

INTERFACE DE USUARIO: LAYOUTS



Programación multimedia
y dispositivos móviles
2º DAM

ÍNDICE DE CONTENIDOS

1. Interface de usuario. Clases asociadas
2. Componentes y layouts
3. Linear Layout
4. Relative Layout
5. Table Layout
6. Grid Layout
7. Absolute Layout
8. Frame Layout
9. Constraint Layout
10. Scroll View
11. Horizontal Scroll View

1. INTERFACE DE USUARIO. CLASES ASOCIADAS

El interfaz de una aplicación Android es aquello que aparece en pantalla, que por tanto el usuario puede ver, y por tanto, interactuar con él. Todos los componentes que nos ofrece Android derivan de la clase View, y podremos definirlo en el archivo XML del directorio res/layout.

1. INTERFACE DE USUARIO. CLASES ASOCIADAS

La clase View tiene una serie de atributos que serán heredados por todos los elementos que de ella provienen.

<https://developer.android.com/reference/android/view/View>

Hay dos atributos obligatorios para cualquier componente/layout:

- layout_width
- layout_height

1. INTERFACE DE USUARIO. CLASES ASOCIADAS

La clase View tiene una serie de atributos que serán heredados por todos los elementos que de ella provienen.

<https://developer.android.com/reference/android/view/View>

Hay dos atributos obligatorios para cualquier componente/layout:

- layout_width
- layout_height

Estos atributos establecen el tamaño que ocupan en la pantalla.

1. INTERFACE DE USUARIO. CLASES ASOCIADAS

Formas de establecer el tamaño:

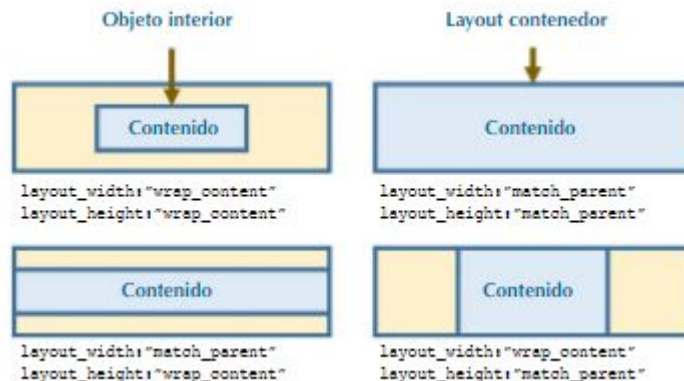
- En pulgadas (in)
- En milímetros (mm)
- En puntos (pt)
- En píxeles (px)
- **En píxeles independientes de la densidad (dp)**
- En píxeles independientes de la escala (sp)

<https://developer.android.com/guide/topics/resources/more-resources#Dimension>

1. INTERFACE DE USUARIO. CLASES ASOCIADAS

Otra forma de indicarlo es con:

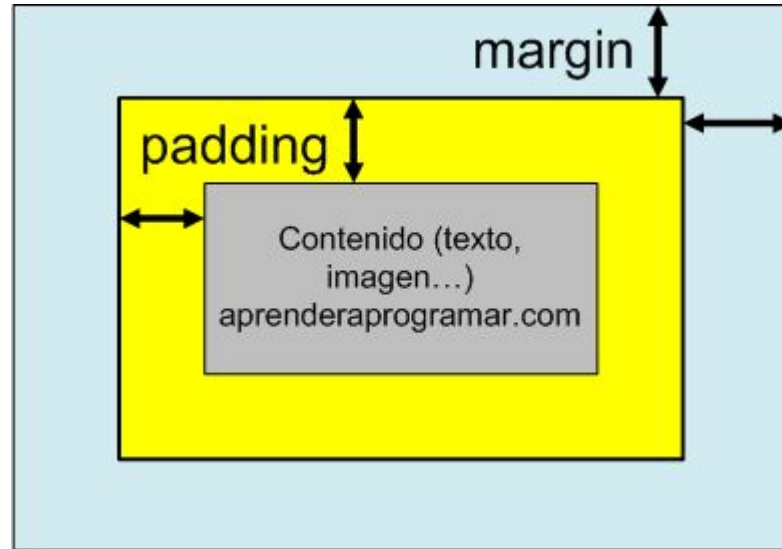
- match_parent
- wrap_content



1. INTERFACE DE USUARIO. CLASES ASOCIADAS

<i>Atributos de posicionamiento</i>	layout_width	ancho
	layout_height	alto
<i>Atributos para los márgenes</i>	layout_margin	cuatro márgenes
	layout_marginBottom	margen inferior
	layout_marginLeft	margen izquierdo
	layout_marginRight	margen derecho
	layout_marginTop	margen superior
<i>Atributos para el espaciado</i>	android:padding	espaciado a los cuatro lados
	android:paddingTop	espaciado superior
	android:paddingBottom	espaciado inferior
	android:paddingLeft	espaciado izquierdo
	android:paddingRight	espaciado derecho

