

Curso de Desarrollo en Android

Layouts

Roberto Calvo Palomino

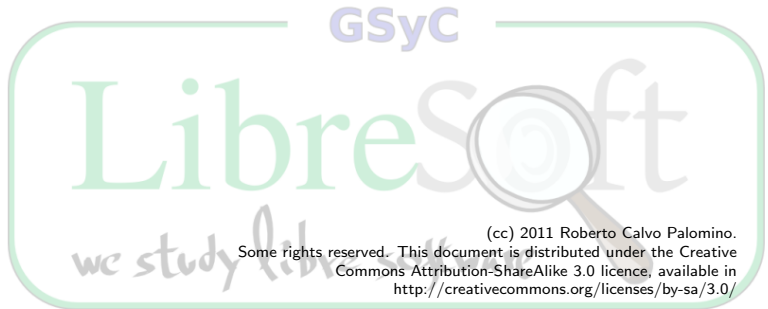
rocapal@libresoft.es
GSyC/Libresoft

March 11, 2011

we study libre software



Universidad
Rey Juan Carlos



(cc) 2011 Roberto Calvo Palomino.
Some rights reserved. This document is distributed under the Creative
Commons Attribution-ShareAlike 3.0 licence, available in
<http://creativecommons.org/licenses/by-sa/3.0/>

Index

1 Nomenclatura

2 Tipos de Layouts

3 Depuración de Vistas



Medición en Pantalla

- px (píxeles)
- dip (píxeles independientes del dispositivo)
- sp (píxeles escalados, mejor para tamaños de texto)
- pt (puntos)
- in (inches)
- mm (milímetros)

Diferentes Pantalla

- Tamaño
- Densidad (dependiente o no del pixel)
- Aspecto de Ratio
- Versión de plataforma

Screens Support (Link)

Index

- 1 Nomenclatura
- 2 Tipos de Layouts
- 3 Depuración de Vistas

GSyC

LibreSoft
we study libre software

Introduccion

- Unidad elemental de la interfaz gráfica.
- Contenedores de widgets que ocupan la pantalla
- Relación fuerte entre layouts para conseguir la interfaz deseada
- `fill_parent`: Ocupa la anchura o altura máximo que le deje su padre
- `wrap_content`: Ocupa la anchura o altura mínima necesaria

FrameLayout

- Está diseñado para ocupar un area de la pantalla para mostrar un solo elemento
- Puedes añadir muchos hijos a este FrameLayout, pero todos los hijos aparecerán alineados arriba a la izquierda de este contenedor.
- En este grupo de vistas no se puede especificar la posición de los hijos
- No son los layouts más utilizados


```
1 <?xml version='1.0' encoding='utf-8'?>
2 <FrameLayout
3   xmlns:android='http://schemas.android.com/apk/res/android'
4     android:layout_width='150dip'
5     android:layout_height='150dip'
6     android:background='#FF00FF'
7   >
8     <Button
9       android:layout_width='wrap_content'
10      android:layout_height='wrap_content'
11      android:text='Botón'
12    />
13    <TextView
14      android:layout_width='wrap_content'
15      android:layout_height='wrap_content'
16      android:text='Hola que tal como estás. Este texto aparece encima
17      del botón'
18    />
19  </FrameLayout>
```

we study libre software

LinearLayout

- Dispone sus hijos en una sola columna o en una sola fila (configurable)
- La dirección de cada fila puede ser especificada mediante *setOrientation()* en código o mediante *android:orientation* en el XML. Por defecto, es horizontal
- También puedes especificar gravedad, la cual indica la alineación de todos los elementos hijos
- Soporta asignar peso a los hijos de forma individual. Este valor permite a los hijos que se expandan hasta llenar cualquier espacio libre de la pantalla.

```
1 <?xml version='1.0' encoding='utf-8'?>
  <LinearLayout
3     xmlns:android='http://schemas.android.com/apk/res/android'
        android:orientation='horizontal'
5         android:layout_width='fill_parent'
        android:layout_height='wrap_content'
7         android:background='#FF00FF'
        >
9     <TextView
        android:layout_width='wrap_content'
11        android:layout_height='wrap_content'
        android:text='Texto prueba'
13        android:background='#FFFFFF'
        android:layout_weight='5'
15        />
        <Button
17        android:layout_width='wrap_content'
        android:layout_height='wrap_content'
19        android:text='Botón'
        android:layout_weight='1'
21        />
    </LinearLayout>
```

