

PRACTICA 3: CREACIÓN DE UN LAYOUT CON CONSTRAINTLAYOUT

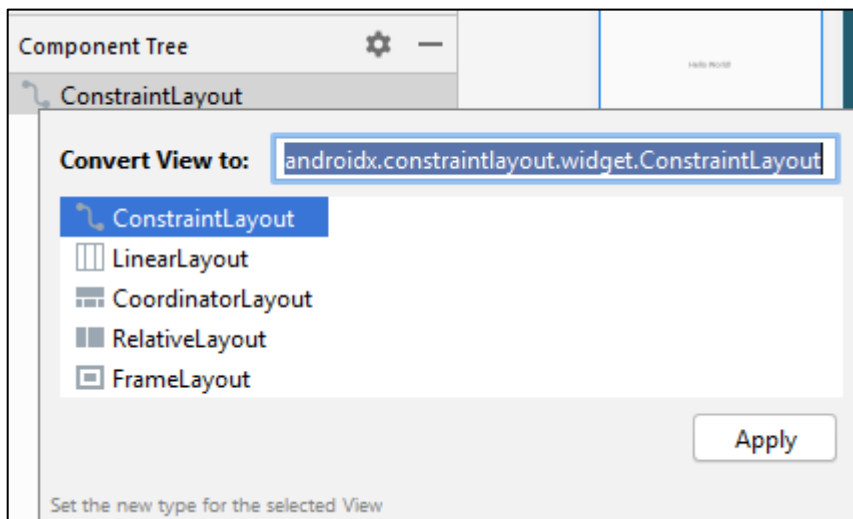
Parte I: Creación de un layout con ConstraintLayout

Paso 1. Abre el proyecto PrimerasVistas o crea uno nuevo llamado UsoConstraintLayout.

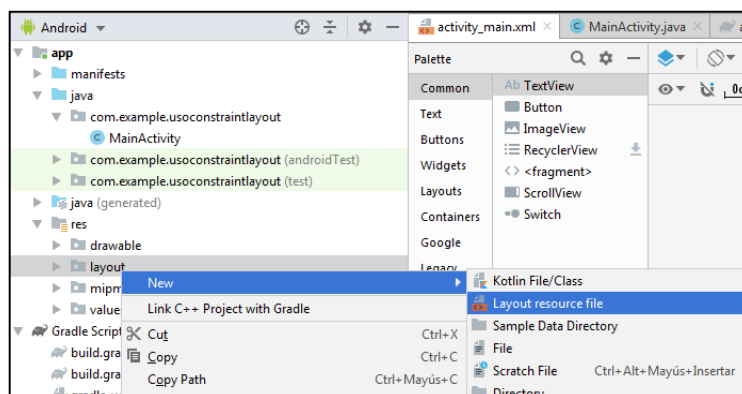
Paso 2. En Gradle Scripts/Buid.gradle (Module:app) ha de estar la dependencia:

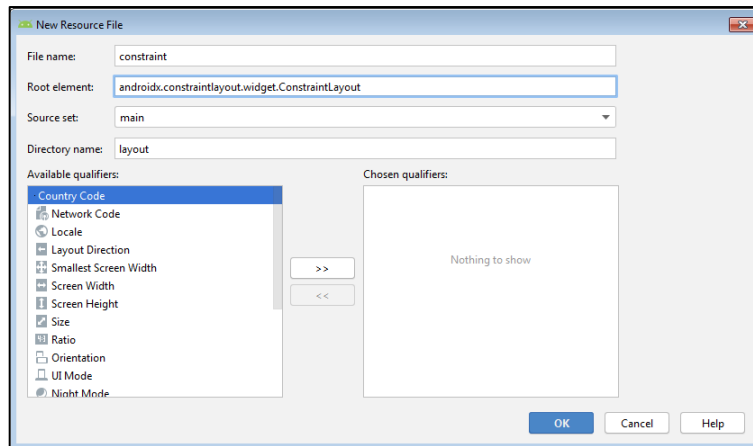
```
dependencies {  
    implementation fileTree(dir: 'libs', include: ['*.jar'])  
    implementation 'androidx.appcompat:appcompat:1.1.0'  
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'androidx.test:runner:1.2.0'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'  
}
```

Paso 3. Abre el layout activity_main.xml creado en el ejercicio anterior. Pulsa con el botón derecho en Component Tree y seleccionamos la opción Convert View to ConstraintLayout. Esta herramienta nos permite convertir nuestros viejos diseños que se basaban en LinearLayout y RelativeLayout en este nuevo tipo de layouts.

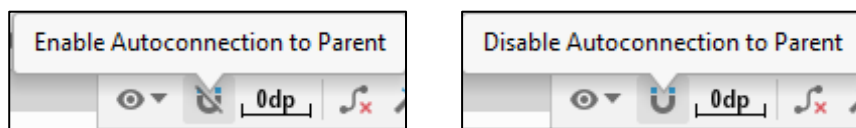


Paso 4. Crea un nuevo layout. Para ello, pulsa con el botón derecho sobre app/res/layout y selecciona New/Layout resource file. Como nombre introduce “constraint” y en Root element: androidx.constraintlayout.widget.ConstraintLayout



