

# Interfaces de Usuario

### Interfaces de usuario

#### Layouts

- LinearLayout
- RelativeLayout
- FrameLayout

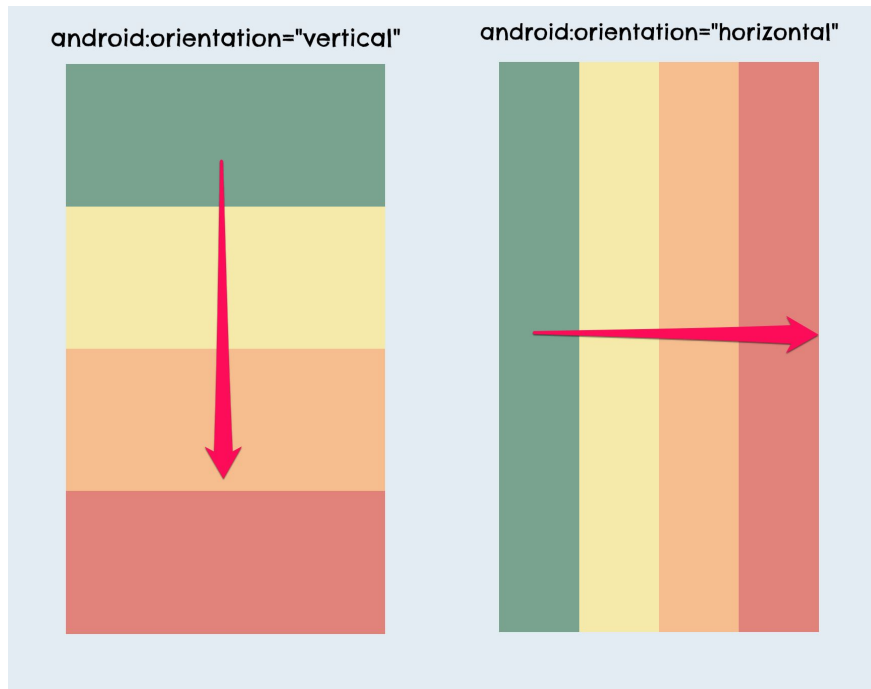
#### Views

- TextView
- Button
- EditText

### Linear Layout

Muestra los elementos uno tras otro, dependiendo de su orientación.

```
<LinearLayout  
  android:layout_width="match_parent"  
  android:layout_height="match_parent"  
  android:orientation="vertical"  
  android:background="#CCCCFC">  
</LinearLayout>
```



### Más propiedades

¿Cómo podríamos definir **tamaños de manera proporcional**?

### Weight

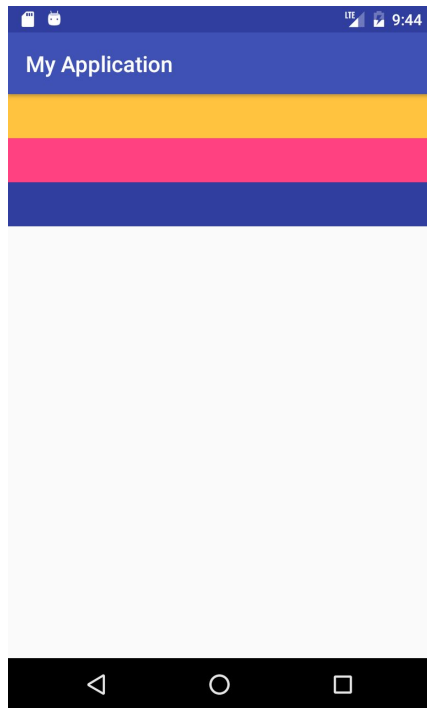
Especifica la importancia de un componente dentro de un view.

Es decir, cuánto espacio debe ocupar en la pantalla.

