

|  |
| --- |
| http://www.becas.sep.gob.mx/images/logo.pngEnsayo de Articulo  TECNOLÓGICO NACIONAL DE MÉXICO  Instituto Tecnológico de Chihuahua II |
|  |
| Layouts |
|  |
|  |
| **26/08/2016**  Unidad 1  Ulises Manuel De la Rosa Contreras  13550362  Sistemas Operativos Para Dispositivos Móviles |

DESCRIPION DE LAYOUTS

### LinearLayout

Posiciona las vistas en filas o columnas dependiendo de su atributo orientation. Por defecto sitúa un elemento al lado del otro en horizontal.

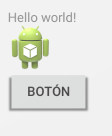
[](http://www.isgarsi.com/wp-content/uploads/2015/01/LinearLayout.png)

Orientación Horizontal

<**LinearLayout**  
xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:tools="http://schemas.android.com/tools"  
android:layout\_width="match\_parent"  
android:layout\_height="match\_parent"  
android:paddingLeft="@dimen/activity\_horizontal\_margin"  
android:paddingRight="@dimen/activity\_horizontal\_margin"  
android:paddingTop="@dimen/activity\_vertical\_margin"  
android:paddingBottom="@dimen/activity\_vertical\_margin"  
tools:context=".MainActivity"  
***android:orientation="horizontal"***>  
<**TextView**  
android:id="@+id/textView"  
android:text="@string/hello\_world"  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content" />  
<**ImageView**  
android:id="@+id/imagen"  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content"  
android:src="@drawable/ic\_launcher"/>  
<**Button**  
android:text="Botón"  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content" />  
<**/LinearLayout**>

Lo mas importante en este layout es indicar la orientación para situar los elementos en horizontal o en vertical. Después claramente podemos añadir margenes, paddings y todo esto a las vistas por separado o en una hoja de estilos como veremos mas adelante.

Tan solo cambiando el horizontal por vertical, este seria su nuevo aspecto:

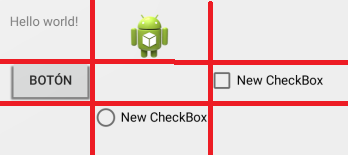
[](http://www.isgarsi.com/wp-content/uploads/2015/01/LinearLayout2.png)

Orientación Vertical

Hecha un vistazo a la [documentación oficial](http://developer.android.com/reference/android/widget/LinearLayout.LayoutParams.html)

### TableLayout

Posiciona las vistas en forma de tabla. Utilizaremos la etiqueta **<TableRow>** cada vez que queramos añadir una nueva fila. Cada vista dentro del tablerow sera una columna

[](http://www.isgarsi.com/wp-content/uploads/2015/01/TableLayout2.png)

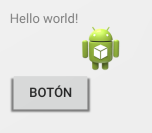
<**TableLayout**  
xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:tools="http://schemas.android.com/tools"  
android:layout\_width="match\_parent"  
android:layout\_height="match\_parent"  
android:paddingLeft="@dimen/activity\_horizontal\_margin"  
android:paddingRight="@dimen/activity\_horizontal\_margin"  
android:paddingTop="@dimen/activity\_vertical\_margin"  
android:paddingBottom="@dimen/activity\_vertical\_margin"  
tools:context=".MainActivity"  
android:orientation="vertical">  
**<TableRow>**  
<**TextView**  
android:id="@+id/textView"  
android:text="@string/hello\_world"  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content" />  
<**ImageView**  
android:id="@+id/imagen"  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content"  
android:src="@drawable/ic\_launcher"/>  
**</TableRow>**  
**<TableRow>**  
<**Button**  
android:text="Botón"  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content" />  
<**CheckBox**  
android:id="@+id/checkBox"  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content"  
android:text="New CheckBox"  
***android:layout\_column="2"*** />  
***</TableRow>***  
***<TableRow>***  
<**RadioButton**  
android:id="@+id/radioButton"  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content"  
android:text="New CheckBox"  
**android:layout\_column="1"** />  
***</TableRow>***  
</TableLayout>

Como podemos ver, tenemos 3 filas y 3 columnas.

