

# Desarrollo de aplicaciones Android (161811FP07)

---

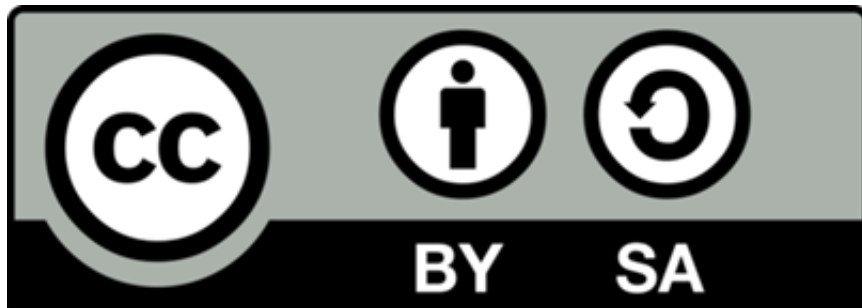
Marzo-Abril de 2016. CEP de Granada

---

José Antonio Vacas @javacasm

---

[javacasm@gmail.com](mailto:javacasm@gmail.com)

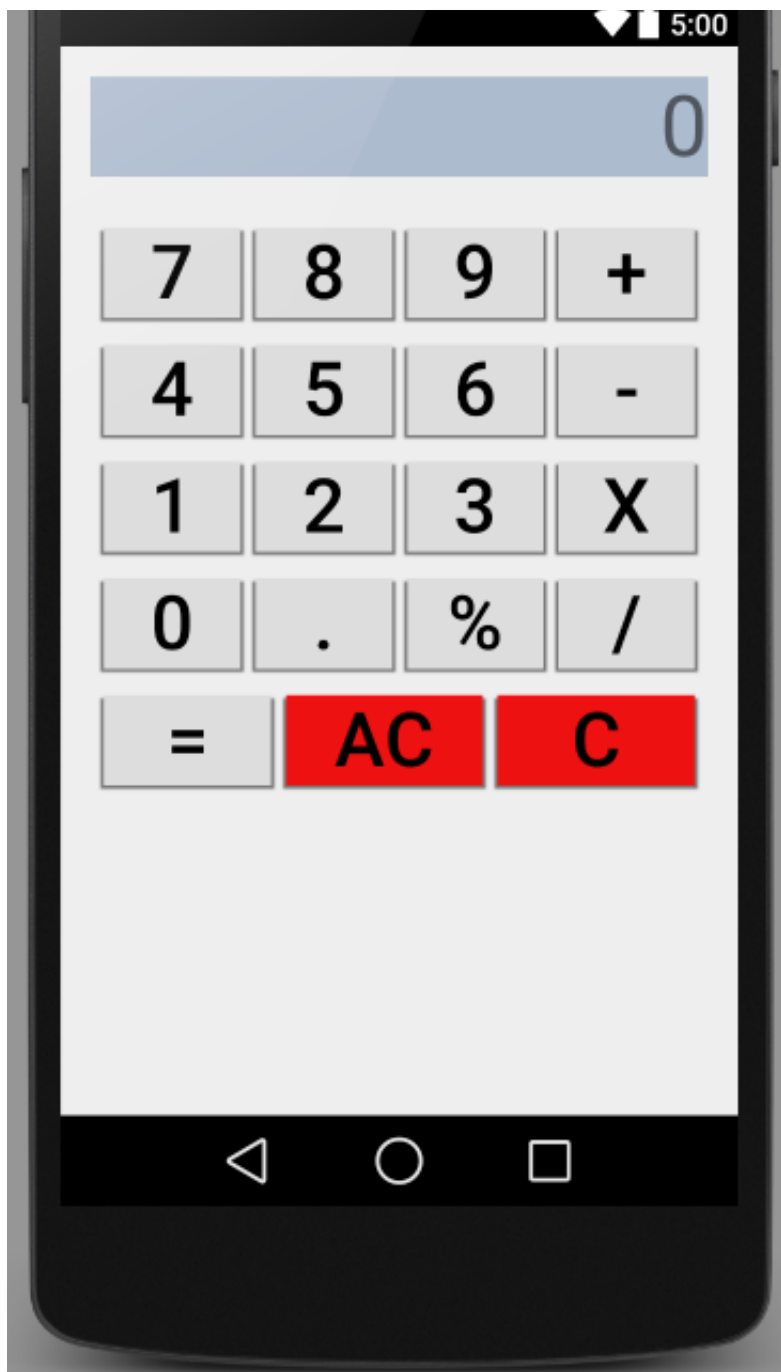


## Proyecto 1: Calculadora

---

Vamos a comenzar diseñando el interfaz de una calculadora como la siguiente





# Analizando

---

En primer lugar vamos a analizar los tipos de controles que usaremos:

- Una caja de texto, donde el usuario podrá leer los números. No es necesario que el usuario pueda editar los contenidos.
- Un conjunto de botones con los escribiremos los números y con los que indicamos las operaciones a realizar.

Viendo la disposición de los botones podemos pensar que lo más adecuado es usar una disposición tabular. Eso lo podemos conseguir con el `TableLayout`

## Layout:

---

Una Layout es una forma de disponer los controles. Existe diferentes tipos:

- `FrameLayout`
- `AbsoluteLayout`
- `RelativeLayout`
- `LinearLayout`
- `TableLayout`

## TableLayout

---

Está constituido por filas que contienen otros elementos. Dentro de ellas los controles se colocan unos a continuación de los otros

# Estilos

---

Dado que son muchos los botones que vamos a utilizar y casi todos tienen un aspecto muy similar, definiremos un estilo que aplicaremos a todos. De esta manera cualquier cambio que queramos hacer se aplicará a todos.

Como extraer un estilo Herencia en los estilos

Vamos a comenzar dando el aspecto que queramos al botón

Las propiedades siempre se nombra con la parte del control al que se aplican y luego con la propiedad:

TextSize TextColor Background (comentar el uso de imágenes para hacer botones de distintas formas) TextAlignment  
Márgenes ¿Border?

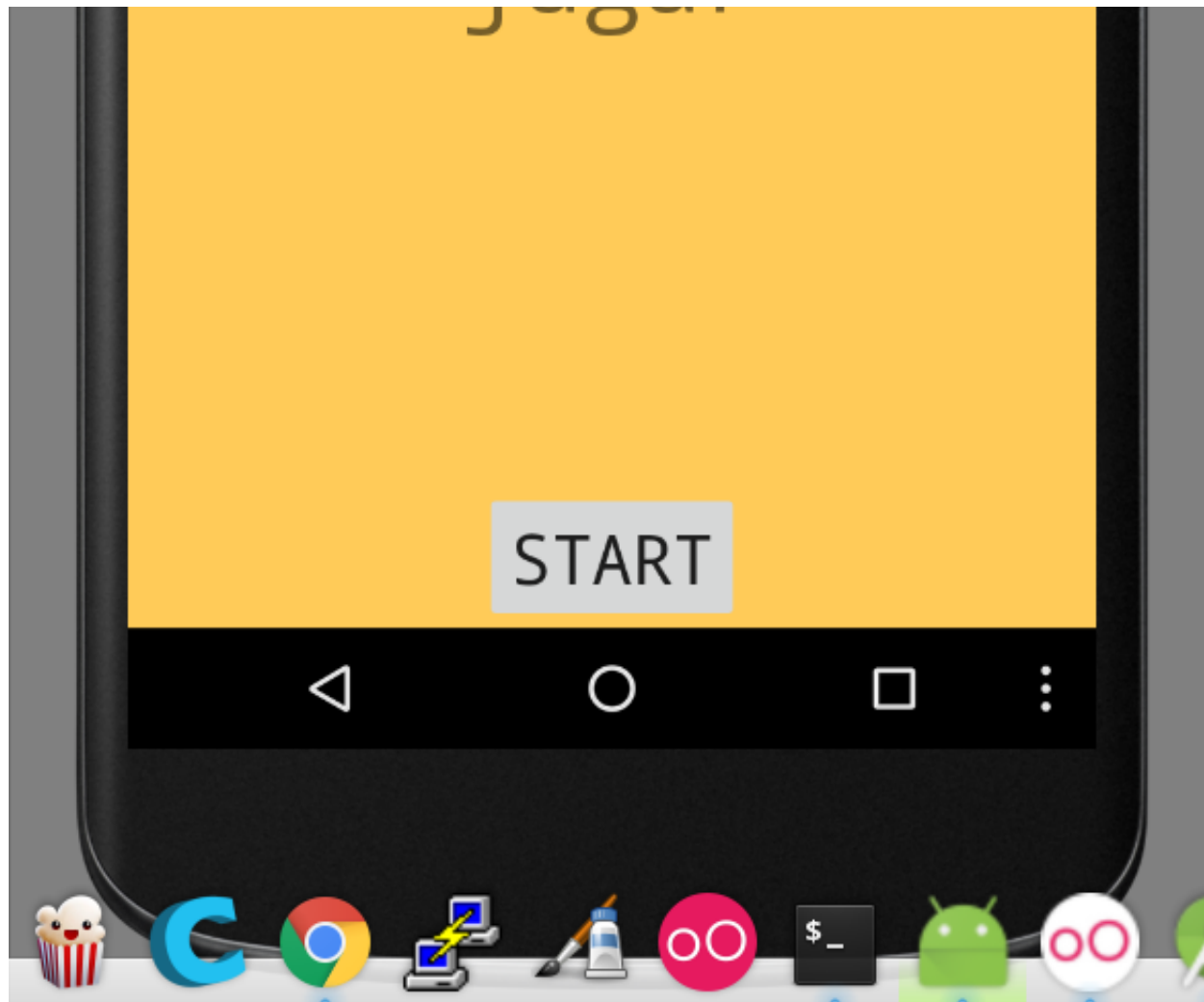
- Extraemos el estilo
- Creamos tablelayout y lo colocamos (con márgenes)
- Añadimos un tablerow y le añadimos el botón
- Explicar que mejor añadir al Arbol de controles
- Copiamos y pegamos otros 3 botones
- Usamos weight para que se repartan el espacio
- Revisamos el xml del layout para evitar propiedades innecesarias

