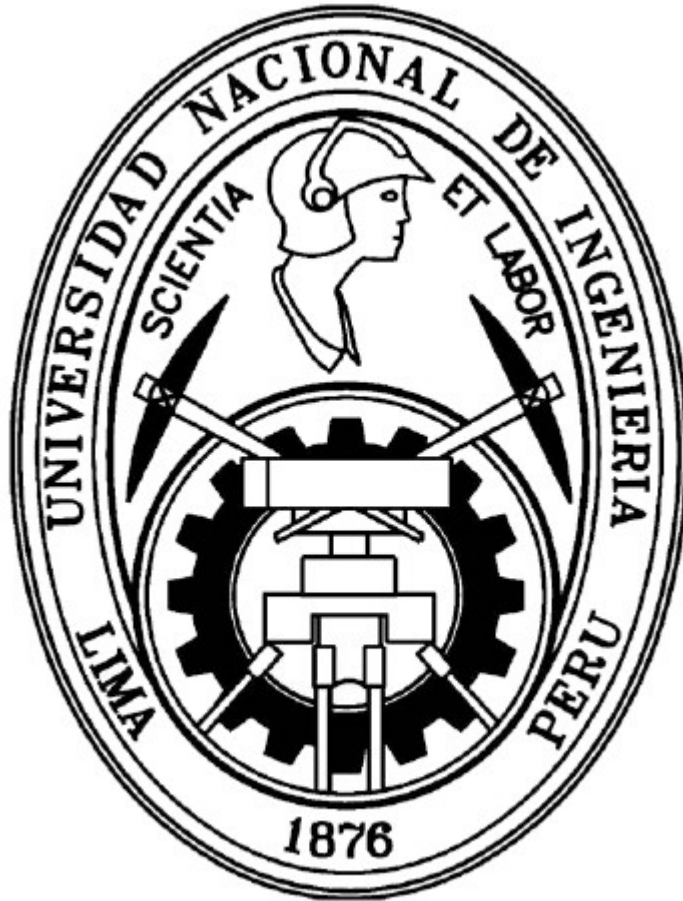


Laboratorio 5.2



Apellidos: Moreno Vera

Nombres: Felipe Adrian

Código: 20120354I

**Asignatura: Programación en Dispositivos Móviles
(CC481)**

2016 - I

Indice

Actividad 1 (3)

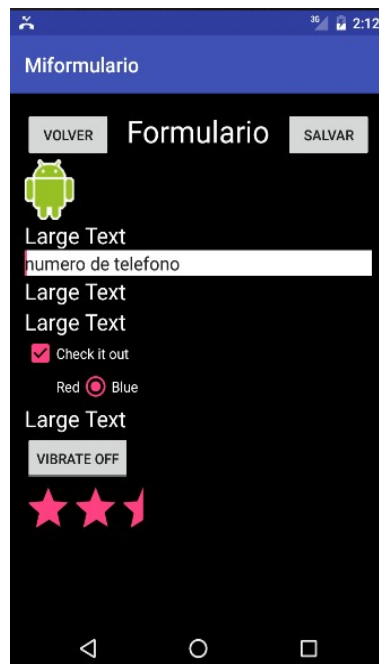
Actividad 2 (4)

Actividad 1

1. En esta actividad se pide crear un formulario igual que se ve en la imagen. Use el Layout adecuado para el y la posición de cada elemento. Tenga en cuenta las propiedades que se exponen, por ejemplo el teléfono. Observe que la imagen es un botón con el fondo una imagen que se encontrará en drawable.



Solución:



Actividad 2

1. Lo primero que tenemos que hacer es declarar nuestros elementos del formulario.

Ejemplo: `private Button imagebutton;`

2. En el método onCreate lo que vamos a realizar ahora es llamar a un método que configura las vistas. Por ejemplo:

`setUpViews();` Aquí se configura el Toast para el mensaje cuando apretas el ImageButton

3. Creamos nuestro método que configurará las vista.

4. Lo primero que tenemos que realizar es identificar de la parte vista nuestros elementos. Ejemplo:

`Imagebutton = (Button) findViewById (R.id.button);`

5. Para el botón “Volver” le añadimos en el evento onClickListener() de manera que cuando haga click que finalice la actividad con el método finish().

