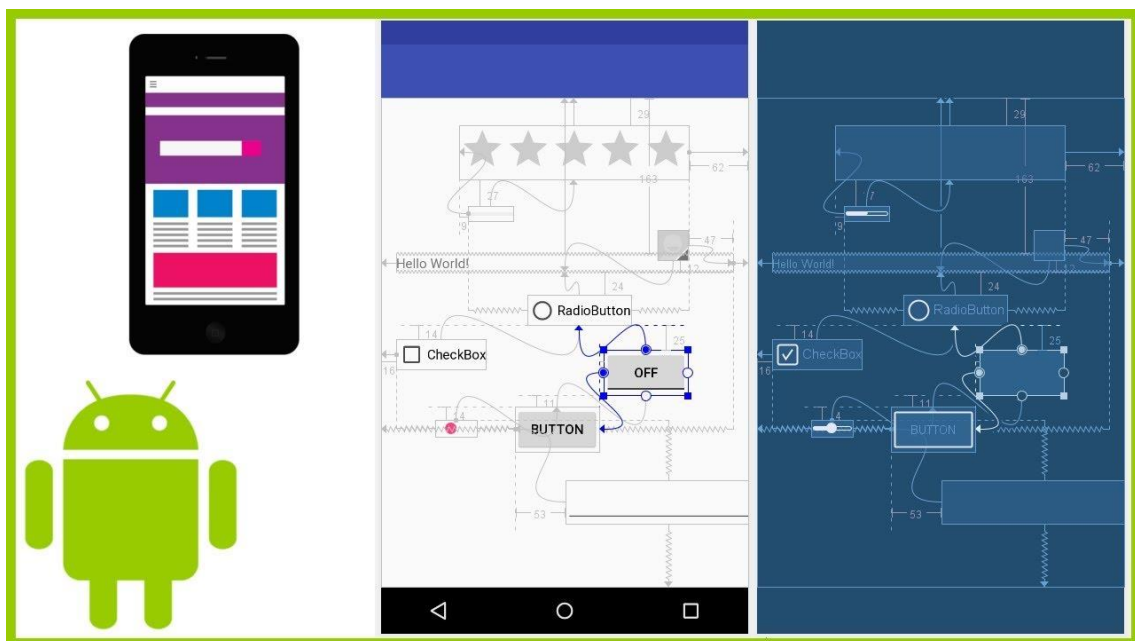
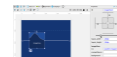


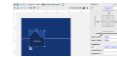
# Constraint Layout





## Contenidos

Introducción .....	3
Conociendo ConstraintLayout.....	4
Menú de opciones y propiedades de las vistas.....	5
Restricción.....	5
Sesgo .....	6
Tamaño.....	7
Alineación de texto .....	7
Alinear vistas .....	8
Distribución equitativa .....	9
Directrices o líneas guías.....	10
Barreras .....	10
Creación automática de restricciones.....	11
Otras opciones .....	12
Eliminar restricciones.....	12
Atributos XML.....	12
Prácticas propuestas .....	12



