CFGS DAM

IES MONTE NARANCO

UT-2

Programación de aplicaciones Android

Interfaz de Usuario – Guías e imágenes

Guías (Guidelines)

- Las **Guidelines** (Líneas Guía) son uno de los *Helpers* más útiles y sencillos de **ConstraintLayout**.

- Son líneas invisibles que solo existen para **servir como puntos de referencia** para otras *Views*. No se ven en el diseño final ni afectan el rendimiento de la aplicación.

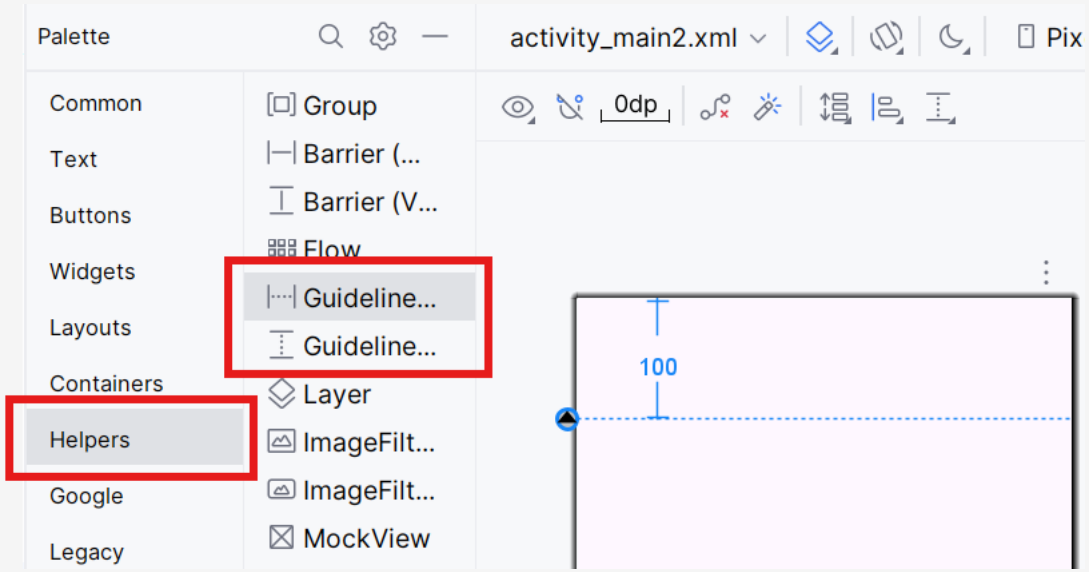
Guías (Guidelines)

- Una **Guideline** debe tener una orientación y un **ID** para que otras **Views** puedan referenciarla.

Atributo	**Descripción**
android:id	Necesario para que otras *Views* se restrinjan a ella.
android:orientation	Define si es una línea **vertical** u **horizontal**.
layout_constraintGuide_begin o layout_constraintGuide_end o layout_constraintGuide_percent	Define la posición de la línea. **Solo se usa uno de estos tres.**

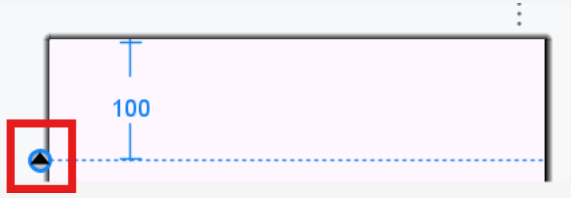
Guías (Guidelines)

- Las Guías se encuentran en el apartado Helpers de la Paleta de componentes.
- Podemos arrastrar una guía al diseño para crearla.