1. INTERFACE DE USUARIO. CLASES ASOCIADAS



1. INTERFACE DE USUARIO. CLASES ASOCIADAS

Otro atributo muy importante (no obligatorio) es el identificador.

```
android:id="@+id/texto"
```

De esta forma, establecemos un nombre para cada uno de los elementos gráficos, para después poder utilizarlos.

- A nivel de diseño. Ejemplo: indicar que un elemento va a la derecha de otro.
- A nivel de código. Ejemplo: usar el elemento seleccionado de un spinner.

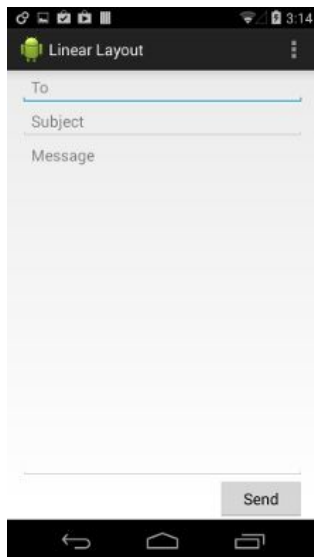
2. CONTROLES Y LAYOUT.

- Los controles son los elementos que componen la interfaz de usuario de una aplicación . Entre ellos, se encuentran los botones.
- Un layout es un conjunto de vistas agrupadas de una determinada forma. Los layouts solo definen la disposición de los elementos, no son visibles en la actividad.

Tanto los controles como los layout heredan de la clase View, por lo que se pueden definir tanto desde código Java, como desde XML.

3. LINEAR LAYOUT

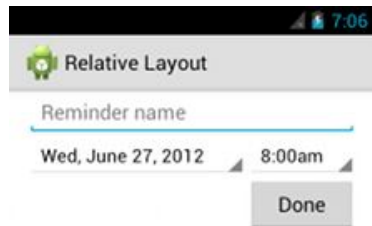
LinearLayout es un grupo de vista que alinea todos los campos secundarios en una única dirección, de manera vertical u horizontal. Puedes especificar la dirección del diseño con el atributo `android:orientation`.



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:orientation="vertical" >
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/to" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/subject" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="top"
        android:hint="@string/message" />
    <Button
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="@string/send" />
</LinearLayout>
```

4. RELATIVE LAYOUT

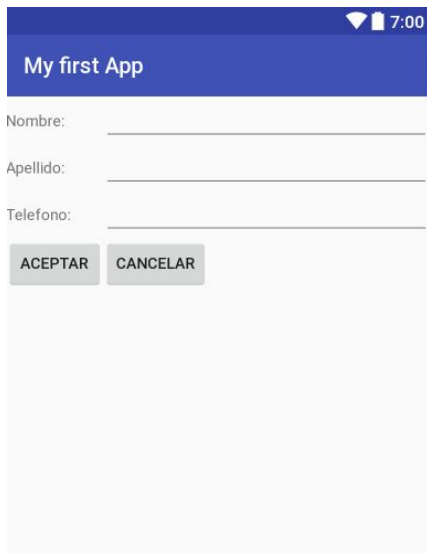
Permite colocar elementos en cualquiera de las cuatro esquinas del contenedor e ir añadiendo otros elementos pegados a estos.



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp" >
    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/reminder" />
    <Spinner
        android:id="@+id/dates"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/name"
        android:layout_alignParentLeft="true"
        android:layout_toLeftOf="@+id/times" />
    <Spinner
        android:id="@+id/times"
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/name"
        android:layout_alignParentRight="true" />
    <Button
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/times"
        android:layout_alignParentRight="true"
        android:text="@string/done" />
</RelativeLayout>
```