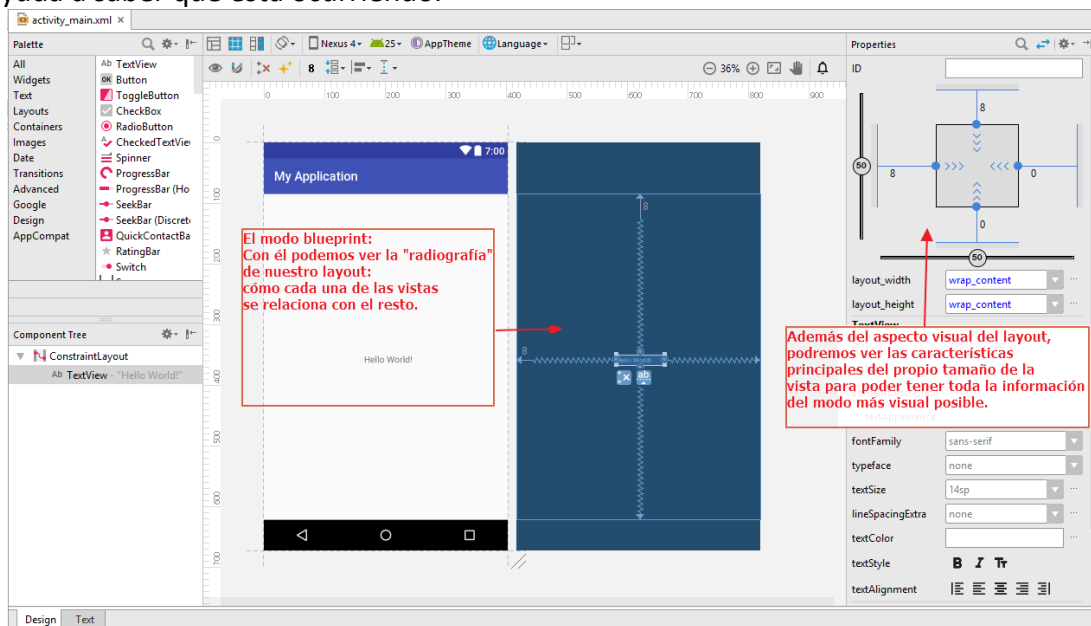
## Introducción

*ConstraintLayout* es una de las últimas novedades añadidas para diseñar la interfaz gráfica de usuario. En las últimas versiones de AS es el layout por defecto.

Similar a *RelativeLayout* pero mucho más potente y versátil, agiliza el diseño y mejora el rendimiento en tiempo de ejecución (ya que permite crear interfaces de usuario complejas sin necesidad de anidar múltiples *ViewGroup*).

El *ConstraintLayout* basa su funcionamiento en "constraints" (restricciones) que anclan unas vistas a otras, a posiciones de la pantalla, etc. En este layout la posición de las vistas será relativa a otra de ellas o al propio layout, de manera que se ajusten correctamente independientemente del tamaño de la pantalla del dispositivo.

Aquí es donde el modo *blueprint* (la zona "azul" del editor de layouts o plano) nos ayuda a saber qué está ocurriendo.

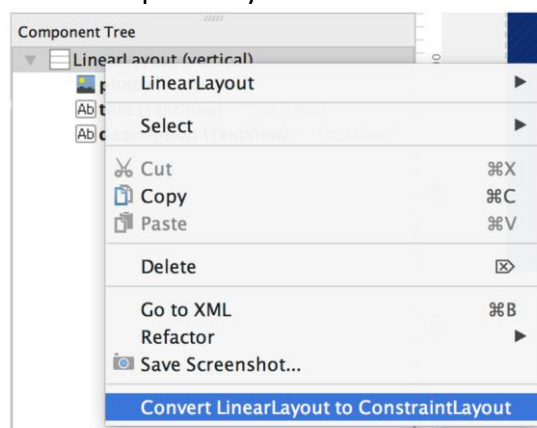


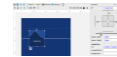
Además, el editor puede calcular de forma inteligente algunas de estas restricciones.

*ConstraintLayout* está disponible desde Android 2.3. Como suele ocurrir, Google ha decidido incluir este layout en la librería de soporte para ponerlo disponible para todos en versiones anteriores a Android Nougat. Concretamente está disponible desde la API 9 (Android 2.3) siempre y cuando incluyamos el paquete (AS ya lo hace por nosotros):

implementation **'androidx.constraintlayout:constraintlayout:1.1.3'**

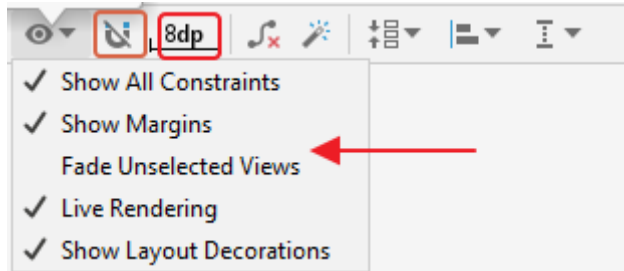
Además, para promover el uso de este nuevo layout, Google nos ofrece un modo automático para convertir otro tipo de layout a un *ConstraintLayout*:





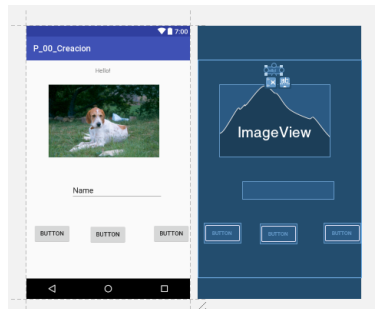
## Conociendo ConstraintLayout

Trabaja con el primer proyecto P\_00\_Creacion, en el layout activity\_main.xml y borra el TextView que había creado el asistente de AS al escoger el tipo de plantilla Empty. Para comprender el concepto de restricción comprueba que las características del menú de opciones están de la siguiente manera:

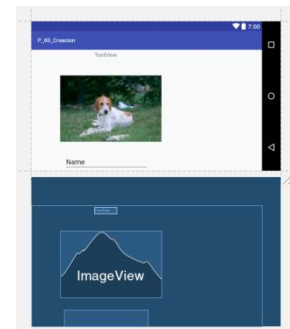


Es decir, hemos decidido que nos muestre todas las restricciones en el desplegable, hemos deshabilitado la herramienta de creación automática de restricciones y hemos cambiado la medida por defecto de 0 a 8.

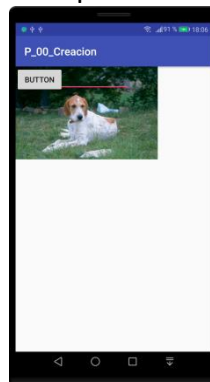
Añade unas cuantas vistas (TextView, ImageView, EditText y 3 botones por ejemplo) sin restricciones:

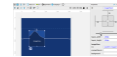


Este diseño de UI que "tanto trabajo" nos ha costado que quedase "bonito" en cuanto giramos la pantalla se nos "descuadra", ya no vemos todos los elementos en la pantalla porque no están fijadas las restricciones (distancias al borde de la pantalla o a otras vistas, etc.):



Es más, si ejecutamos la aplicación en un dispositivo, las vistas al no tener restricciones aparecen todas en la esquina superior izquierda:





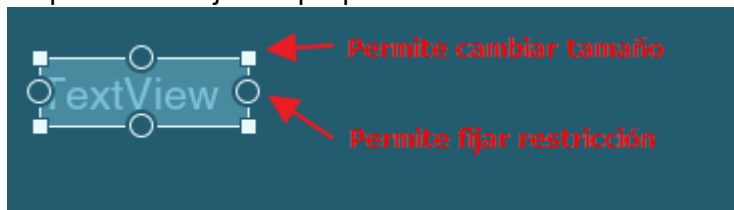
## Menú de opciones y propiedades de las vistas

Observa el menú de opciones que tenemos en el editor de layout cuando trabajamos con ConstraintLayout:



Iremos viendo todas las propiedades, pero ahora nos centramos en el "8dp". Indica el tamaño por defecto que tendrán nuestras restricciones. La guía de diseño Material Design recomienda que, por ejemplo, para márgenes sea 8dp, pero obviamente podremos cambiar cuando nos interese que tenga otro valor.

Empecemos a fijar las propiedades del TextView:



## Restricción

Cada restricción de una vista representa una conexión o alineación o posicionamiento a otra vista del layout, al propio layout o una guía invisible. Cada restricción define la posición de la vista a lo largo ya sea el eje vertical o el horizontal.

Al crear restricciones, hay que recordar las siguientes reglas:

- Cada vista debe tener al menos dos restricciones: una horizontal y otra vertical (observa que ya nos está advirtiendo de que hay un montón de errores ya que no hemos fijado ninguna), aunque a menudo son necesarias más.
- Pueden crearse restricciones sólo entre puntos de restricción que comparten el mismo plano. Por lo que un plano vertical (los lados izquierdo y derecho) de un punto de vista puede ser limitado solamente a otro plano vertical; y las líneas de base pueden limitar sólo a otras líneas de base.
- Cada punto de restricción puede ser utilizado para fijar una sola restricción, pero se pueden crear múltiples limitaciones (desde diferentes puntos de vista) para el mismo punto de anclaje.

