



Unidad 2 / Escenario 3

Lectura fundamental

Interfaz de usuario

Contenido

- 1 Características de la interfaz de usuario
- 2 Diseño de la interfaz de usuario

Palabras clave: *layout, vista, XML.*

1. Características de la interfaz de usuario

Una interfaz gráfica de usuario de una aplicación de Android está formada por una jerarquía de Vistas y Grupos de Vistas (ver Figura 1). Las vistas son, generalmente, la interfaz de usuario, como botones y cajas de texto, en tanto que los grupos de vistas representan los contenedores de vistas que definen el orden de estas:

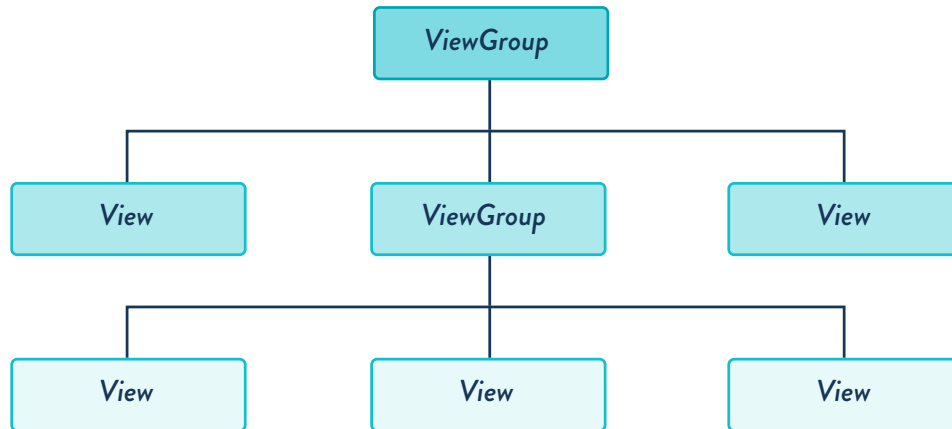


Figura 1. Jerarquía de Vistas

Fuente: Politécnico Grancolombiano

Para agrupar varios elementos tipo Vista se utilizan los objetos *layouts*, los cuales contienen una o varias vistas. Los *layouts* controlan el comportamiento y la ubicación de cada uno de los elementos dentro de la interfaz gráfica. A continuación, se explican los diferentes *layouts* disponibles en Android Studio (ver Figura 2), los cuales se encuentran en la paleta del diseño de la interfaz, localizada en el archivo XML de la carpeta *layout*, ubicada en la ventana *Project* de Android Studio.

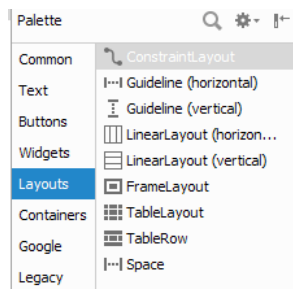


Figura 2. Captura de pantalla Layouts Android Studio en uso de software gratuito.

Fuente: elaboración propia

1.1. *ConstraintLayout*

Crea diseños extensos con una jerarquía sin grupos de vistas anidadas; aquí, cada una de las vistas se alinea con respecto a las que se encuentran en el mismo nivel o con las que se encuentran inmediatamente en el nivel superior.

1.2. *Guideline (horizontal/vertical)*

Representa una guía visual que sirve para alinear las vistas de forma horizontal o vertical.

1.3. *LinearLayout (horizontal/vertical)*

Agrupar un conjunto de vistas de forma horizontal o vertical, es decir, una al lado de la otra o una debajo de la otra, respectivamente.

1.4. *FrameLayout*

Coloca todas las vistas hijas alineadas en la esquina superior izquierda, de tal forma que cada vista queda oculta para la siguiente, a no ser que la última tenga transparencia.

1.5. *TableLayout*

Distribuye las vistas en una tabla o matriz, con filas y columnas. Cada fila está representada por *TableRow* y las columnas se van ajustando a medida que se adicionan vistas a cada fila.

1.6. *TableRow*

Permite agregar nuevas filas a un *TableLayout* para organizar las vistas de forma horizontal.

1.7. Space

Deja crear espacios adicionales entre vistas.

