



PROYECTO FINAL – SENSORES

Estudiantes: Eimy Guerra 3-745-950
Andrew González 8-954-1126
Reynalis López 3-743-1190
Ary Sánchez 8-904-1189

Grupo: 11L143

Sensor de Proximidad

Modificación en AndroidManifest.xml

Dentro del código del archivo de AndroidManifest.xml se agregó la siguiente línea de comando para solicitar el permiso de uso del sensor de proximidad.

```
<uses-feature android:name="android.hardware.sensor.proximity"
    android:required="true" />
```

Código de MainActivity.java

```
package com.example.proximity_sensor;

import android.graphics.Color;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    private SensorManager sensorManager;
    private Sensor proximitySensor;
    private SensorEventListener proximitySensorListener;
    TextView indicator;
    ImageView alert, good;

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

        indicator=findViewById(R.id.txt_indicator);
        alert=findViewById(R.id.img1);
        good=findViewById(R.id.img2);

        sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
```

```

        //seleccionamos el tipo de sensor que utilizaremos
        proximitySensor =
        sensorManager.getDefaultSensor(Sensor.TYPE_PROXIMITY);

        //verificamos si el EL DISPOSITIVO tiene el sensor, si no tiene
        sensor finalizamos el activity
        if (proximitySensor == null) {
            Toast.makeText(this, "Proximity sensor is not available !",
            Toast.LENGTH_LONG).show();
            finish();
        }

        //agregamos nuestro evento
        proximitySensorListener = new SensorEventListener() {
            /*este metodo detecta cuando los valores del sensor han cambiado
            cuando nos alejamos o nos acercamos*/
            @Override
            public void onSensorChanged(SensorEvent sensorEvent) {
                //verificamos que la distancia al sensor sea menor al rango
                maximo
                if (sensorEvent.values[0] <
                proximitySensor.getMaximumRange()) {
                    // al cumplirse signifca que estamos dentro del rango del
                    sensor
                    getWindow().getDecorView().setBackgroundColor(Color.RED);
                    indicator.setVisibility(View.VISIBLE);
                    indicator.setText("ACTION - NEAR THE SENSOR");
                    good.setVisibility(View.INVISIBLE);
                    alert.setVisibility(View.VISIBLE);
                } else {
                    getWindow().getDecorView().setBackgroundColor(Color.GREEN);
                    indicator.setVisibility(View.VISIBLE);
                    indicator.setText("STATIC - FAR FROM SENSOR");
                    alert.setVisibility(View.INVISIBLE);
                    good.setVisibility(View.VISIBLE);
                }
            }
            @Override
            public void onAccuracyChanged(Sensor sensor, int i) {
            }
        };
    }
    @Override
    protected void onResume() {
        super.onResume();
        sensorManager.registerListener(proximitySensorListener,
        proximitySensor,
        2 * 1000 * 1000);
    }

    @Override
    protected void onPause() {
        super.onPause();
        sensorManager.unregisterListener(proximitySensorListener);
    }
}

```

Código de activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/txt_indicator"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TextView"
        android:textColor="#000000"
        android:textSize="24sp"
        android:textStyle="bold"
        android:visibility="invisible"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.196"
        tools:visibility="visible" />

    <ImageView
        android:id="@+id/img1"
        android:layout_width="218dp"
        android:layout_height="197dp"
        android:visibility="invisible"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.497"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.408"
        app:srcCompat="@drawable/alert"
        tools:visibility="visible" />

    <ImageView
        android:id="@+id/img2"
        android:layout_width="218dp"
        android:layout_height="197dp"
        android:visibility="invisible"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.497"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.408"
        app:srcCompat="@drawable/good"
        tools:visibility="visible" />
</androidx.constraintlayout.widget.ConstraintLayout>
```