RelativeLayout

- El más flexible de todos los tipos de layouts. Puedes construir casi cualquier cosa con ellos.
- La posición de los hijos son relativas al padre e incluso a otros hijos.
- Nos permite realizar interfaces complejas y portables a diferentes tamaños de pantalla.

```
1 <?xml version='1.0' encoding='utf-8'?>
2 <RelativeLayout
3     xmlns:android='http://schemas.android.com/apk/res/android'
4         android:layout_width='fill_parent'
5         android:layout_height='fill_parent'
6     >
7         <TextView
8             android:id='@+id/hello'
9             android:layout_width='fill_parent'
10            android:layout_height='wrap_content'
11            android:text='@string/hello'
12        />
13        <TextView
14            android:id='@+id/black_on_red'
15            android:layout_width='fill_parent'
16            android:layout_height='wrap_content'
17            android:layout_below='@id/hello'
18            android:background='#ff0000'
19            android:textColor='#000000'
20            android:text='black on red'
21        />
22        <TextView
23            android:id='@+id/update_me'
24            android:layout_width='fill_parent'
25            android:layout_height='wrap_content'
26            android:layout_below='@id/black_on_red'
27            android:background='#00ff00'
28            android:textColor='#000000'
29            android:text='update Text'
30        />
31        <TextView
32            android:id='@+id/left_wrap'
33            android:layout_width='wrap_content'
```

```
34     android:layout_height='''wrap_content'''  
35     android:layout_below='''@id/update_me'''  
36     android:background='''#0000ff'''  
37     android:textColor='''#000000'''  
38     android:text='''wrap'''  
39     />  
40 <TextView  
41     android:id='''@+id/right_fill'''  
42     android:layout_width='''fill_parent'''  
43     android:layout_height='''wrap_content'''  
44     android:layout_below='''@id/update_me'''  
45     android:layout_toRightOf='''@id/left_wrap'''  
46     android:background='''#ff0000'''  
47     android:textColor='''#000000'''  
48     android:text='''fill'''  
49     />  
50 </RelativeLayout>
```



TableLayout

- Layout específico para mostrar tablas
- Se puede realizar también con RelativeLayouts
- Columnas y Filas son sus principales propiedades

we study libre software

```
1 <?xml version='1.0' encoding='utf-8'?>
2 <TableLayout xmlns:android='http://schemas.android.com/apk/res/android'
3     android:layout_width='wrap_content'
4     android:layout_height='fill_parent'
5     android:stretchColumns='1'>
6     <TableRow>
7         <TextView
8             android:text='prueba'
9             android:padding='3dip' />
10        <TextView
11            android:text='prueba2'
12            android:padding='3dip' />
13    </TableRow>
14    <TableRow>
15        <TextView
16            android:text='prueba3'
17            android:padding='3dip' />
18        <TextView
19            android:text='prueba4'
20            android:padding='3dip' />
21    </TableRow>
22 </TableLayout>
```

GSyc

libreSoft
we study libre software

Index

- 1 Nomenclatura
- 2 Tipos de Layouts
- 3 Depuración de Vistas



HierarchyViewer y LayoutOpt

- Herramientas para depurar nuestras interfaces
- Útil para poder optimizar recursos en nuestras aplicaciones
- Se encuentra en **\$DIR_SDK_ANDROID/tools/**

Depurar Interfaces (Link)

Optimización

- Nunca usar `AbsoluteLayout`
- Usar el mínimo número de vistas posibles
- Nunca más de 80 layouts en una misma vista, y nunca más de 10 layouts anidados
- Pantallas y vistas sencillas
- Utilizar recursos apropiados para cada tipo de pantalla