

Interfaces de Usuario

Interfaces de usuario

Layouts

- LinearLayout
- RelativeLayout
- FrameLayout

Views

- TextView
- Button
- EditText

XML

Como vimos en clases anteriores **la estructura y la jerarquía de las interfaces de usuario** va a estar descrita usando XML.

```
<TextView  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:text="@string/texto"  
  android:id="@+id/textView" />
```

Propiedades

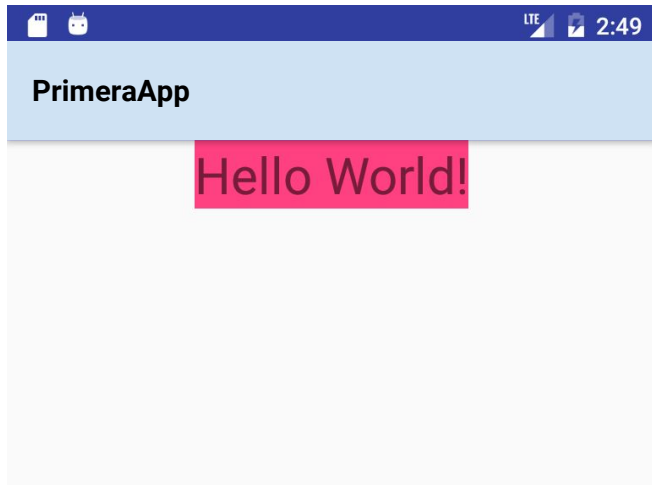
La mayoría de los elementos tienen las mismas propiedades:

- **Width:** Especifica el ancho
`android:layout_width`
- **Height:** Especifica el alto
`android:layout_height`

Especificando **ancho y alto**

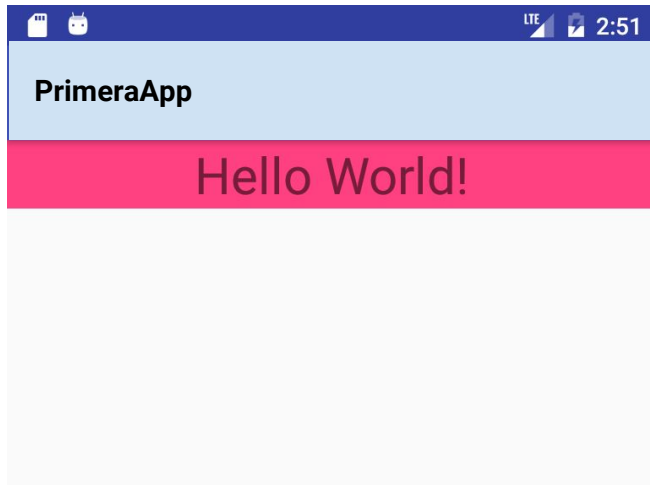
wrap_content

Especifica que el tamaño de la vista sea igual a su contenido.



match_parent

Especifica que el tamaño de la vista será el de su padre.



Especificando **ancho y alto**

En el caso que quisiéramos que un elemento tuviera un tamaño específico, se le puede asignar un valor de **dp** (density points).

Equivale a una cantidad de pixeles
según la resolución del dispositivo

```
<TextView  
  android:layout_width="90dp"  
  android:layout_height="50dp"  
  android:text="@string/texto"  
  android:id="@+id/textView" />
```



Propiedades

Margin

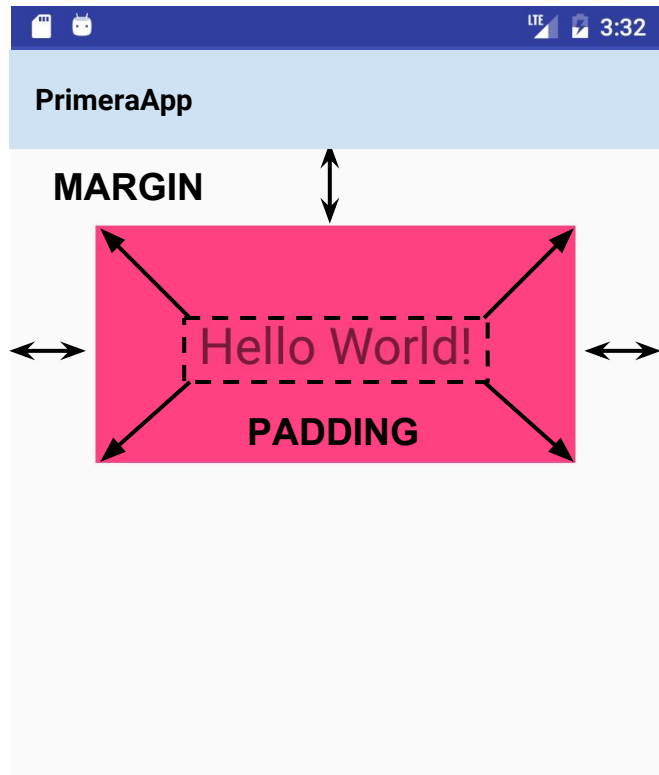
Distancia entre los elementos de un mismo layout.

`android:layout_margin`

Padding

Distancia entre el borde y el texto.

