



Ambiente de desarrollo y sus elementos de configuración

Los assets de un proyecto Android (Parte I)

Implementar una interfaz de usuario utilizando buenas prácticas en el manejo de estilos para brindar un aspecto visual e interacciones acordes a lo requerido

- **Unidad 1:** Ambiente de desarrollo y sus elementos de configuración.
- **Unidad 2:** Elementos de la interfaz, navegación e interacción.
- **Unidad 3:** Fundamentos de GIT y GitHub.



Te encuentras aquí



¿Qué aprenderás en esta sesión?

Reconoce mecanismos para la incorporación de assets en un proyecto Android

Pregunta abierta:
¿Qué es un asset?



**/* Unidades de medida para vistas,
recursos y textos */**

Densidades de pantalla en Android

Manejar recursos de acuerdo a la densidad de pantalla

A diferencia de los teléfonos iOS, donde el tamaño físico y la densidad se mantienen estables, en Android hay una gran cantidad de modelos, cada uno con tamaños y densidades particulares, generando muchas combinaciones.

- El tamaño físico es el tamaño de la pantalla medido en diagonal, en pulgadas.
- La densidad de la pantalla significa cuántos píxeles (puntos) se muestran en una pulgada, puntos por pulgada (dpi).
- Por ejemplo, un teléfono de 6" y 1080 x 2400 píxeles tiene mejor resolución que uno de 6" con 480 x 960 píxeles, a pesar de tener el mismo tamaño físico.

Unidades de medida

dp	Píxeles independientes de la densidad	Unidad abstracta que se basa en la densidad física de la pantalla. Estas unidades son relativas a una pantalla de 160 DPI (puntos por pulgada), en la que 1 dp es aproximadamente 1 px. Cuando se utiliza una pantalla de mayor densidad, el número de píxeles que se usa para dibujar 1 dp se escala por un factor apropiado con respecto a los DPI de la pantalla.
sp	Píxeles independientes de la escala	Similar a dp, pero también se ajusta según la preferencia de tamaño de fuente del usuario. Recomendado para tamaños de fuente (Densidad de pantalla + preferencia del usuario).
px	Píxeles	Corresponde a los píxeles reales en la pantalla (NO recomendada).

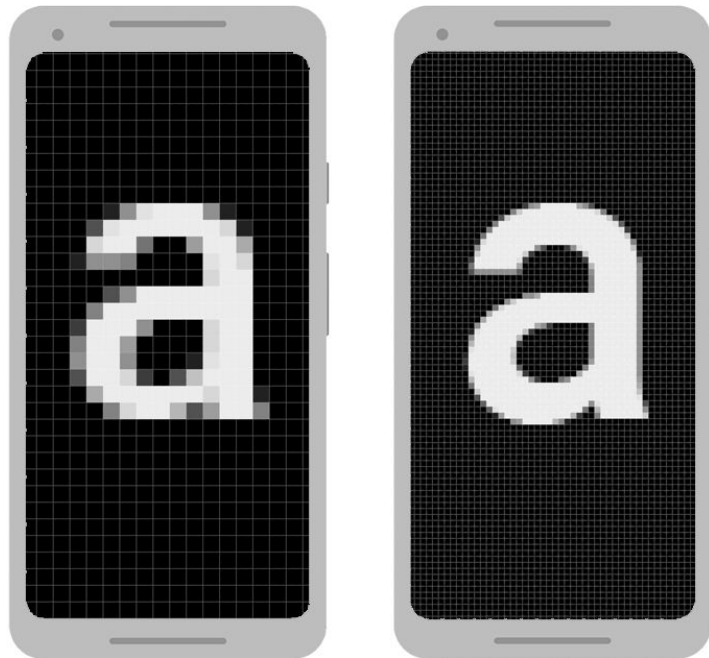
Fuente: developer.android.com/Dimension

dp

Cómo se manejan los recursos de acuerdo a la densidad de pantalla del dispositivo

