

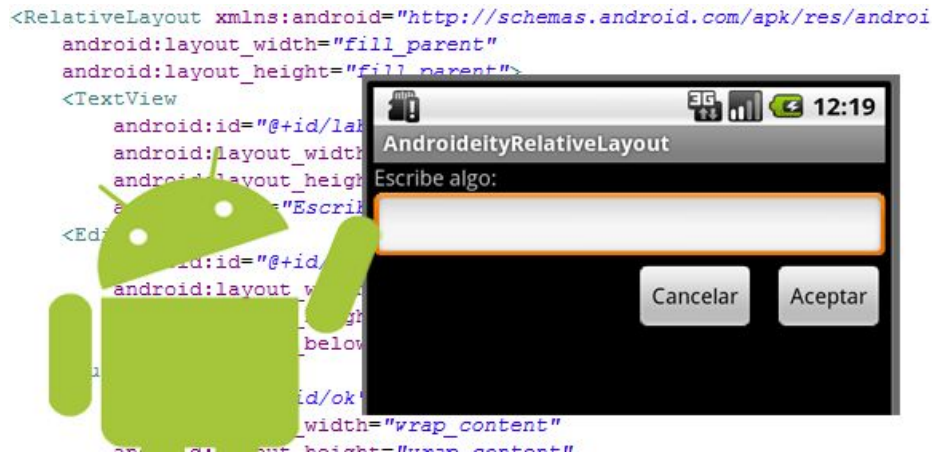
Interfaces de Usuario

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="0dp"

android:text="@string/some_text"

android:id="@+id/textView"

/>

Propiedades

La mayoría de los elementos tienen las mismas propiedades:

1) Width

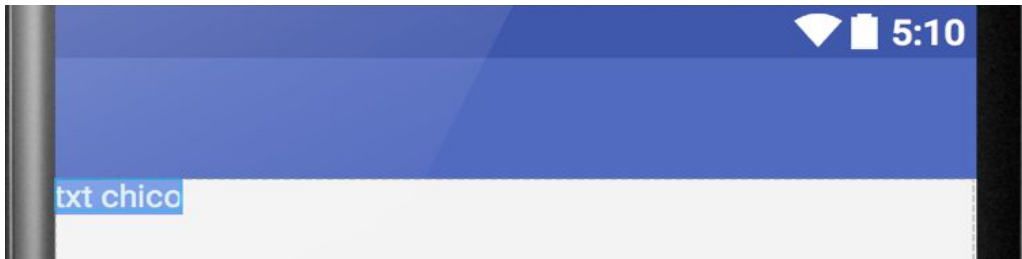
- Especifica el ancho
- `android:layout_width`

2) Height

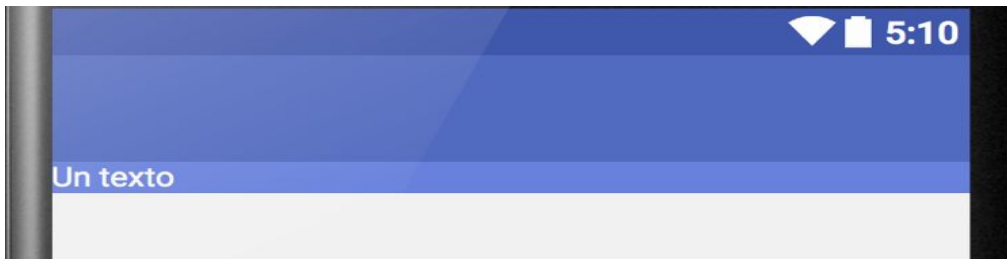
- Especifica el alto
- `android:layout_height`

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

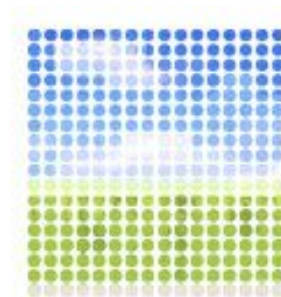
- DPI

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

Según la resolución del device equivale a una cantidad de pixeles.

<TextView

```
    android:layout_width="90dp"  
    android:layout_height="wrap_content"  
    android:text="Un texto"  
    android:textColor="#f5f5f5"  
    android:background="#5467d7"  
/>
```