5. TABLE LAYOUT

Distribuye los elementos de forma tabular, similar a las tablas HTML. Se utiliza la etiqueta `<TableRow>` para crear una nueva fila.



```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TableRow>
        <TextView
            android:text="Nombre: " />
        <EditText android:id="@+id/edit_message"
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:hint="" />
    </TableRow>

    <TableRow>
        <TextView
            android:text="Apellido: " />
        <EditText android:id="@+id/edit_message2"
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:hint="" />
    </TableRow>

    <TableRow>
        <TextView
            android:text="Telefono: " />
        <EditText android:id="@+id/edit_message3"
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:hint="" />
    </TableRow>

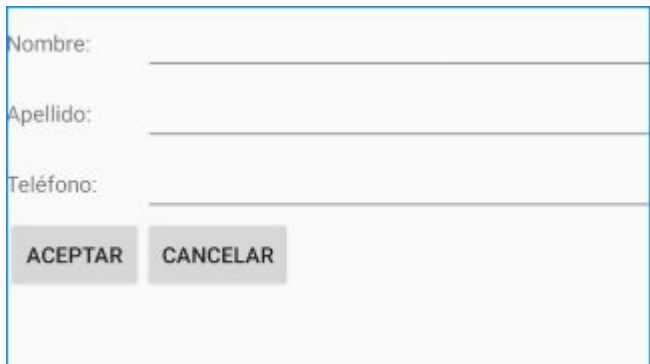
    <TableRow>
        <Button
            android:text="Aceptar" />

        <Button
            android:text="Cancelar" />
    </TableRow>
</TableLayout>
```

6. GRID LAYOUT

Similar a Table Layout, utiliza un interfaz de forma tabular, distribuido en filas y columnas mediante:

- rowCount
- columnCount

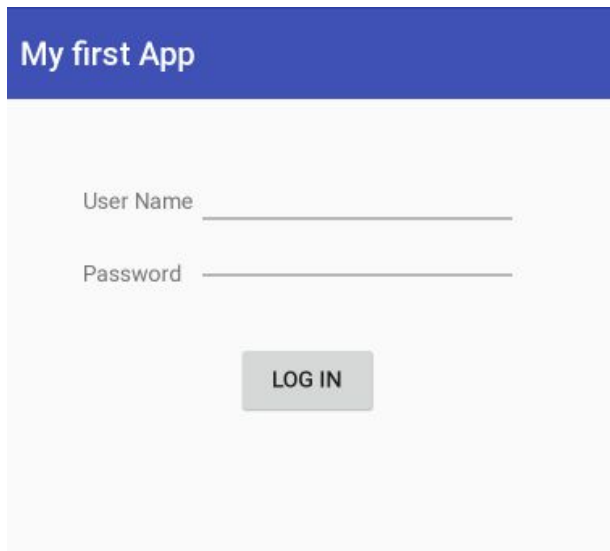


```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity"
    android:rowCount="4"
    android:columnCount="2">

    <TextView
        android:text="Nombre:"/>
    <TextView
        android:text="Apellido:"/>
    <TextView
        android:text="Teléfono:"/>
    <Button
        android:text="Aceptar"/>
    <EditText
        android:layout_height="wrap_content"
        android:layout_width="match_parent"
        android:id="@+id/edit_message"
        android:hint=""/>
    <EditText
        android:layout_height="wrap_content"
        android:layout_width="match_parent"
        android:id="@+id/edit_message2"
        android:hint=""/>
    <EditText
        android:layout_height="wrap_content"
        android:layout_width="match_parent"
        android:id="@+id/edit_message3"
        android:hint=""/>
    <Button
        android:text="Cancelar"/>
</GridLayout>
```

7. ABSOLUTE LAYOUT

Permite fijar las coordenadas X,Y en las que se visualizará cada elemento de forma arbitraria. Su uso está totalmente no recomendado. (DEPRECATED!!)



```
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <TextView
        android:layout_x="110px"
        android:layout_y="110px"
        android:text="User Name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <EditText
        android:layout_x="250px"
        android:layout_y="80px"
        android:width="100px"
        android:layout_width="200dp"
        android:layout_height="wrap_content" />

    <TextView
        android:layout_x="110px"
        android:layout_y="200px"
        android:text="Password"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <EditText
        android:layout_x="250px"
        android:layout_y="150px"
        android:width="100px"
        android:layout_width="200dp"
        android:layout_height="wrap_content" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Log In"
        android:layout_x="300px"
        android:layout_y="300px"/>
</AbsoluteLayout>
```