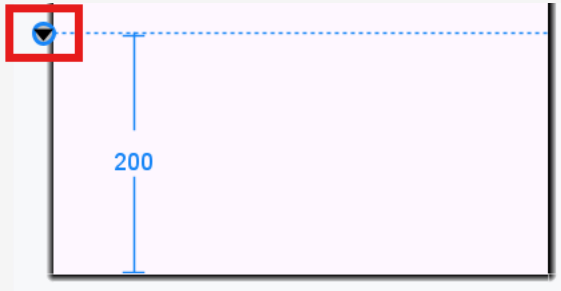
Guías (Guidelines)

- Una vez colocada podemos cambiar el tipo de guía pulsando en el botón que aparece pegado a un extremo de la guía:



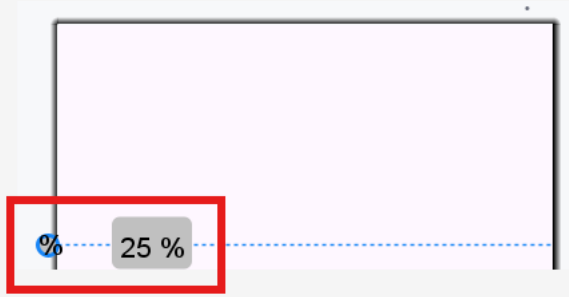

```
app:layout_constraintGuide_begin="100dp"
```

Indicamos la posición como una distancia en píxeles desde la parte superior




```
app:layout_constraintGuide_end="200dp"
```

Indicamos la posición como una distancia en píxeles desde la parte inferior

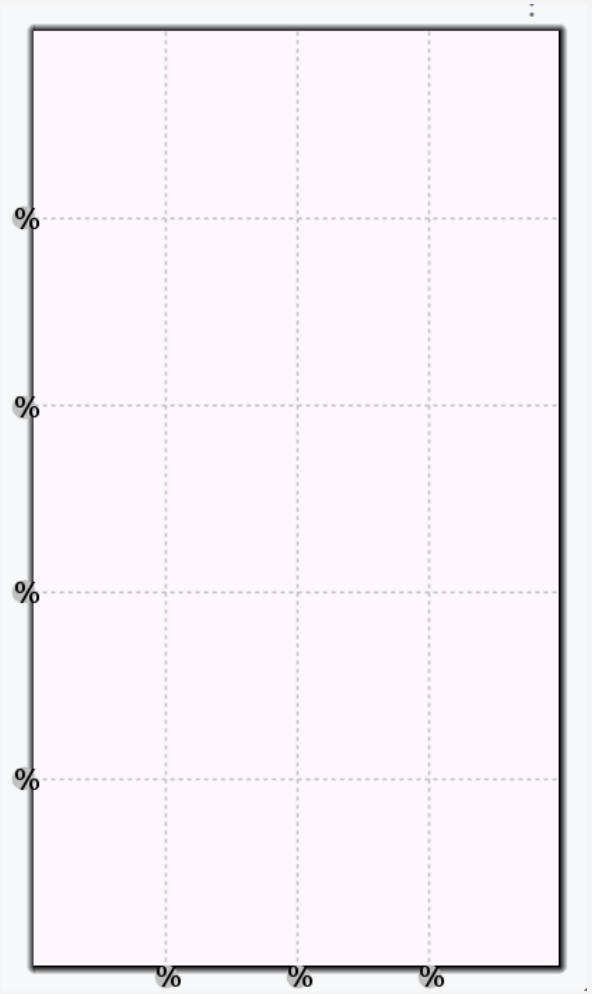



```
app:layout_constraintGuide_percent="0.25"
```

Indicamos la posición en forma de porcentaje que oscila entre 0 y 1

Ejercicio

- Crea un nuevo proyecto que se llame **GuiasImagenes**
- En el diseño del activity principal crea las siguientes guías horizontales:
  - 20% con id **guidelineh1**
  - 40% con id **guidelineh2**
  - 60% con id **guidelineh3**
  - 80% con id **guidelineh4**
- Crea las siguientes guías verticales
  - 25% con id **guidelinev1**
  - 50% con id **guidelinev2**
  - 75% con id **guidelinev3**