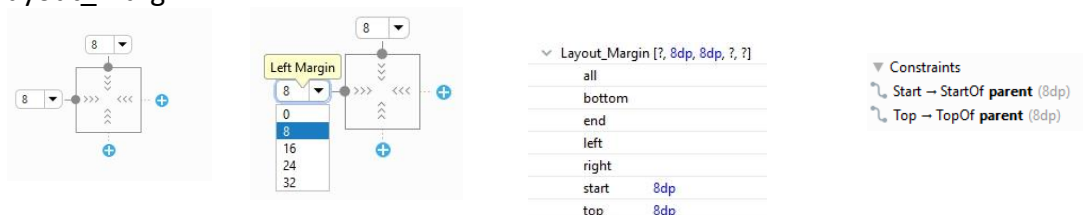
Añadimos una restricción al TextView para que esté a 8 dp del margen superior del layout y a 8dp del margen izquierdo:

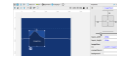
- sitúa el ratón en el círculo superior, observa que se pone "rellena" y arrastra el círculo superior hasta el borde superior del layout



- haremos lo mismo con el círculo de la izquierda hasta el borde izquierdo del layout

Observa que dichos valores quedan reflejados en el panel de propiedades y que pueden cambiarse pulsando sobre ellos y que fijan los valores de las propiedades Layout\_Margin:





## Sesgo

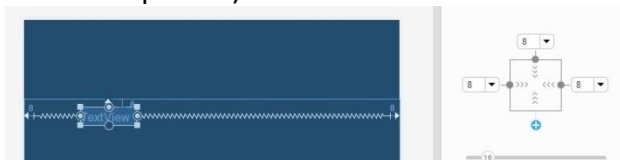
Añadimos también restricciones a la derecha:



Nos aparece un "sesgo" en horizontal (si también hubiésemos añadido restricción hacia abajo nos hubiese salido también un sesgo en vertical).

A menos que el TextView mida lo mismo que el layout (menos 16dp), es imposible que puedan cumplirse ambas restricciones. Se puede pensar en el número sesgo como la representación de la fuerza de un resorte que está tirando de la vista seleccionada en ese momento a la parte superior o el derecho de la disposición.

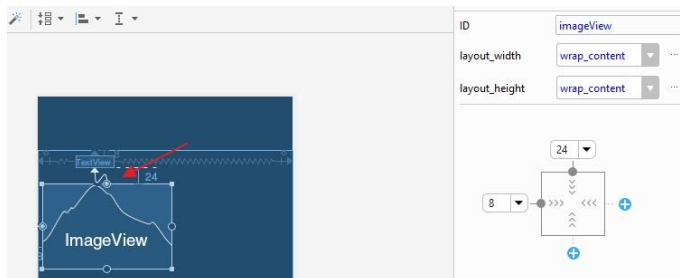
Por defecto, siempre centra (50%) pero podemos desplazar el valor para favorecer un determinado lado sobre otro (en el ejemplo del 16%, el TextView aparecerá por tanto hacia la izquierda):