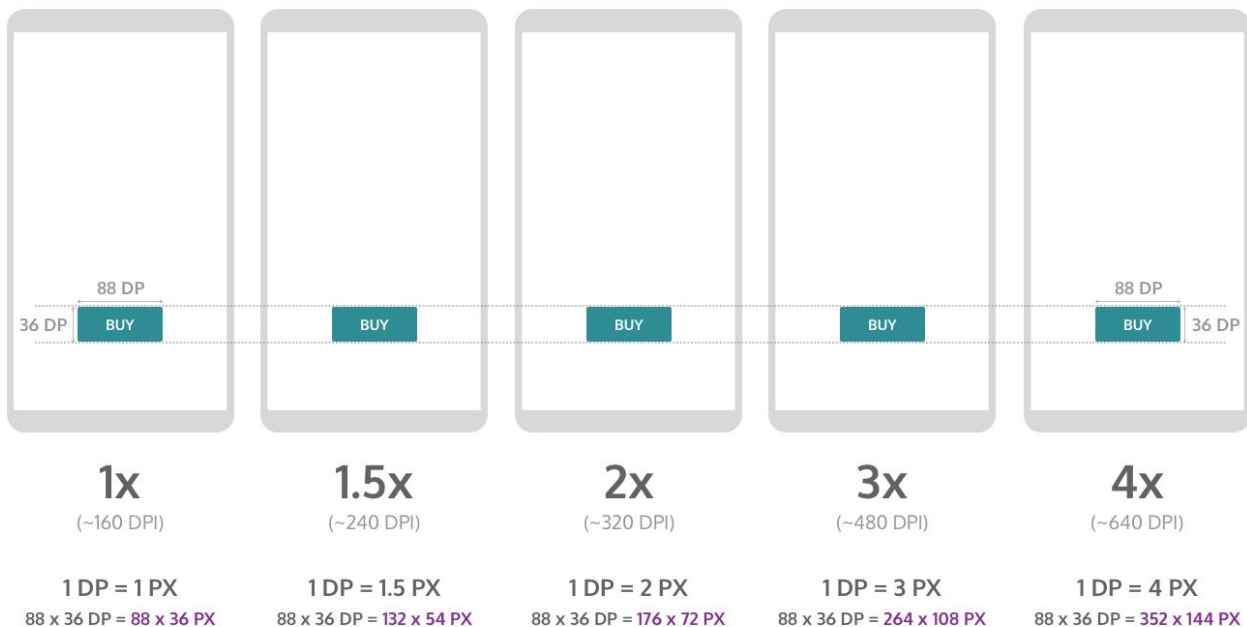
Android traduce el valor en dp a la cantidad apropiada de píxeles reales para cada densidad.

Si definieras una vista de "100 px" de ancho, parecerá mucho más grande en el dispositivo de la izquierda. Por lo tanto, debes usar "100 dp" para asegurarte de que aparezca en ambas pantallas.



DP

Density-independent-pixels (DP)



/* Importar recursos gráficos */

Importar recursos en formato rasterizado

Image Asset

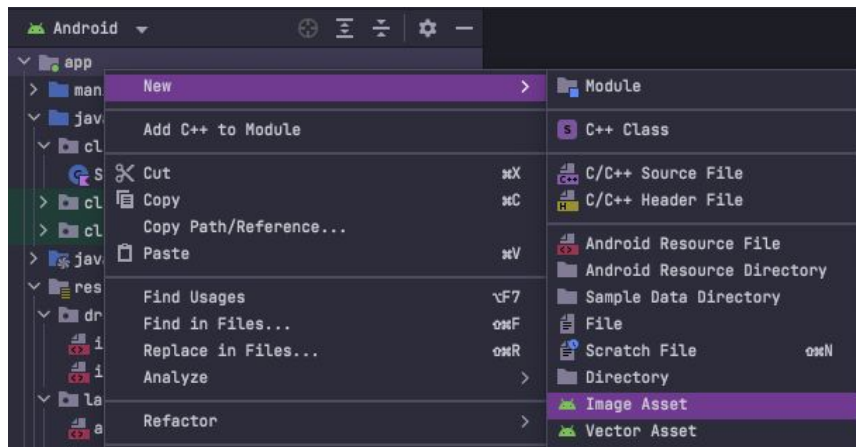
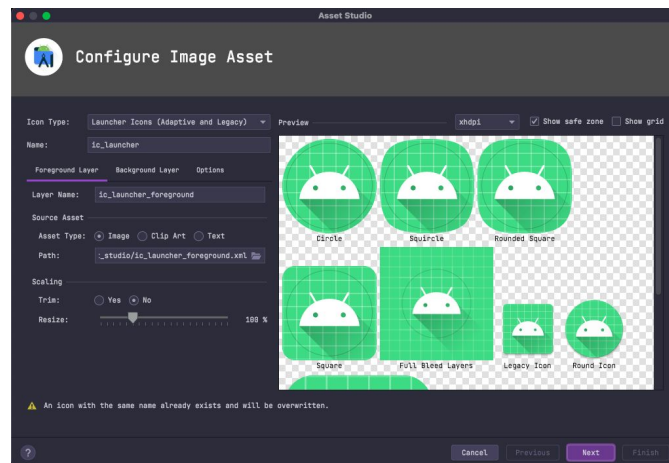