Para cada uno de los *layouts* vistos anteriormente, existen una serie de atributos para las diferentes vistas que los contienen. En la Tabla 1 se describen algunos:

Tabla 1. Atributos de las vistas para interfaz gráfica

Categoría	Atributo	Descripción
Posicionamiento relativo	<code>layout_constraintTop_toTopOf</code>	Alinea la parte superior de una vista con la parte superior de otra.
	<code>layout_constraintTop_toBottomOf</code>	Alinea la parte superior de una vista con la parte inferior de otra.
	<code>layout_constraintBottom_toTopOf</code>	Alinea la parte inferior de la vista con la parte superior de otra.
	<code>layout_constraintBottom_toBottomOf</code>	Alinea la parte inferior de una vista con la parte inferior de otra.
	<code>layout_constraintLeft_toTopOf</code>	Alinea la parte izquierda de una vista con la parte superior de otra.
	<code>layout_constraintLeft_toBottomOf</code>	Alinea la parte izquierda de una vista con la parte inferior de otra.
	<code>layout_constraintLeft_toLeftOf</code>	Alinea la izquierda de una vista a la izquierda de otra.
	<code>layout_constraintLeft_toRightOf</code>	Alinea la izquierda de una vista a la derecha de otra.
	<code>layout_constraintRight_toTopOf</code>	Alinea la derecha de una vista a la parte superior de otra.
	<code>layout_constraintRight_toBottomOf</code>	Alinea la derecha de una vista a la parte inferior de otra.
	<code>layout_constraintRight_toLeftOf</code>	Alinea la derecha de una vista a la izquierda de otra.
	<code>layout_constraintRight_toRightOf</code>	Alinea la derecha de una vista a la derecha de otra.

Márgenes	android:layout_margin	Especifica el espacio adicional a los lados izquierdo, superior, derecho e inferior de una vista.
	android:layout_marginBottom	Especifica el espacio en la parte inferior de una vista.
	android:layout_marginEnd	Especifica el espacio en el lado final de una vista.
	android:layout_marginHorizontal	Especifica el espacio en los lados izquierdo y derecho de una vista.
	android:layout_marginLeft	Especifica el espacio en el lado izquierdo de una vista.
	android:layout_marginRight	Especifica el espacio en el lado derecho de una vista.
	android:layout_marginStart	Especifica el espacio en el lado de inicio de una vista.
	android:layout_marginTop	Especifica el espacio en el lado superior de una vista.
	android:layout_marginVertical	Especifica el espacio en los lados superior e inferior de una vista.
Posicionamiento circular	layout_constraintCircle	Hace referencia a otro ID de la vista.
	layout_constraintCircleRadius	Especifica la distancia al centro de la vista.
	layout_constraintCircleAngle	Especifica en qué ángulo debe estar la vista (en grados, de 0 a 360).
Restricciones de dimensiones	android:minWidth	Establece el ancho mínimo para la vista.
	android:minHeight	Define la altura mínima para la vista.
	android:maxLength	Establece el ancho máximo para la vista.
	android:maxLength	Establece la altura máxima para la vista.
Lineal	android:orientation.	Define si las vistas secundarias se muestran en una fila o columna.
	android:layout_height	Especifica la altura básica de la vista.
	android:layout_width	Establece el ancho básico de la vista.
	android:layout_weight	Especifica cómo el diseño lineal divide el espacio restante entre las vistas que contiene.
Marco	android:foreground	Define las características de la primera vista.
	android:foregroundGravity	Define la ubicación de la vista.
	android:measureAllChildren	Determina si se miden todas las vistas o solo aquellas que están visibles o invisibles.
	android:visibility	Determina si la vista es visible o no.
Tabla	android:collapseColumns	Une dos o más columnas en una.
	android:shrinkColumns	Reduce el ancho de la columna.
	android:stretchColumns	Expande el ancho de la columna.

Fuente: elaboración propia

2. Diseño de la interfaz de usuario

Luego de conocer las características de la interfaz gráfica en una aplicación para Android con Android Studio, es necesario identificar algunas vistas con atributos que se pueden tomar para realizar un diseño de la interfaz gráfica de usuario.

Después de crear un proyecto en Android Studio, en la ventana *Project*, bajo la carpeta *layout*, ubicada en *res* de la aplicación en la ventana del proyecto, se encuentra el archivo XML de la actividad principal, en donde se puede trabajar en vista de diseño o con comandos XML.

A continuación, se muestran los diferentes *layouts* con algunas vistas que contienen algunos de sus atributos.

2.1. *ConstraintLayout*

La Tabla 2 muestra el *ConstraintLayout* con dos vistas: un *textView* (etiqueta de texto para mostrar datos) y un *editText* (campo de texto para captura de datos).