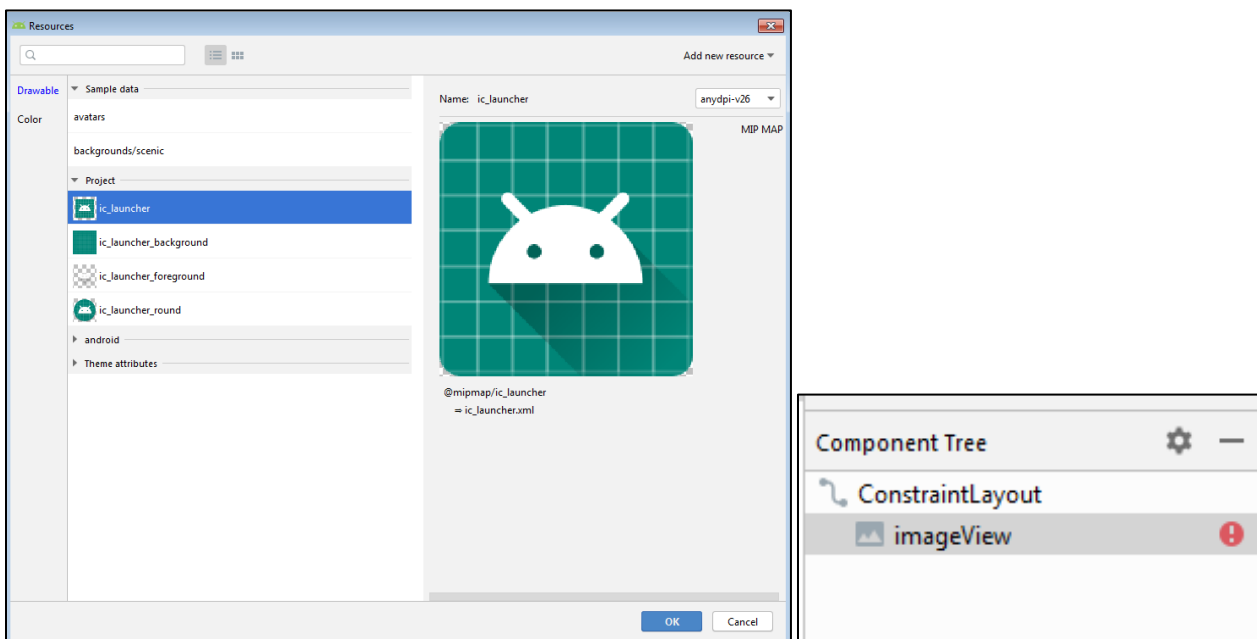


Paso 5. Vamos a desactivar la opción de Autoconnect de la barra de acciones del ConstraintLayout. Es el segundo icono con forma de imán:

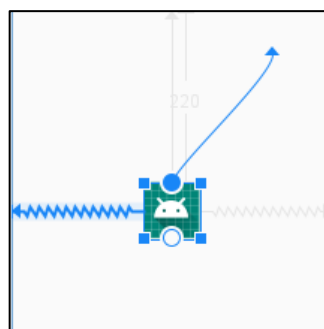


Tener activa esta opción es útil para diseñar más rápido los layouts. No obstante, a la hora de aprender a usar los constraint, es mejor ir haciéndolos de uno en uno.

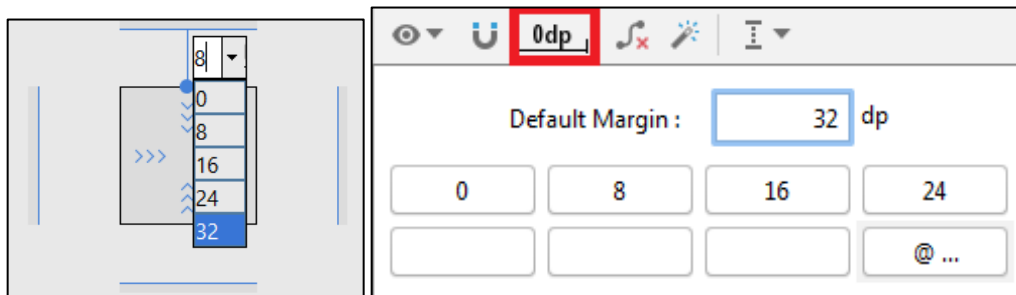
Paso 6. Dentro del área Palette, selecciona Commom y arrastra una vista de tipo ImageView al área de diseño. Se abrirá una ventana con diferentes recursos Drawable. Selecciona en Project, ic_launcher.



Paso 7. Para definir el primer constraint, pulsa sobre el punto de anclaje que aparece en la parte superior del ImageView y arrástralo hasta el borde superior del contenedor:

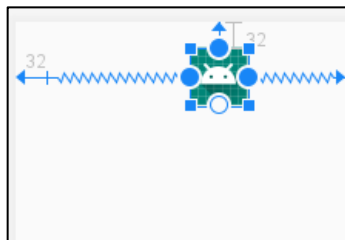


Paso 8. En la parte superior derecha nos aparece un editor visual para los constraint. Por defecto la distancia seleccionada ha sido 8dp. Pulsa sobre este número y cámbialo a 32dp:

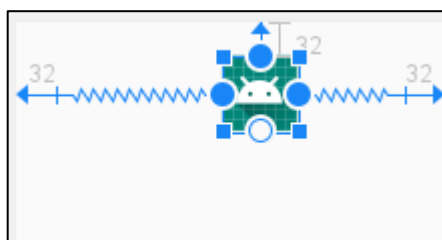


NOTA: Según las recomendaciones de Material Design los márgenes y tamaños han de ser un múltiplo de 8dp.

Paso 9. Realiza la misma operación con el punto de anclaje izquierdo, arrastrándolo al borde izquierdo. Ya tenemos la restricción horizontal y vertical por lo que la vista está perfectamente ubicada en el layout.

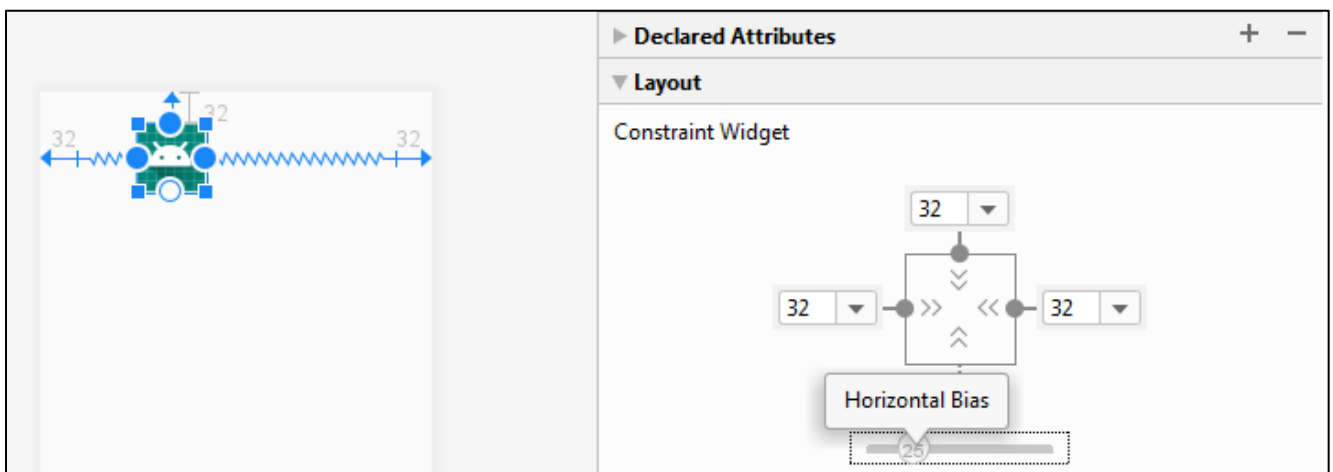


Paso 10. Arrastra el punto de anclaje derecho al borde derecho, introduciendo una distancia de 32dp:



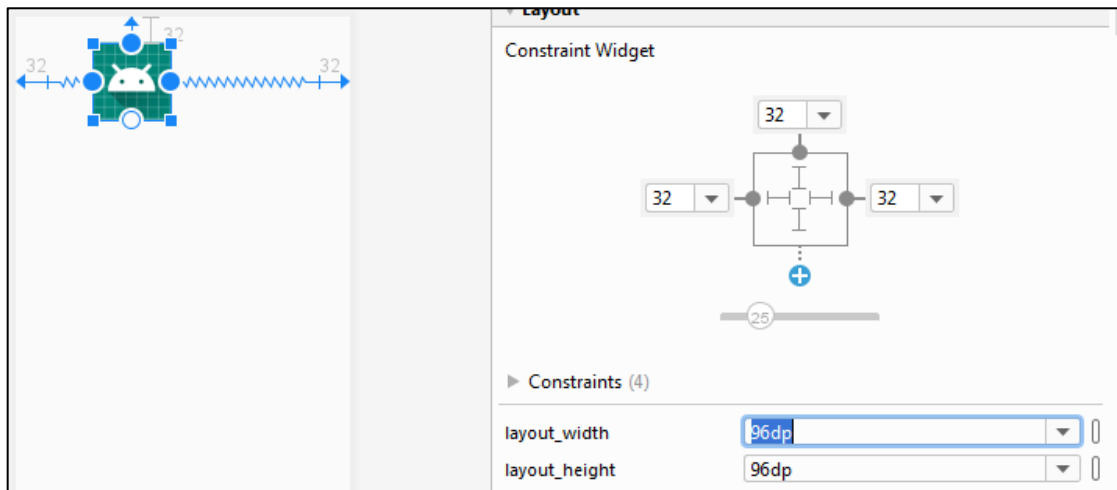
Observa como en este caso, al tener que cumplir simultáneamente dos constraint horizontales la imagen es centrada horizontalmente. Esto se representa con la línea en zigzag, representando un muelle, que estira de la vista desde los dos lados. El pequeño botón con una cruz roja que aparece, nos permite borrar todos los constraint de la vista.

Paso 11. Si en lugar de querer la imagen centrada la queremos en otra posición, podemos ir al editor de constraint y usar la barra deslizante (Horizontal Bias). Desplazarla a la posición 25.

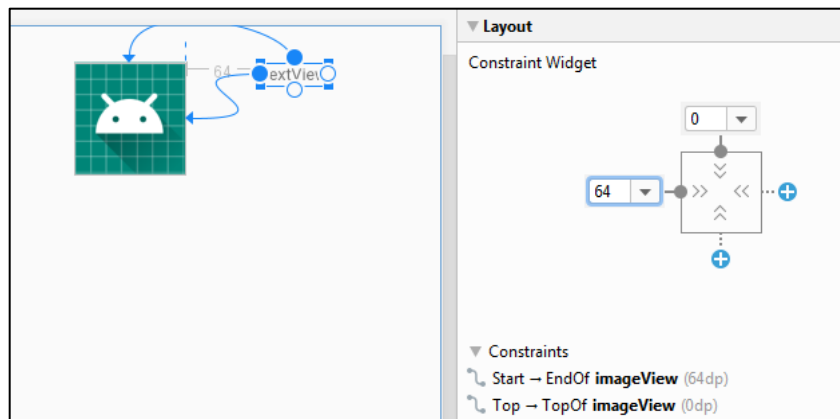


Observa en el área de trabajo, como la longitud del muelle de la izquierda es un 25%, frente al 75% del muelle de la derecha.

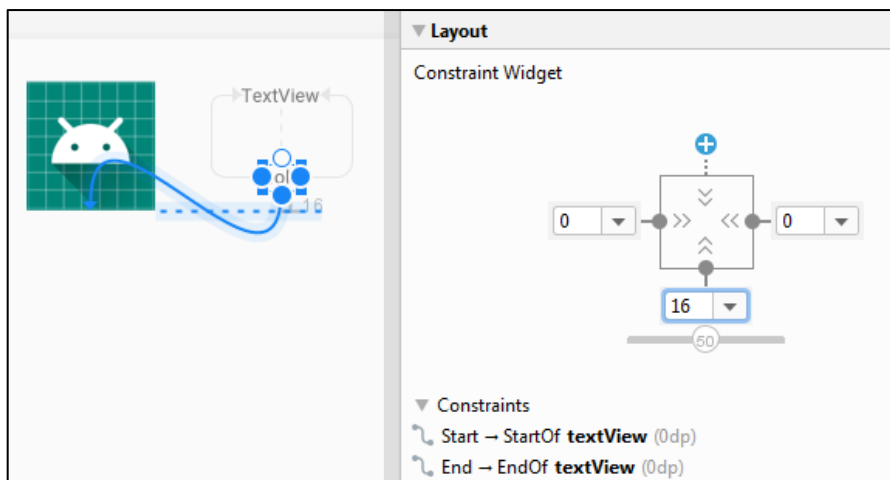
Paso 12. Seleccionando el ImageView en el área de trabajo, arrastra el cuadrado azul de la esquina inferior derecha, hasta aumentar su tamaño hasta 96x96 dp (múltiplos de 8). Otra alternativa es modificar los valores `layout_width` y `layout_height` en el área Properties.