`android:padding`

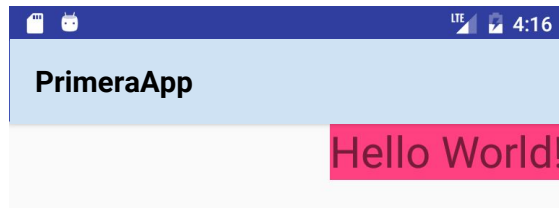
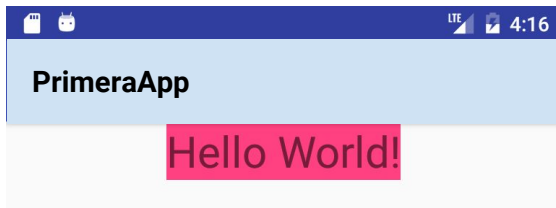
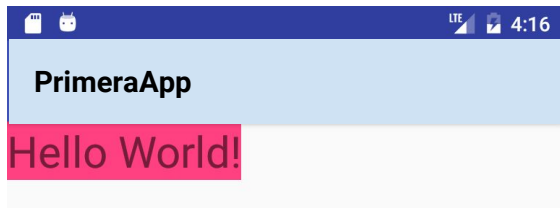


Propiedades

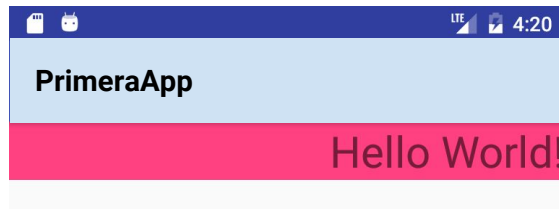
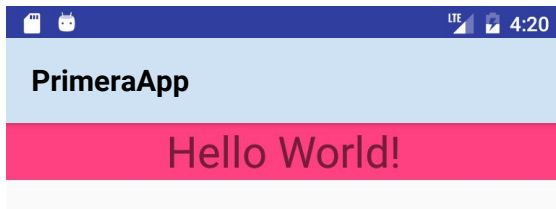
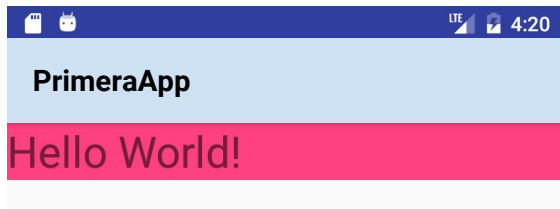
Gravity

Especifica la posición del contenido dentro de una vista.

android:layout_gravity



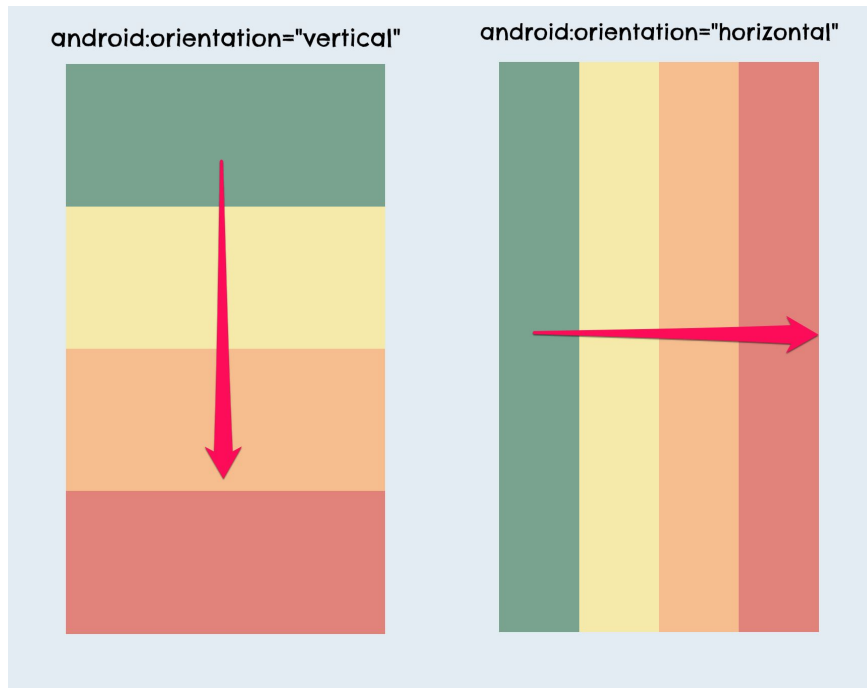
android:gravity



Linear Layout

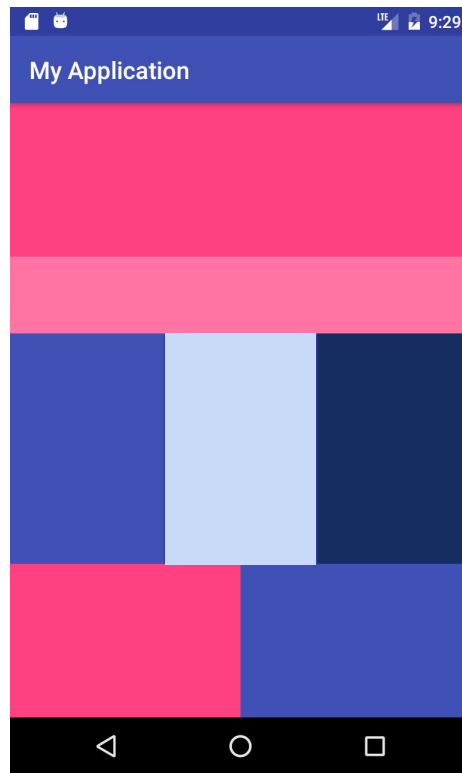
Muestra los elementos uno tras otro, dependiendo de su orientación.

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



Linear Layout

Como un LinearLayout organiza sus elementos y se pueden armar **estructuras jerárquicas**, nada nos impide que uno de estos elementos sea otro LinearLayout.



Ejercicio

Implementar el siguiente layout:

