

UI: Parte II

Desarrollo para plataformas móviles



Borja Martin Herrera
Borja.herrera@u-tad.com

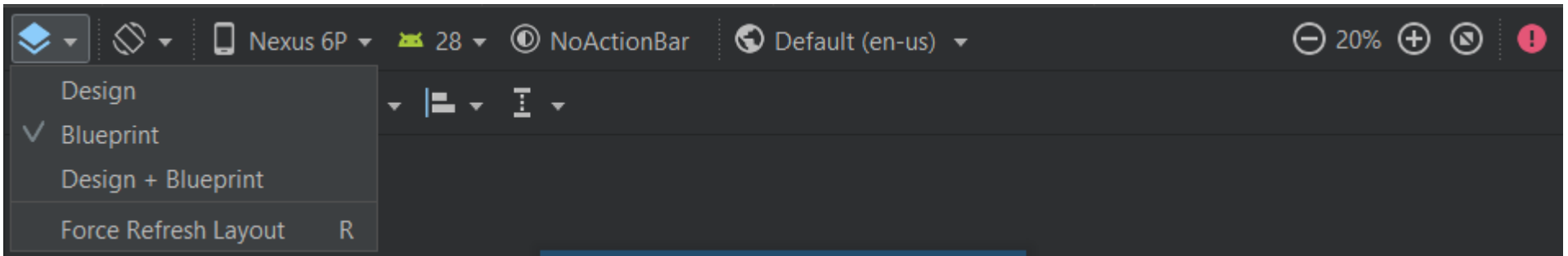
Constraint Layout

- Constraint Layout:
 - Hasta ahora hemos visto cómo organizar el layout utilizando *LinearLayout* y *FrameLayout*.
 - *ConstraintLayout* es un tipo de View más complejo pero mucho más flexible.
 - Nos permite editarlo de manera visual en *AndroidStudio* y no escribiendo en el XML.
 - Para vistas complejas, es mucho más eficiente que crear un jerarquía de Linear y Frame Layouts.