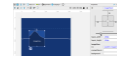
```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:text="TextView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.16"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Añadimos restricciones al ImageView:

- está a 8dp del margen izquierdo del layout
- está a 24 dp del **margen inferior del TextView** (las restricciones pueden ser hacia otras vistas)



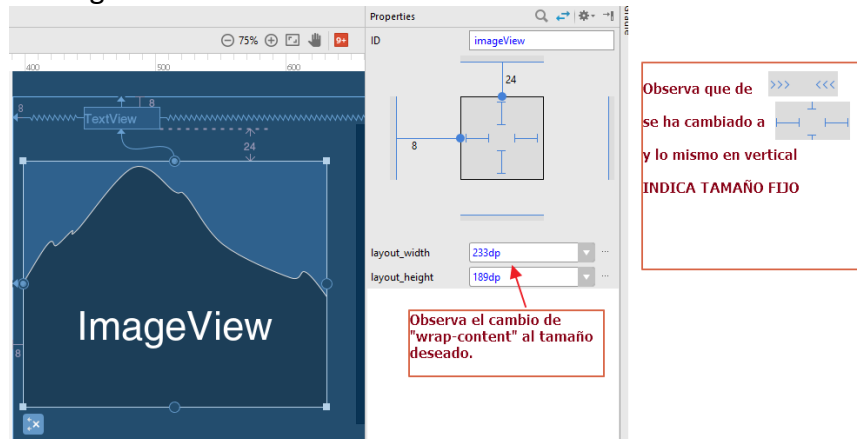
```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="24dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView"
    app:srcCompat="@drawable/perro" />
```



## Tamaño

Cambiar el tamaño de una vista (por ejemplo, del ImageView) es muy sencillo.

- Para escalar a un determinado tamaño, haremos uso de cualquiera de los rectángulos de los vértices de la vista

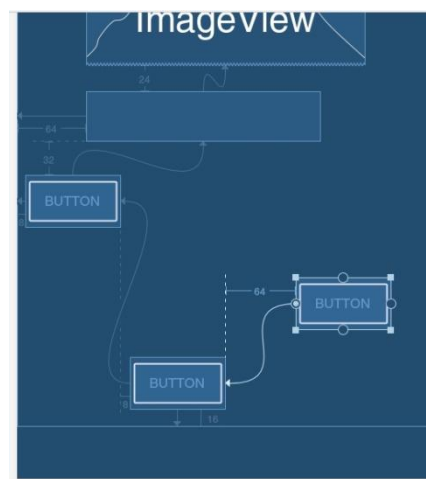


- Pulsando sobre el símbolo (tanto en vertical como en horizontal) podemos cambiar el tamaño para que sea "wrap-content" (>>> <<<), fijo (<=>) o que "coincida con restricciones" (similar a "match-parent") (H H H H)
- No se debe utilizar match\_parent en una ConstraintLayout. En su lugar hay que utilizar "coincida con restricciones" (0dp)
- Cuando se ha escogido que "coincida con restricciones", es posible ajustar el tamaño de la vista a una proporción. Para habilitar la relación, se clicka en el conmutador de relación de aspecto y se fija la relación "ancho:alto" deseada:

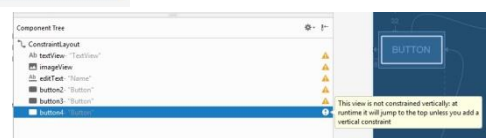


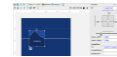
## Alineación de texto

Añadimos restricciones al EditText y a los botones para que queden semejantes a la siguiente imagen:



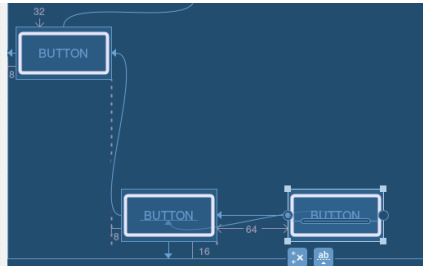
Observa que Lint nos dice que faltan restricciones (el tercer botón no tiene las dos mínimas)



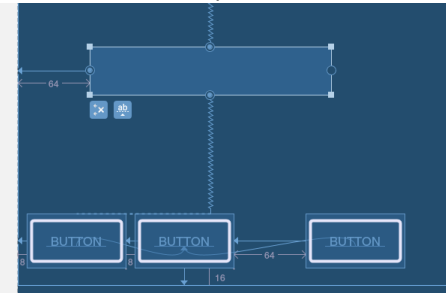


Además del citado error, las restricciones anteriores hacen que los textos de los botones no queden alineados.