Propiedades

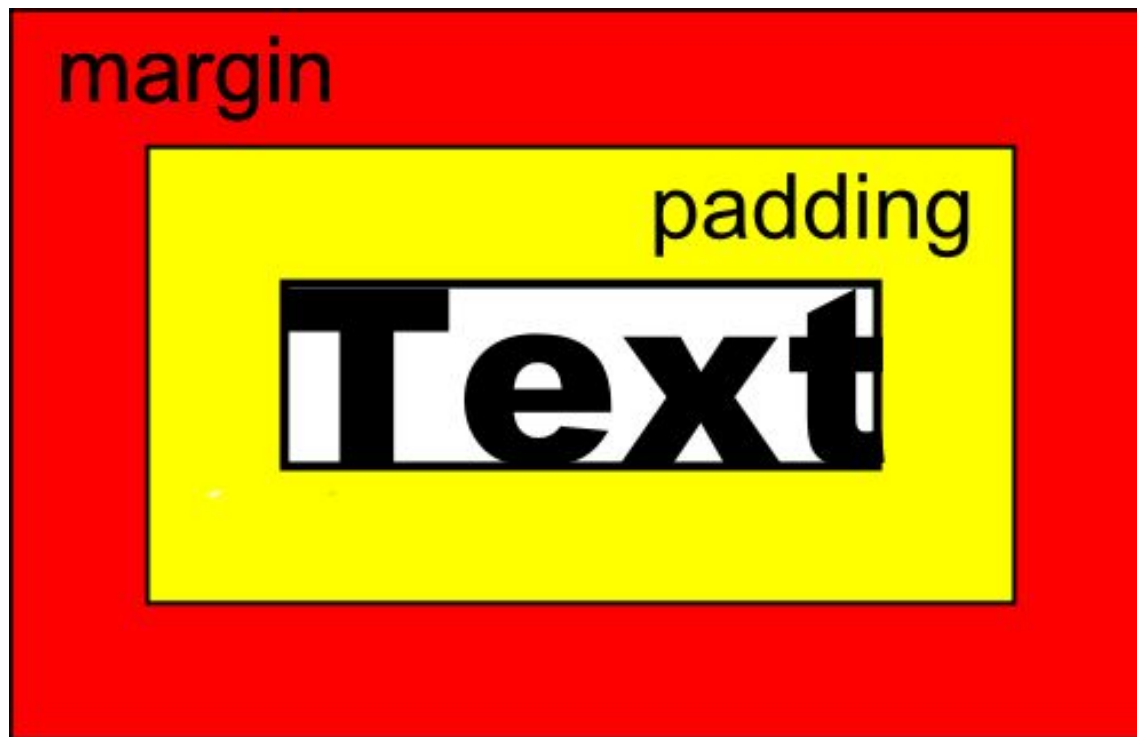
3. Margin

- Distancia entre los elementos de un mismo layout.
- `android:layout_margin`

4. Padding

- Distancia entre el borde y el texto.
- `android:padding`

Especificando margin y padding



Propiedades

5. Gravity

- Especifica la posición del contenido dentro de una vista.
- `android:gravity`
- `android:layout_gravity`



Linear Layout

Muestra los elementos uno tras otro, dependiendo de su orientación

```
<LinearLayout
```

```
    android:layout_width="match_parent"
```

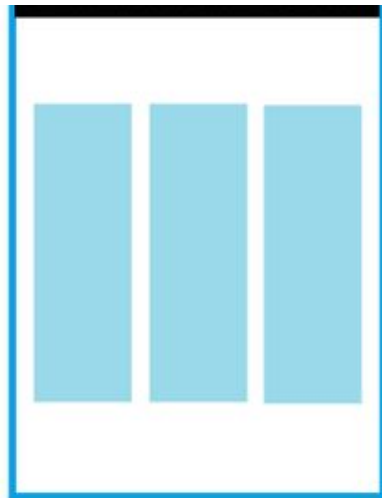
```
    android:layout_height="match_parent"
```

```
    android:orientation="vertical"
```

```
    android:background="#CCCCFC"
```

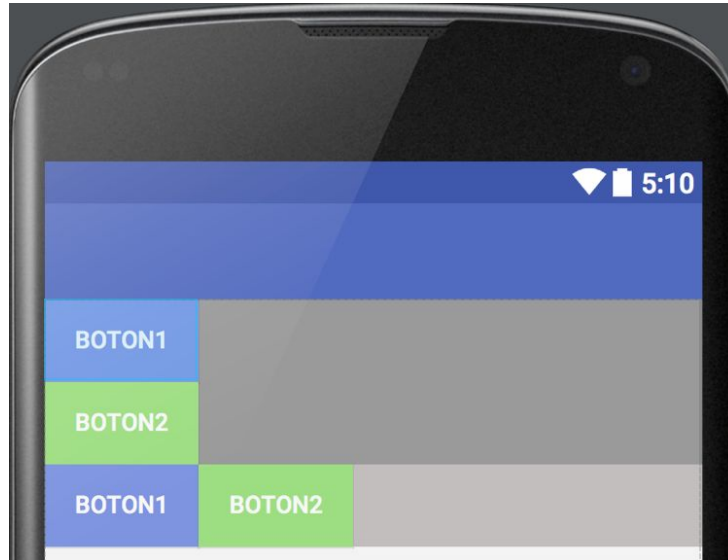
```
>
```

```
</LinearLayout>
```



Linear...

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



Ejercicio

Implementar el siguiente layout



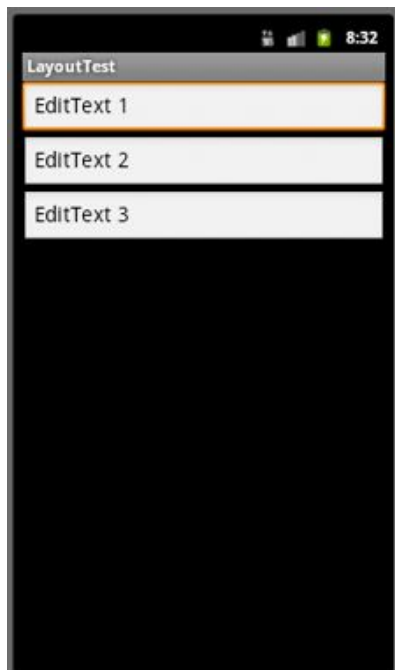
Más propiedades

¿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.

Más propiedades



Default weights



Weights 1,1,2