Laboratorio 4.1

Apellidos: Moreno Vera
Nombres: Felipe Adrian
Código: 20120354I
Asignatura: Programación en Dispositivos Móviles (CC481)

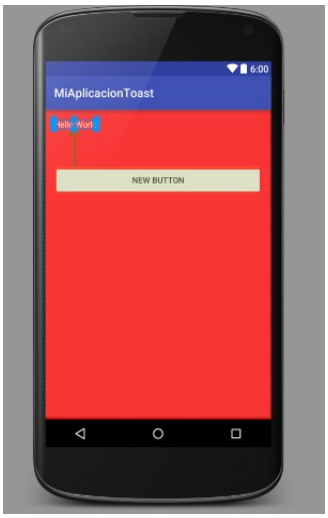
2016 - I

Indice

Actividad 1

1. Nos centramos en laActivity 1, Activity a la que vamos a insertar un Button y abrimos el fichero xml en vista diseño.
2. Modificamos algunas de sus propiedades:

1.Hacemos que el Button se adapte al tamaño de la pantalla con la propiedad "layout:width" en el valor "match_parent". Vemos que se ajusta al ancho.
2. Cambiamos el color de fondo a color rojo.
3. El color de texto lo modificamos a blanco.
4. Cambiamos algunos valores en la spropiedades de layout y describimos lo que muestra el diseño en cada uno de sus valores que se ponen.
Por ejemplo se cambia el layout_margintop, y se desplaza en vertical según el valor que le demos, modificamos el android:layout_alignParentStart="true" a "false"
y ya no esta alineado al textView. Si quitamos la etiqueta layout_below, se podra pegar encima del texto.



3. Abrimos el código Java de nuestra Activity 1. Creamos el método encargado de administrar las opciones de Button, por ejemplo, "mostrarAlerta".
4. En Android las alertas del sistema se llaman Toast. Creamos el método por tanto.

1. El método makeText() genera un texto en pantalla. Explique los parámetros de dicho método.
Toma como referencia el contexto o actividad, el mensaje a mostrar, y un tiempo a mostrar, los sugeridos son Toast.LENGTH_SHORT o LONG.
2. El método getApplicationContext() es el contexto de mi aplicación, Qué significa esto?
Reconoce desde que contexto o actividad se esta llamando al metodo para iniciar el toast.

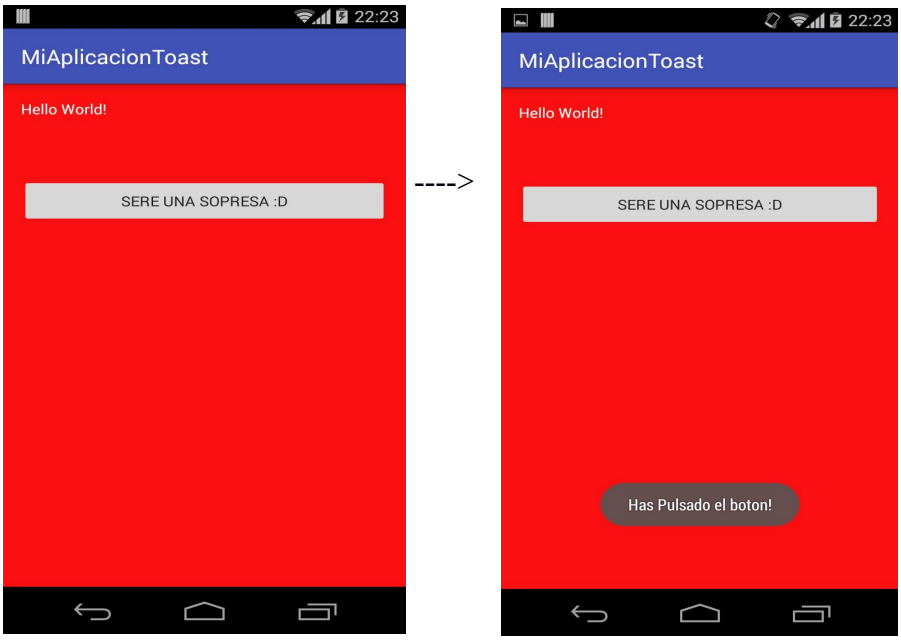
3. El método show() es el encargado de que se muestre por pantalla.
Public void mostrarAlerta(View view){
Toast toast = Toast.makeText(getApplicationContext(),
“Has Pulsado el boton!”, Toast.LENGTH_LONG);
toast.show();
}

```
import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void mostrarAlerta(View view){
        Toast toast = Toast.makeText(getApplicationContext(),
         'Has Pulsado el boton!", Toast.LENGTH_LONG);
        toast.show();
    }
}
```

5. Acuérdese de que Buton hay que relacionarlo con el método correcto.
Terminando el código en el Main Activity.

```
private Button btnAlerta;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    btnAlerta = (Button) findViewById(R.id.btnAlerta);
    btnAlerta.setOnClickListener(new View.OnClickListener(){
        @Override
        public void onClick(View view){
         mostrarAlerta(view);
        }
    });
}
public void mostrarAlerta(View view){
    Toast toast = Toast.makeText(getApplicationContext(),
     "Has Pulsado el boton!", Toast.LENGTH_LONG);
    toast.show();
}
```

Ahora Ejecutando la app en mi moto G.



Actividad 2

1. Creamos un nuevo proyecto llamado y vamos al código Java de la clase principal y hacemos que herede de ActionBarActivity. Explicar esta clase padre.
Como no hay nombre, le puse “MiActionBar”. La clase ActionBarActitivy esta deprecated, y nos dice que usemos AppCompatActivity en su lugar, Es el titulo que aparece actualmente en todas las apps que creamos, antes en el API 11 no había, por eso se tenía una clase dedicada a esa.

2. Con el editor hacemos click derecho y Generate --> Override Methods y seleccionamos los siguientes métodos: onStart(), onRestart(), onPause(), onResume() y onDestroy().Vemos que se han escrito por pantalla.
3. Para cada método escribimos la etiqueta cambiando el string “onRestart” por el método en el que estemos:

```
Log.d(“Hello World”,”onRestart”);
```

4. Compilamos la aplicación y vemos los resultados en el Log. Para que la aplicación finalice pulsamos el botón de “Back” del teléfono y verificamos los métodos en el emulador.

Despues de correr la aplicación y salir de ahi a otra pestaña, aparece:

```
01-17 23:42:06.603 1156-1156/com.example.jbot.miactionbar W/InputEventReceiver: Attempt
01-17 23:42:07.324 1156-1156/com.example.jbot.miactionbar D/Hello World: onPause
01-17 23:42:19.724 1156-1156/com.example.jbot.miactionbar D/Hello World: onRestart
01-17 23:42:19.735 1156-1156/com.example.jbot.miactionbar D/Hello World: onResume
01-17 23:42:19.743 1156-1156/com.example.jbot.miactionbar I/Adreno-EGL: <qeglDrvAPI_egl
```

y finalmente al salir con el botón flecha atrás de la pantalla.

```
01-17 23:42:36.859 1156-1156/com.example.jbot.miactionbar D/Hello World: onPause
01-17 23:42:37.434 1156-1156/com.example.jbot.miactionbar D/Hello World: onDestroy
```

Link del github con los códigos del laboratorio:

https://github.com/Jenazad/PDM/tree/master/Laboratorio_4

Referencias

http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=603