshot shows the Android Studio interface with the XML code for a RelativeLayout on the left and its visual representation in the Design tab on the right.

XML Code:

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

Design Tab Preview:

The preview shows a blue rectangular container labeled "Layouts". Inside, there is a gray button with the number "1" centered vertically. The button's bottom edge is aligned with the bottom edge of the container, demonstrating the effect of the `layout_alignParentBottom` attribute.

Toolbars and Status Bar:

At the bottom, there are toolbars for "Design" and "Text", and a status bar showing "Gradle build finished in 1s 980ms (yesterday 23:23)", "27:1 CRLF", "UTF-8", "Context: <no context>", and a lock icon.

RelativeLayout

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

The screenshot shows the Android Studio interface with two main panes: the Text pane on the left and the Design pane on the right.

Text Pane (XML Code):

```
1 <RelativeLayout android:layout_width="match_parent"
2     android:layout_height="match_parent"
3     xmlns:android="http://schemas.android.com/apk/res/android">
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 Pane (Layout Preview):

The preview shows a blue rectangular button with the number "1" centered inside. The button is positioned at the bottom-right corner of the parent container, demonstrating the effect of the `layout_alignParentBottom` and `layout_alignParentRight` attributes.

At the bottom of the screen, there are tabs for "Design" and "Text", and a status bar with build information and file navigation icons.

RelativeLayout

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

The screenshot shows the Android Studio interface with two panes. On the left is the XML code editor, and on the right is the layout preview window.

XML Code:

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

Layout Preview:

The preview shows a blue rectangular container labeled "Layouts". Inside it, a gray button is positioned at the bottom-right corner of the container. The button has the number "1" centered on it. The entire layout is contained within a white area with a black border at the bottom.

At the bottom of the screen, there are tabs for "Design" and "Text", and a status bar with the text "Gradle build finished in 1s 980ms (yesterday 23:23)" and "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

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

The screenshot shows the Android Studio interface with two main panes: the Text pane on the left and the Design pane on the right.

Text Pane:

```
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 Pane:

The Design pane displays a layout preview with a blue rectangular button centered in the middle of the screen. The button has the number "1" on it. The layout is defined by a dashed border. The top edge of the button is aligned with the top edge of the dashed border, demonstrating the effect of `layout_alignParentTop="true"`.

At the bottom of the Design pane, there is a toolbar with icons for navigation and zoom.

At the very bottom of the interface, there are tabs for "Design" and "Text", and status bars for "Gradle build finished in 1s 980ms (yesterday 23:23)", "22:1 CRLF", "UTF-8", "Context: <no context>", and page numbers "41".

RelativeLayout

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

The screenshot shows the Android Studio interface with the XML code for a RelativeLayout on the left and the Design tab on the right.

XML Code:

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

Design Tab Preview:

The Design tab shows a blue rectangular container labeled "Layouts". Inside, there are two buttons: button 1 (a gray square) and button 2 (a darker gray square). Button 2 is positioned above button 1, demonstrating the effect of the `android:layout_above` attribute. The layout is constrained by a vertical dashed line and a horizontal dashed line, with a coordinate system ranging from 0 to 500 on both axes.

At the bottom of the screen, the status bar displays "26:1 CRLF UTF-8 Context: <no context>" and the footer shows "Gradle build finished in 3s 641ms (2 minutes ago)" and "42".

RelativeLayout

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

The screenshot shows the Android Studio interface with the XML code for a RelativeLayout on the left and the corresponding visual representation in the Design tab on the right.

