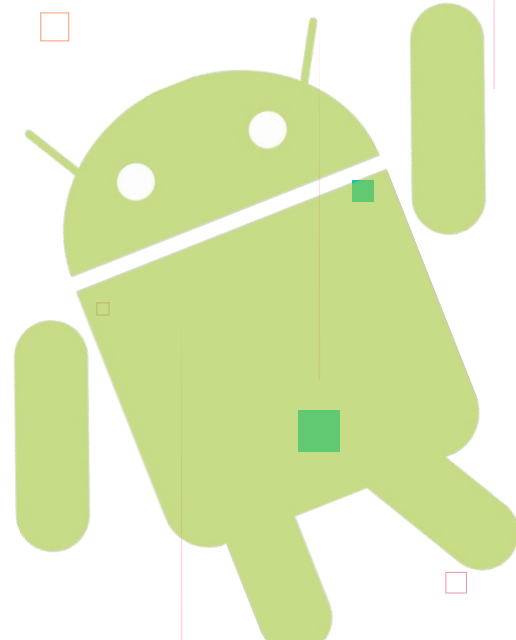


UT-03: Componentes de Interfaz de Usuario

Programación Multimedia y
Dispositivos Móviles



OBJETIVOS Y CRITERIOS

Objetivos

(RA-2) Desarrolla aplicaciones para dispositivos móviles analizando y empleando las tecnologías y librerías específicas.

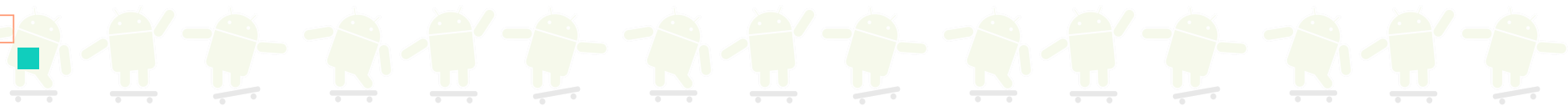
Criterios

- Se ha generado la estructura de clases necesarias para la aplicación.
- Se han analizado y utilizado las clases que modelan ventanas, menús, alertas y controles para el desarrollo de aplicaciones gráficas sencillas.
- Se han realizado pruebas de interacción usuario-aplicación para optimizar las aplicaciones desarrolladas a partir de emuladores.
- Se han empaquetado y desplegado las aplicaciones desarrolladas en dispositivos móviles reales.
- Se han documentado los procesos necesarios para el desarrollo de las aplicaciones.

INTERFAZ DE USUARIO

La interfaz de usuario (UI) de una app es todo aquello que el usuario puede ver y con lo que puede interactuar en ella.

Android ofrece una variedad de componentes UI previamente compilador y además, permite crear componentes personalizados.



INTERFAZ DE USUARIO

Diseño

Un diseño define la estructura de una interfaz de usuario en una aplicación, por ejemplo, una actividad.