Paso 13. Desde la paleta de widgets, añade un TextView a la derecha del ImageView. Arrastra el punto de anclaje de la izquierda hasta el punto de la derecha de la imagen y establece un margen de 64 dp. Arrastra el punto de anclaje superior del TextView hasta el punto superior de la imagen:

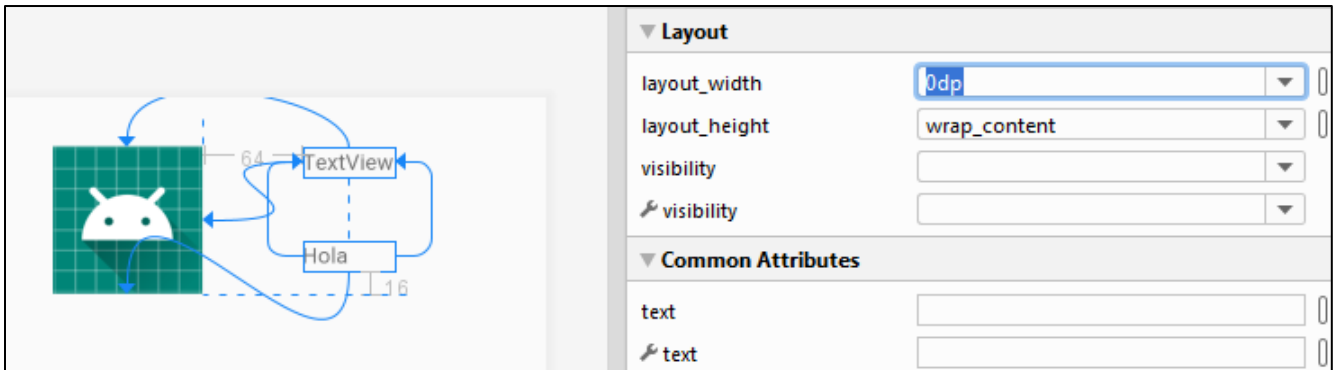


Paso 14. Añade un nuevo TextView bajo el anterior, con texto “hola”. Introduce tres constraint, usando los puntos de anclaje inferior, izquierdo y derecho, tal y como se muestra en la siguiente figura:

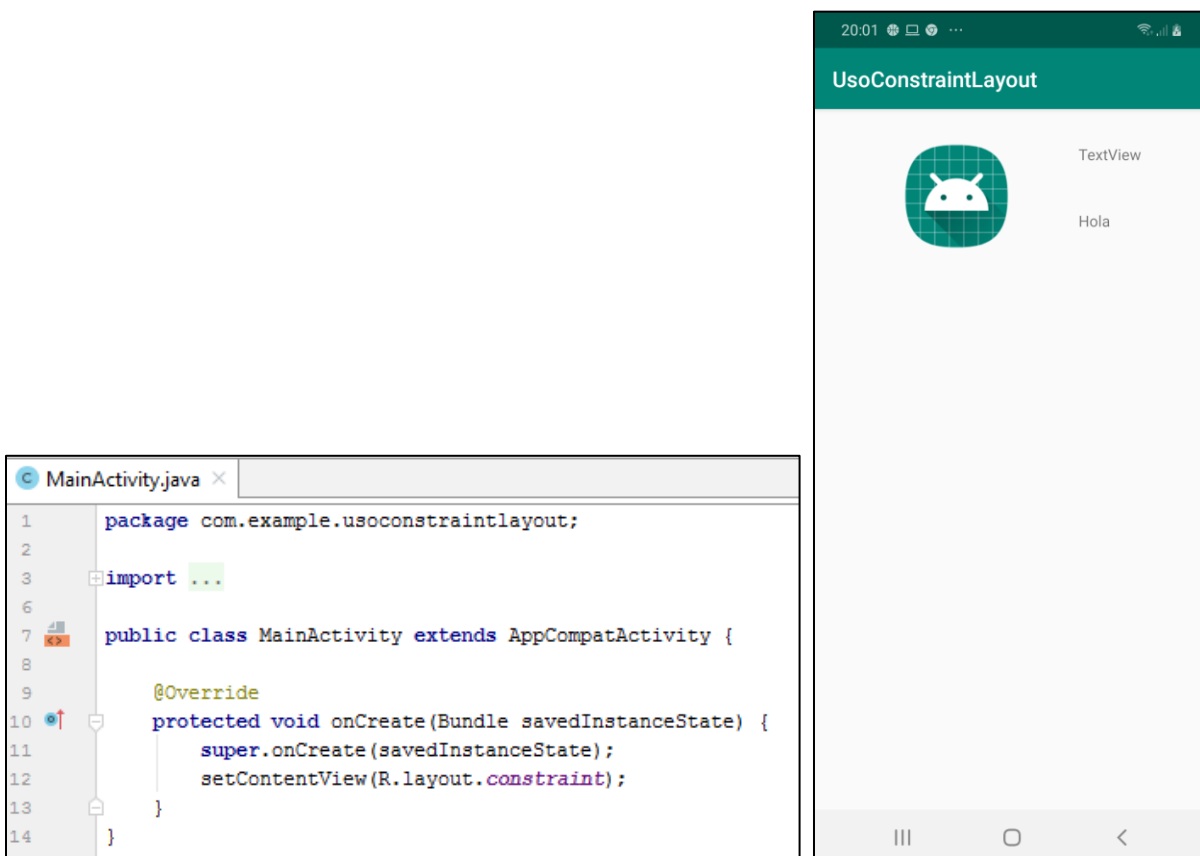


El margen inferior ha de ser 16dp y el izquierdo y derecho 0. De esta forma hemos centrado horizontalmente los dos TextView.

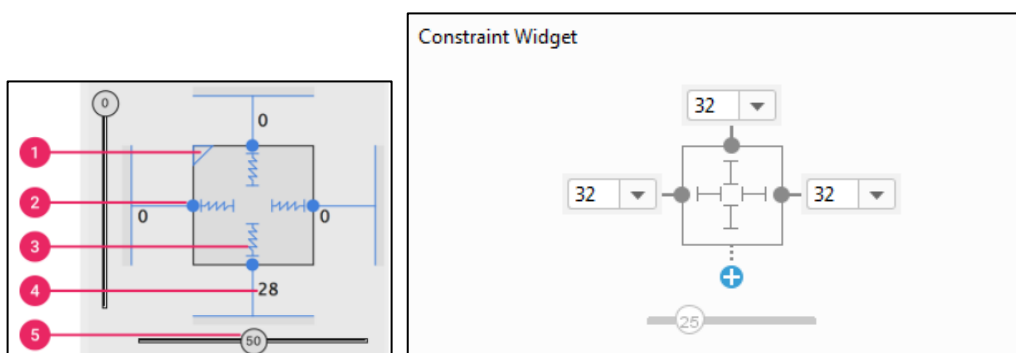
Paso 15. También podemos conseguir que el ancho del nuevo TextView coincida con el superior. Para ello selecciona la vista y en el campo `layout_width` introduce `match_constraint` ó `0dp`. Con esto, hacemos que el ancho se calcule según las restricciones de los constraint. Quita los márgenes laterales para que los anchos de las dos vistas coincidan.