XML 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: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>
```

Design Tab Preview:

The preview shows a blue rectangular container labeled "Layouts". Inside, there are two gray rectangular buttons. The top button is labeled "1" and has a bounding box of approximately [100, 100, 250, 250]. The bottom button is labeled "2" and has a bounding box of approximately [300, 300, 450, 450]. A dashed vertical line aligns the top edge of button "2" with the top edge of button "1".

Toolbars and Status Bar:

At the bottom, there are toolbars for "Design" and "Text", and a status bar showing "Gradle build finished in 6s 549ms (today 19:09)", "17:44", "CRLF", "UTF-8", "Context: <no context>", and icons for file operations.

RelativeLayout

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

The screenshot shows the Android Studio interface with the XML code for a RelativeLayout on the left and the corresponding visual representation in the Design tab on the right.

XML Code:

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

Design Tab: The Design tab displays a layout container with two buttons. Button 1 is centered in the parent with a size of 150dp by 150dp. Button 2 is positioned below Button 1 and is aligned to its baseline. Both buttons have a text size of 50dp.

Bottom Status Bar:

Design Text

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

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

44

RelativeLayout

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

The screenshot shows the Android Studio interface with the XML code for a RelativeLayout on the left and the Design tab on the right.

XML Code:

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

Design Tab Preview:

The preview shows a blue header bar at the top. Below it, there are two buttons labeled "1" and "2". Button "1" is positioned higher than button "2". A dashed blue line connects the bottom edge of button "1" to the bottom edge of button "2", illustrating the `layout_alignBottom` constraint.

Toolbars and Status Bar:

At the bottom, there are tabs for "Design" and "Text", with "Text" currently selected. The status bar at the very bottom shows: Gradle build finished in 3s 641ms (6 minutes ago), 16:37, CRLF, UTF-8, Context: <no context>, and icons for battery, signal, and search.

RelativeLayout

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

The screenshot shows the Android Studio interface with the XML code for a RelativeLayout on the left and the corresponding visual layout on the right.

XML Code:

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

Design Tab Preview:

The preview shows two buttons labeled "1" and "2". Button "1" is positioned at the top center of the screen. Button "2" is positioned below button "1", specifically aligned to the bottom edge of button "1". The layout uses a relative positioning strategy where button "2" is placed below button "1".

Bottom Status Bar:

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

46

RelativeLayout

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

The screenshot shows the Android Studio interface with the XML code for a RelativeLayout on the left and the Layout Editor on the right.

XML Code:

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

Layout Editor:

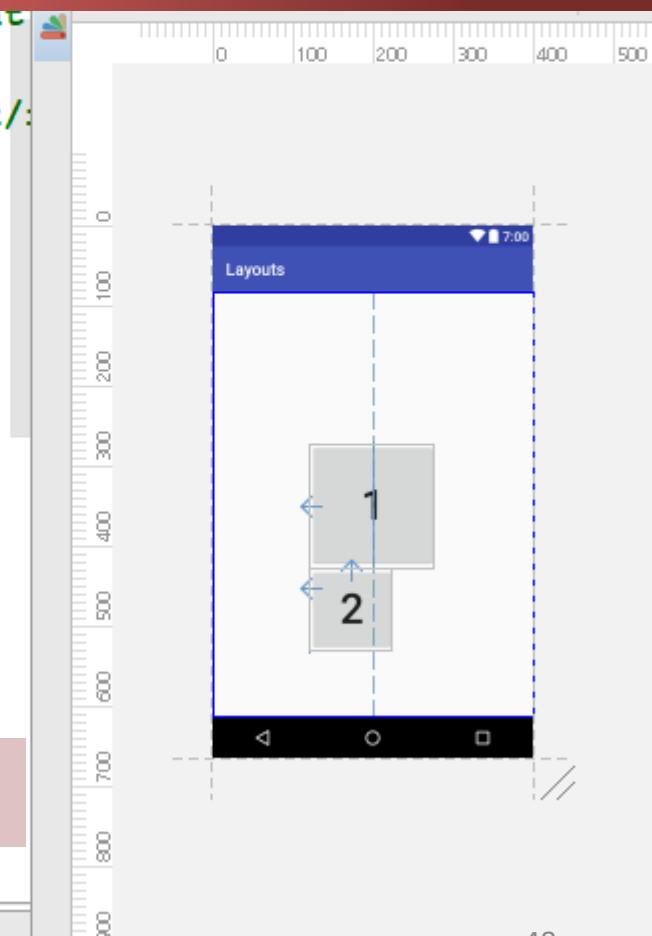