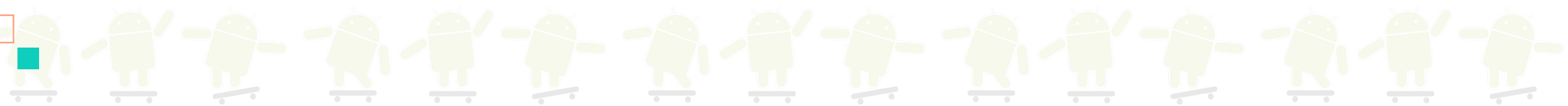
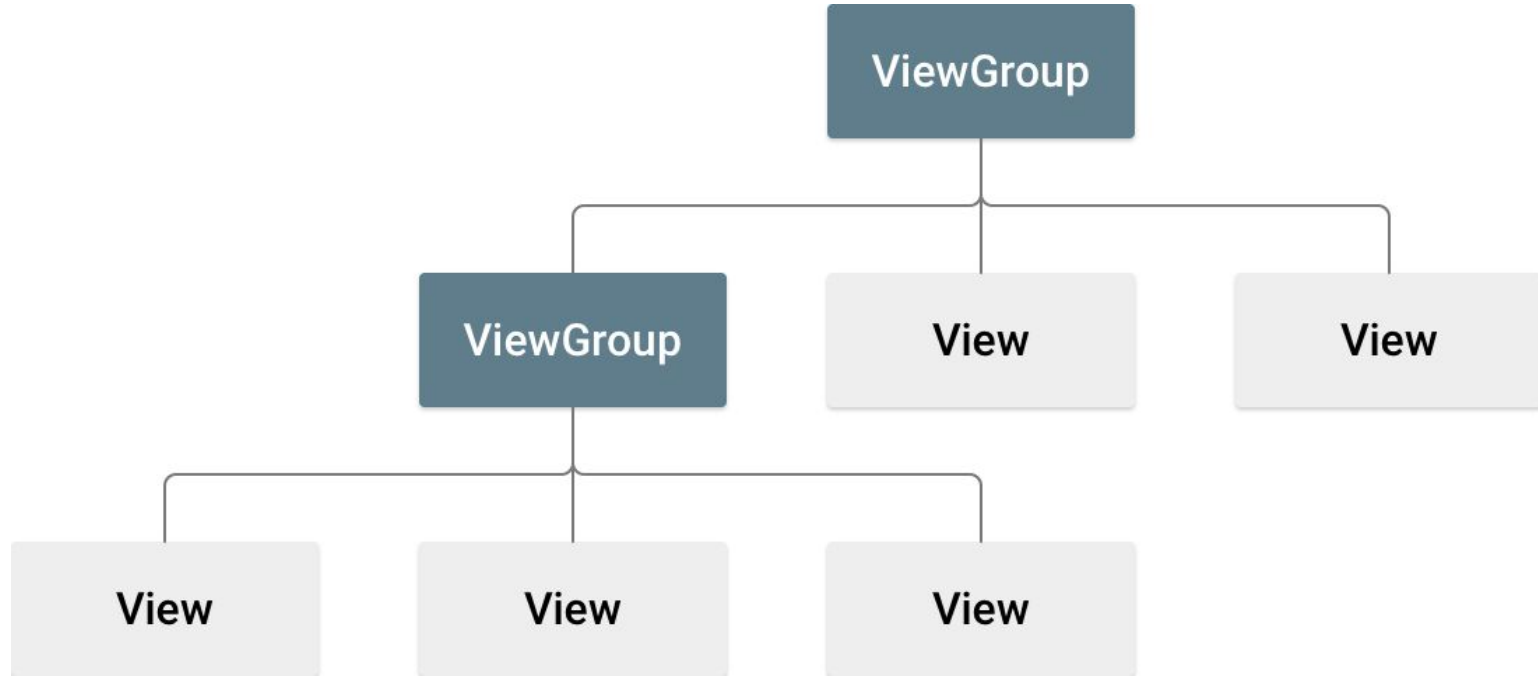
Todos los elementos de diseño se crean usando una jerarquía de objetos **View** y **ViewGroup**.

Una **View** muestra un elemento que el usuario puede ver y con el que puede interactuar.

Un **ViewGroup** es un contenedor invisible que define la estructura de diseño de View y otros objetos ViewGroup.



INTERFAZ DE USUARIO



INTERFAZ DE USUARIO

Diseño

Un diseño se puede declarar de dos maneras:

- Declarar elementos de la UI en XML.
- Crear una instancia de elementos de diseño durante el tiempo de ejecución, es decir, mediante programación.



INTERFAZ DE USUARIO

Diseño

Declarar elementos en la UI

```
<TextView  
    android:layout_width="0dp"  
    android:layout_height="match_parent"  
    android:layout_weight="2"  
    android:textAlignment="center"  
    tools:text="Hola" />
```



INTERFAZ DE USUARIO

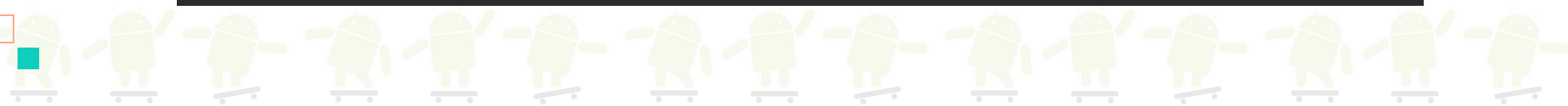
Diseño

Declarar elementos en mediante programación:

```
val parentLayout : RelativeLayout = findViewById(R.id.layout_root)

val textView: TextView = TextView(context: this)
textView.text = "Hola"
textView.setTextColor(ContextCompat.getColor(context: this, R.color.black))

parentLayout.addView(textView)
```



INTERFAZ DE USUARIO

Diseño

- Declarar la UI en XML permite separar la presentación del código de la app que controla el comportamiento.
- Declarar la UI en XML facilita la creación de distintos diseños para distintas pantallas y orientaciones.