Para solucionarlo, usaremos la propiedad que permite alinear la línea base del texto de un objeto con la línea base del texto de otro. Para ello debemos mostrarla usando click derecho. Primero alinearemos el texto del tercer botón con el del segundo (con lo cual además de conseguir la alineación, le otorgamos la restricción que le faltaba!). Lo podíamos haber hecho también de otra manera, seleccionando ambos, click dcho. y alinear por línea de base.



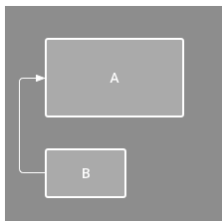
Luego alinearemos el texto del primer botón con el del segundo. Solo lo podemos hacer de la primera manera ya que hay restricciones incompatibles (32dp del primer botón al EditText).



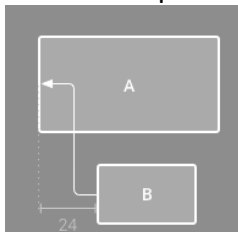
## Alinear vistas

Podemos alinear una vista con otra:

- Por un lado (el izquierdo por ejemplo)

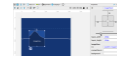


- Por un lado pero con margen

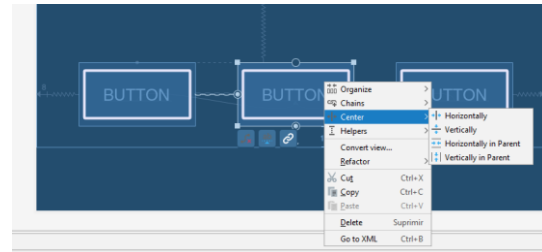
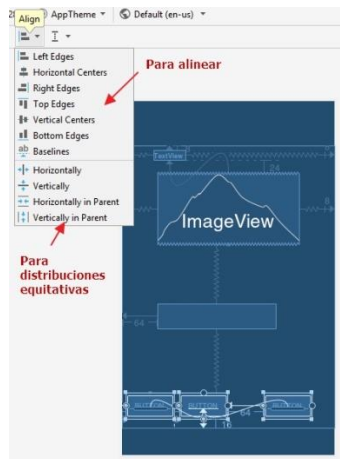


- Centrado: creando una restricción en ambos lados



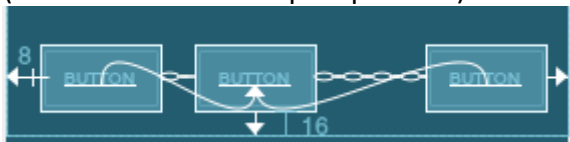


- Si son muchas vistas a alinear (una serie de botones, como en nuestro caso), se seleccionan todas (con May+click) y se hace uso de la opción del menú o click dcho.:



## Distribución equitativa

Selecciona los tres botones, borra las restricciones que tienen, crea la vertical para uno de ellos, alínealos por la línea de base y distribúyelos centrándolos horizontalmente (observa las cadenas que aparecen):



Si seleccionamos alguna de las vistas encadenadas y clicamos botón dcho. puede cambiarse el tipo de encadenamiento:

Horizontal Chain Style (spread)

Tipos:

- 1 **Spread:** Las vistas se distribuyen uniformemente (después de márgenes se contabilizan). Este es el valor predeterminado.
- 2 **Corre el interior:** La primera y última vista se fijan a las limitaciones de cada extremo de la cadena y el resto se distribuye de manera uniforme.
