

Capítulo 8 Elementos gráficos en Android

Los elementos visuales contienen siempre 2 propiedades, sin importar, el comportamiento, el tamaño, el color

Esas propiedades son:

- Ancho → width } Estas propiedades pueden tener 3 valores
- Alto → height } distintos: match-parent, fill-parent, wrap-content

- * match-parent: Esto quiere decir que tendrá el ancho de su padre
- * fill-parent: Mismo concepto que "match-parent", ya no se utiliza.
- * wrap-content: Se ajusta de manera dinámica.

id: Sirve para poder acceder desde el código Java al elemento.

El código del ID será el siguiente: `android:id="@+id/textView"`

- Interfaz gráfica: En la parte inferior podemos ver que tenemos dos pestañas: Design, Text.

- Pestaña Design: lo que vemos es el archivo desde la interfaz gráfica, esto nos permite poder modificar el archivo de forma muy similar a una herramienta "Drag & Drop" (arrastrar & soltar) podemos arrastrar los elementos del bdo izquierdo y soltarlos en la posición que los necesitamos

- Pestaña Text: Esta pestaña se basa en el código duro y para si necesitamos colocar un elemento utilizando esta pestaña veremos código XML ventajas de usar esta pestaña es que es más ligero de usar y contiene menos bugs, ya que solo debe de interpretar el código que el desarrollador está colocando.

- TextView: Es un elemento que nos sirve para colocar texto dentro de una aplicación

Propiedades del TextView:

- text Size: Nos permite asignarle un tamaño diferente al texto de este elemento Android, no utiliza px como es esto de medida sino que utiliza sp que significa Scale-independent Pixels

- text Color: Permite cambiar el color del texto de nuestro textView
`textColor="#FF0000" ← forma hexadecimal`

► TextColor utilizando el archivo colors.xml

Otra forma de asignar color es utilizando el archivo colors.xml res/values/colors.xml. Es buena práctica colocar aquí los colores y asignarles un nombre dentro de colors.xml

```
<color name="red">#FF0000</color>
```


Para utilizarlo dentro de nuestros componentes basta con ponerlo en las propiedades así

```
android:textColor="@color/red"
```


► Button El botón por lógica es el elemento que se debe de utilizar como una entrada de información por el usuario.

► Edit Text Este elemento es el más utilizado junto a button y textView, permite escribir información por medio del usuario. La propiedad hint dentro del edit text nos permite indicar al usuario que escribir ahí y a la hora de digitar

► Margin y Padding

► Margin  tenemos este escenario; si quisiéramos distancia entre ellos de por medio es de colocar la propiedad margin al primer o al segundo

```
android:layout_marginRight="100dp"
```

 Padding también nos sirve para mantener un espacio, pero podemos que padding es internamente y margin es externamente.

► Peso y distribución

Si en nuestro layout padre colocamos `android:weightSum="2"` y en nuestros botones colocamos `android:layout_weight="1"` les asignamos a nuestros botones el peso 1 y 1 en otras palabras 50%.

Capítulo 9 Densidades y Tamaño de Pantalla

► Densidades.

En Android contamos con 6 densidades de pantalla:

ldpi - densidad baja

mdpi - densidad media

hdpi - densidad alta

xhdpi - extra alta densidad

xxhdpi - extra extra alta densidad

xxxhdpi - extra extra extra alta densidad.

En valores numéricos cada densidad representaría lo siguiente equivalencia:

ldpi 120 dpi

mdpi 160 dpi

hdpi 240 dpi

xhdpi 320 dpi

xxhdpi 480 dpi

xxxhdpi 640 dpi

► Tamaños de Pantalla: Podemos tener 4 tamaños de pantalla hoy en día en el mundo Android: • Pequeño, • Normal, • Largo, • Extra largo.

► Un tamaño pequeño de pantalla va de los 2 plg. a los casi 3,7 plg.
el tamaño normal va de los 3,5 plg. a los 4,7 plg. un tamaño largo de pantalla va de los 4,2 plg. a los 7 plg. y un tamaño extra largo va de los 7 plg. en adelante.

Relación densidad y tamaño de pantalla:

La densidad de la pantalla y mejor calidad de la imagen depende de muchos factores y no tanto del dispositivo; Pero entre mayor densidad tiene un dispositivo mayor calidad tiene su pantalla.

hoy en día las 2 densidades con mayor calidad son **xxhdpi**, **xxxhdpi**
la densidad = a los pixeles que se tiene por pulgada.

Cuanto más grande es la pantalla de un teléfono se pierde la calidad, ya que los pixeles se distribuyen de una manera más amplia y se pierde densidad.

Capítulo 10 Medidas en ANDROID

Las diferentes medidas que tenemos en Android sirven para asignar valores de tamaño, espacio a los elementos, etc.

- DP (Density - independent Pixels) Cada una de estas medidas tiene usos diferentes dependiendo de la situación o del elemento que lo requiera.
- PT (Points)
- Px (Pixels)
- MM (Millimeters)
- IN (inch)
- SP (Scale independent Pixels)

▶ Píxeles (Px) El píxel es la medida universal para la representación gráfica en la informática pero no sirve en android

▶ Pulgadas (inch) No sirve en android

▶ Milímetros (mm) No sirve en android

▶ Puntos (pt) los puntos son matemáticamente $1/72$ de una plg. solamente debemos de tomar en cuenta el tamaño de la pantalla

Píxeles de densidad independiente (dp) En la documentación oficial los encontramos como una unidad abstracta, eso se debe a que físicamente no los podemos medir con una regla o con cualquier dispositivo. Se tiene que calcular.

un píxel en dp es equivalente a un píxel físico en una pantalla de 160 DPI

DPI o en español los puntos por plg. son la cantidad de píxeles que existen en una plg. (Densidad)

En valores numéricos cada densidad representaría la siguiente equivalencia:

• 1dpi 120 dpi

• mdpi 160 dpi

• hdpi 240 dpi

• xhdpi 320 dpi

• xxhdpi 480 dpi

• xxxhdpi 640 dpi

Volviendo al concepto de dp, recordemos que la equivalencia de 1 dp sería lo mismo que un píxel físico en una pantalla de 160 dpi

Resolución	DPI	Escala	Tamaño ejemplo	Imagen
mdpi	160	1.0	100 x 100	
hdpi	240	1.5	150 x 150	
xhdpi	320	2.0	200 x 200	
xxhdpi	480	3.0	300 x 300	
xxxhdpi	640	4.0	400 x 400	