Paso 16. Haz que desde MainActivity se visualice este layout y ejecuta el proyecto en un dispositivo.




Una vez familiarizados con los conceptos básicos de los constraint vamos a ver con más detalle las herramientas disponibles. Veamos en editor de constraint:




(1) relación de tamaño: Puede establecer el tamaño de la vista en una proporción, por ejemplo 16:9, si al menos una de las dimensiones de la vista está configurada como "ajustar a constraint" (0dp). Para activar la relación de tamaño, haz clic donde señala el número 1.

(2) eliminar constraint: Se elimina la restricción para este punto de anclaje.

(3) establecer alto/ancho: Para cambiar la forma en la que se calcula las dimensiones de la vista, pulsa en este elemento. Existen tres posibilidades:

 ajustar a contenido: equivale al valor `warp_content`. (Ej. 1er TextView)

 ajustar a constraint: equivale a poner 0dp. (Ej. 2 TextView)

 tamaño fijo: equivale a poner un valor concreto de dp. (Ej. ImageView)


Aunque se representan 4 segmentos, realmente podemos cambiar 2, los horizontales para el ancho y los verticales al alto.


(4) establecer margen: Podemos cambiar los márgenes de la vista.

(5) Sesgo del constraint: Ajustamos como se reparte la dimensión sobrante.


También es importante repasar las acciones disponibles cuando trabajamos con `ConstraintLayout`:




 Ocultar constraint: Elimina las marcas que muestra las restricciones y los márgenes existentes.

 Autoconectar: Al añadir una nueva vista se establecen unos constraint con elementos cercanos de forma automática.

 Definir márgenes por defecto

 Borrar todos los constraint: Se eliminan todas las restricciones del layout.

 Crear automáticamente constraint: Dada una vista seleccionada, se establecen unos constraint con elementos cercanos de forma automática.

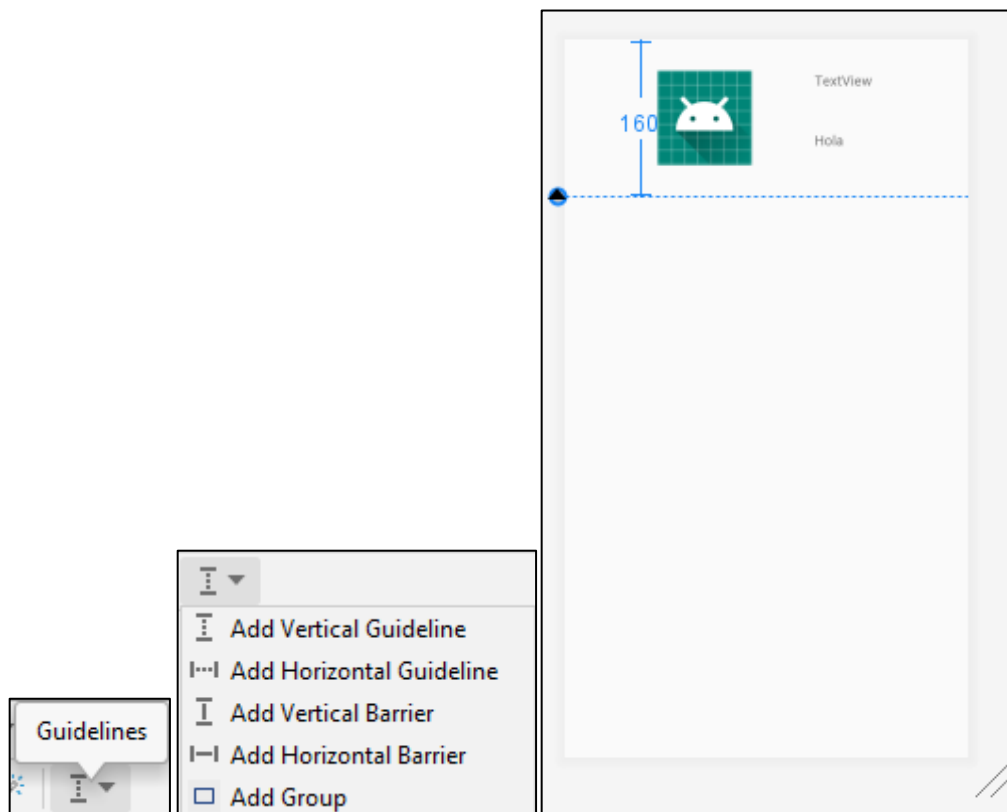
 Empaquetar / expandir: Se agrupan o se separan los elementos.

 Alinear: Centra o justifica los elementos seleccionados.

 Añadir línea de guía: Se crea una nueva línea de referencia.