**android:layout\_weight**

### Weight

Pesos por  
default



Pesos  
1, 1, 2



### Interfaces de usuario

#### Layouts

- LinearLayout
- RelativeLayout
- FrameLayout

#### Views

- TextView
- Button
- EditText
- RadioButton
- Checkbox

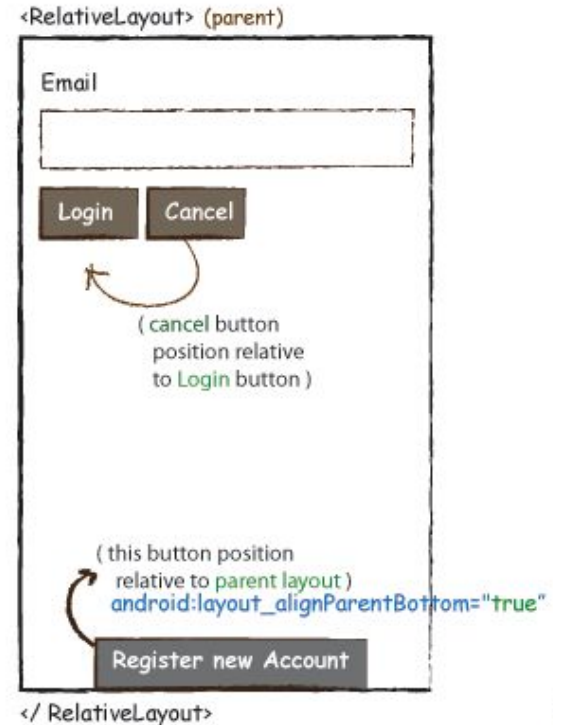
### Frame Layout

Está diseñado para mostrar un único componente en su interior, a modo de contenedor. **Permite superponer elementos (capas).**



### Relative Layout

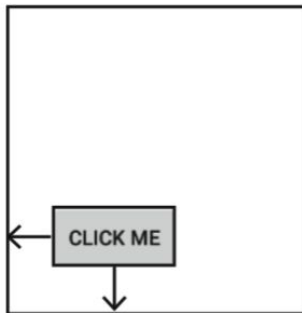
Permite ordenar componentes en base a distintas **posiciones relativas**.



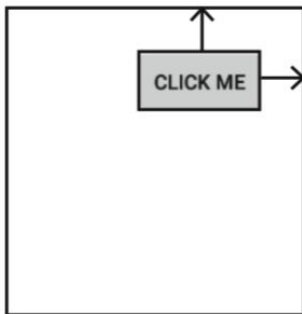


### Propiedades (alineación respecto al padre)

**android:**  
**layout\_alignParentBottom**



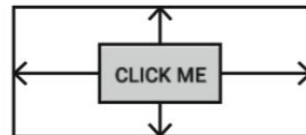
**android:**  
**layout\_alignParentLeft**



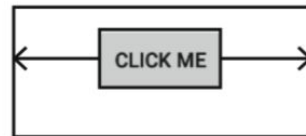
**android:**  
**layout\_alignParentRight**

**android:**  
**layout\_alignParentTop**

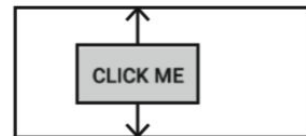
**android:**  
**layout\_centerInParent**



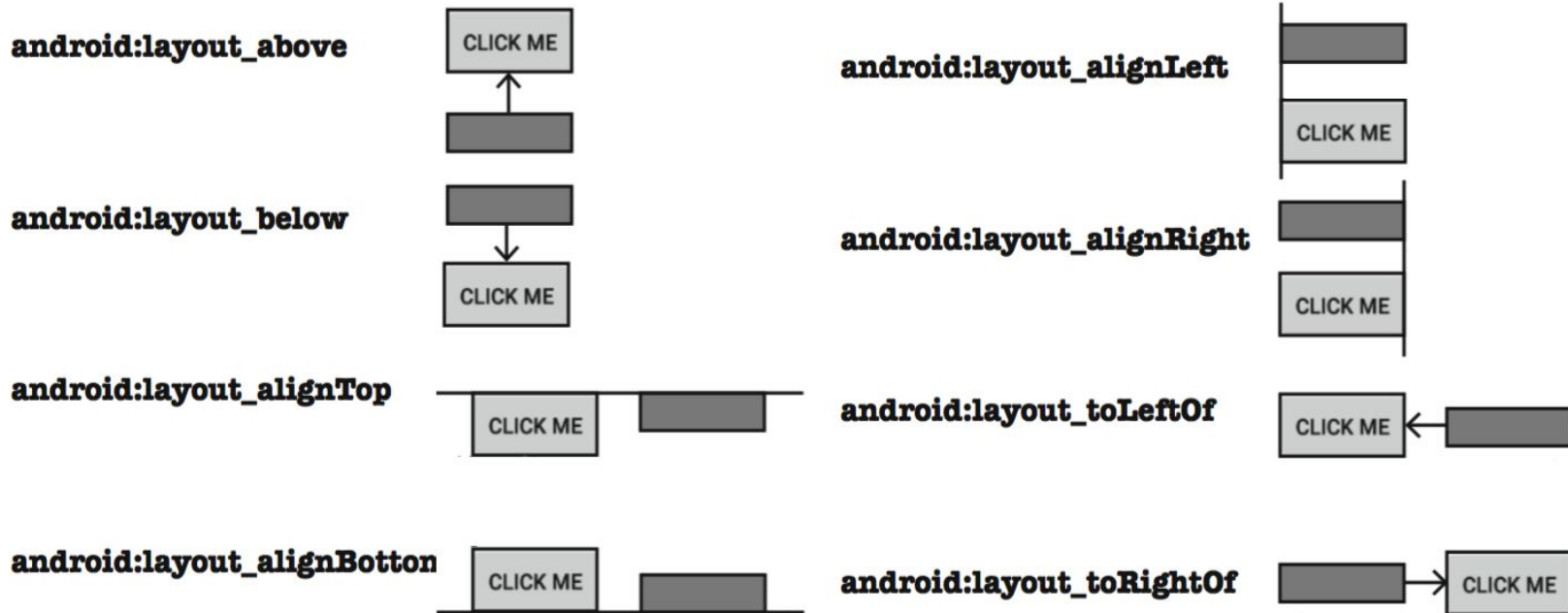
**android:**  
**layout\_centerHorizontal**