The Layout Editor displays a 500x800 grid. It contains two buttons: button 1 (top-right) and button 2 (bottom-left). A dashed blue vertical line represents the constraint from button 2 to the left edge of button 1. A blue arrow points from button 2 to this dashed line, indicating the direction of the constraint.

Toolbars and Status Bar:

At the bottom, there are tabs for "Design" and "Text". The status bar at the very bottom shows: Gradle build finished in 3s 641ms (9 minutes ago), 17:45, CRLF, UTF-8, Context: <no context>, and icons for battery, signal, and search.

RelativeLayout

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



The screenshot shows the Android Studio interface with the Layout Editor open. On the left, the XML code for a RelativeLayout is displayed:

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

The XML code includes two buttons. The second button has its `layout_alignLeft` attribute set to the value of the first button's `id`. In the Layout Editor, the second button is positioned to the left of the first button, demonstrating the effect of the `layout_alignLeft` attribute. The Layout Editor also shows a vertical dashed line representing the alignment constraint between the two buttons.

RelativeLayout

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

The screenshot shows the Android Studio interface with the XML code for a RelativeLayout on the left and the Design tab on the right.

XML Code:

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

Design Tab Preview:

The Design tab shows a blue rectangular container labeled "Layouts". Inside, there are two buttons. The top button is labeled "1" and has a width of 150dp and a height of 150dp. The bottom button is labeled "2" and has a width of 100dp and a height of 100dp. A vertical dashed line represents the right edge of button "1". Button "2" is positioned below button "1" and is aligned to the right of button "1", matching its right edge. The bottom of button "2" is aligned with the bottom of button "1".

Bottom Status Bar:

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

49

RelativeLayout

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

The screenshot shows the Android Studio interface with the XML code for a RelativeLayout on the left and the Design tab on the right.

XML Code:

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

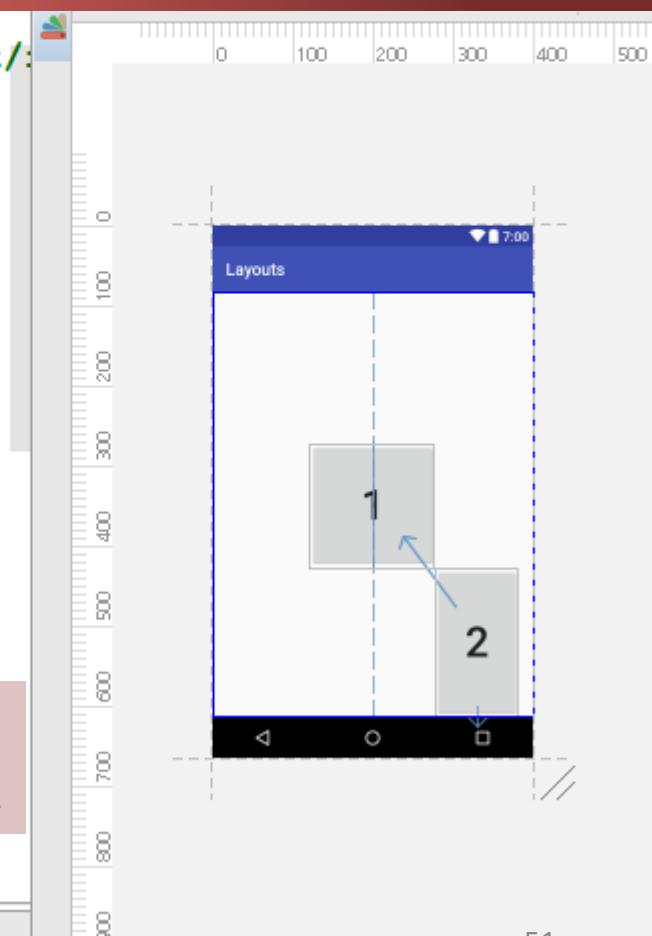
Design Tab Preview:

The Design tab shows a blue rectangular container labeled "Layouts". Inside, there are two gray square buttons. The top button is labeled "1" and is centered. The bottom button is labeled "2" and is positioned below and to the right of button "1". A dashed blue vertical line extends from the right edge of button "1" to the left edge of button "2", illustrating the `layout_toRightOf` constraint.

At the bottom of the screen, the status bar displays: Gradle build finished in 3s 641ms (13 minutes ago), 21:1, CRLF, UTF-8, Context: <no context>, and page number 50.

RelativeLayout

Se pueden combinar distintas disposiciones para obtener resultados muy variados



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

The screenshot shows the Android Studio interface with the XML code on the left and the corresponding visual representation in the Design tab on the right. The XML code defines a RelativeLayout containing two buttons. The first button is centered in the parent with dimensions 150dp by 150dp and a text size of 50dp. The second button is positioned below and to the right of the first, also with dimensions 100dp by 100dp and a text size of 50dp. It uses relative positioning to align with the first button and is specifically aligned to the bottom of the parent layout.