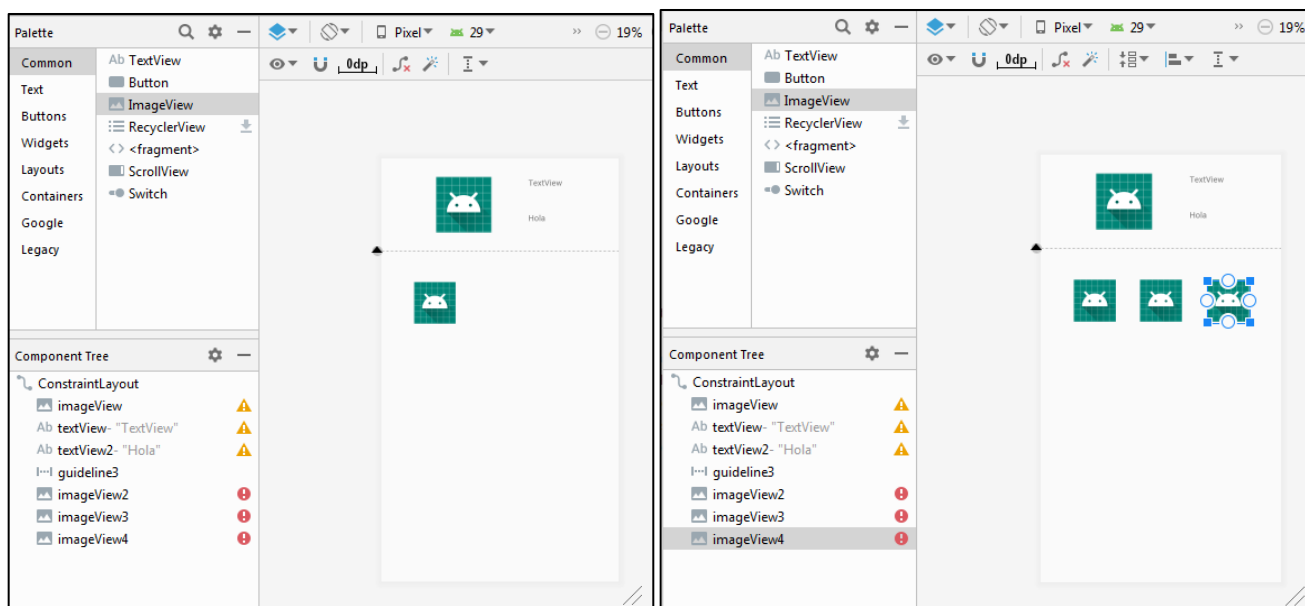
Parte II: Líneas guía y cadenas en ConstraintLayout


Paso 1. Siguiendo con el Layout del ejercicio anterior. Pulsa en la acción Guideline, y selecciona Add Horizontal GuideLine. Aparecerá un círculo gris con un pequeño triángulo pegado al borde izquierdo, desde donde sale una línea guía horizontal. Arrástrala hacia abajo, hasta separarla una distancia de 160dp.

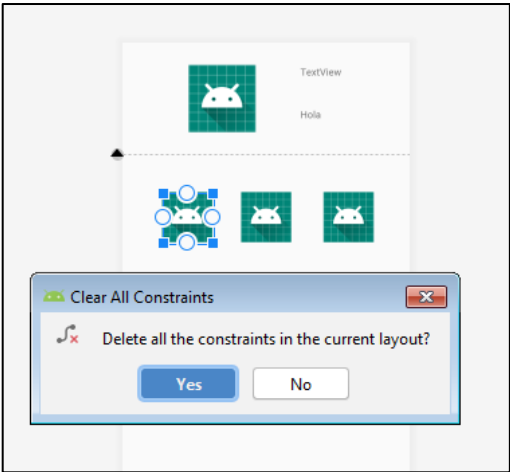


Paso 2. Esta guía nos permite dividir el layout en dos áreas, la superior donde ya hemos realizado una especie de cabecera y la inferior. A partir de ahora los elementos de la parte inferior los colocaremos en relación a esta línea guía.

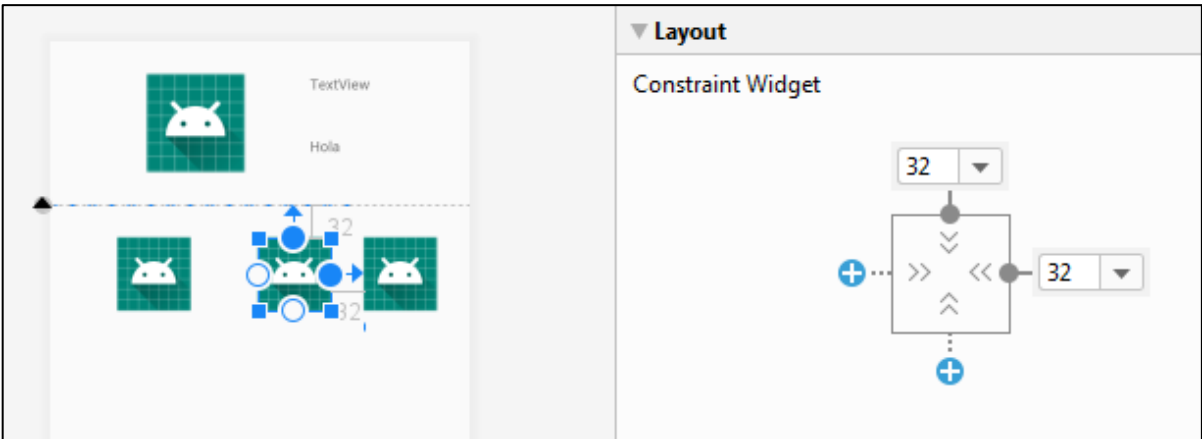
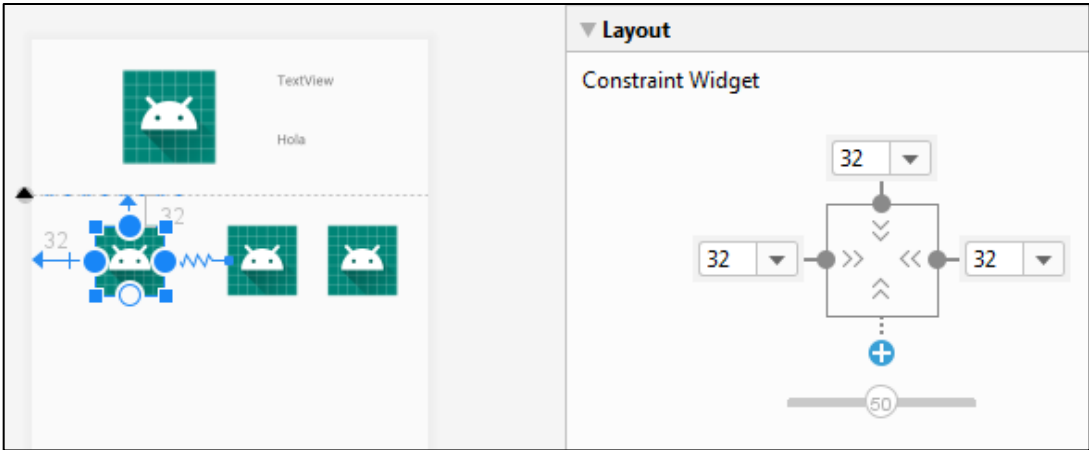
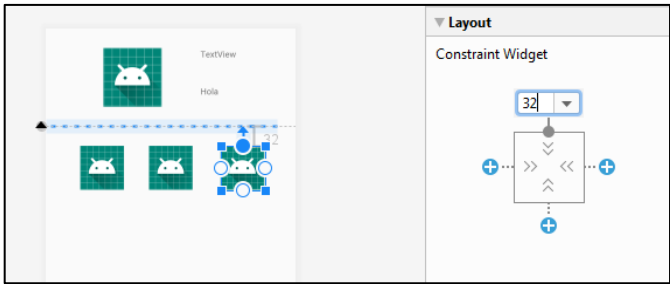
Paso 3. Selecciona el ImageView, cópialo (Ctrl+C) y pégalo tres veces (Ctrl+V). Acabamos de hacer tres copias de la imagen, pero no son visibles al estar en la misma posición. En el área Component Tree, selecciona imageView2 y arrastrarlo bajo la línea guía, pegado a la izquierda. Coloca imageView3 bajo la línea guía, en el centro e imageView4, a la derecha de este