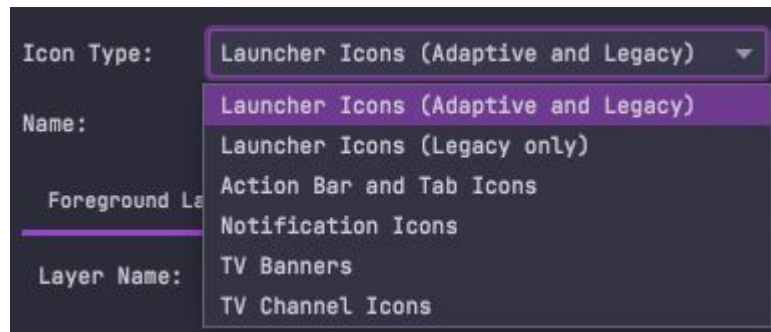
Image Asset Studio te permite **crear diferentes tipos de íconos en distintas densidades** y te muestra la ubicación exacta de tu proyecto en la que se dispondrán.



Importar recursos en formato rasterizado

Image Asset - Icon Type

- Launchers Icons es el icono de la app.
- Los iconos agregados quedan disponibles para ser utilizados en toda la app.



Importar recursos en formato rasterizado

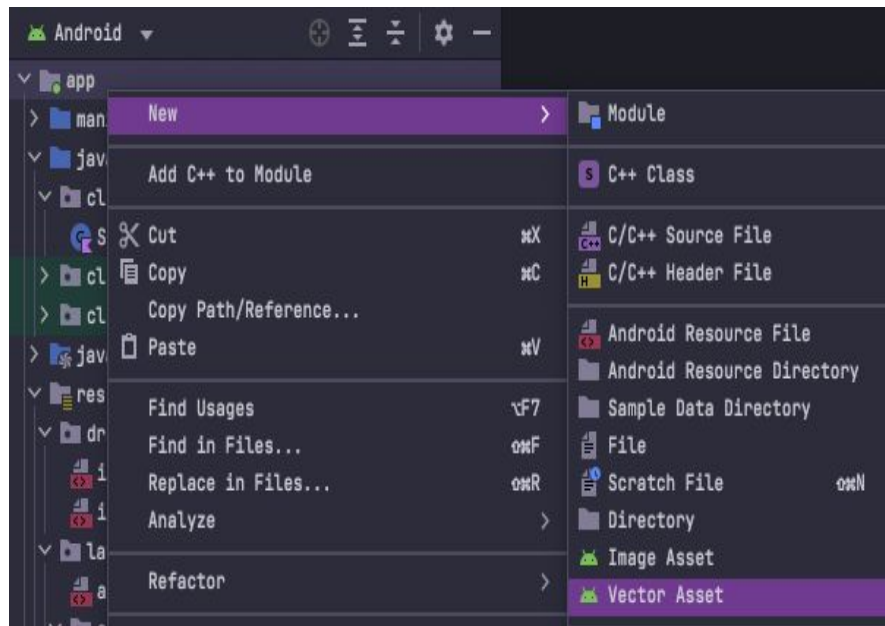
Image Asset - Asset Type



- Image: PNG (preferido), JPG (aceptable) y GIF (desaconsejado).
- Clip Art: Para especificar una imagen del [conjunto de íconos de material design](#).
- Text: Permite escribir una string de texto en diferentes fuentes y la dispone en un ícono.