INTERFAZ DE USUARIO

Diseño

Las vistas en XML tienen multitud de atributos que se van heredando de un widget a otro.

```
<TextView
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="2"
    android:textAlignment="center"
    tools:text="Hola" />
```



INTERFAZ DE USUARIO

contenedores
de elementos

INTERFAZ DE USUARIO

Contenedores (ViewGroup)

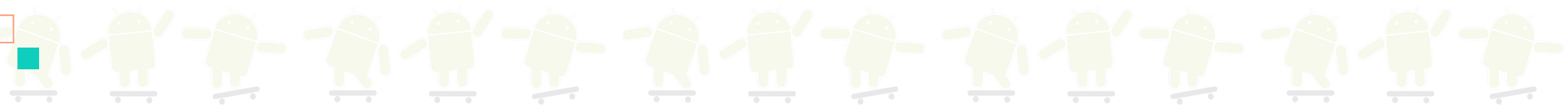
- **LinearLayout:** Las vistas se organizan de forma lineal: horizontal o vertical.
- **RelativeLayout:** Las vistas se organizan tomando como referencia su padre o hermanas.
- **FrameLayout:** Las vistas se cargan una encima de otra.
- **ConstraintLayout:** Igual que un RelativeLayout pero con nuevos atributos y preparado para hacerlo de forma visual.



INTERFAZ DE USUARIO

Contenedores (ViewGroup)