**android:**  
**layout\_centerVertical**

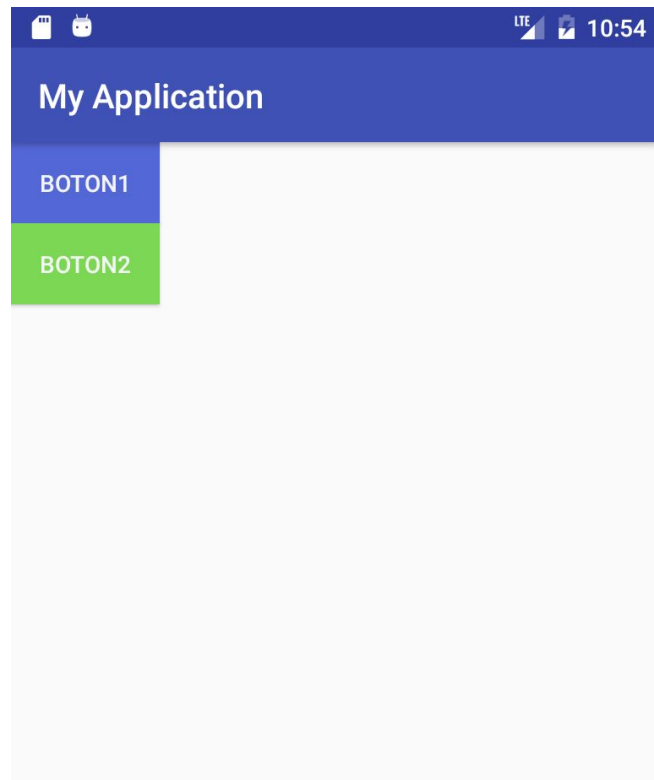


### Propiedades (alineación respecto a otros componentes)



### Ejemplo

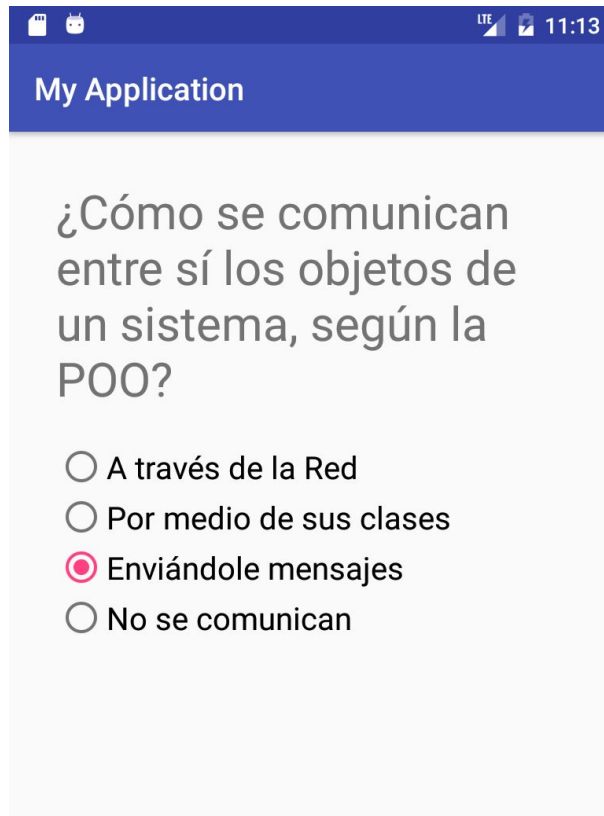
```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:background="#5467d7"
        android:text="Boton1"
        android:textColor="#f5f5f5" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/button2"
        android:layout_below="@id/button"
        android:background="#7bd754"
        android:text="Boton2"
        android:textColor="#f5f5f5" />
</RelativeLayout>
```



### Radio Button

Permite al usuario seleccionar una **única** opción entre varias.

Este componente se define como un **RadioGroup** y cada elemento es como **RadioButton**.



### Check Box

Los Checkboxes permiten al usuario seleccionar **una o más opciones** de un set en simultáneo.

Para saber si el checkbox está seleccionado usamos el método **isChecked()** (devuelve un Boolean)

