

# Atributos más usados en Layouts Android

## Tamaño y espacio

---

### layout\_width

- **Qué hace:** Define el ancho de un view dentro de su contenedor.
  - Valores más frecuentes:
    - `match_parent`: ocupa todo el ancho disponible.
    - `wrap_content`: ocupa solo lo necesario para su contenido.
    - Ej.: `android:layout_width="wrap_content"`
- 

### layout\_height

- **Qué hace:** Define la altura de un view.
  - Mismos valores que `layout_width`.
  - Ej.: `android:layout_height="match_parent"`
- 

### layout\_margin

- **Qué hace:** Espacio **externo** alrededor del view (separación con otros elementos).
- Puede usarse en direcciones específicas:
  - `layout_marginTop`
  - `layout_marginBottom`
  - `layout_marginLeft`
  - `layout_marginRight`
- Ej.:

```
xml
CopiarEditar
android:layout_margin="16dp"
android:layout_marginTop="8dp"
```

---

## padding

- **Qué hace:** Espacio **interno** entre el borde del view y su contenido.
- También existe por lados:
  - paddingTop
  - paddingBottom
  - paddingLeft
  - paddingRight

- Ej.:

```
xml
CopiarEditar
android:padding="10dp"
```

---

## Posicionamiento

---

### layout\_gravity

- **Qué hace:** Posiciona el view **dentro del layout** padre.
- Usado en LinearLayout, FrameLayout.
- Ej.: center, left, right.
- Ej.:

```
xml
CopiarEditar
android:layout_gravity="center_horizontal"
```

---

## gravity

- **Qué hace:** Posiciona el **contenido** dentro del view.
- Ej.:

```
xml
CopiarEditar
android:gravity="center"
```

---

## orientation

- **Qué hace:** Define si los hijos de LinearLayout se colocan en **fila o columna**.

- Valores:

- `vertical`
- `horizontal`

- Ej.:

```
xml
CopiarEditar
android:orientation="vertical"
```

---

## layout\_weight

- **Qué hace:** Distribuye espacio extra entre hijos de LinearLayout.

- Ej.: Si dos vistas tienen `layout_weight="1"`, se reparten el espacio disponible al 50%.

- Ej.:

```
xml
CopiarEditar
android:layout_weight="2"
```

---

## Atributos específicos de RelativeLayout

---

### layout\_below

- **Qué hace:** Ubica el view **debajo** de otro view.

- Ej.:

```
xml
CopiarEditar
android:layout_below="@id/otroView"
```

---

### layout\_above

- **Qué hace:** Ubica el view **arriba** de otro view.
- 

### layout\_toRightOf

- **Qué hace:** Ubica el view **a la derecha** de otro view.
-

## layout\_toLeftOf

- **Qué hace:** Ubica el view **a la izquierda** de otro view.
- 

## layout\_alignParentTop

- **Qué hace:** Alinea el borde superior del view con el borde superior del contenedor.
- 

## layout\_alignParentBottom

- **Qué hace:** Alinea el borde inferior del view con el borde inferior del contenedor.
- 

## layout\_centerInParent

- **Qué hace:** Centra el view tanto horizontal como verticalmente en el contenedor.
- 

## Aspecto Visual

---

### background

- **Qué hace:** Define color o imagen de fondo del view.
- Puede ser un color o un recurso drawable.
- Ej.:

```
xml
CopiarEditar
android:background="@color/purple_200"
android:background="#FF0000"
```

---

### textColor

- **Qué hace:** Define el color del texto en vistas como TextView, Button, etc.
- Ej.:

```
xml
CopiarEditar
android:textColor="@color/white"
```

---

### textSize

- **Qué hace:** Define tamaño del texto.
- Ej.:

```
xml
CopiarEditar
android:textSize="18sp"
```

---

## src

- **Qué hace:** Imagen mostrada en un ImageView.
- Ej.:

```
xml
CopiarEditar
android:src="@drawable/ic_launcher"
```

---

## hint

- **Qué hace:** Texto gris que aparece en un EditText cuando está vacío.
- Ej.:

```
xml
CopiarEditar
android:hint="Ingrese su nombre"
```

---

## id

- **Qué hace:** Identificador único del view para poder manipularlo en Kotlin o Java.
- Ej.:

```
xml
CopiarEditar
android:id="@+id/botonAceptar"
```

---

## visibility

- **Qué hace:** Controla si el view está visible o no.
- Valores:
  - `visible` → se ve.
  - `invisible` → no se ve, pero ocupa espacio.
  - `gone` → no se ve y **no ocupa espacio**.
- Ej.:

```
xml
CopiarEditar
android:visibility="gone"
```

---

## Tips MC

- `layout_gravity`  $\neq$  `gravity` → el primero afecta al view dentro del layout; el segundo afecta al contenido interno.
  - `wrap_content` → sólo lo necesario.
  - `match_parent` → ocupa todo el espacio disponible.
  - `RelativeLayout` → atributos comienzan con `layout_` y suelen hacer referencia a otros views por id.
-