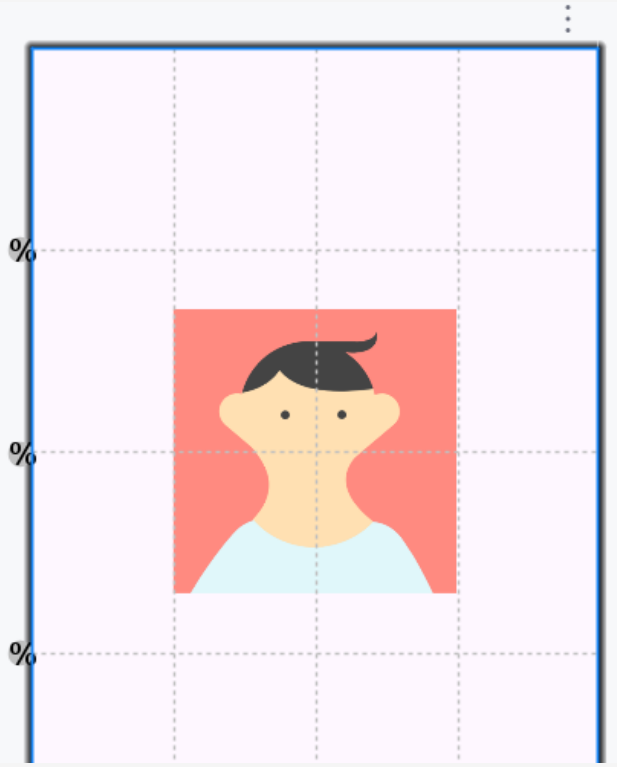
ImageView

- Es una subclase de View que se usa para mostrar un archivo de imagen.

Atributo	Descripción
android:id	Identificador único.
android:layout_width	Ancho del componente
android:layout_height	Altura del componente.
app:srcCompat	**Especifica la fuente de la imagen** (el recurso) que se cargará por defecto. Se refiere a un archivo dentro de la carpeta **res/drawable**. Se puede usar **android:src** pero presenta problemas de compatibilidad con versiones Android anteriores a 5.0(Lollipop)
android:contentDescription	**Importante para accesibilidad.** Describe el contenido de la imagen para usuarios con lectores de pantalla.

ImageView - Ejercicio

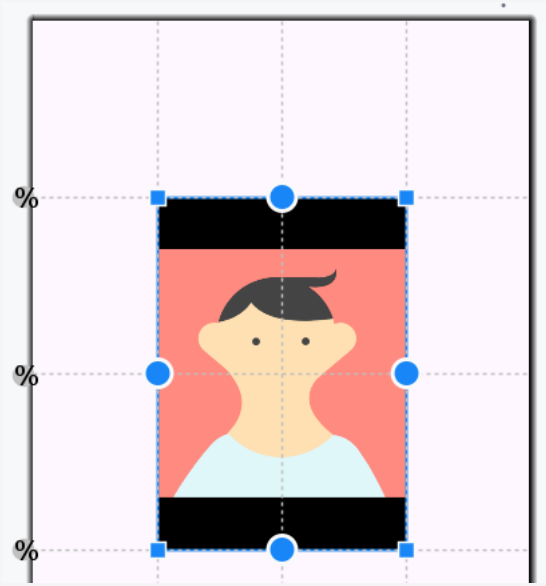
- Añade un ImageView al proyecto que tenga las siguientes constraints:
  - Por arriba a **guidelineh1**
  - Por abajo a **guidelineh3**
  - Por la derecha a **guidelinev3**
  - Por la izquierda a **guidelinev1**
- **android:id** ivImagen
- **android:layout_width** "0dp“
- **android:layout_height** "0dp"



ImageView – Atributo background

- El atributo **android:background** se usa para establecer el fondo de un ImageView. Podemos establecer un color o un elemento de diseño en el fondo de un ImageView.

- Establecemos el valor **android:background** = "#000"

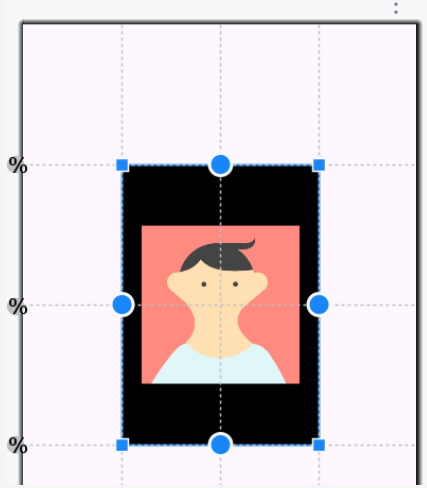


ImageView – Atributo padding

- El atributo padding se utiliza para establecer un margen desde la izquierda, la derecha, la parte superior o la parte inferior de la vista de imagen.

  - **paddingRight**: establece el margen desde el lado derecho de la vista de la imagen.
  - **paddingLeft**: establece el margen desde el lado izquierdo de la vista de la imagen.
  - **paddingTop**: establece el margen desde la parte superior de la vista de la imagen.
  - **paddingBottom**: establece el margen desde la parte inferior de la vista de la imagen.
  - **padding**: establece el margen desde todos los lados de la vista de la imagen.

- Establecemos: **android:padding**="20dp"



ImageView – Atributo scaleType

- El atributo **android:scaleType** permite controlar como redimensionar o mover el contenido de la imagen para que coincida con el rectángulo del **ImageView**.

- **android:scaleType="center"** Centra la imagen en el view sin realizar escalado

ImageView – Atributo scaleType

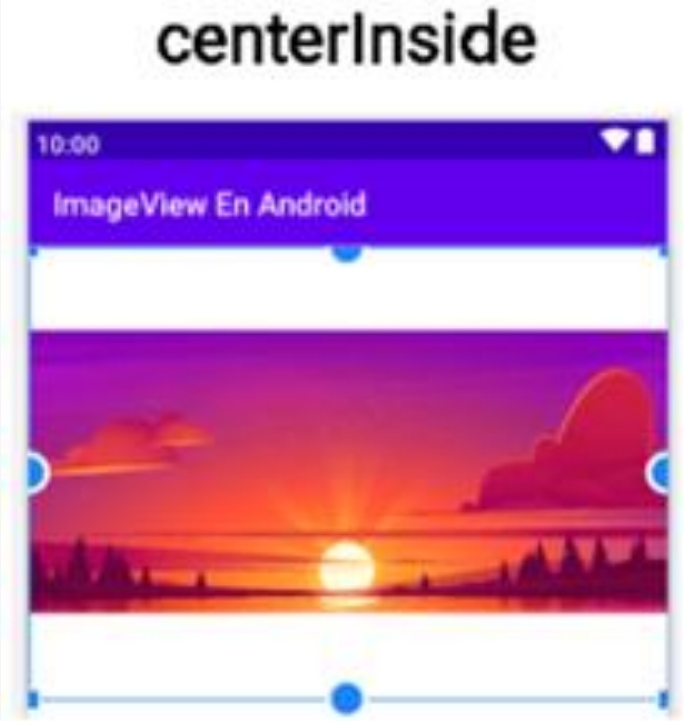
- **android:scaleType="centerCrop"**

Escala el ancho y alto de la imagen, manteniendo la relación de aspecto. De tal forma que ambas dimensiones sean iguales o mayores a las del view. Luego es centrada.

ImageView – Atributo scaleType

- **android:scaleType="centerInside"**

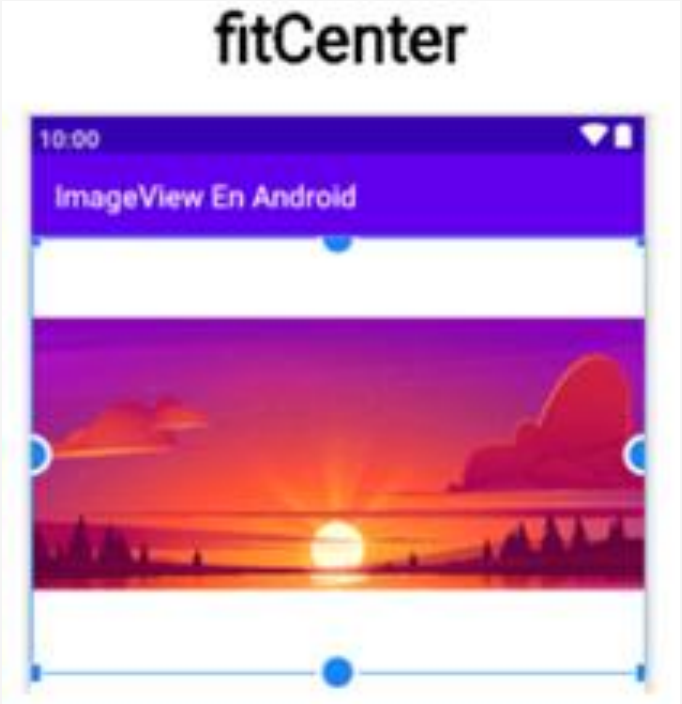
Igual que **centerCrop**, sólo que el escalado hace que las dimensiones de la imagen sean iguales o menores a las del View.



ImageView – Atributo scaleType

- **android:scaleType=“fitCenter"**

Computa una matriz que mantenga el ratio de la imagen, asegurándose que encaje en el view por completo. Al menos uno de los ejes (X o Y) será ajustado y luego se centrará el resultado final.



ImageView – Atributo scaleType

- **android:scaleType=“fitEnd"**

Igual que **fitCenter**, solo que se ajusta la imagen hacia el borde inferior derecho del view.

ImageView – Atributo scaleType

- **android:scaleType=“fitStart"**

Igual que **fitCenter**, solo que se ajusta la imagen hacia el borde superior izquierdo del view.

ImageView – Atributo scaleType

- **android:scaleType=“fitXY"**

Escala el alto y ancho independientemente para que coincidan con el tamaño del view. No se conservará el ratio.

ImageView – Atributo adjustViewBounds

- Este atributo permite conservar la relación de aspecto de la imagen manteniendo el control de una dimensión.

- Cuando establecemos adjustViewBounds a true, le estamos diciendo al ImageView (no al drawable) que ajuste sus límites para preservar la relación de aspecto de su drawable.

```
android:layout_width="wrap_content"
android:layout_height="18dp"
android:adjustViewBounds="true"
```

- Fijamos el alto y dejamos que el ancho se ajuste al contenido.
- Al activar adjustViewBounds el ImageView adaptará sus dimensiones para mantener las proporciones.

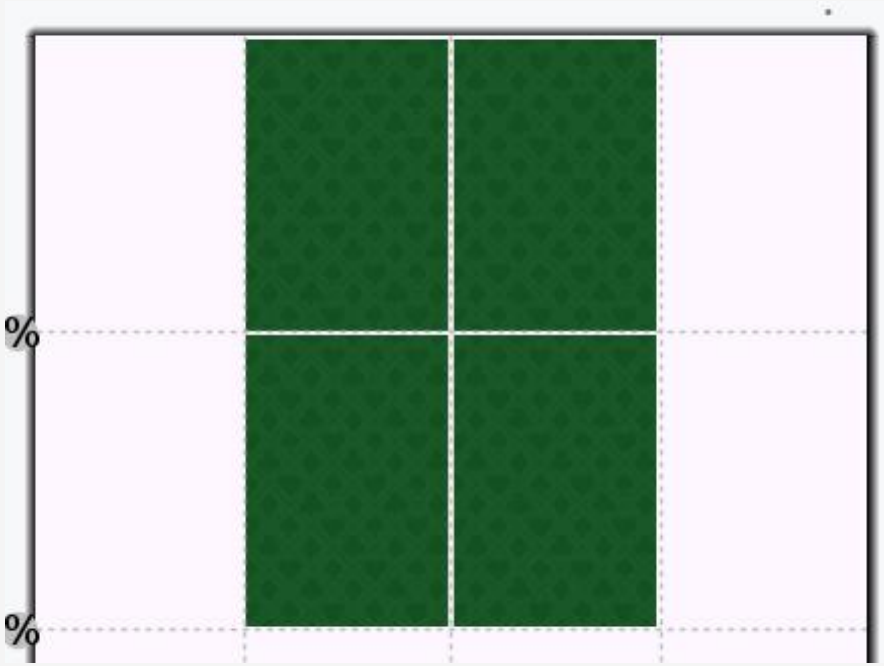
ImageView – Atributo scaleType

- **android:scaleType=“matrix"**

Usa la matriz de transformación que se proporcione a través del método setImageMatrix(Matrix)

Ejercicio

- Añade 4 **ImageView** y llamálos:
  - ivCasilla1
  - ivCasilla2
  - ivCasilla3
  - ivCasilla4
- Añade restricciones respecto a las guías para colocarlos como se ve en la imagen.
- Deja 1 pixel de margen por cada lado de cada ImageView.
- Establece como imagen de cada ImageView **fondoMemorion.png.**
- Utiliza la propiedad **scaleType** para que la imagen se ajuste como en la imagen ocupando todo el espacio dentro de las restricciones.



Ejercicio

```
<ImageView
    android:id="@+id/ivCasilla1"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginLeft="1dp"
    android:layout_marginTop="1dp"
    android:layout_marginRight="1dp"
    android:layout_marginBottom="1dp"
    android:adjustViewBounds="true"
    android:scaleType="centerCrop"
    app:layout_constraintBottom_toTopOf="@+id/guidelineh1"
    app:layout_constraintLeft_toRightOf="@id/guidelinev1"
    app:layout_constraintRight_toLeftOf="@id/guidelinev2"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/fondomemorion" />

<ImageView
    android:id="@+id/ivCasilla2"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="1dp"
    android:layout_marginTop="1dp"
    android:layout_marginEnd="1dp"
    android:layout_marginBottom="1dp"
    android:adjustViewBounds="false"
    android:scaleType="centerCrop"
    app:layout_constraintBottom_toTopOf="@+id/guidelineh1"
    app:layout_constraintEnd_toStartOf="@+id/guidelinev3"
    app:layout_constraintStart_toStartOf="@+id/guidelinev2"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/fondomemorion" />
```

Ejercicio

```
<ImageView
    android:id="@+id/ivCasilla3"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="1dp"
    android:layout_marginTop="1dp"
    android:layout_marginEnd="1dp"
    android:layout_marginBottom="1dp"
    android:scaleType="centerCrop"
    app:layout_constraintBottom_toTopOf="@+id/guidelineh2"
    app:layout_constraintEnd_toStartOf="@+id/guidelinev2"
    app:layout_constraintStart_toStartOf="@+id/guidelinev1"
    app:layout_constraintTop_toTopOf="@+id/guidelineh1"
    app:srcCompat="@drawable/fondomemorion" />

<ImageView
    android:id="@+id/ivCasilla4"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="1dp"
    android:layout_marginTop="1dp"
    android:layout_marginEnd="1dp"
    android:layout_marginBottom="1dp"
    android:scaleType="centerCrop"
    app:layout_constraintBottom_toTopOf="@+id/guidelineh2"
    app:layout_constraintEnd_toStartOf="@+id/guidelinev3"
    app:layout_constraintStart_toStartOf="@+id/guidelinev2"
    app:layout_constraintTop_toBottomOf="@+id/guidelineh1"
    tools:srcCompat="@drawable/fondomemorion" />
```

Ejercicio

Declaramos variables para hacer referencia a los ImageView

```
    lateinit var ivCasilla1: ImageView
    lateinit var ivCasilla2: ImageView
    lateinit var ivCasilla3: ImageView
    lateinit var ivCasilla4: ImageView
```

Declaramos una lista con los identificadores de las imágenes de frutas

```
  val frutas=listOf(R.drawable.fruta1,R.drawable.fruta2,R.drawable.fruta3,R.drawable.fruta4)
```

Declaramos una lista con valores booleanos. Si el valor es **false** querrá decir que la casilla correspondiente está boca abajo y si es **true** boca arriba

```
  val estado=mutableListOf(false,false,false,false)
```

Ejercicio

- Inicializa las variables que referencian a los ImageView
- Crea una función que se llame inicializar que ponga todos los ImageView mostrando la imagen de casilla tapada: **fondoMemorion.png** e invócala después de inicializar las variables de referencia.
- Implementa el evento **onClick** de cada **ImageView** de modo:
  - Si la casilla booleana correspondiente es **false** (la casilla está boca abajo) mostraremos la imagen de la fruta correspondiente del array de frutas.
  - Si la casilla booleana correspondiente es **true** (casilla boca arriba) mostraremos la imagen correspondiente a la casilla tapada.