Importar recursos en formato vectorizado

Vector Asset

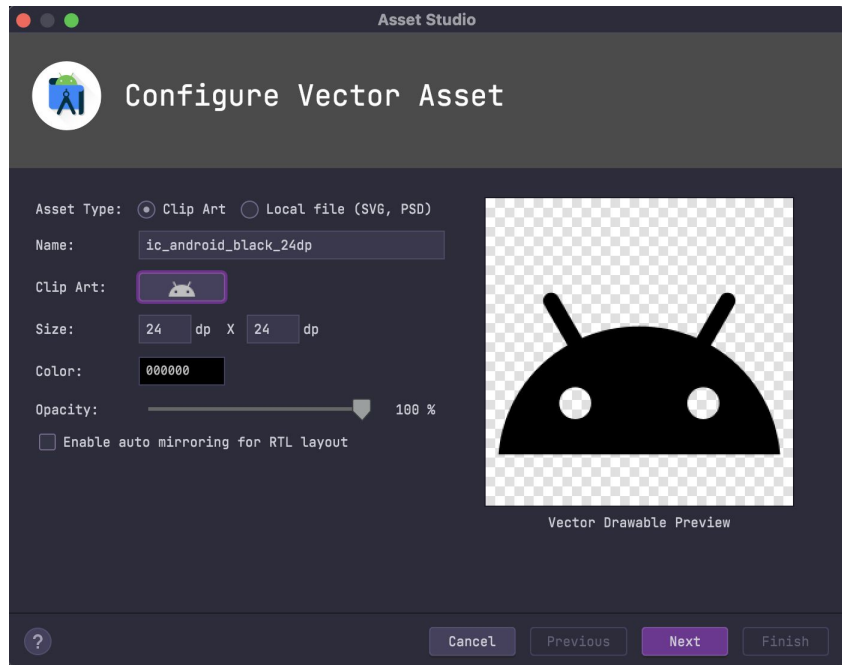


Permite agregar íconos de material e importar archivos de Gráficos vectoriales escalables (SVG) y de Adobe Photoshop Document (PSD) a tu proyecto como recursos de elementos de diseño vectoriales.

El uso de elementos de diseño vectoriales en lugar de mapas de bits reduce el tamaño de tu APK porque se puede cambiar el tamaño del mismo archivo según las diferentes densidades de pantalla sin perder la calidad de la imagen.

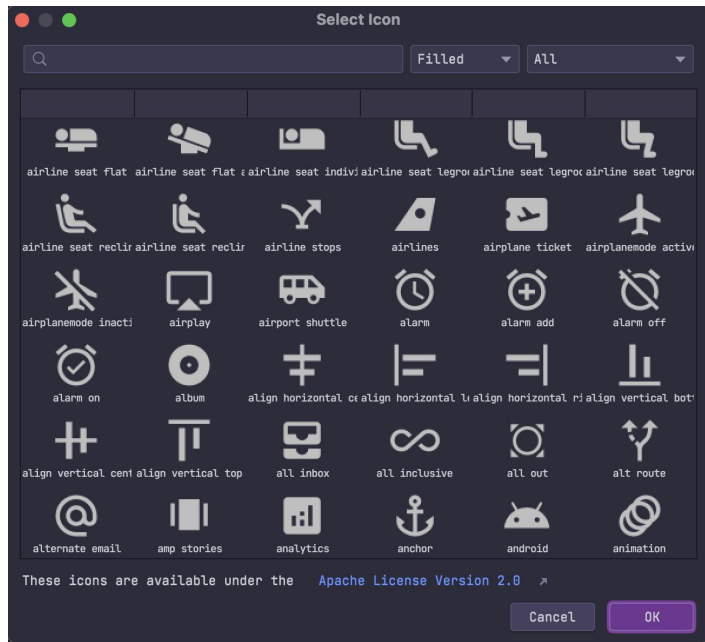
Importar recursos en formato vectorizado

Vector Asset - Asset Type



- **Clip Art:** Al igual que Image Asset, permite especificar una imagen del conjunto de íconos de material design.
- **Local file:** Gráficos vectoriales escalables (SVG) y Adobe Photoshop Document (PSD).

Conjunto de iconos de material design



Android Studio incluye el conjunto de iconos de material con una gran variedad de iconos para utilizar en la app.

Para conocer más sobre **material design** te recomendamos visitar su [sitio oficial](#).

Compatibilidad

Generación de imágenes PNG

Android 5.0 (nivel de API 21) y las versiones posteriores proporcionan compatibilidad con interfaces dibujables en vector.

- Si la app tiene un nivel mínimo de API inferior, Vector Asset Studio agrega el archivo de elemento de diseño vectorial a tu proyecto.
- Durante el tiempo de compilación, Gradle crea imágenes de trama PNG con varias resoluciones.