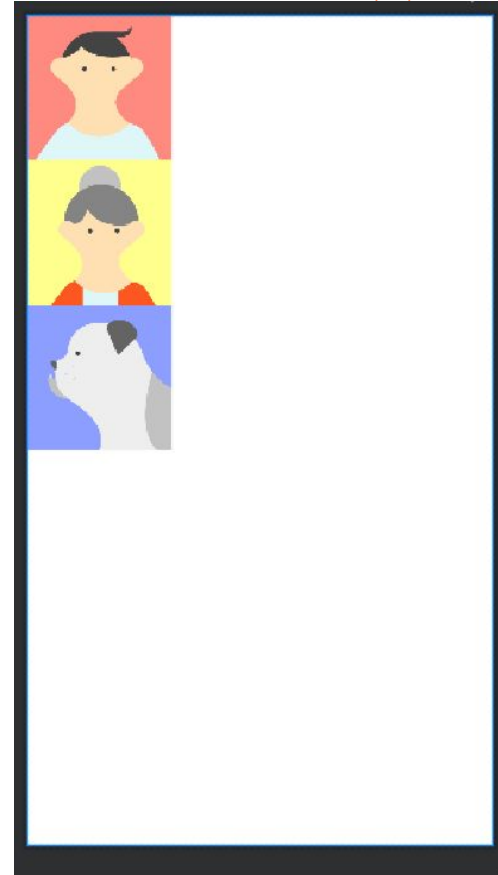
- TableLayout
- GridLayout



INTERFAZ DE USUARIO

LinearLayout

- Vertical



INTERFAZ DE USUARIO

LinearLayout

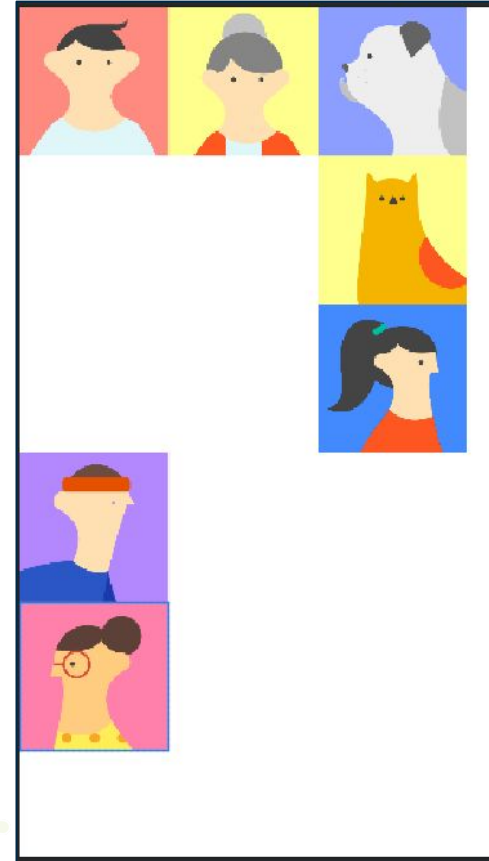
- Horizontal



INTERFAZ DE USUARIO

LinearLayout

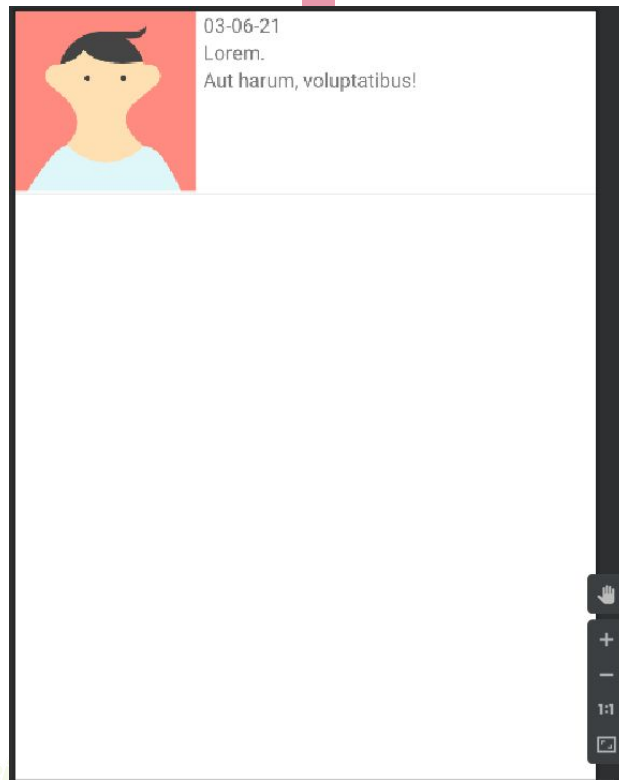
- Vertical
- Horizontal



INTERFAZ DE USUARIO

LinearLayout

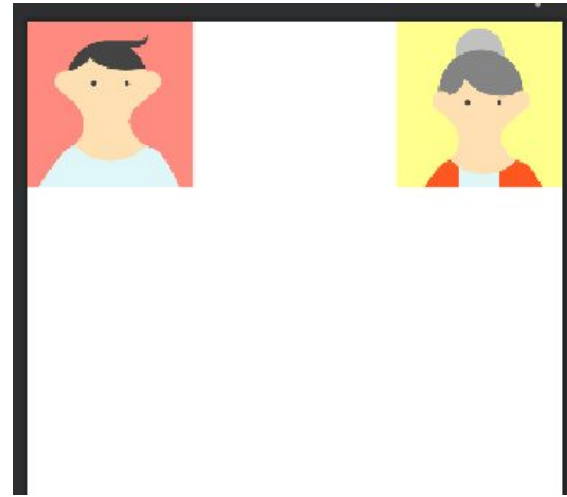
- Vertical
- Horizontal
- Imagenes
- Texto



INTERFAZ DE USUARIO

LinearLayout

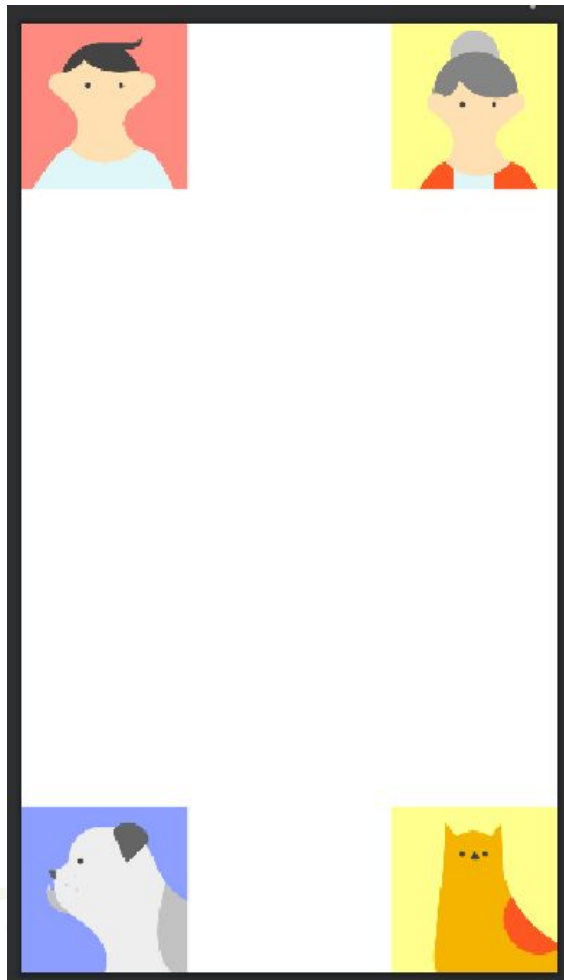
¿?