6. Vamos a capturar el texto del elemento EditText.

1. Para ello creamos el método sendClick en Java, teniendo en cuenta la propiedad onClick de XML

2. Luego dinamizamos en Java.

```
public void sendClick(View v) {  
    // Perform action on clicks  
    String allText = new String("campo:" + edittext.getText());  
    allText = allText + ":checkbox:";  
    if (checkbox.isChecked()) {  
        allText = allText + "Checked:";  
    } else {  
    }  
    allText = allText + "Not Checked:";  
    allText = allText + ":toggle:";  
    if (togglebutton.isChecked()) {  
        allText = allText + "Checked:";  
    } else {  
        allText = allText + "Not Checked:";  
    }  
    allText = allText + "radios:rojo:";  
    String redtext = "";  
    if (radio_red.isChecked()) {  
        redtext = "pulsado:";  
    } else {  
        redtext = "no pulsado:";  
    }  
}
```

```

    }
    allText = allText + redtext;
    allText = allText + "azul";
    String bluetext = "";
    if (radio_blue.isChecked()) {
        bluetext = "pulsado:";
    } else {
        bluetext = "no pulsado:";
    }
    allText = allText + bluetext;
    allText = allText + "rating:";
    float f = ratingbar.getRating();
    allText = allText + Float.toString( f ) + ":";
    Log.d("app", allText);
    Toast.makeText(this, allText, Toast.LENGTH_LONG).show();
}

```

7. Vamos a configurar ahora el botón para que cambie la imagen de Android de manera que cree una animación al pulsar.

1. Para ello tenemos que poner dos ficheros PNG en la carpeta drawable en la que cambie por ejemplo el color de la imagen (descargar del aula virtual).

2. Además en dicha carpeta deberemos de crear un XML con las siguientes propiedades. Tenga en cuenta el nombre PNG de la imágenes.

```

<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@drawable/android_pressed"
        android:state_pressed="true" />
    <item android:drawable="@drawable/android_normal" />
</selector>

```

Creamos un selector.xml en drawable y en ImageButton, hacemos
 android:src="@drawable/selector"

3. Por último en el XML de la actividad habrá que añadir la propiedad para que llame al XML de drawable.

android:background="@drawable/android_button"

8. Vamos ahora al CheckBox, para que cuando esté activo se active el botón de enviar y cuando no se quede sin poder pulsar.

1. Para ello creamos el método checkBoxClick en Java, teniendo en cuenta la propiedad onClick de XML

2. A Continuación, dinamizados en Java.

```
public void checkBoxClick(View v) {
    String text = "";
    if (checkbox.isChecked()) {
        text = "Selected";
        send.setEnabled(true);
        Toast.makeText(this, "Ya puedes Salvar", Toast.LENGTH_LONG).show();
    } else {
        send.setEnabled(false);
        Toast.makeText(this, "Hasta que no marques la casilla no podrás salvar", Toast.LENGTH_LONG).show();
        text = "Not selected";
    }
    Toast.makeText(this, text, Toast.LENGTH_SHORT).show();
}
```

9. En cuanto a los RadioButton. Se pide realizar un método en el que cuando se pulse un RadioButton genere un Toast con el color del identificador del Radio seleccionado. Hay que tener en cuenta que la propiedad onClick de los dos elementos tendrá el mismo nombre.

10. Por último vamos a modificar el TaggleButton.

1. Damos la propiedad cuando está encendido:

android:textOff="Vibrate off"

android:textOn="Vibrate on"

2. Dar la propiedad cuando está apagado.

3. Escribir el código Java de manera que aparezca un Toast que muestre el el texto puesto en la propiedad.

Link del github con los códigos del laboratorio:

https://github.com/Jenazad/PDM/tree/master/Laboratorio_5

Referencias

<http://stackoverflow.com/questions/10135499/android-checkbox-style>

<http://www.mkyong.com/android/android-rating-bar-example/>

<http://www.iteramos.com/pregunta/5934/como-funciona-exactamente-el-androidonclick-atributo-xml-difieren-de-setonclicklistener>

<http://www.mkyong.com/android/android-imagebutton-example/>

<http://developer.android.com/intl/es/reference/android/widget/ImageButton.html>

<http://www.mkyong.com/android/android-radio-buttons-example/>