Densidades de pantalla en Android

Tabla 1: Calificadores de configuración para diferentes densidades de píxeles

Calificador de densidad	Descripción
ldpi	Recursos para pantallas de densidad baja (<i>ldpi</i>) (120 dpi)
mdpi	Recursos para pantallas de densidad media (<i>mdpi</i>) (~160 dpi; esta es la densidad de referencia)
hdpi	Recursos para pantallas de densidad alta (<i>hdpi</i>) (~240 dpi)
xhdpi	Recursos para pantallas de densidad muy alta (<i>xhdpi</i>) (~320 dpi)
xxhdpi	Recursos para pantallas de densidad muy, muy alta (<i>xxhdpi</i>) (~480 dpi)
xxxhdpi	Recursos para usos de densidad extremadamente alta (<i>xxxhdpi</i>) (~640 dpi)
nodpi	Recursos para todas las densidades. Estos son recursos independientes de la densidad. El sistema no escala recursos etiquetados con este calificador, independientemente de la densidad de la pantalla actual.
tvdpi	Recursos para pantallas entre mdpi y hdpi; de aproximadamente 213 dpi. Este no se considera un grupo de densidad "principal". Se usa más que nada para televisiones, y la mayoría de las apps no lo necesitarán ya que con recursos mdpi y hdpi será suficiente; el sistema los escalará según corresponda. Si crees que es necesario proporcionar recursos tvdpi, debes ajustar su tamaño con un factor de 1.33*mdpi. Por ejemplo, una imagen de 100 x 100 píxeles para pantallas mdpi debe ser de 133 x 133 píxeles para tvdpi.

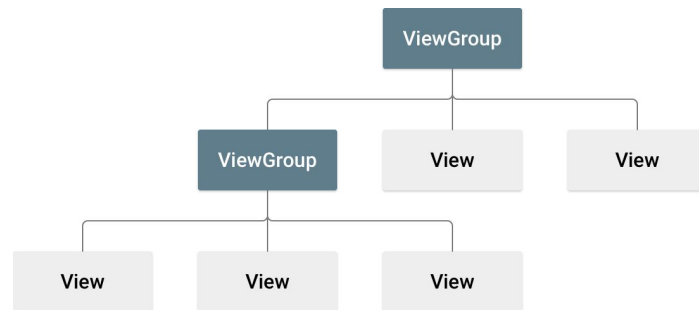
- 36 x 36 (0.75x) para densidad baja (ldpi)
- 48 x 48 (referencia de 1.0x) para densidad media (mdpi)
- 72 x 72 (1.5x) para densidad alta (hdpi)
- 96 x 96 (2.0x) para densidad muy alta (xhdpi)
- 144 x 144 (3.0x) para densidad muy, muy alta (xxhdpi)
- 192 x 192 (4.0x) para densidad extremadamente alta (xxxhdpi)

/* Introducción a las Vistas */

Vistas

¿Qué son?

- Todas las vistas en una pantalla están agrupadas en un solo árbol.
- Se pueden agregar vistas por código o especificando un árbol de vistas en al menos un archivo XML.
- En Android, existe la clase View que representa un componente visual individual o un conjunto de otras View que sirve para agruparlas: ViewGroup.
- Hay muchas subclases de View que actúan como controles, por ejemplo un botón, o que muestra textos, imágenes u otros contenidos.



Vistas

Attributes

Cada vista define una serie de atributos que se aplican a la vista. Algunos de los campos más importantes son:

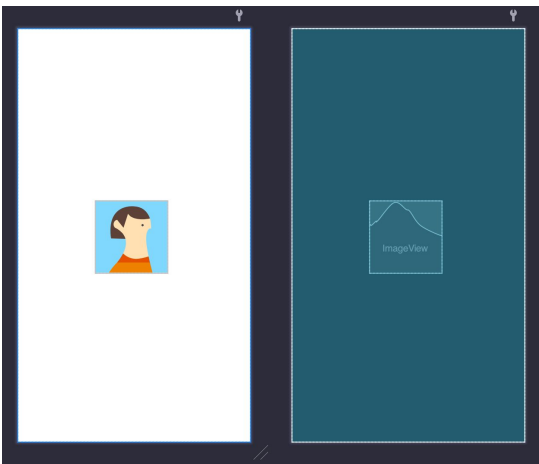
- [Recomendado] **android:id**: Define un nombre de recurso único para el elemento, que puedes usar a fin de obtener una referencia al elemento.
- [Obligatorio] **android:layout_width**: Indica el ancho (width) de la vista.
- [Obligatorio] **android:layout_height**: Indica el alto (height) de la vista.