Tabla 2. Atributos de las vistas *textView* y *editText*

	
Código XML	Comentario
<code><?xml version="1.0" encoding="utf-8"?></code>	
<code><android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"</code>	Especificación del tipo de <i>layout</i> .
<code>xmlns:app="http://schemas.android.com/apk/res-auto"</code>	
<code>xmlns:tools="http://schemas.android.com/tools"</code>	
<code>android:layout_width="match_parent"</code>	Ancho con base en la vista padre.

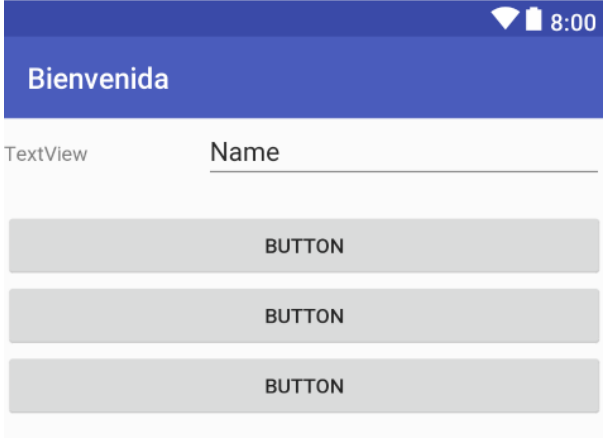
<code>android:layout_height="match_parent"</code>	Alto con base en la vista padre.
<code>android:orientation="vertical"</code>	Orientación vertical.
<code>tools:context=".MainActivity"></code>	
<code><TextView</code>	Tipo de vista.
<code>android:id="@+id/textView"</code>	Identificador de la vista.
<code>android:layout_width="wrap_content"</code>	Ancho con base en su contenido.
<code>android:layout_height="29dp"</code>	Ancho de 29 pixeles.
<code>android:layout_marginBottom="8dp"</code>	Margen inferior de 8 pixeles.
<code>android:text="Nombre"</code>	Texto que se muestra en la interfaz.
<code>app:layout_constraintBottom_toBottomOf="@+id/editText"</code>	Alinear la parte inferior de la vista actual (<i>textView</i>) con la parte inferior de la vista con identificador <i>editText</i> .
<code>tools:layout_editor_absoluteX="16dp" /></code>	Posición con respecto al eje x de 16 pixeles.
<code><EditText</code>	Tipo de la vista.
<code>android:id="@+id/editText"</code>	Identificador de la vista.
<code>android:layout_width="wrap_content"</code>	Ancho con base en su contenido.
<code>android:layout_height="wrap_content"</code>	Alto con base en su contenido.
<code>android:ems="10"</code>	Tamaño de la fuente 10 emes.
<code>android:inputType="textPersonName"</code>	Tipo de entrada.
<code>android:text="Name"</code>	Texto que se muestra en el diseño.
<code>tools:layout_editor_absoluteX="124dp"</code>	Posición en eje x 124 pixeles.
<code>tools:layout_editor_absoluteY="0dp" /></code>	Posición respecto al eje y de 0 pixeles.
<code></></code>	

Fuente: elaboración propia

2.2. *LinearLayout* (horizontal/vertical)

En la Tabla 3 se muestra un ejemplo de *LinearLayout* de forma horizontal, el cual contiene a otro *LinearLayout* de forma vertical.

Tabla 3. Atributos de las vistas para el botón de comandos

	
Codigo Xml	Comentario
<code><LinearLayout</code>	
<code> android:layout_width="match_</code> <code>parent"</code>	Ancho tan grande como la vista padre.
<code> android:layout_height="match_</code> <code>parent"</code>	Alto tan grande como la vista padre.
<code> android:orientation="horizontal"</code>	Orientación de las vistas horizontalmente.
<code> tools:layout_editor_</code> <code>absoluteX="0dp"</code>	Posición en eje x es 0 pixeles.
<code> tools:layout_editor_</code> <code>absoluteY="85dp"></code>	Posición en eje y es 85 pixeles.
<code> <TextView</code>	Tipo de la vista.
<code> android:id="@+id/</code> <code>textView"</code>	Nombre de la variable.
<code> android:layout_</code> <code>height="wrap_content"</code>	Alto con base en su contenido.
<code> android:layout_weight="1"</code>	El peso con respecto a las otras vistas dentro de su <i>layout</i> .
<code> android:text="TextView"</code> <code>/></code>	Texto que se muestra.
<code> <EditText</code>	Tipo de control.
<code> android:id="@+id/</code> <code>editText"</code>	Nombre de la variable.
<code> android:layout_</code> <code>height="wrap_content"</code>	Alto con base en su contenido.
<code> android:layout_weight="2"</code>	El peso con respecto a las otras vistas dentro de su <i>layout</i> .
<code> android:ems="10"</code>	Tamaño de la fuente (10 emes).
<code> android:inputType="textPersonName"</code>	Tipo de datos de entrada.
<code> android:text="Name" /></code>	Texto que se muestra en el diseño.
<code></LinearLayout></code>	