Design Text

Gradle build finished in 3s 641ms (16 minutes ago)

21:1 CRLF UTF-8 Context: <no context>

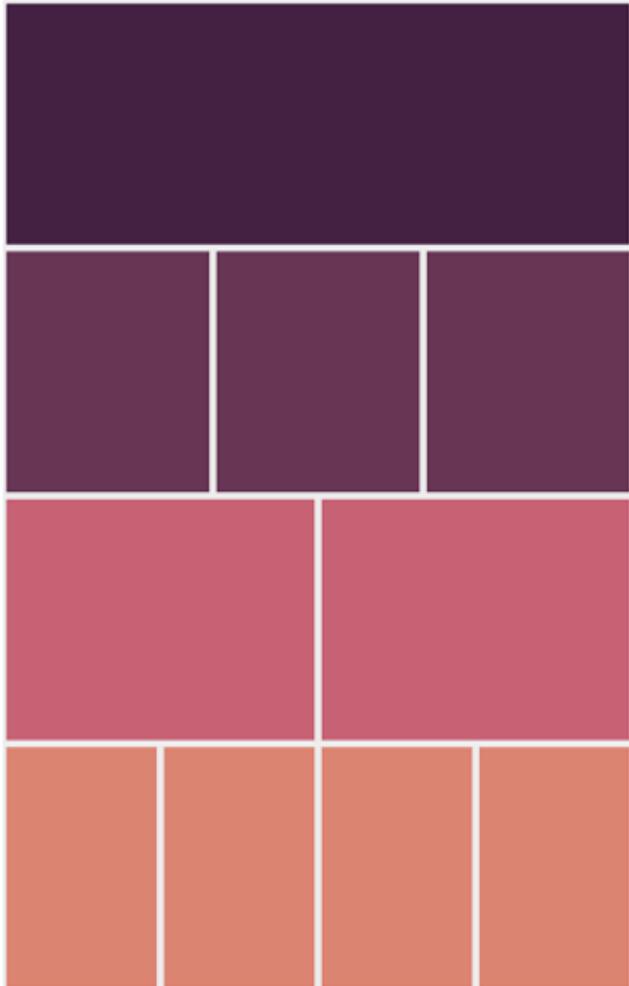
51

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

TableLayout

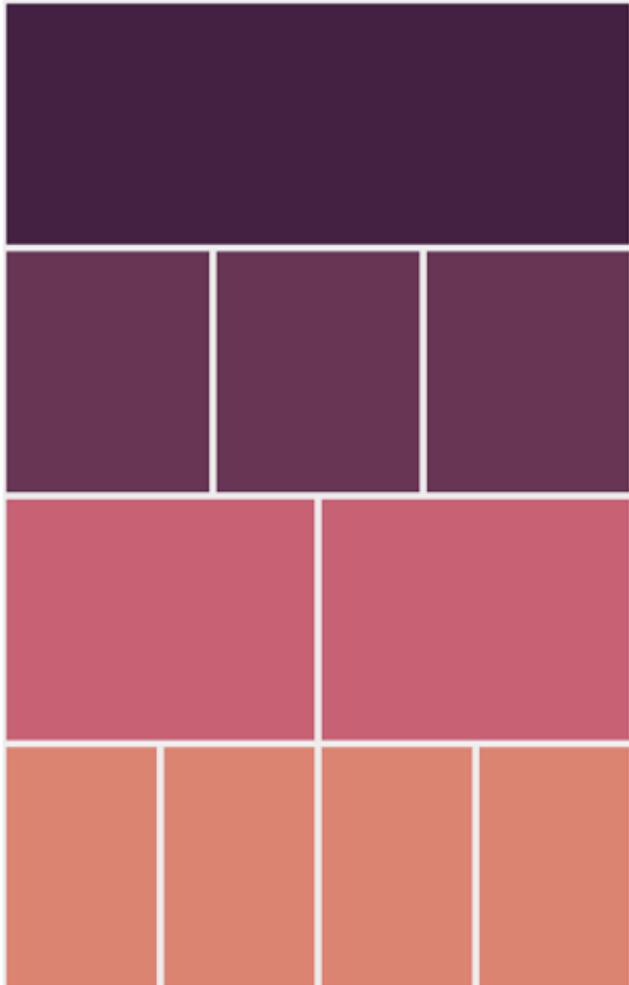


Es un **ViewGroup** que ubica a sus hijos en filas y columnas

Generalmente está compuesto de objetos **TableRow**.

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

TableLayout

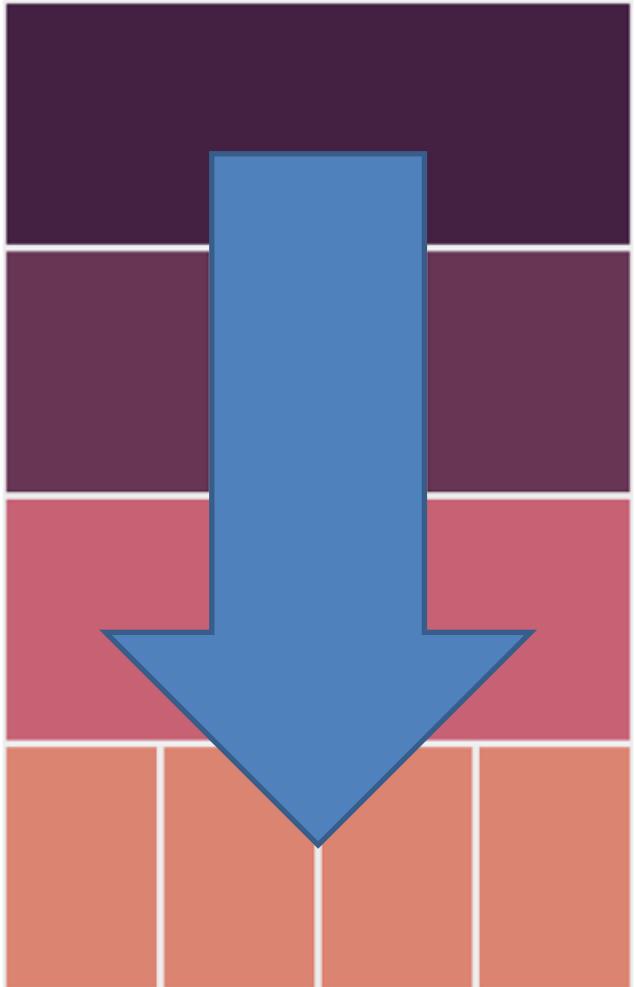


La tabla tendrá tantas columnas como aquel **TableRow** con más celdas.

Una celda podría expandirse ocupando más de una columna.

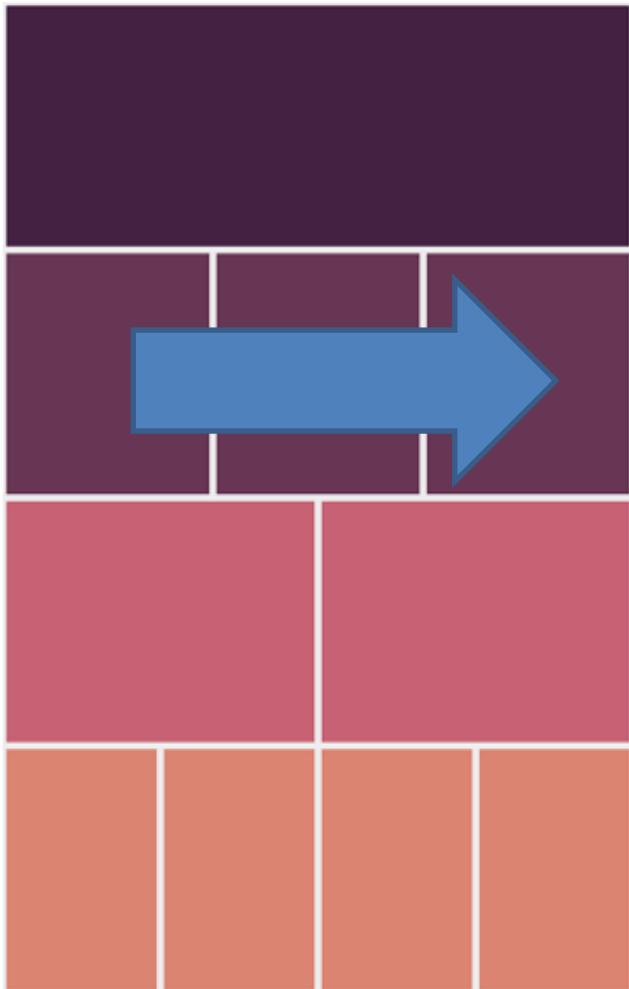
Un **TableLayout** no contempla bordes. Simplemente organiza elementos visuales

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

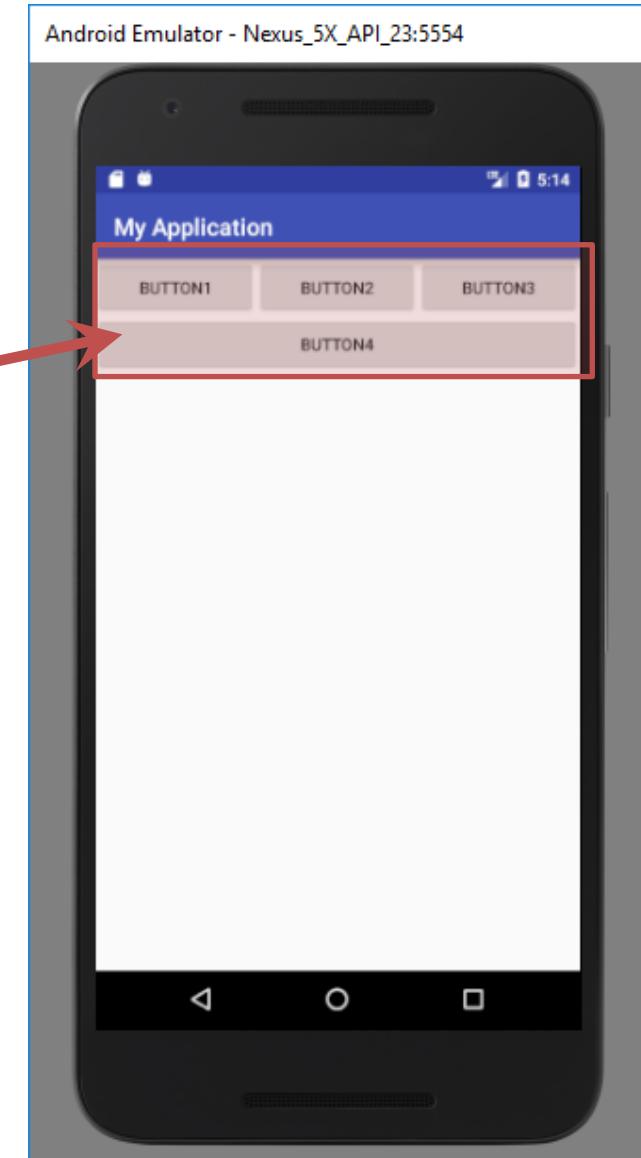
```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:stretchColumns="*">
    <TableRow>
        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            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:layout_weight="1"/>
        <Button
            android:id="@+id/button2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"/>
        <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:layout_weight="1"/>
    </TableRow>
</TableLayout>
```

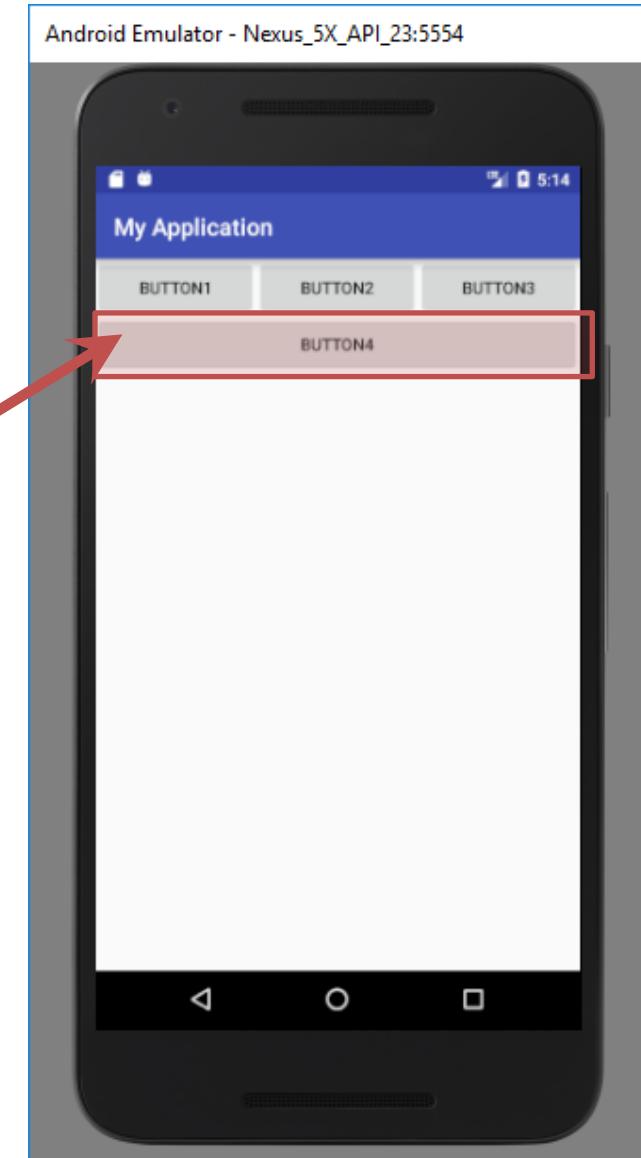
Estira las columnas para ocupar todo el ancho del TableLayout



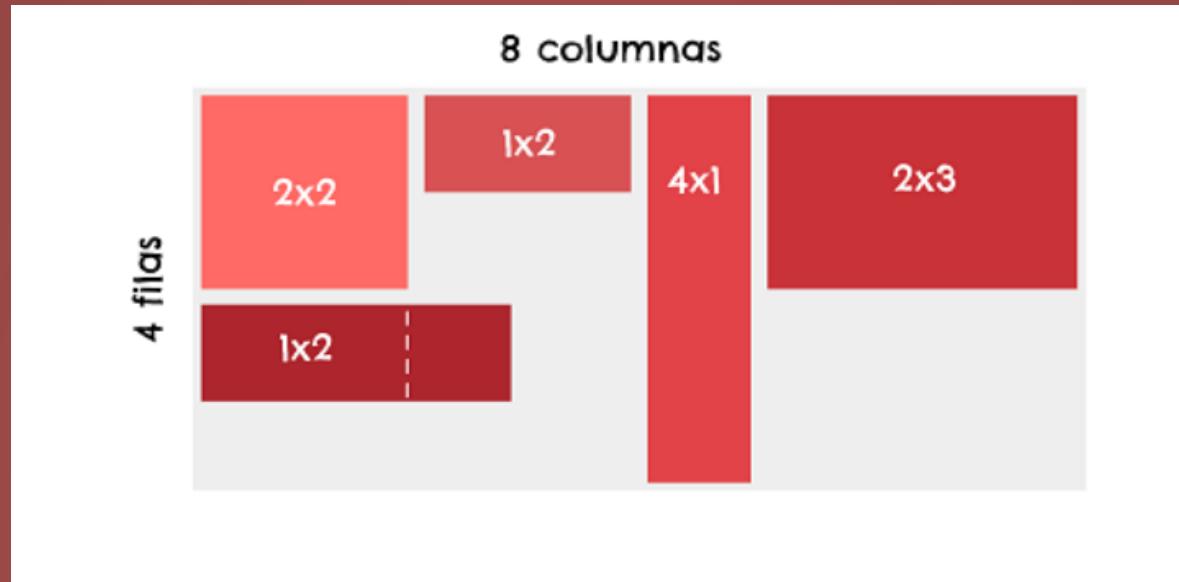
Ejemplo de TableLayout