Los valores posibles para el ancho y el alto son:

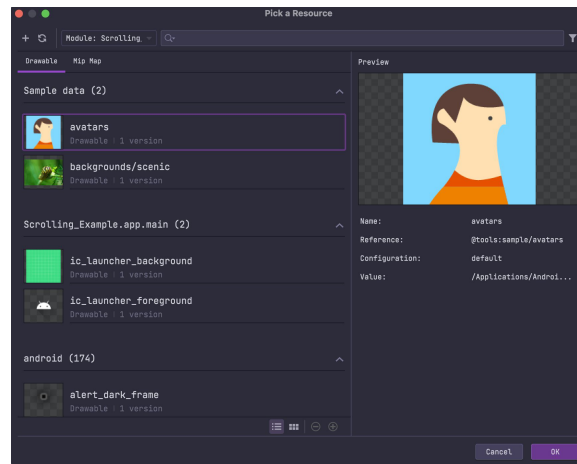
- `match_parent`: Se ajusta al tamaño del contenedor.
- `wrap_content`: Se ajusta al tamaño del contenido
- `0dp`: Delega la responsabilidad de calcular el tamaño a `ConstraintLayout`.
- `Xdp`: Tamaño en dp, por ejemplo, `100dp`

ImageView

Es un contenedor que permite mostrar una imagen, generalmente vectorial.



Tiene un selector donde se muestran los recursos disponibles para mostrar.



ImageView

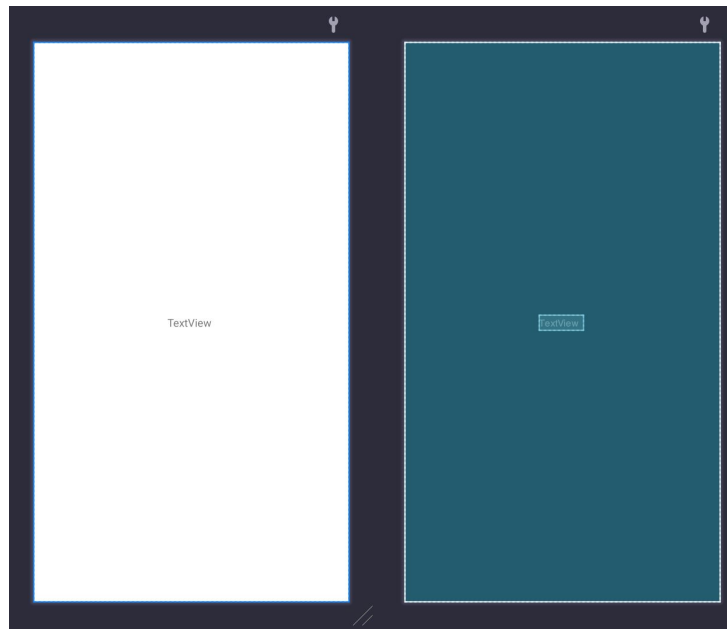
```
<ImageView
    android:id="@+id/image_view_id"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/my_image"
    android:contentDescription="@string/my_image_description"
/>
```

- **id:** Define el id de la vista para ser referenciado por otras vistas.
- **src:** La ubicación de la imagen, referenciando a los drawables.
- **contentDescription:** Utiliza internacionalización para los textos.

TextView

- Permite mostrar un texto en la pantalla.
- El texto debe estar definido en strings.xml.