INTERFAZ DE USUARIO

RelativeLayout

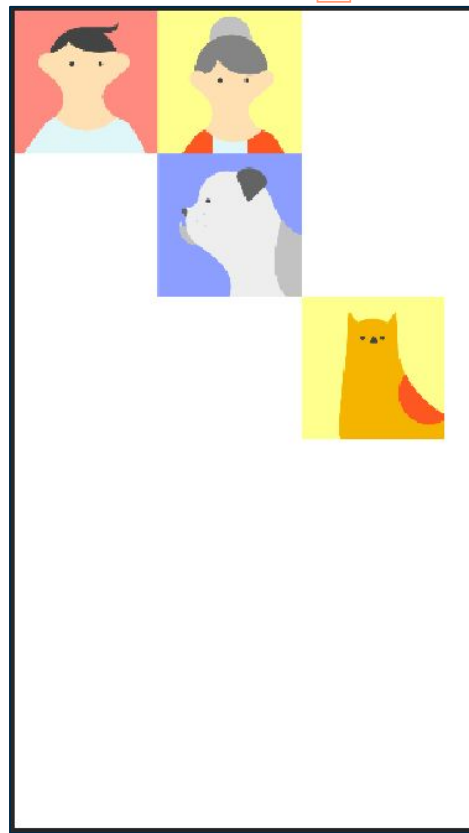
- alignParentEnd
- alignParentBottom
- alignParentStart



INTERFAZ DE USUARIO

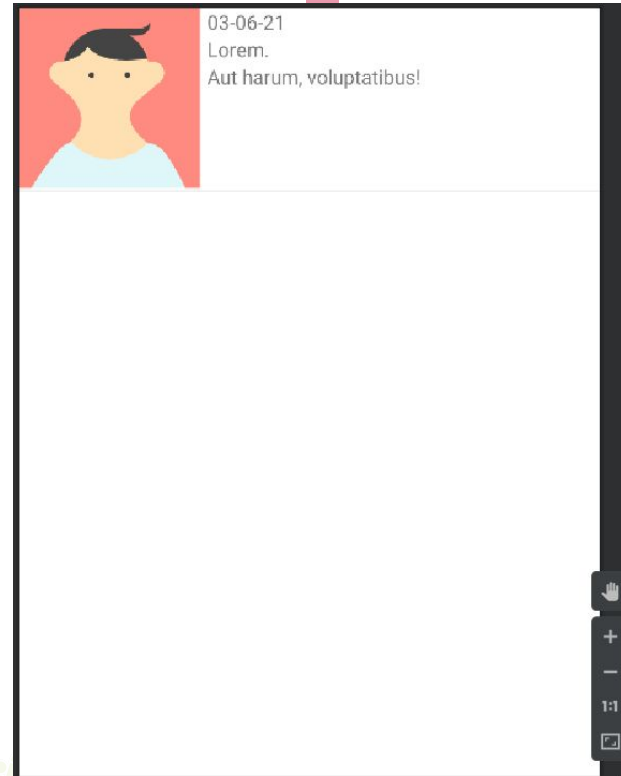
RelativeLayout

- alignParentEnd
- alignParentBottom
- alignParentStart



INTERFAZ DE USUARIO

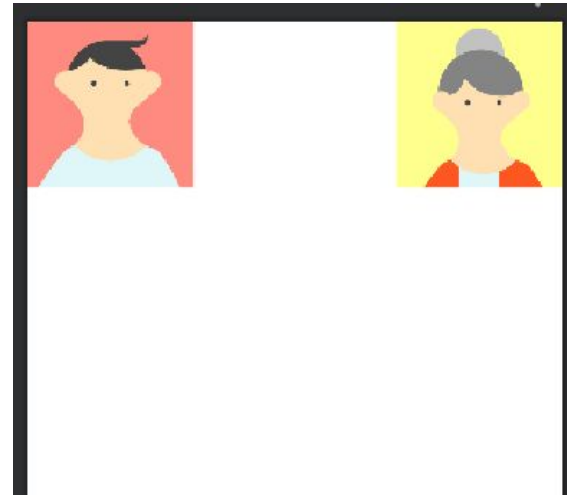
RelativeLayout



INTERFAZ DE USUARIO

FrameLayout

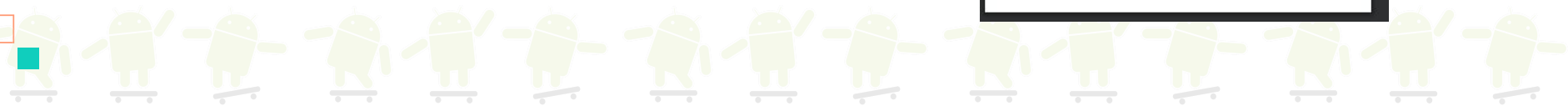
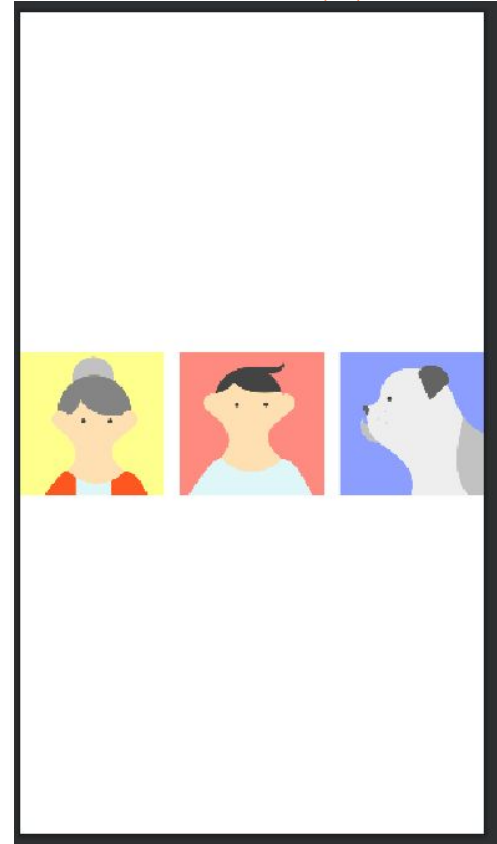
¿?