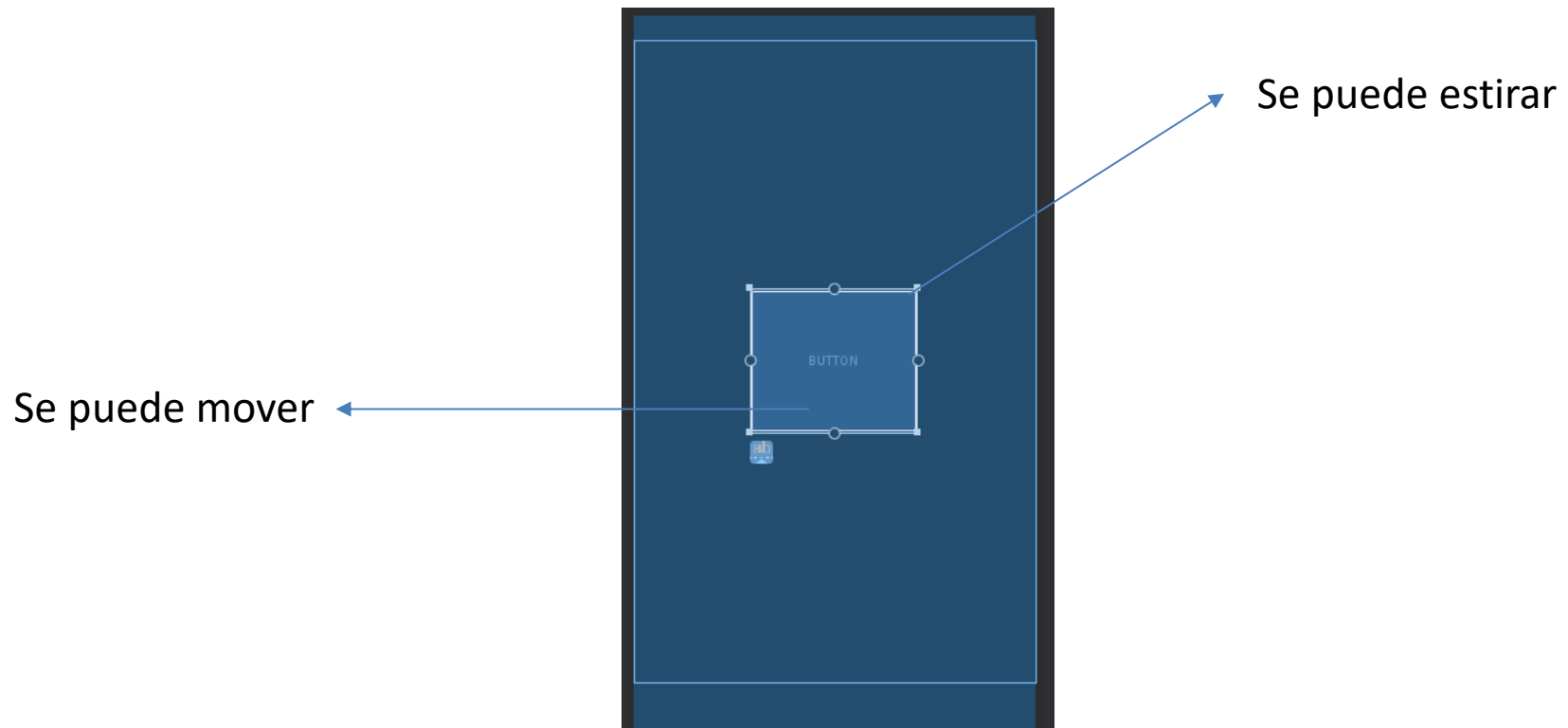
Constraint Layout

- Constraint Layout:
 - Lo primero para utilizarlo es seleccionar la vista *Blueprint* del *AndroidStudio*:



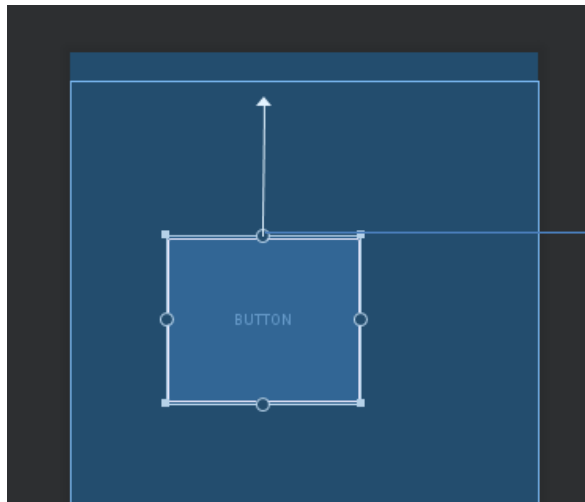
Constraint Layout

- Ahora vamos a poder editar los elementos del *layout* cómo si fuera un programa de edición de imagen usando el ratón:



Constraint Layout

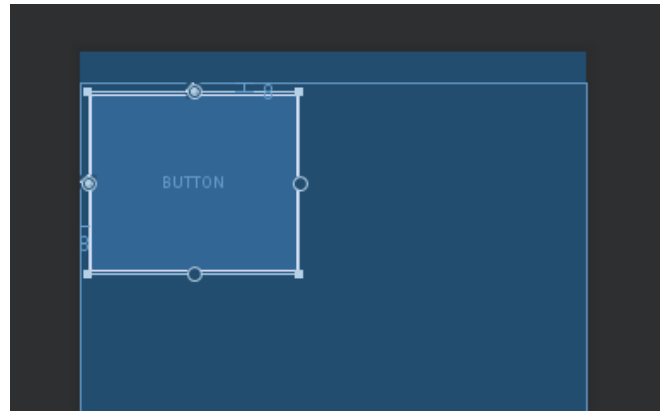
- Para añadir un nuevo elemento al *Layout*, tenemos que arrastrarlo a la vista del *Blueprint*.
- Al hacerlo, el botón está en el “aire”. Si ejecutamos la app, vemos como se pega a la posición 0,0 de pantalla (*left-top*).
- Para evitar esto, tenemos que anclarlo al resto de vistas:



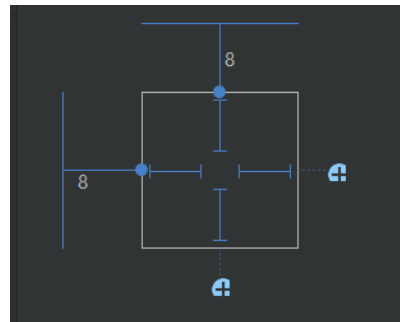
Haciendo *click* en los círculos del *view* podemos anclarlo a su padre o a otros elementos.