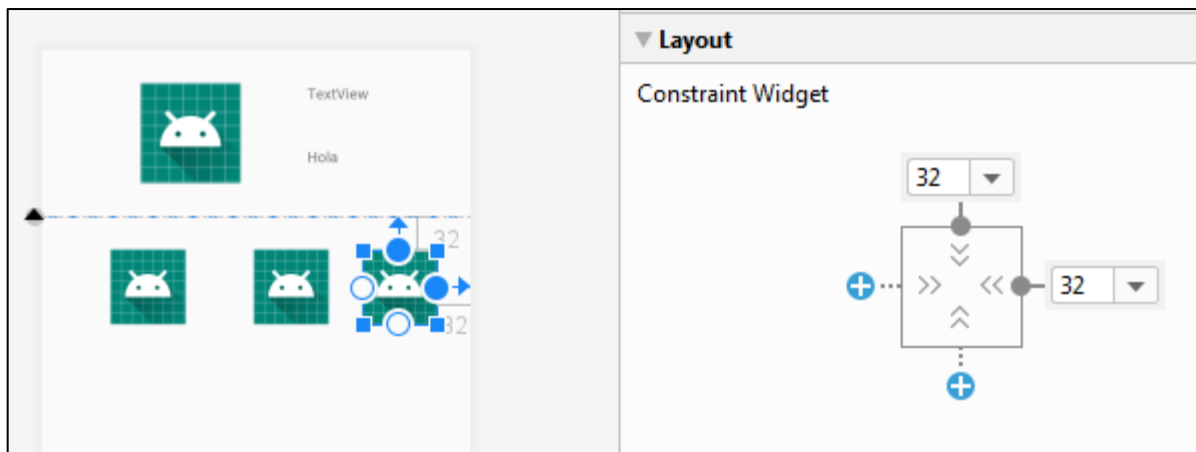



Paso 4. Selecciona imageView2, y borra todos sus constraint pulsando en el botón  que aparece bajo la vista. Repite esta operación para imageView3 e imageView4.

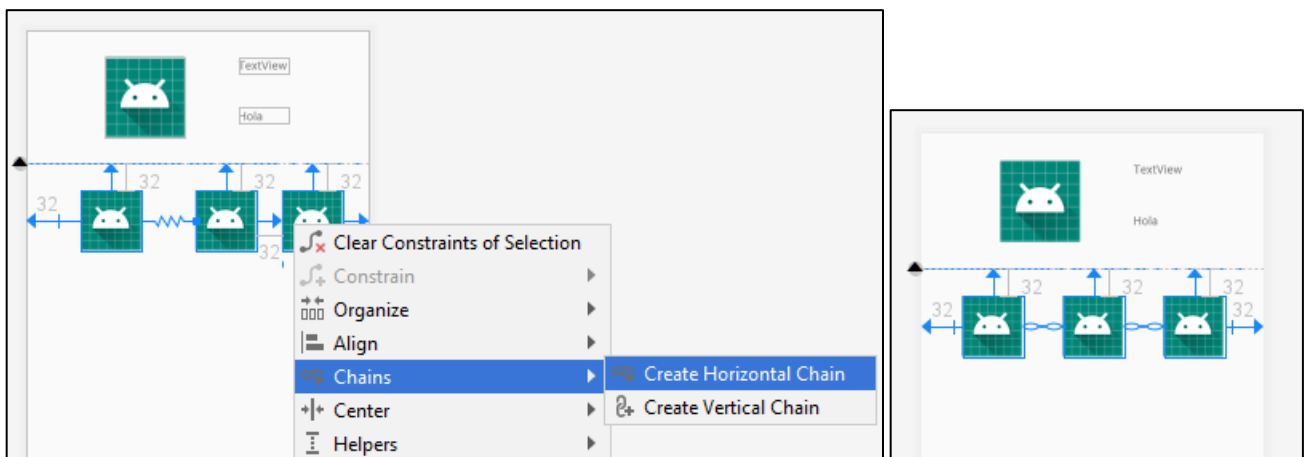


Paso 5. Las tres imágenes han de tener un constraint desde el punto de anclaje superior a la línea guía con un margen de 32dp. Desde el punto de anclaje izquierdo de imageView2, establece un constraint con el borde izquierdo y en las otras dos imágenes, al punto de anclaje derecho de la vista de la derecha. El punto de anclaje derecho de la tercera vista únelo al borde derecho. El resultado ha de ser el siguiente:






Paso 6. Para conseguir que estas tres vistas formen una cadena, selecciónalas y utiliza la acción Align  / Horizontally o el botón derecho Chains / Create Horizontal Chain:



Observa cómo las vistas ahora están unidas por medio de un conector de cadena y aparece un icono con dos eslabones en la parte inferior.

Si abres la lengüeta Text, para estudiar el XML, puedes comprobar que para establecer la cadena se han añadido dos constraint, desde el punto de anclaje derecho de las dos primeras vista hacia la vista de su izquierda. Es decir, una restricción mutua, de $A \rightarrow B$ y de $B \rightarrow A$.

