

Nombre	Carnet	Grupo
Gabriela Michelle Álvarez Flores	AF160428	01L

Capítulo 8: Elementos gráficos de android

- **match-parent**: hereda el ancho de su padre.
 - **fill-parent**: hereda el ancho de su padre. Ya no se usa esta desfasado.
 - **wrap-content**: coloque un tamaño dependiendo del contenido
-
- **Interfaz Gráfica**: nos permite representar las vistas de nuestras actividades. ~~una~~ escritas en archivos con extensión XML.
 - **Pestaña "Design"**: Vemos el archivo desde la interfaz gráfica lo que nos permite arrastrar y soltar componentes.
 - **Pestaña "Text"**: Aquí se colocan componentes en la interfaz a base de código XML.
 - **TextView**: elemento que sirve para colocar texto.
 - **Orientación LinearLayout**: inicialmente coloca los elementos uno abajo del otro, si colocamos la orientación en horizontal los elementos se apilan uno sobre otro por lo que es necesario colocar el valor wrap-content en los elementos.
 - **Propiedades TextView**:
 - ↳ **Text Size**: Tamaño del fuente
 - ↳ **TextColor**: Cambio de color del fuente
 - **Button**: captura o ejecuta una acción al momento de ser presionado.
 - **Edit Text**: nos permite escribir datos para que la app los pueda usar.
 - **Practicando Relative Layout**: podemos colocar un orden o posicionamiento indicando que un elemento se coloque abajo de otro. Podemos centrar los elementos y agregarle un padding o margin.
 - **Margin y Padding**: El margin se le coloca a un elemento y este toma espacio entre sí mismo y el otro elemento. El padding es un espacio que se genera en el interior del elemento.
 - **Pesos y distribución**: Colocamos el atributo weightSum para poder repartir el tamaño que queremos que ocupe cada elemento a partir de su valor dado. Básicamente es el método más sencillo de ordenar nuestros elementos en la pantalla.

Capítulo 9: Densidades y tamaños de pantalla

Debido al creciente aumento de diferentes pantallas creadas por los fabricantes, hoy en día es bastante complejo el tema de las densidades.

• Densidades:

→ 1 dpi	densidad baja	→ 1 dpi 120 dpi
→ mdpi	densidad media	→ mdpi 160 dpi
→ hdpi	densidad alta	→ hdpi 240 dpi
→ xhdpi	extra alta densidad	→ xhdpi 320 dpi
★ → xxhdpi	extra extra alta densidad	→ xxhdpi 480 dpi
★ → xxxhdpi	extra extra extra alta densidad	→ xxxhdpi 640 dpi

valores numéricos de cada densidad

Estas dos son las densidades más altas. Las podemos encontrar en dispositivos de gama alta o gama media super dotados.

• Tamaños de pantalla

- Pequeños [2 pulg a 3,7 pulg]
- Normal [3.5 pulg a 4.7 pulg]
- Largo [4.2 pulg a 7 pulg]
- Extra Largo [7 pulg a ∞]

La calidad de imagen de una pantalla depende de la densidad de esta, a mayor densidad mayor será la calidad de imagen.

Si el contenido tiene baja calidad entonces no importa si se tiene mayor densidad ya que es el contenido el que está con baja calidad.

Hay que tener en cuenta que la densidad depende de los píxeles que se tienen por pulgada.

Capítulo 10: Medidas en android.

Medidas que podemos utilizar para aplicar tamaños o espacios a los elementos.

- **Píxeles (px):** medida física y por lo tanto inmutable, es decir no varían por lo que no es la mejor opción, para implementar tamaños.
- **Pulgada (inch):** por la misma razón que los píxeles las pulgadas no son una opción de medidas que se utilicen en android.
- **Milímetros (mm):** los mm están en la misma categoría que los px y inch, estos no se utilizan.
- **Puntos (pt):** se trata de una medida que matemáticamente es $1/72$ de una pulgada. Solo hay que tener en cuenta el tamaño de la pantalla.
- **Píxeles de densidad independiente (dp)**
Esta unidad no se mide, más bien se calcula.
1 px en dp es equivalente en un px físico en una pantalla de 160 DPI. Los DPI son básicamente la cantidad de píxeles que hay en 1 inch. (densidad)

El valor de altura de 2dp se adaptará al tamaño de la pantalla en el que el elemento se esté mostrando.