<code><LinearLayout</code>	
<code>android:layout_width="match_parent"</code>	Ancho con base en la vista padre.
<code>android:layout_height="wrap_content"</code>	Alto con base en su contenido.
<code>android:layout_weight="1"</code>	Peso con respecto a las demás vistas dentro del <i>layout</i> .
<code>android:orientation="vertical"</code>	Orientación vertical
<code>tools:layout_editor_absoluteX="120dp"</code>	Posición con respecto al eje x es 120 pixeles.
<code>tools:layout_editor_absoluteY="63dp"></code>	Posición con respecto al eje y es 63 pixeles.
<code><Button</code>	Tipo de vista.
<code>android:id="@+id/button1"</code>	Identificador de la vista.
<code>android:layout_width="match_parent"</code>	Ancho con base en su vista padre.
<code>android:layout_height="wrap_content"</code>	Alto con base en su contenido.
<code>android:text="Button" /></code>	Texto que se muestra en el diseño.
<code><Button</code>	
<code>android:id="@+id/button2"</code>	
<code>android:layout_width="match_parent"</code>	
<code>android:layout_height="wrap_content"</code>	
<code>android:text="Button" /></code>	
<code><Button</code>	
<code>android:id="@+id/button3"</code>	
<code>android:layout_width="match_parent"</code>	
<code>android:layout_height="wrap_content"</code>	
<code>android:text="Button" /></code>	
<code></LinearLayout></code>	
<code></></code>	

Fuente: elaboración propia

2.3. *FrameLayout*

La Tabla 4 muestra en ejemplo de *FrameLayout* utilizando diferentes tipos de gravedad.

Tabla 4. Atributos de las vistas *textView* dentro de un *FrameLayout*

	
Codigo Xml	Comentario
<code><FrameLayout</code>	
<code> android:layout_width="409dp"</code>	
<code> android:layout_height="190dp"></code>	
<code> <TextView</code>	
<code> android:layout_width="wrap_content"</code>	
<code> android:layout_height="wrap_content"</code>	
<code> android:background="#0000ff"</code>	Color de fondo de la vista.
<code> android:text="Sin gravedad"</code>	Texto que muestra.
<code> android:textColor="#ffffff"</code>	Color de la fuente.
<code> android:textSize="15dp"</code>	Tamaño de la fuente.
<code>></code>	
<code> <TextView</code>	
<code> android:layout_width="wrap_content"</code>	
<code> android:layout_height="wrap_content"</code>	
<code> android:layout_gravity="bottom right"</code>	Ubicada en la parte inferior derecha del <i>layout</i> .
<code> android:background="#0000ff"</code>	
<code> android:text="Inferior derecha"</code>	
<code> android:textColor="#ffffff"</code>	
<code> android:textSize="15dp"</code>	
<code>></code>	
<code><TextView</code>	


<code>android:layout_width="wrap_content"</code>	
<code>android:layout_height="wrap_content"</code>	
<code>android:layout_gravity="center"</code>	Ubicada en el centro del <i>layout</i> .
<code>android:background="#0000ff"</code>	
<code>android:text="centro"</code>	
<code>android:textColor="#ffffff"</code>	
<code>android:textSize="15dp"</code>	
<code>/></code>	
<code></FrameLayout></code>	

Fuente: elaboración propia

2.4. *TableLayout*

En la Tabla 5 se presenta una serie de vistas organizadas en forma de matriz (filas y columnas), cada una con unas propiedades que se describen al frente de las líneas XML.

Tabla 5. Atributos de botones de comandos dentro para un *TableLayout*

	
Codigo Xml	Comentario
<code><TableLayout</code>	
<code>android:layout_width="264dp"</code>	Ancho de <i>TableLayout</i> .
<code>android:layout_height="254dp"</code>	Alto de <i>TableLayout</i> .
<code>tools:layout_editor_absoluteX="-72dp"</code>	Posición en el eje x.
<code>tools:layout_editor_absoluteY="-16dp"></code>	Posición en el eje y.
<code><TableRow</code>	Definición de una fila.