- 3 **Ponderado:** Cuando la cadena se establece en la **propagación o difundir en el interior**, se puede llenar el espacio restante mediante el establecimiento de una o más vistas a "coincidir con limitaciones" (0dp). Por defecto, el espacio se distribuye en partes iguales entre cada vista que se ajusta a "coincidir con limitaciones", pero se puede asignar un peso de importancia para cada vista mediante el `layout_constraintHorizontal_weighty` `layout_constraintVertical_weight` atributos. Si está familiarizado con `layout_weight`un **diseño lineal**, esto funciona de la misma manera. Por lo que la vista con el valor de peso más alto recibe la mayor cantidad de espacio; vistas que tienen el mismo peso reciben la misma cantidad de espacio.
- 4 **Embalado:** Las vistas son embaladas en común (después de los márgenes se contabilizan). A continuación, puede ajustar el sesgo de toda la cadena (izquierda / derecha o arriba / abajo) cambiando cabeza ver el sesgo de la cadena.

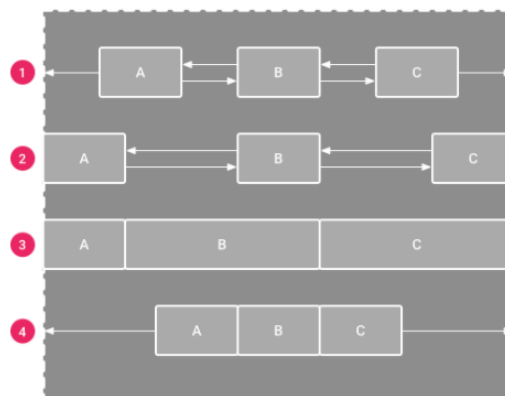
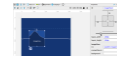



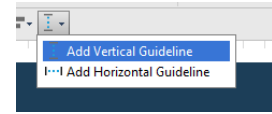
Figura 14. Ejemplos de cada estilo de la cadena



## Directrices o líneas guías

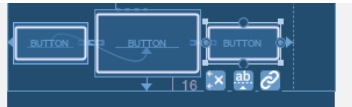
Puede añadirse una línea guía vertical u horizontal a la que puede limitar vistas y la directriz será invisible para los usuarios de la aplicación. La guía puede colocarse en el diseño de acuerdo a las unidades dp o % siempre con respecto al borde del layout.

Para crear una guía, se clicka en  de la barra de herramientas y se escoge el tipo de guía.



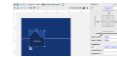
Se arrastra la línea de puntos para cambiar su posición y se clicka en el círculo en el borde de la guía para cambiar el modo de medición.

A partir de este momento, cualquier vista puede tener una restricción con anclaje a esta directriz.




## Barreras


Similar a una guía, una barrera es una línea invisible a la que puede restringir las vistas. Excepto que una barrera no define su propia posición; la posición de barrera se mueve en función de la posición de las vistas que contiene. Esto es útil cuando desea restringir una vista al conjunto de vistas en lugar de a una vista específica.

