

PRACTICA # 04.

OBJETIVO GENERAL.

Crear y desarrollar una aplicación que de una bienvenida usando un boton (Button), una etiqueta (TextView) y un EditText de tipo nombre persona, para conocer y aplicar la interacción entre ellos mediante el uso de eventos (Listener) y cambios de propiedades en tiempo de ejecución de acuerdo a las especificaciones de la figura.



OBJETIVOS ESPECÍFICOS:

- Utilizar LinearLayout como contenedor principal de la aplicación.
- Crear objetos:
 - Un TextView : Utilizado para desplegar la bienvenida
 - Un EditText : Para introducir un nombre de una persona
 - Un Button : Utilizado para desplegar el mensaje de bienvenida.
- Crear un evento que modifique el texto del TextView y le dé la bienvenida a la persona que escribió su nombre en el EditText.

Programación de Dispositivos Móviles.
Delio Coss Camilo.

- NOTA: Cree mediante el SDK de Android una aplicación vacía (consulte la practica 1 con los pasos necesarios para la realización de esta práctica).

CREACION DE LA INTERFACE GRAFICA.

Usaremos el código siguiente para la interface gráfica.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="20dp">

    <TextView
        android:id="@+id/txtMensaje"
        android:layout_margin="5dp"
        android:text="Escribe tu nombre"
        android:textSize="24sp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <EditText
        android:id="@+id/edtMensaje"
        android:layout_margin="5dp"
        android:textSize="24sp"
        android:inputType="textPersonName"
        android:hint="Escribe tu nombre"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <Button
        android:id="@+id/btnSaludar"
        android:text="Saludar"
        android:textSize="24sp"
        android:layout_margin="5dp"
        android:onClick="saludarPersona"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</LinearLayout>
```

DESARROLLO DE LA LOGICA DE APLICACIÓN.

El código inicial de nuestra actividad principal se verá:

```
package com.example.delio.app01a;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Es aquí donde iremos agregando líneas de código para darle funcionalidad a nuestra aplicación.

DECLARACION DE VARIABLES

En primer lugar debemos de crear variables que serán usadas para relacionar la parte grafica con la lógica de la aplicación, en este caso, son todos los elementos que tendrán acción al funcionar la aplicación.

```
private TextView txtMensaje;
private EditText edtMensaje;
private Button btnSaludar;
```

Después de la declaración de variables, DEBEMOS de relacionar los elementos de la interface gráfica con nuestras variables de nuestra lógica de aplicación.

```
txtMensaje = (TextView) findViewById(R.id.txtMensaje);
edtMensaje = (EditText) findViewById(R.id.edtMensaje);
btnSaludar = (Button) findViewById(R.id.btnSaludar);
```

Vamos a crear la lógica para el botón al evento de tocar el botón:

```
public void saludarPersona(View view){
    String nombre = edtMensaje.getText().toString();
    txtMensaje.setText("Te doy la bienvenida al curso " + nombre +
        ", espero sea de utilidad para ti");
    edtMensaje.setText("");
}
```

Este método que usamos es llamado desde el archivo XML mediante la siguiente instrucción, es una forma diferente de crear eventos a un objeto de la aplicación.

```
android:onClick="saludarPersona"
```

Que se localiza en el marcado XML del botón, debemos de mencionar dos detalles importantes si deseamos relacionar los eventos desde el archivo XML y son:

- El método debe ser público y recibir como parámetros un View

Programación de Dispositivos Móviles.

Delio Coss Camilo.

Con todo esto ya tenemos nuestra aplicación funcional, solo deberá de ejecutarla dentro de su emulador.

```
1 package com.example.delio.app04;
2 import ...
8
9 public class MainActivity extends AppCompatActivity {
10     private TextView txtMensaje;
11     private EditText edtMensaje;
12     private Button btnSaludar;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18
19         txtMensaje = (TextView) findViewById(R.id.txtMensaje);
20         txtMensaje.setText("");
21         edtMensaje = (EditText) findViewById(R.id.edtMensaje);
22         btnSaludar = (Button) findViewById(R.id.btnSaludar);
23     }
24     public void saludarPersona(View view){
25         String nombre = edtMensaje.getText().toString();
26         txtMensaje.setText("Te doy la bienvenida al curso " + nombre +
27             ", espero sea de utilidad para ti");
28         edtMensaje.setText("");
29     }
30 }
```

Código completo de referencia.