# Diseño de aplicación de juego Memory

---

Pantalla inicial





Pantalla del juego





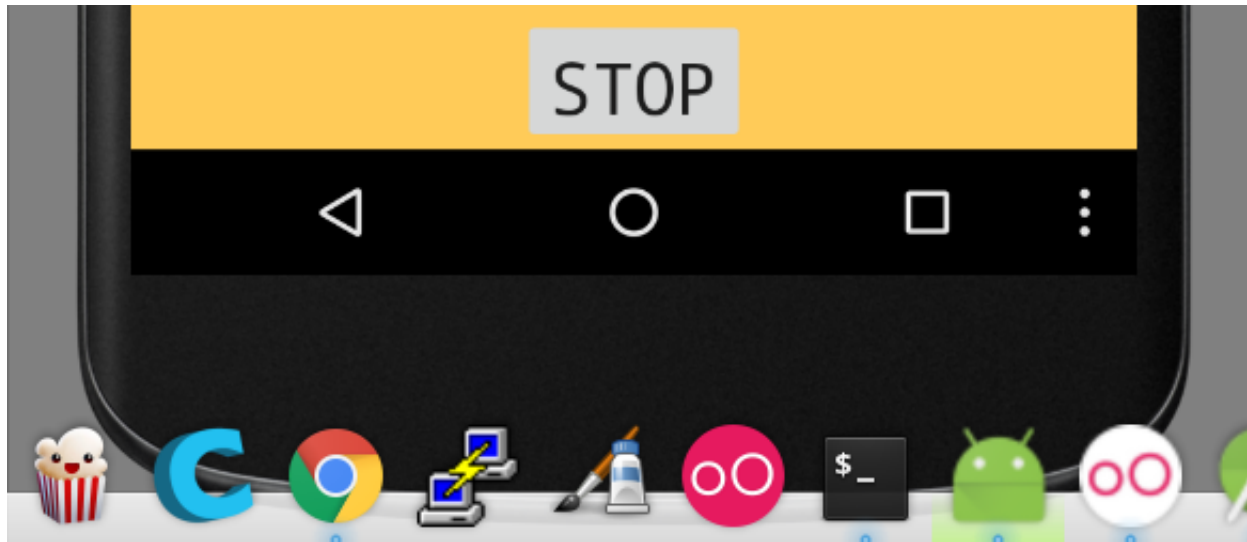
3G 1:43

## Memory 2014

Puntuación:5



01:14



Parece más fácil usar un Tablelayout para las casillas

- Jugaremos con la visibilidad de algunos elementos

```
btStop.setVisibility(View.VISIBLE);
```

- Usaremos el acceso a recursos como imagen

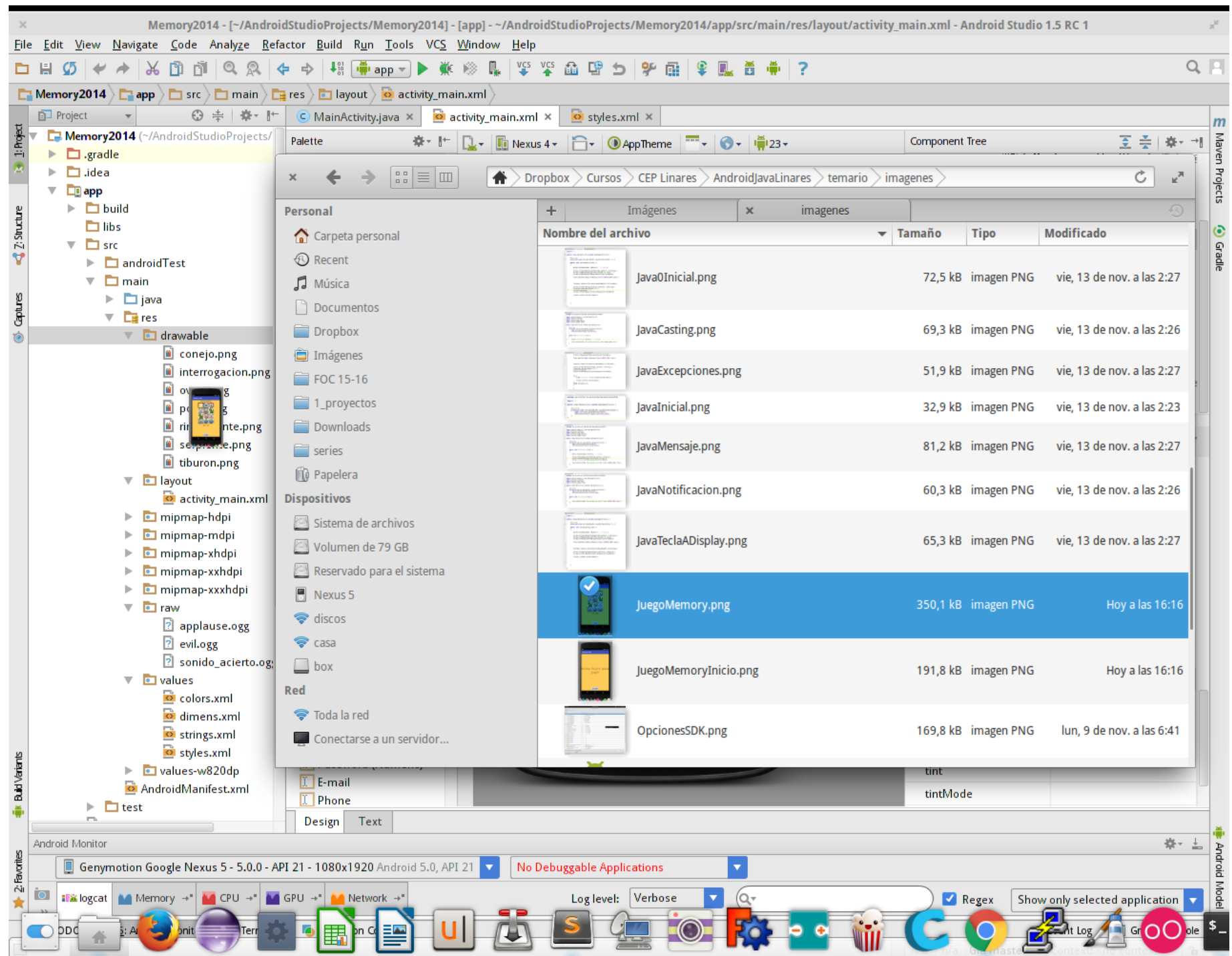
```
ivCartaSegunda.setImageResource(R.drawable.Interrogacion);
```

- Utilizaremos sonidos para reforzar la interacción

## Construcción:

- Añadiremos un control de tipo ImageView (podríamos usar ImageButton, pero tendríamos peor aspecto por usar borde)
- Añadimos la imágenes a la carpeta drawable (podemos hacerlo con copiar/pegar o arrastrando)





Usaremos la propiedad `src` para asignarle una imagen. Veremos que se adapta al tamaño de esta, pero que es superior al que queremos.

Hay varias formas de establecer el tamaño:

i. Marcamos un ancho y alto, pero por defecto no se pone a ese tamaño sino que tenemos que activar la propiedad

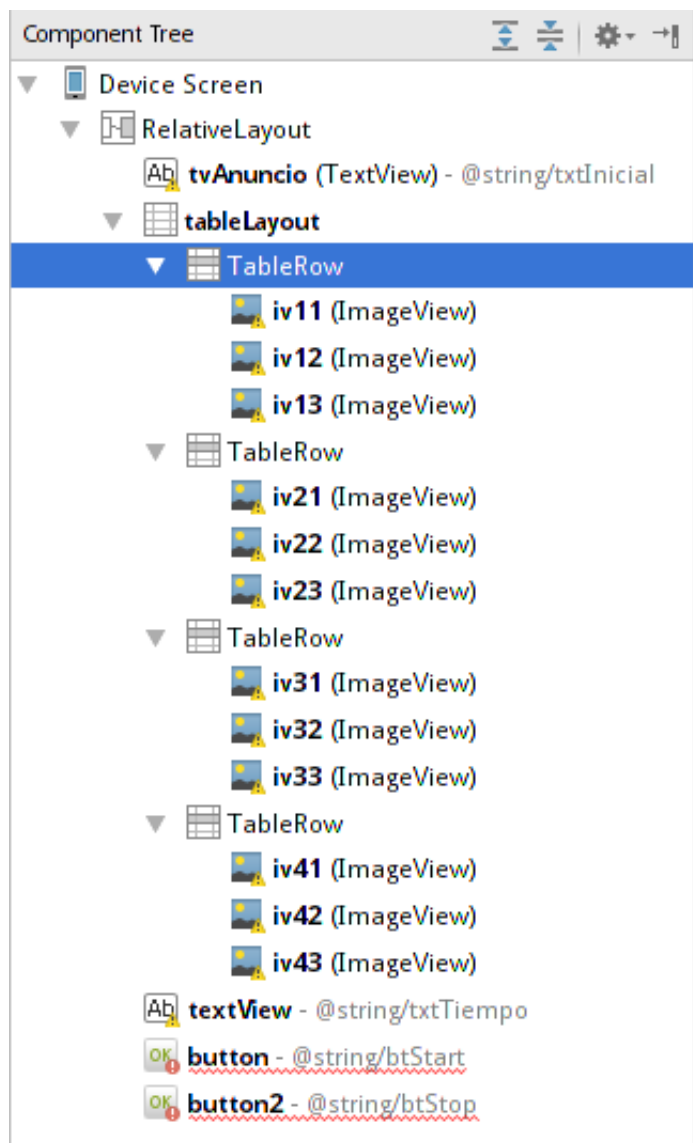
`AjustViewBounds=true maxHeight=100dp maxWidth=100dp`

ii. Establecemos el `layout_Width` y el `layout_Height` a 100dp

- Ahora generamos un style (en el que podemos incluir la propiedad `Src`) y ya podemos crear los restantes botones que incluiremos en nuestro layout



donde podemos ver que los controles están organizados según el siguiente árbol



Destacar que hemos usado una manera de nombrar sistemática a los ImageViews para que sea sencillo localizarlos (usando dos números: uno para indicar la fila y otro la columna)

- Veamos las propiedades que definen este formato

android:layout\_centerVertical="true" android:layout\_centerHorizontal="true"

Para centrar los componentes y dejando en los controles ImageView sólo estas propiedades:

```
<ImageView  
    android:id="@+id/iv11"  
    style="@style/imagen" />
```