* Primera Fila: He añadido 2 vistas(TextView y ImageView), sin ningún modificador, por tanto se rellenan las columnas 0 y 1.
* Segundo Fila: Tambien contiene 2 vistas (Button y CheckBox), pero en este caso si modificamos el CheckBox para que se ponga en la columna 2, por tanto la tercera ya que siempre empezamos a contar por la 0 y dejamos la 1 vacía.
* Tercera Fila: Solo añadimos una vista y la modificamos para que se situe en la columna 1.

[Documentación Oficial TableLayout](http://developer.android.com/reference/android/widget/TableLayout.LayoutParams.html)

### RelativeLayout

Es el que pone Android Studio por defecto al crear un proyecto. Posiciona las vistas en relación a otro o al padre.  
[](http://www.isgarsi.com/wp-content/uploads/2015/01/Captura.png)  
  
<**RelativeLayout**  
xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:tools="http://schemas.android.com/tools"  
android:layout\_width="match\_parent"  
android:layout\_height="match\_parent"  
android:paddingLeft="@dimen/activity\_horizontal\_margin"  
android:paddingRight="@dimen/activity\_horizontal\_margin"  
android:paddingTop="@dimen/activity\_vertical\_margin"  
android:paddingBottom="@dimen/activity\_vertical\_margin"  
tools:context=".MainActivity">  
<**TextView**  
android:id="@+id/textView"  
android:text="@string/hello\_world"  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content" />  
<**ImageView**  
android:id="@+id/imagen"  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content"  
android:src="@drawable/ic\_launcher"  
***android:layout\_below="@id/textView"***  
***android:layout\_toRightOf="@id/textView"***/>  
<**Button**  
android:text="Botón"  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content"  
***android:layout\_below="@+id/imagen"***  
***android:layout\_alignLeft="@id/textView"*** />  
<**/RelativeLayout**>

Si nos fijamos en la imagen, tenemos un texto, una imagen y un botón, uno debajo de otro y alineados de forma distinta.

Ahora si nos fijamos en el código:

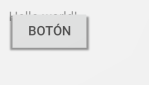
* TextView: Al ser el primero, no tiene ningún modificador de posición pero también podríamos alinearlo a la derecha o ponerle un margen superior para que baje…
* ImageView: Al ser un layout relativo, le decimos que se ponga debajo del texto, indicando su id. Además le decimos que se ponga a la derecha del texto. Vamos que estamos indicando la posición relativa al texto que hay arriba.
* Button: Indicamos tambien que este debajo, pero  en este caso de la imagen y que se alinea por la izquierda con el texto de arriba.

Claramente este tipo de layout es muy potente y tenemos muchos mas atributos disponibles que podemos consultar en la [documentación oficial](http://developer.android.com/reference/android/widget/RelativeLayout.LayoutParams.html)

### FrameLayout

Posiciona las vistas usando todo el contenedor. Suele usarse cuando queremos que varias vistas ocupen un mismo lugar. Después podemos jugar con la visibilidad para mostrar una o otra con la propiedad *visibility*.

Si no usáramos la propiedad visibility en las vistas, veríamos como el botón se solapa encima del texto.

[](http://www.isgarsi.com/wp-content/uploads/2015/01/FrameLayout1.png)

Ahora bíen si añadimos la propiedad visibility a invisible quedaría así:

[FrameLayout](http://www.isgarsi.com/wp-content/uploads/2015/01/FrameLayout3.png)

propiedad visibility en botón

[](http://www.isgarsi.com/wp-content/uploads/2015/01/FrameLayout2.png)

propiedad visibility en texto

<**FrameLayout**  
xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:tools="http://schemas.android.com/tools"  
android:layout\_width="match\_parent"  
android:layout\_height="match\_parent"  
android:paddingLeft="@dimen/activity\_horizontal\_margin"  
android:paddingRight="@dimen/activity\_horizontal\_margin"  
android:paddingTop="@dimen/activity\_vertical\_margin"  
android:paddingBottom="@dimen/activity\_vertical\_margin"  
tools:context=".MainActivity">  
<**TextView**  
android:id="@+id/textView"  
android:text="@string/hello\_world"  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content"  
/>  
<**Button**  
android:text="Botón"  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content"  
***android:visibility="invisible"***/>  
<**/FrameLayout**>

**NOTA**: Visibility puede tener 3 valores:

* **visible**: Valor por defecto, no hace falta ponerlo.
* **invisible**: No se ve la vista pero si ocupa el espacio.
* **gone**: No se ve la vista y además no ocupa espacio en el layout.