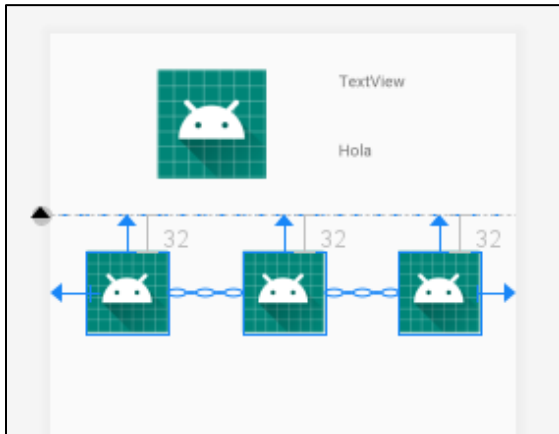
Nota: En la versión actual del editor visual, parece que establecer la cadena indicando estos constraint individualmente no parece posible. Hay que usar la acción Align  / Center Horizontally o Chains/Create Horizontal Chain.

Paso 7. También es posible otras distribuciones de cadena, pulsando sobre cualquier botón con dos eslabones (se muestra en la figura anterior). Podemos hacer que los márgenes se distribuyan solo entre las vistas o solo en los extremos izquierdo y derecho:



Si abres la lengüeta Text para estudiar el XML puedes comprobar que estas dos nuevas configuraciones de cadena se consiguen añadiendo en la primera distribución vista el atributo:

`app:layout_constraintHorizontal_chainStyle="spread_inside"` para la primera distribución:

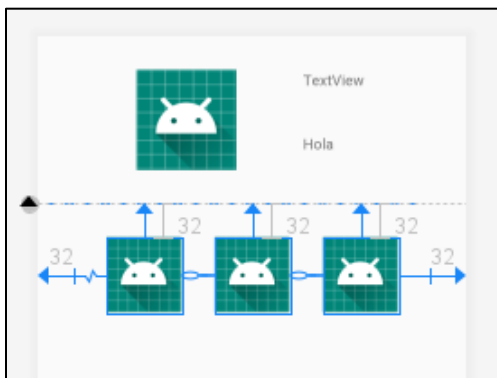


```
<ImageView
    android:id="@+id/imageView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:layout_marginEnd="32dp"
    app:layout_constraintEnd_toStartOf="@+id/imageView4"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toEndOf="@+id/imageView2"
    app:layout_constraintTop_toTopOf="@+id/guideline3"
    app:layout_constraintHorizontal_chainStyle="spread inside"
    app:srcCompat="@mipmap/ic_launcher" />

<ImageView
    android:id="@+id/imageView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginTop="32dp"
    android:layout_marginEnd="32dp"
    app:layout_constraintEnd_toStartOf="@+id/imageView3"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline3"
    app:layout_constraintHorizontal_chainStyle="spread inside"
    app:srcCompat="@mipmap/ic_launcher" />
```

Y para la segunda distribución:

app:layout_constraintHorizontal_chainStyle="packed"





```
<ImageView
    android:id="@+id/imageView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginTop="32dp"
    android:layout_marginEnd="32dp"
    app:layout_constraintEnd_toStartOf="@+id/imageView3"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/guideline3"
    app:layout_constraintHorizontal_chainStyle="packed"
    app:srcCompat="@mipmap/ic_launcher" />

<ImageView
    android:id="@+id/imageView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:layout_marginEnd="32dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toEndOf="@+id/imageView3"
    app:layout_constraintTop_toTopOf="@+id/guideline3"
    app:layout_constraintHorizontal_chainStyle="packed"
    app:srcCompat="@mipmap/ic_launcher" />
```

Para la distribución del punto anterior el valor es "spread" o no indicar nada.

Paso 8. Existe otra distribución en la que los márgenes desaparecen y se ajusta el ancho de las vistas hasta cubrir todo el espacio disponible. Selecciona la vista central y en el editor de constraint pulsa

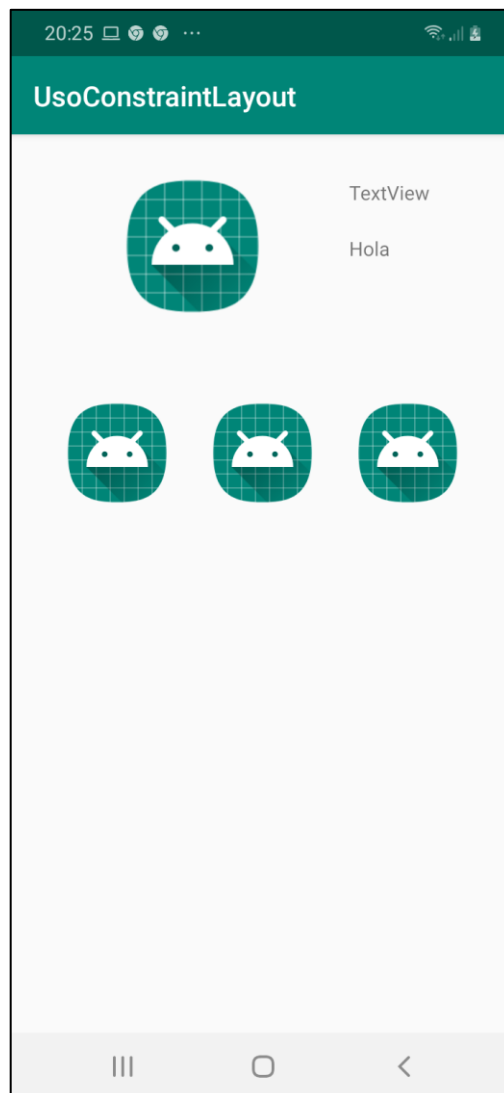
sobre el icono  hasta que aparezca . Recuerda que esta acción es equivalente a poner 0 en `layout_width`. El resultado se muestra a la izquierda:



Para conseguir el resultado de la derecha, hemos repetido la operación con las otras dos imágenes.

Si en lugar de repartir los anchos por igual, quieres otra configuración, puedes usar el atributo `layout_constraintHorizontal_weight` (funciona igual que `layout_weight` en un `LinearLayout`)

Paso 9. Ejecuta el proyecto en un dispositivo.



Al crear constraint, recuerde las siguientes reglas:

- Cada vista debe tener al menos dos restricciones: una horizontal y otra vertical.
- Sólo puede crear restricciones que compartan el mismo plano. Así, el plano vertical (los lados izquierdo y derecho) de una vista puede anclarse sólo a otro plano vertical.
- Cada manejador de restricción puede utilizarse para una sola restricción, pero puede crear varias restricciones (desde vistas diferentes) al mismo punto de anclaje.