```
<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

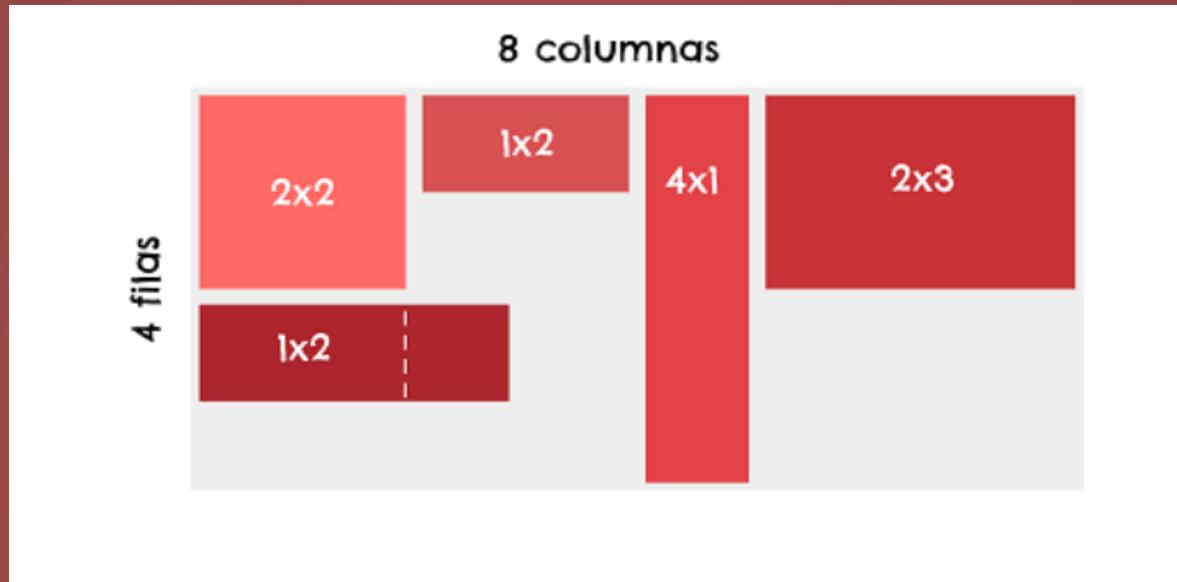


GridLayout



- 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>
```

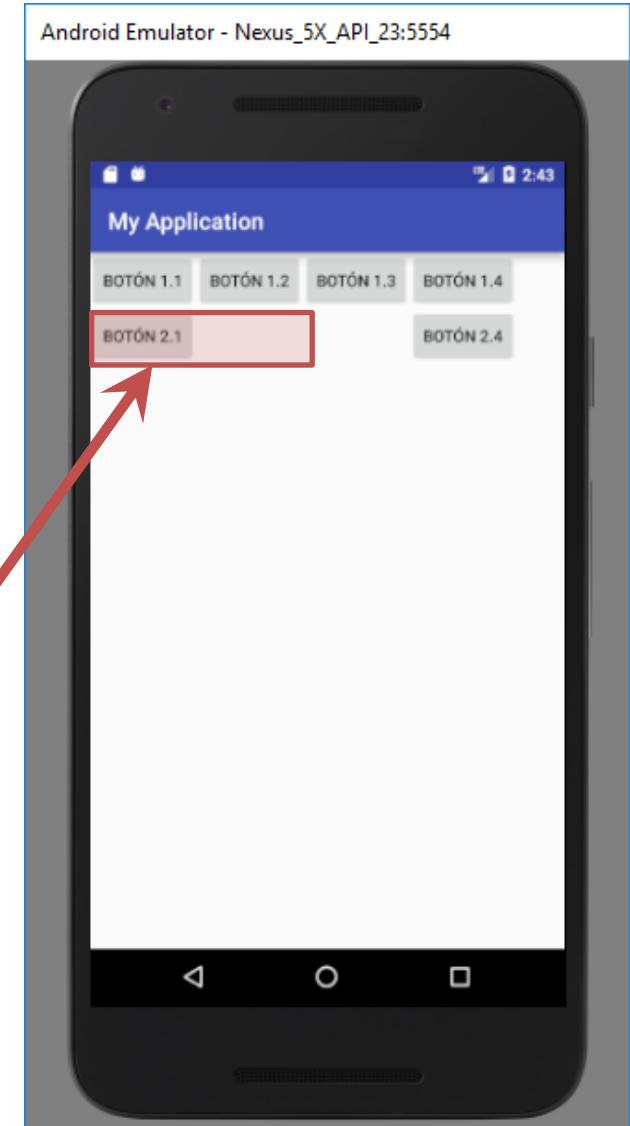
La vista ocupa 2 filas



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>
```

