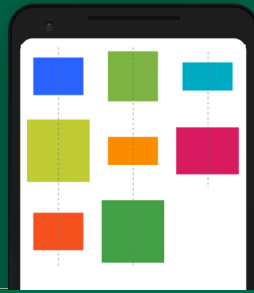


# PROGRAMACIÓN MULTIMEDIA Y DISPOSITIVOS MÓVILES

**CURS 2025-2026**

## UD3: Componentes de una aplicación Android



Profesorado	
Ana Sanchis	<a href="mailto:ag.sanchisperales@edu.gva.es">ag.sanchisperales@edu.gva.es</a>
Juan Mateu	<a href="mailto:j.mateuruzafa@edu.gva.es">j.mateuruzafa@edu.gva.es</a>
Joan Carrillo	<a href="mailto:jr.carrilloforonda@edu.gva.es">jr.carrilloforonda@edu.gva.es</a>

### Instrucciones iniciales

- Las tareas se subirán a la plataforma Aules en la fecha establecida y con el formato indicado en los siguientes puntos.
- Para el desarrollo de las siguientes tareas, se creará un único proyecto nuevo con Android Studio.

### Objetivos de la actividad

1. Conocer los principales componentes de una aplicación Android.
2. Conocer y utilizar las principales vistas en una app Android (TextView, Button, ImageView...)
3. Utilizar los diferentes tipos de contenedores o layouts en Android.

### Temporalización

La tarea tiene una estimación aproximada de dos sesiones lectivas de 55 minutos cada sesión.

## COMPONENTES DE UNA APLICACIÓN



1. En la presente práctica se quiere realizar el diseño de la interfaz gráfica que se muestra a continuación:

- Para posicionar los elementos utilizaremos el Constraint Layout.



### Pasos a seguir

1. Crear un nuevo proyecto **t3a1-PrimerApellido-Nombre**.
2. Diseñamos la interfaz con el editor xml, y si lo precisas puedes utilizar también el editor visual.
3. Para posicionar los elementos utilizaremos Guidelines, en concreto:
  - a. **Horizontal guideline**: al 33% de la pantalla
  - b. **Vertical guideline**: al 50% de la pantalla

Tras ello, vamos a añadir los siguientes elementos:

4. **ImageView:** será una imagen en formato jpg, ubicada en la carpeta drawable, la imagen irá pegada al margen superior y en los lados ocupará todo el espacio disponible. Utilizaremos el atributo `scaleType`, `centerCrop` para centrar la imagen. Además, para indicar que independientemente del tamaño de la pantalla queremos que la imagen ocupe el 33% de toda la pantalla. Referenciamos la parte inferior de la imagen respecto a la guideline y le decimos que verticalmente también ocupe todo el espacio disponible.
5. **TextViews:** cambiar los ids a autor y título del producto así como los textos. También las propiedades del texto: `textAppearance: @style/TextAppearance.MaterialComponents.Headline3`, y cambiar color para que se visualice correctamente.
6. **Ubicar los TextViews.** El Título a 16 de la izquierda, y a 8 de la guideline.
7. **TextViews:** para la descripción y el texto de la descripción.
8. **Posicionar TextViews:** Título Descripción a 16 dp de la horizontal guideline y 16 del margen izquierdo. Contenido Descripción: 8 dp del título y a 16 dp del margen izquierdo, respecto a la vertical guideline a 8 dp y que el contenido ocupe todo el tamaño disponible según el dispositivo.
9. **TextViews** para los ingredientes, y los posicionamos como los de descripción.
10. **TextViews** para la receta y los posicionamos a 16 dp tanto horizontal como verticalmente.

### Aspectos a tener en cuenta de éste tipo de Layout

#### Modificar tamaño



- Los cuadrados azules sirven para cambiar el tamaño del elemento.

#### Referenciar elementos

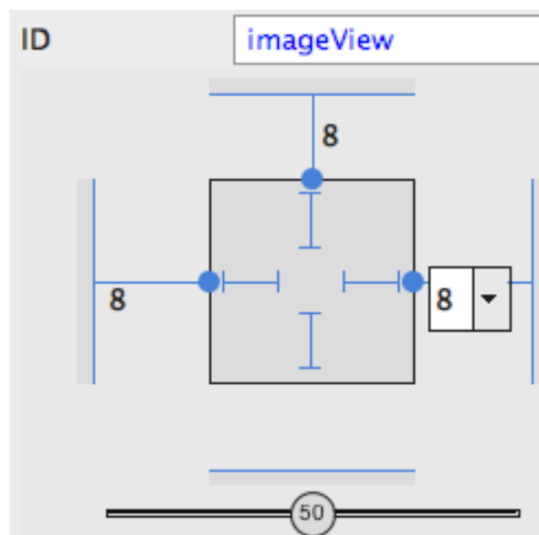
- Los círculos rojos los utilizaremos para hacer referencia, ya sea a los componentes hermanos o al padre.
- Es importante saber que cuando insertamos un componente se debe establecer al menos una relación horizontal y otra vertical, bien con el padre o con algún hermano.

### Posicionar elementos



- Las flechas en modo de sierra indican que existe una definición de esas delimitaciones en una dirección opuesta.

### Ocupar el espacio



- **Fixed:** Líneas rectas que indican, ocupan el tamaño indicado por layout\_width y layout\_height
- **Match constraints:** Líneas dentadas que indican que va a ocupar todo el tamaño disponible

- **Wrap content:** Tamaño que tenga la imagen

#### Guía de referencia

- Sirve para medir la cantidad de espacio que hay disponible, tanto vertical como horizontalmente.