INTERFAZ DE USUARIO

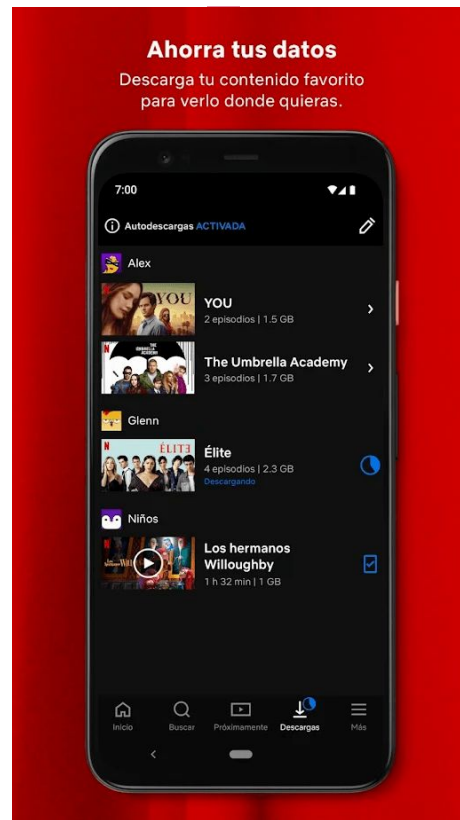
FrameLayout

- `layout_gravity`



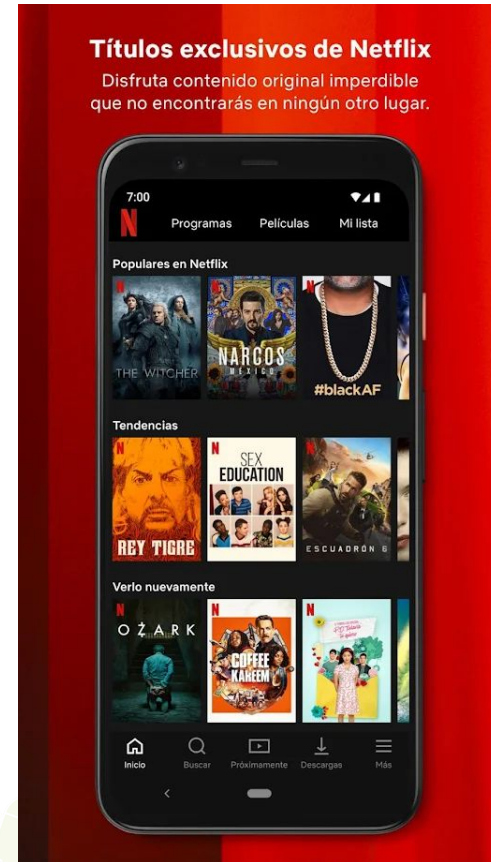
INTERFAZ DE USUARIO

Ejercicio 01



INTERFAZ DE USUARIO

Ejercicio 02



INTERFAZ DE USUARIO

Contenedores (ViewGroup)

ConstraintLayouts

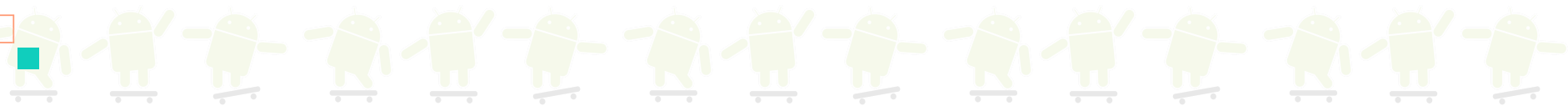
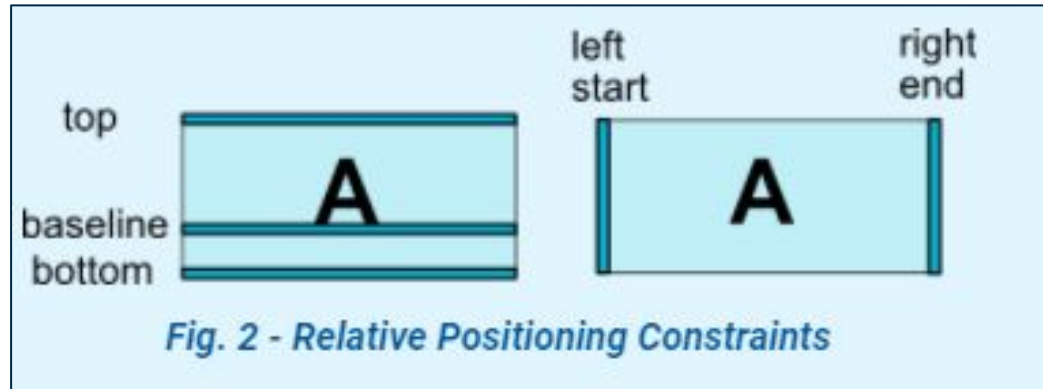
- Son muy parecidos a los RelativeLayouts. Se busca una relación con otras vistas al mismo nivel o superior.



INTERFAZ DE USUARIO

Contenedores (ViewGroup)

ConstraintLayouts



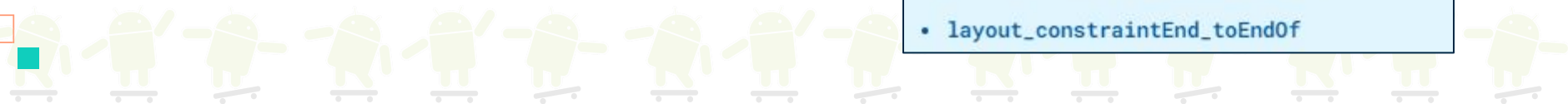
INTERFAZ DE USUARIO

Contenedores (ViewGroup)

ConstraintLayouts

Atributos principales que veremos.

- `layout_constraintLeft_toLeftOf`
- `layout_constraintLeft_toRightOf`
- `layout_constraintRight_toLeftOf`
- `layout_constraintRight_toRightOf`
- `layout_constraintTop_toTopOf`
- `layout_constraintTop_toBottomOf`
- `layout_constraintBottom_toTopOf`
- `layout_constraintBottom_toBottomOf`
- `layout_constraintBaseline_toBaselineOf`
- `layout_constraintStart_toEndOf`
- `layout_constraintStart_toStartOf`
- `layout_constraintEnd_toStartOf`
- `layout_constraintEnd_toEndOf`



INTERFAZ DE USUARIO

Contenedores (ViewGroup)

ConstraintLayouts

- Ampliar conceptos aquí o tomar como referencia de consultas (importante):

<https://developer.android.com/reference/androidx/constraintlayout/widget/ConstraintLayout>

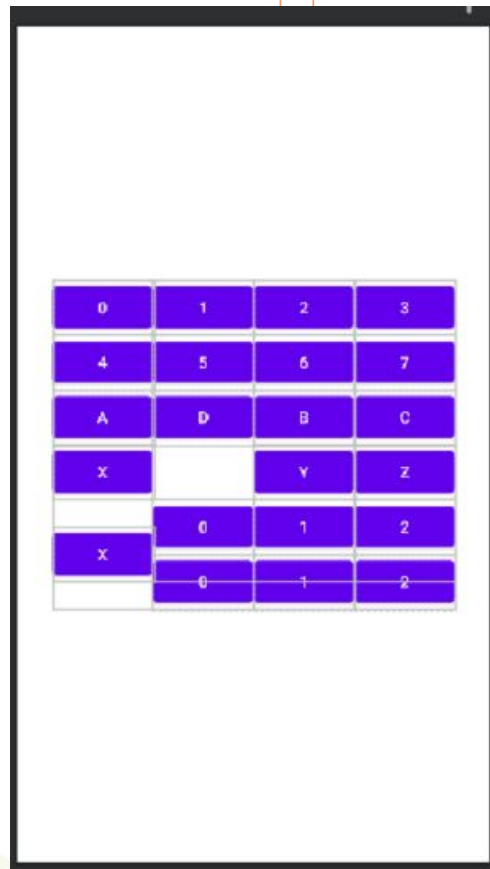


INTERFAZ DE USUARIO

Contenedores (ViewGroup)

GridLayouts

(ver repositorio en GitHub)



INTERFAZ DE USUARIO

Contenedores (ViewGroup)

TableLayouts

(ver repositorio en GitHub)

Columna 1	Columna 2	
Columna 1	Columna 2	
Columna 3		
Columna 3		
Columna 1	Columna 2	
Columna 1	Columna 2	
	Columna 1	
Columna 1	Columna 2	
Columna 1	Columna 2	Columna 3
Columna 1	Columna 2	





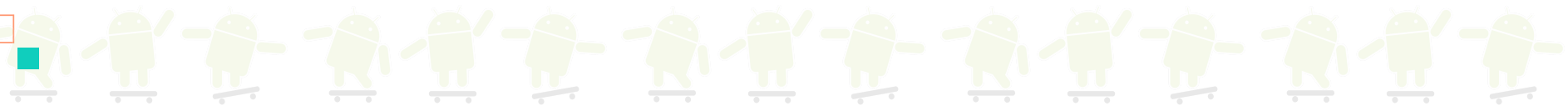
INTERFAZ DE USUARIO

elementos de la vista

INTERFAZ DE USUARIO

Vistas (View)

- **TextView**: Etiquetas para mostrar un texto.
- **EditTextView**: Introducir texto.
- **ImageView**: Visualizar una imagen.
- **Button**: Crear un botón.
- **Spinner**: Lista desplegable.
- **Checkbox**



INTERFAZ DE USUARIO

Vistas (View)

- **RadioButton**
- **RatingBar**
- **Switch**
- **SeekBar**
- **SearchView**
- **ProgressBar** (determinado/indeterminado)



The background features several thin, vertical lines in shades of pink and orange. Scattered throughout are small squares in teal, orange, and pink. The text is centered on the page.

INTERFAZ DE USUARIO

eventos

INTERFAZ DE USUARIO

Eventos

- **OnClickListener**
- **OnLongClickListener**
- **OnFocusChange**
- **OnKeyListener**
- **OnTouchListener**

Fuente: <https://developer.android.com/guide/topics/ui/ui-events?hl=es-419>



INTERFAZ DE USUARIO

Eventos: OnClickListener

Se llama a este método cuando el usuario toca el elemento (en el modo táctil) o se centra en el elemento con las teclas de navegación o la bola de seguimiento y presiona la tecla "Intro" adecuada o la bola de seguimiento.



INTERFAZ DE USUARIO

Eventos: `OnLongClickListener`

Se llama a este método cuando el usuario mantiene presionado el elemento (en el modo táctil) o se centra en el elemento con las teclas de navegación o la bola de seguimiento y mantiene presionada la tecla "Intro" adecuada o la bola de seguimiento (durante un segundo).



INTERFAZ DE USUARIO

Eventos: `OnFocusChangeListener`

Se llama a este método cuando el usuario navega hacia el elemento o sale de él utilizando las teclas de navegación o la bola de seguimiento.

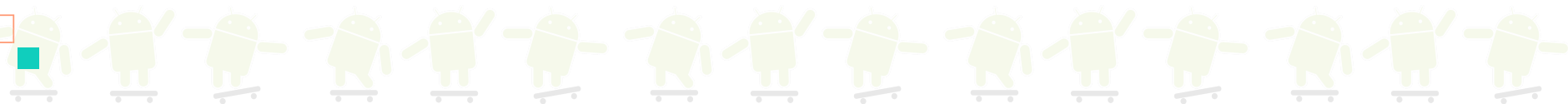


INTERFAZ DE USUARIO

Eventos: `OnFocusChangeListener`

Se llama a este método cuando el usuario navega hacia el elemento o sale de él utilizando las teclas de navegación o la bola de seguimiento.

Reto: Crea un `EditText` y muestra un Log cuando tenga el focus y otro cuando lo pierda.



INTERFAZ DE USUARIO

Eventos: OnKeyListener

Se llama a este método cuando el usuario se centra en el elemento y presiona o suelta una tecla de hardware del dispositivo.

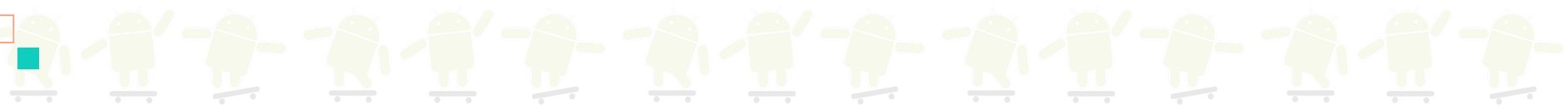
Reto: Crea un EditText y muestra un Log cuando la tecla pulsada sea la 'm'.



INTERFAZ DE USUARIO

Eventos: `OnTouchListener`

Se llama a este método cuando el usuario realiza una acción calificada como evento táctil, por ejemplo, presionar, soltar o cualquier gesto de movimiento en la pantalla (dentro de los límites del elemento).



INTERFAZ DE USUARIO

RecyclerView