Constraint Layout

- Cuando lo anclemos, se va a mover el elemento hacia el margen izquierdo-arriba:

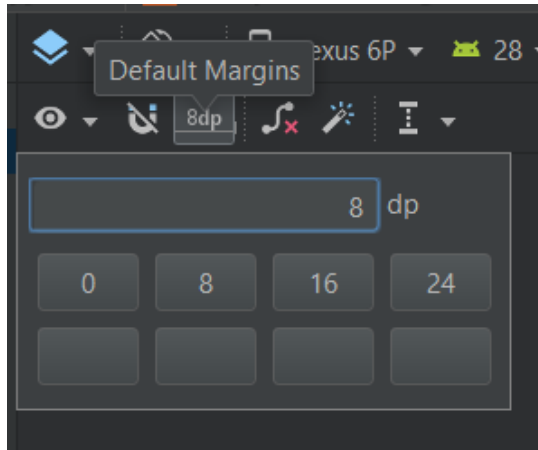


- A la derecha (vista de atributos) podemos ver que se ha anclado con un margen de N dp.

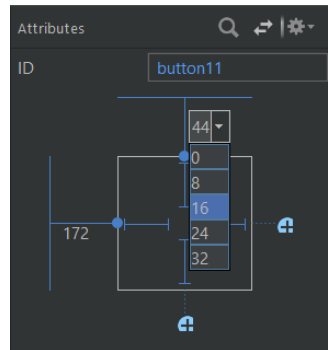


Constraint Layout

- Podemos elegir el margen por defecto que se añade a las vistas pulsando en:

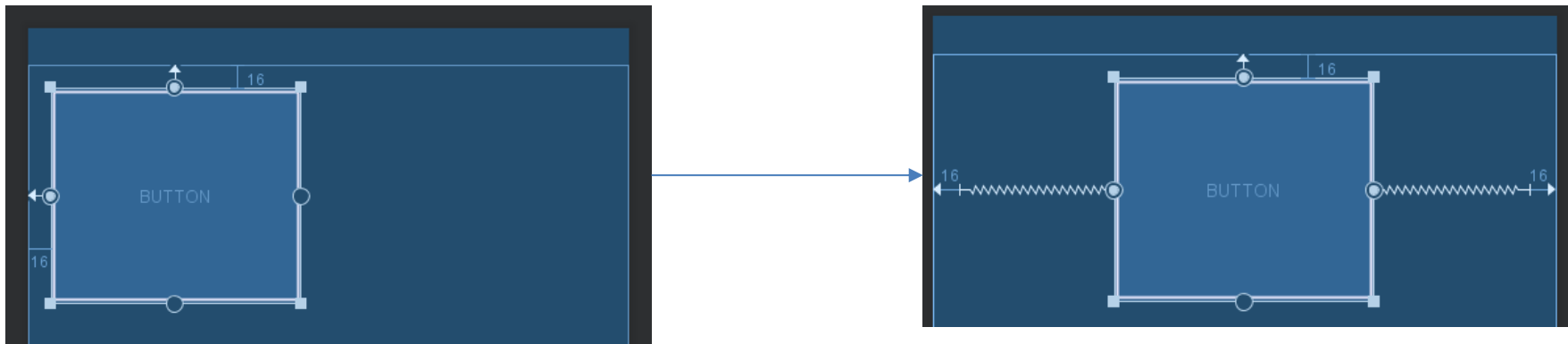


- Se puede modificar este margen tanto moviendo el elemento por el Blueprint como en la vista de atributos.



Constraint Layout

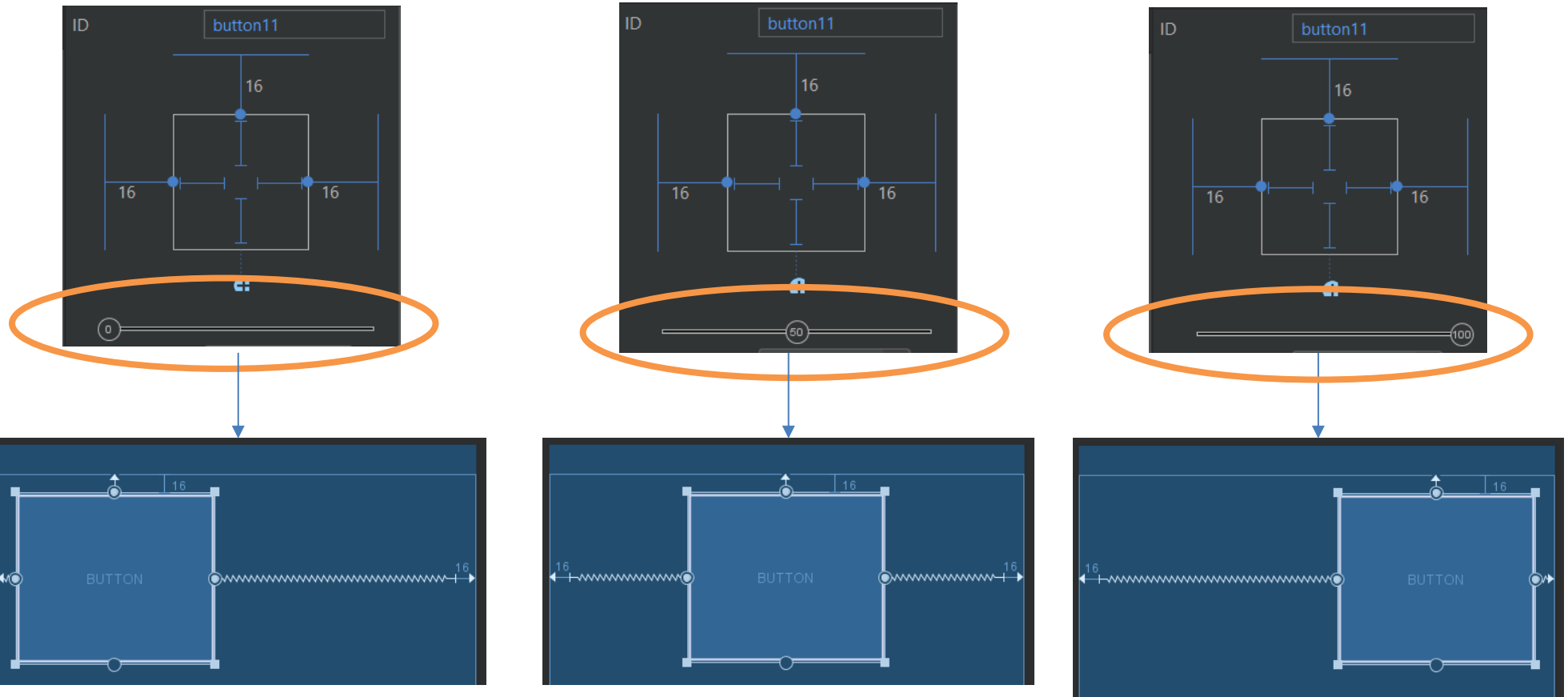
- Si añadimos un constraint por la derecha, la vista se centra:



- Una vez centrado, podemos desplazarlo a derecha e izquierda tanto usando el ratón cómo usando el slider de la vista de atributos.

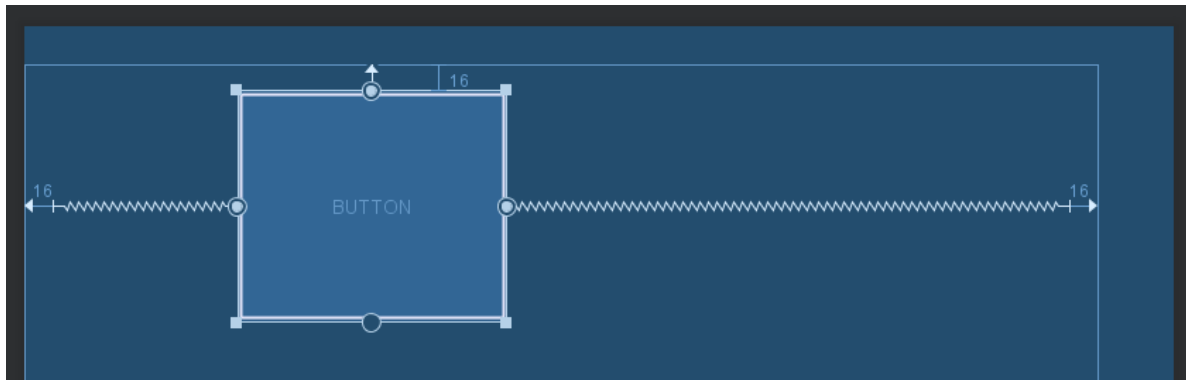
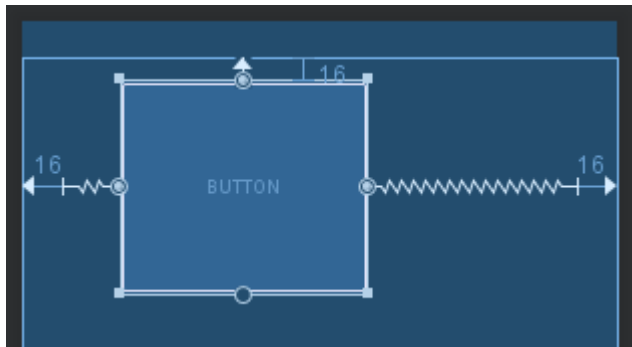
Constraint Layout

- Una vez centrado, podemos desplazarlo a derecha e izquierda tanto usando el ratón como usando el slider de la vista de atributos.



Constraint Layout

- Al hacerlo de esta manera, la vista se ve igual independientemente de si el dispositivo esta en portrait o landscape:



Ajustado al 25%

Constraint Layout

- Según vamos haciendo cambios, estos se ven reflejados en el XML:

```
<Button
    android:id="@+id/button11"
    android:layout_width="171dp"
    android:layout_height="148dp"
    android:layout_marginEnd="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="16dp"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:text="Button"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.25"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

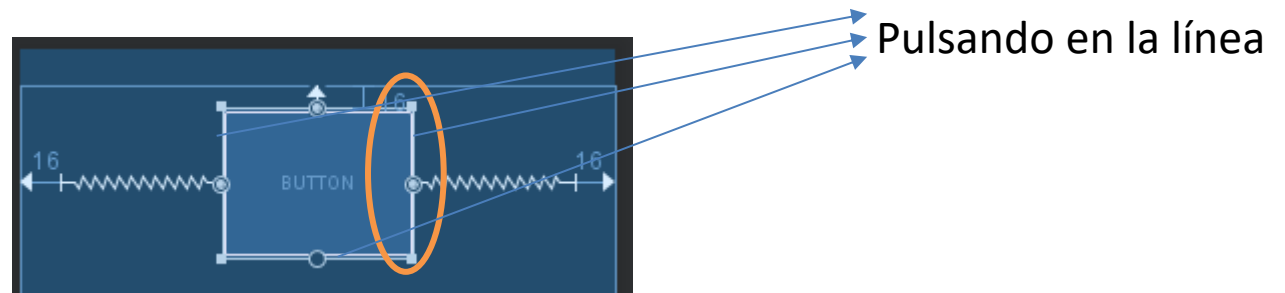
Constraint Layout

- Al igual que podemos modificar la posición del elemento en la vista, podemos cambiar su tamaño.