8. FRAME LAYOUT

Posiciona las vistas usando todo el contenedor sin distribuirlas especialmente. Se utiliza cuando queremos que dos elementos ocupen un mismo lugar. Mostrando uno y ocultando otro.



```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:id="@+id/frameLayout" >

    <TextView
        android:id="@+id/frameText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Imagen de muestra"
        android:textSize="15sp"
        android:textStyle="bold"
        android:visibility="gone"
        android:layout_gravity="center" />

    <ImageView
        android:id="@+id/frameImage"
        android:layout_width="200dp"
        android:layout_height="300dp"
        android:layout_gravity="center"
        android:clickable="true"
        android:src="@mipmap/ic_launcher_round" />

</FrameLayout>
```

9. CONSTRAINT LAYOUT

Constraint Layout permite colocar los widgets de una forma más flexible, y también redimensionarlos.

Cuando en una actividad tenemos un gran anidamiento de Layouts, ralentiza mucho la aplicación, e incluso, puede que algunos dispositivos no consigan mostrar la interfaz.

There are currently various types of constraints that you can use:

- Relative positioning
- Margins
- Centering positioning
- Circular positioning
- Visibility behavior
- Dimension constraints
- Chains
- Virtual Helpers objects
- Optimizer

https://www.youtube.com/watch?time_continue=6&v=XamMbnzI5vE&feature=emb_logo

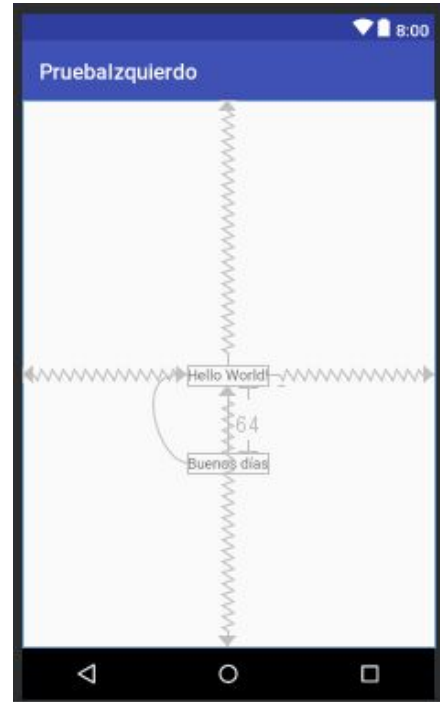
9. CONSTRAINT LAYOUT

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

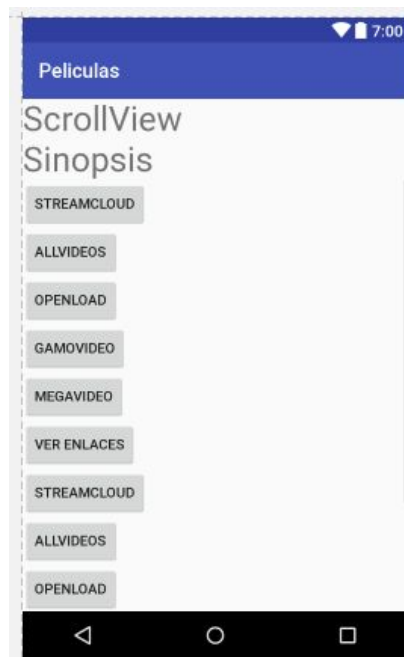
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="64dp"
        android:text="@string/buenos_dias"
        app:layout_constraintStart_toStartOf="@+id/textView"
        app:layout_constraintTop_toBottomOf="@+id/textView"
        app:layout_goneMarginLeft="23dp" />

</android.support.constraint.ConstraintLayout>
```



10. SCROLL VIEW

Visualiza una columna de elementos; cuando estos no caben en pantalla se permite un deslizamiento vertical.



```
<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical" >

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="StreamCloud"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="AllVideos"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="OpenLoad"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="GamoVideo"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="MegaVideo"/>
```

11. HORIZONTAL SCROLL VIEW

Visualiza una fila de elementos; cuando estos no caben en pantalla se permite un deslizamiento horizontal.



```
<HorizontalScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="StreamCloud"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="AllVideos"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="OpenLoad"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="GamoVideo"/>

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="MegaVideo"/>

        <Button
            android:id="@+id/btn_act"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Ver enlaces" >

    </LinearLayout>
</HorizontalScrollView>
```

10. DUDAS Y PREGUNTAS