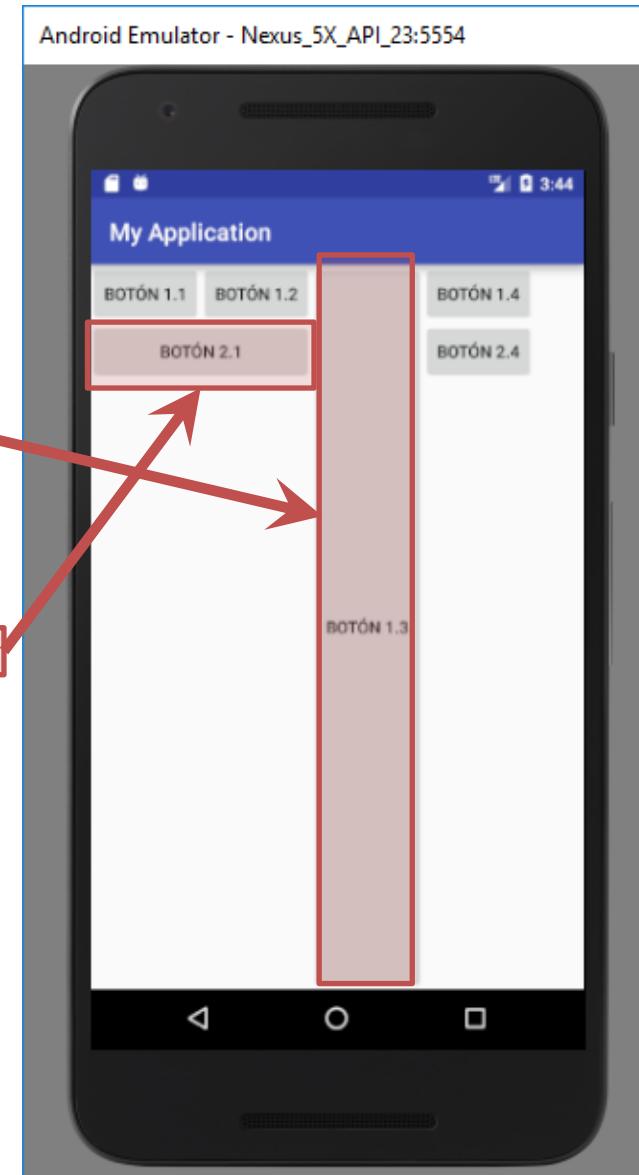
La vista ocupa 2
columnas



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

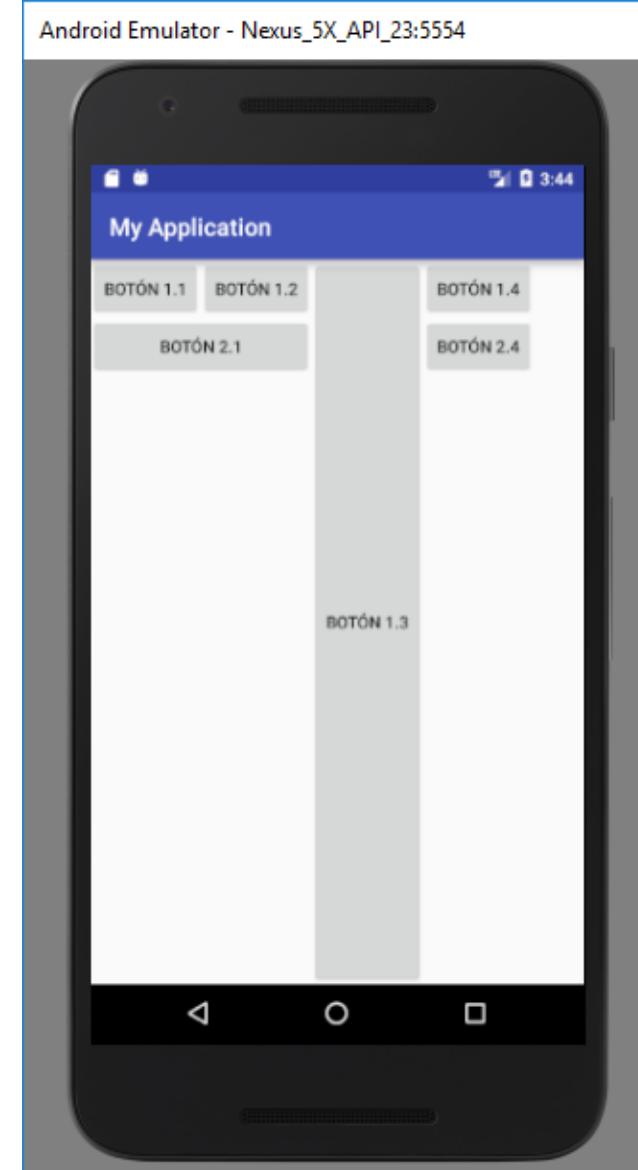
```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:columnCount="4"
    android:orientation="horizontal" >

    <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="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