<code>android:layout_width="311dp"</code>	Ancho de la fila.
<code>android:layout_height="match_parent"></code>	Alto de la fila.
<code><Button</code>	Definición de la vista botón.
<code>android:id="@+id/button1"</code>	Identificador de la vista.
<code>android:layout_width="wrap_content"</code>	Vista ajustada al ancho del contenido.
<code>android:layout_height="wrap_content"</code>	Vista ajustada al alto del contenido.
<code>android:text="Button" /></code>	Fin de la definición de la vista.
<code><Button</code>	
<code>android:id="@+id/button2"</code>	
<code>android:layout_width="wrap_content"</code>	
<code>android:layout_height="wrap_content"</code>	
<code>android:text="Button" /></code>	
<code><Button</code>	
<code>android:id="@+id/button3"</code>	
<code>android:layout_width="wrap_content"</code>	
<code>android:layout_height="wrap_content"</code>	
<code>android:text="Button" /></code>	
<code></TableRow></code>	
<code><TableRow</code>	
<code>android:layout_width="match_parent"</code>	
<code>android:layout_height="match_parent"></code>	
<code><Button</code>	
<code>android:id="@+id/button4"</code>	
<code>android:layout_width="wrap_content"</code>	
<code>android:layout_height="wrap_content"</code>	
<code>android:layout_span="2"</code>	La vista actual se extiende al tamaño de 2 vistas.
<code>android:text="Button" /></code>	
<code><Button</code>	
<code>android:id="@+id/button5"</code>	
<code>android:layout_width="wrap_content"</code>	
<code>android:layout_height="wrap_content"</code>	

<code>android:text="Button" /></code>	
<code></TableRow></code>	
<code><TableRow</code>	
<code>android:layout_width="-</code> <code>match_parent"</code>	
<code>android:layout_height="-</code> <code>match_parent"></code>	
<code><Button</code>	
<code>android:id="@+id/but-</code> <code>ton6"</code>	
<code>android:layout_wid-</code> <code>th="wrap_content"</code>	
<code>android:layout_heigh-</code> <code>t="wrap_content"</code>	
<code>android:layout_weir-</code> <code>ght="1.5"</code>	La vista actual se extiende al tamaño de una y media.
<code>android:text="Button"</code> <code>/></code>	
<code><Button</code>	
<code>android:id="@+id/but-</code> <code>ton7"</code>	
<code>android:layout_wid-</code> <code>th="wrap_content"</code>	
<code>android:layout_heigh-</code> <code>t="wrap_content"</code>	
<code>android:layout_weir-</code> <code>ght="1.5"</code>	La vista actual se extiende al tamaño de una y media.
<code>android:text="Button"</code> <code>/></code>	
<code></TableRow></code>	
<code></TableLayout></code>	

Fuente: elaboración propia

Referencias

Gironés, J. (2013). *El gran libro de Android*. Barcelona, España. Editorial: Marcombo.

Benbourahla, N. (2015). *Android 5. Principios del desarrollo de aplicaciones Java*. Barcelona, España: Eni.

Hebuterne, S. (2016). *Android. Guía de desarrollo de aplicaciones Java para Smartphones y Tabletas*. Barcelona, España: ENI.

Smyth N. (2017). *Android Studio 3.0 Development Essentials. Android 8 Edition*. Payload Media, Inc.

*Texto aclaratorio

En este material se han tomado capturas de pantalla de la página de Android Studio con fines netamente educativos.

Android Studio. (2018). Capturas de pantalla desde *Android Studio* [Capturas de pantalla]. Recuperado de <https://developer.android.com/studio/?hl=es-419#downloads>

INFORMACIÓN TÉCNICA

FACULTAD DE
INGENIERÍA, DISEÑO
E INNOVACIÓN



Módulo: Herramientas de Programación Móvil I

Unidad 2: Interfaces-Layouts

Escenario 3: Interfaz de usuario

Autor: Manuel Báez

Asesor Pedagógico: Angie Laitón

Diseñador Gráfico: Andrés Felipe Figueroa

Asistente: Alejandra Morales

Este material pertenece al Politécnico Gran Colombiano.

Prohibida su reproducción total o parcial.