

Seminario de Lenguajes Android

Práctica 2

1. Crear un nuevo proyecto en Android Studio con una Actividad vacía y modificar el contenido del archivo xml de la actividad creada (en res/layouts), colocar el siguiente código en él y probar en el dispositivo o emulador:

<RelativeLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout_width="match_parent"

android:layout_height="match_parent">

<TextView

android:layout_width="200dp"

android:layout_height="200dp"

android:text="Hola Mundo" />

</RelativeLayout>

2. Investigue la utilidad de los atributos "android:layout_width" y "android:layout_height".
 - a. ¿Qué sucede si se omite alguno de los dos en el elemento RelativeLayout?
 - b. ¿Qué sucede si en lugar de "match_parent" se establece como valor "wrap_content"?
3. Añadir un color de fondo al TextView y modificar el color de las letras.
4. Modificar el ancho del TextView para que siempre se adapte al ancho del contenedor. Luego centrar el texto horizontalmente.
5. Cambiar el tamaño de la tipografía (utilizar la unidad de medida sp)
6. Crear un TextView debajo del *Hola Mundo* con el texto *Aquí Estoy*. ¿Qué ocurrió? ¿Por qué?

7. Reemplazar en el código creado *RelativeLayout* por un *LinearLayout* de la siguiente manera y ver el resultado. ¿Para qué sirve especificar la orientación del *LinearLayout*?

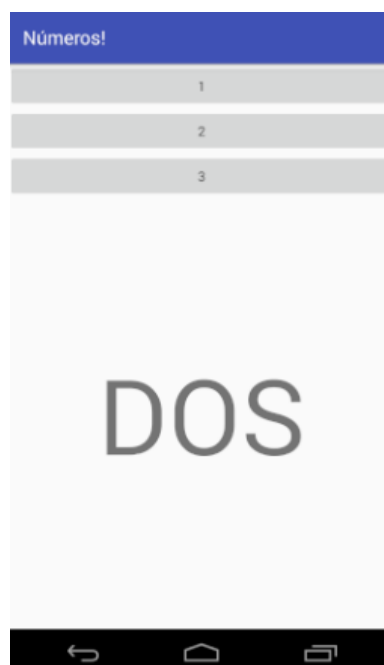
<LinearLayout

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

8. Resuelva el problema visual que surgió en el punto 6 sin utilizar *LinearLayout*.
(Nota: Utilice la propiedad *layout_below*).
9. Genere una disposición de componentes visuales como la siguiente:



10. A partir de la experiencia obtenida con el uso de Layouts.
- ¿Por qué cree que son importantes los Layouts?
 - ¿Cuál hubiera sido la problemática de definir la posición/tamaño de las componentes visuales a través de coordenadas absolutas?
 - ¿Esta problemática solo existe en dispositivos móviles?
11. Implemente una aplicación con un layout como el siguiente:



Cuando se haga click sobre uno de los botones, la aplicación deberá mostrar en pantalla de forma centrada y con una tipografía más grande la representación en letras mayúsculas del número sobre el cual se hizo click.

El ancho de los botones deberá adaptarse al ancho de la pantalla.

El espacio en donde se visualiza el texto deberá adaptarse al espacio disponible y el texto se mostrará centrado horizontal y verticalmente.

- a. Implemente utilizando un manejador de click para cada botón.
- b. Implemente utilizando un único manejador de click para todos los botones.

12. Implemente una aplicación con un layout como el siguiente:



Cada vez que se hace click en el botón "Tirar Dado", la aplicación deberá generar aleatoriamente un valor entre 1 y 6 mostrándolo en pantalla.

Para la generación de números aleatorios investigue el uso de la clase `java.util.Random`

13. Usando un layout similar al desarrollado en el ejercicio 12, agregar una imagen usando el componente **<ImageView>**.

La imagen deberá incluirse en el directorio `/proyecto/app/src/main/res/drawable/`

Nota: puede usar el siguiente código:

```
<ImageView
    android:src="@drawable/imagen"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:scaleType="center" />
```

Práctica 2

Facultad de Informática



OCULTAR

MOSTRAR

- ¿Qué sucede al cambiar el valor del atributo `scaleType` por `"centerCrop"`? ¿Y `"fitXY"`?
- Al hacer click sobre los botones , deberá ocultarse o mostrarse la imagen usando el método `setVisibility(View.GONE)` y `setVisibility(View.VISIBLE)` sobre la imagen.



¿Qué cambia si en vez de ocultar la imagen usando `View.GONE`, usamos `View.INVISIBLE`?

- Agregar el atributo `android:animateLayoutChanges="true"` en el `LinearLayout` que contiene a los elementos. ¿Qué diferencia hay al cambiar la visibilidad de la imagen?