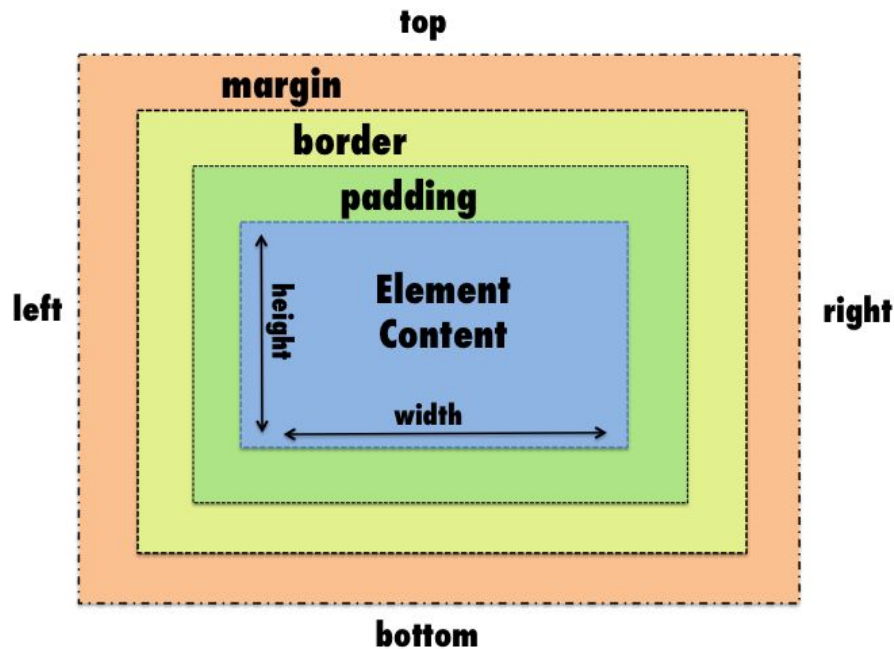
```
<TextView  
    android:id="@+id/text_view_id"  
    android:layout_height="wrap_content"  
    android:layout_width="wrap_content"  
    android:text="@string/hello" />
```



Box Model

Cada vista se maneja con un Box Model, indicando *margin* y *padding*.

Se puede asignar estos valores en forma independiente para top (arriba), bottom (abajo), right (derecha) y left (izquierda).



Box Model

Ejemplo

```
<TextView  
    android:id="@+id/textView"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_margin="16dp"  
    android:paddingTop="24dp"  
    android:paddingLeft="24dp"  
    android:textSize="48sp"  
    android:text="TextView" />
```

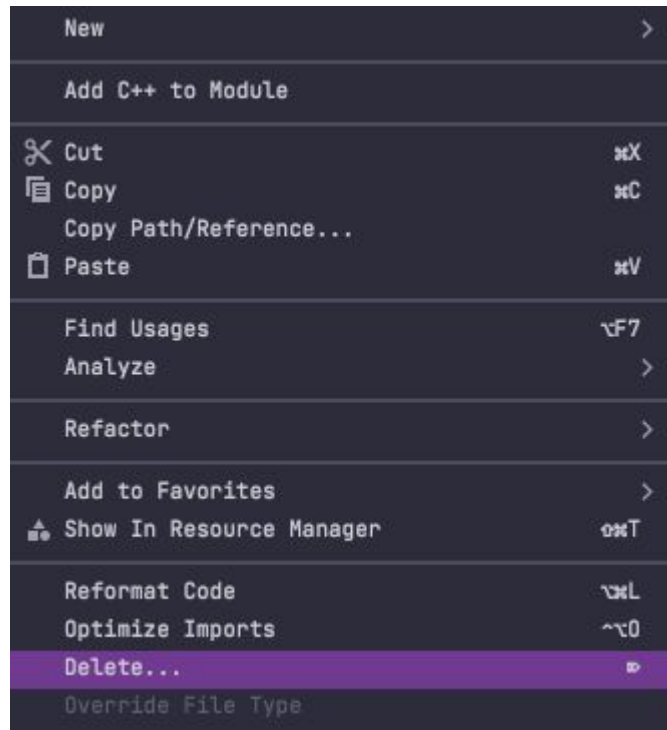


- El tamaño de la vista se ajusta al contenido.
- Se asigna padding solo a la parte superior (top) y a la izquierda (left).
- Se agrega *margin* a toda la vista.
- Se agranda el tamaño de letra para el texto.

**/* Qué pasa si se crea una carpeta o un
archivo erróneamente */**

Eliminar archivo o carpeta

Al presionar sobre cualquier archivo o carpeta, siempre aparece en el menú la opción de borrar, en inglés Delete.



¿Es lo mismo el tamaño de la pantalla que la densidad de la pantalla?





Próxima sesión...

- *Utiliza archivos de assets externos a un proyecto Android obtenidos desde Material Icons para complementar el proyecto*
- *Utiliza componentes adicionales de acuerdo a requerimientos dados*

{desafío}
latam_

*Academia de
talentos digitales*