- Tenemos varias opciones:
 - Poner un tamaño fijo al elemento.
 - Que se adapte al tamaño de los *constraints*.
 - Usar la propiedad *wrapcontent* para que se haga del tamaño mínimo que necesita el elemento.

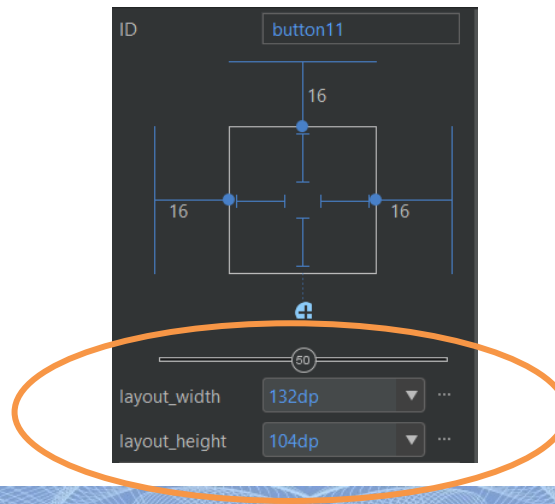


Constraint Layout

- Tamaño fijo:
 - Podemos definir el tamaño usando los manejadores de la vista de blueprints:

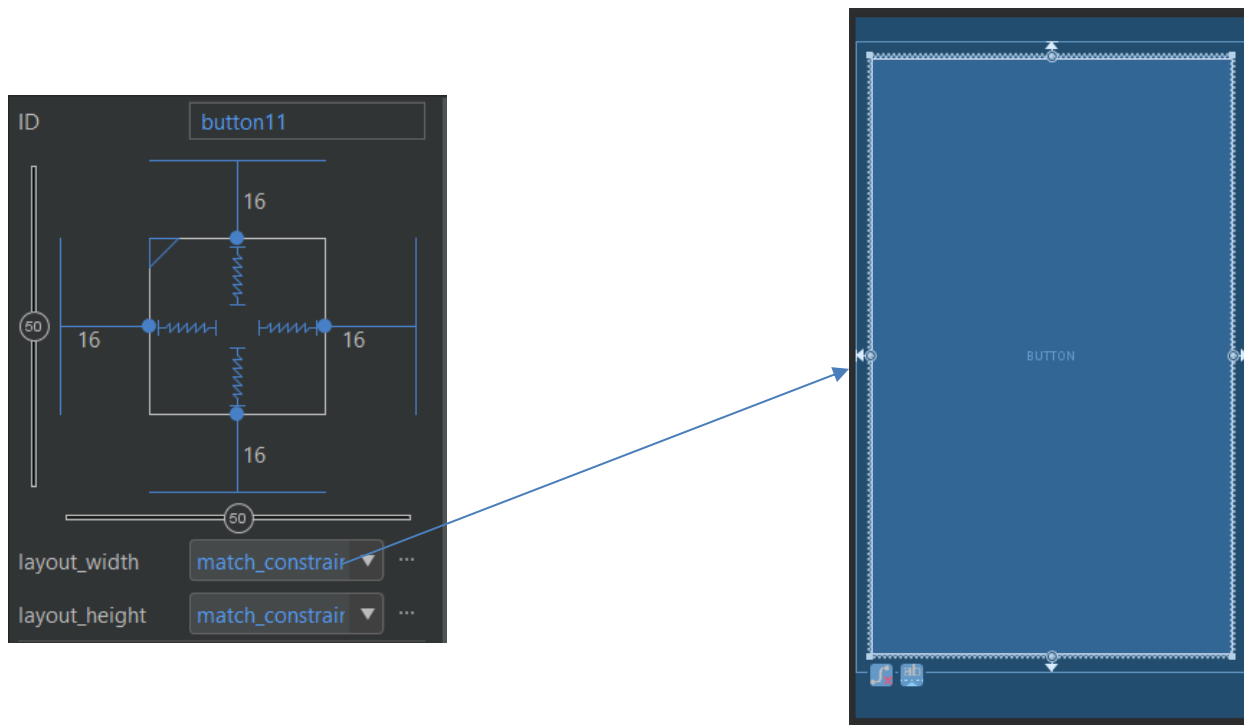


- O en la vista de atributos:



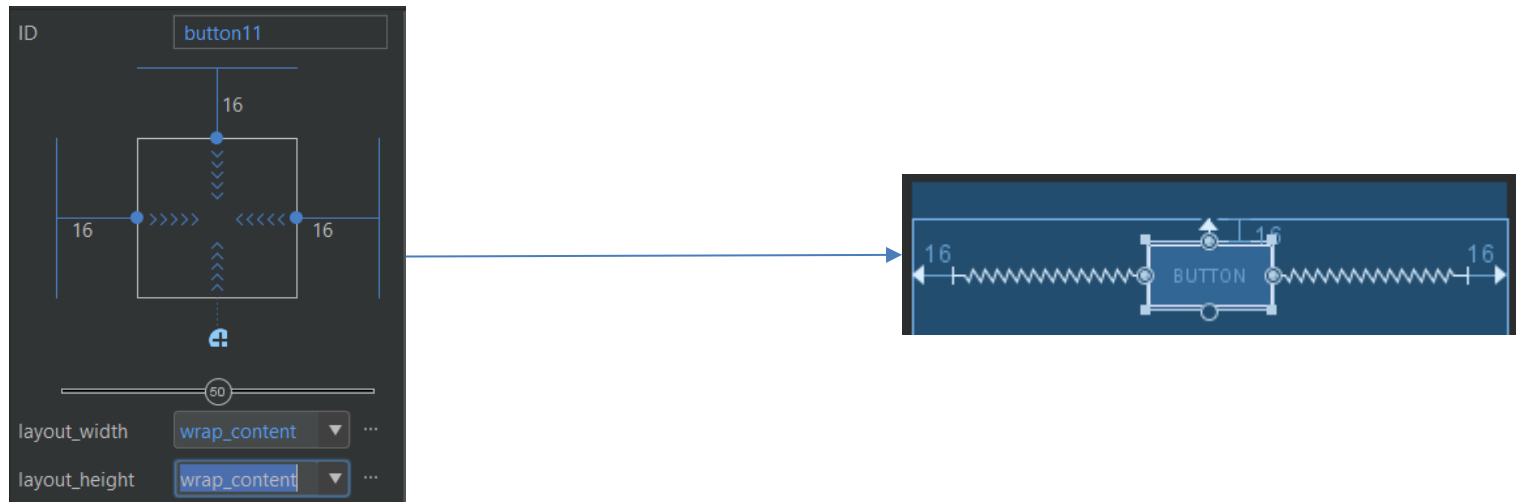
Constraint Layout

- Tamaño fijo:
 - Si usamos como tamaño el valor 0dp (veremos que pone valor *match_constraint*), se adaptara al tamaño de los *constraints*:



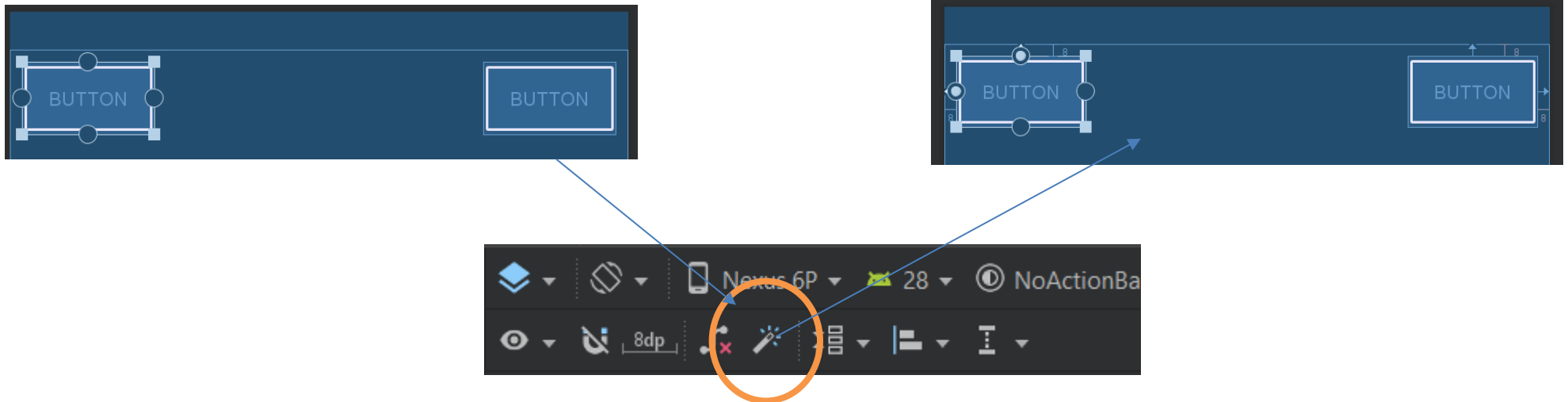
Constraint Layout

- Usando la propiedad *wrapcontent*.
- Se adaptará al tamaño mínimo posible de la vista:



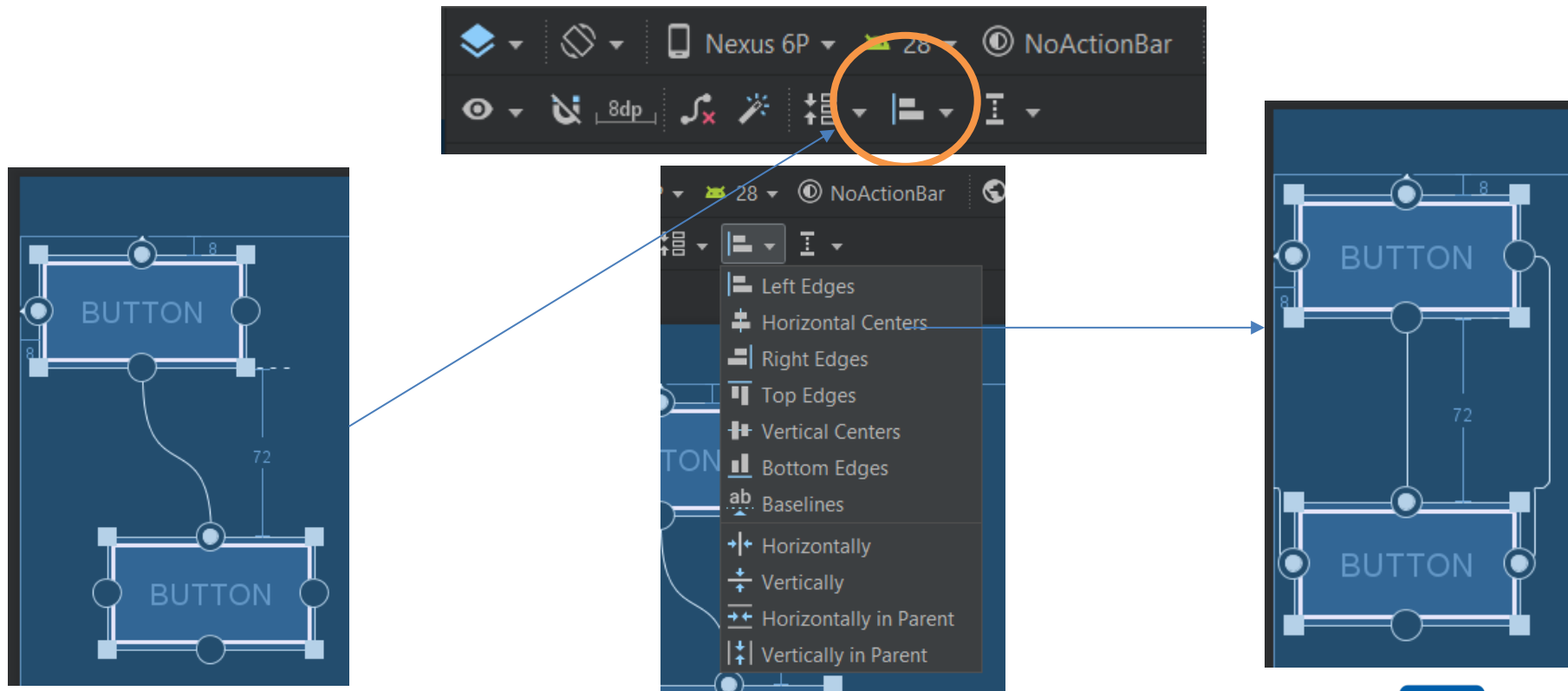
Constraint Layout

- Podemos dejarle a Android que añada automáticamente los *constraints* de las vistas.
- Para ello pulsamos sobre el botón *InferConstraints*



Constraint Layout

- También podemos alinear vistas de manera automática.
- Si tenemos dos botones, podemos alinearlos usando el botón *Align*. Primero tendremos que seleccionar ambos usando la tecla *shift*:



Ejercicio

Ejercicio: Haced la calculadora usando
Constraints.