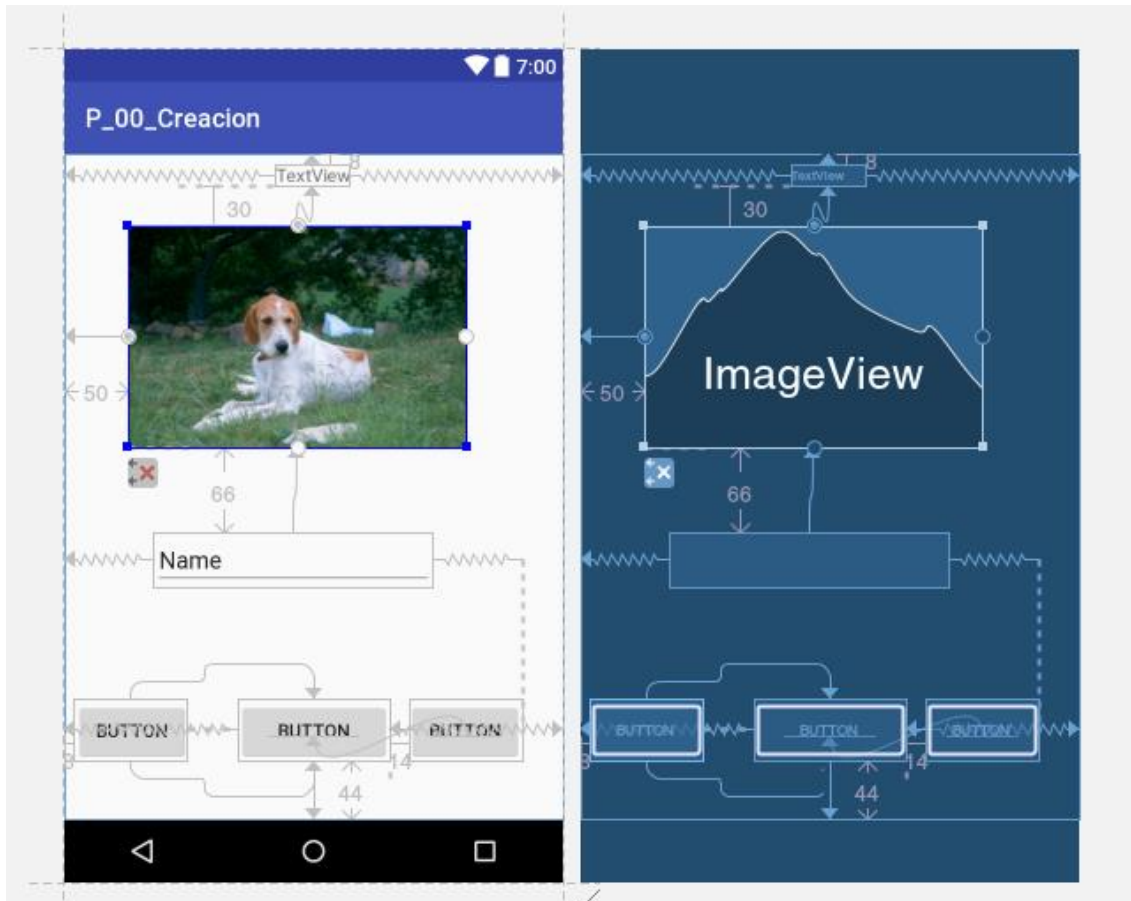


## Creación automática de restricciones

En lugar de añadir restricciones a todas las vistas que se colocan en el diseño (que puede ser trabajoso), puede moverse cada vista a la posición que se desea y, a continuación, hacer clic en **Deducir Restricciones**  para crear automáticamente restricciones.

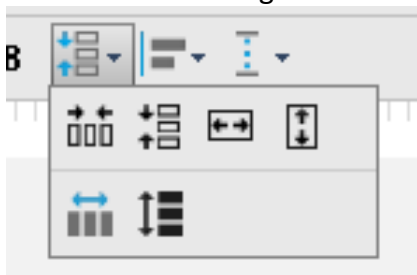
Es posible (bastante!) que luego haya que hacer algunos ajustes para asegurar que el diseño responde para diferentes tamaños de pantalla y orientaciones.

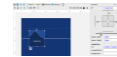
Para probar la creación automática, borra todas las restricciones que hemos estado trabajando , coloca las vistas donde te interese y deja que AS deduzca las restricciones:




Previsualiza en modo "landscape" y comprueba que nuestro diseño no está tan mal gracias a las restricciones.


A veces antes de dejar que se deduzcan las restricciones, es útil establecer algunas características de algunas vistas para ello puede utilizarse la opción del menú





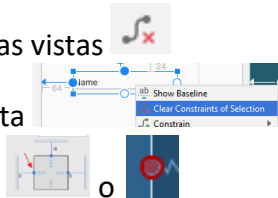
## Otras opciones

Autoconnect  es una utilidad que cuando se activa, se crea automáticamente dos o más restricciones para cada vista a medida que agrega, pero sólo cuando sea necesario para limitar la vista a la disposición de los padres. Autoconnect no crea limitaciones a otros puntos de vista en el diseño.

La opción  permite habilitar/deshabilitar que se muestren o no las restricciones en el editor gráfico

## Eliminar restricciones

- Todas las de todas las vistas
- Todas las de una vista
- Alguna de una vista



## Atributos XML

Observa el layout en la vista texto, se han añadido numerosos [atributos](#) definidos por la plataforma, la ventaja es que lo hemos hecho desde el entorno gráfico con el útil "soltar y arrastrar".

## Prácticas propuestas

Realiza los ejercicios propuestos en el fichero Ejercicios\_01\_Diseño\_básico