

INSTITUTO TECNOLOGICO DE ORIZABA



TEC. DE PROG. EMERGENTES PARA LA WEB

Práctica 06

UNIDAD 2

Manuel Panzi Utrera

9:00 - 10:00

Karyme Michelle Romero Ovando

21011037

Práctica 06 - El Acelerómetro

Expectativas

Conocer el manejo del acelerómetro en un móvil.

Competencias adquiridas al terminar la práctica

Comprender el desarrollo de aplicaciones en dispositivos móviles que nos permitan usar el acelerómetro como interfaz de ingreso de datos.

Planteamiento

Desarrollar una aplicación que nos muestre en la pantalla los valores reportados por el acelerómetro y un aviso para cuando se detecte movimiento en el teléfono.

activity_main.xml

```
<TextView
                          android:id="@+id/txtCoordenadaY"
                          android:layout_width="match_parent"
                          android:layout_height="wrap_content"
                          android:text="Valor de Y:"
                          android:textSize="22sp"
                          android:textColor="#0277BD"/>
                      <TextView
                          android:layout_width="match_parent"
                          android:layout_height="wrap_content"
                          android:text="Valor de Z:"
                          android:textSize="22sp"
                          android:textColor="#0277BD"/>
                  </LinearLayout>
              </androidx.cardview.widget.CardView>
              <TableLayout
                  android:layout_width="match_parent"
                  android:layout_height="wrap_content"
                  android:stretchColumns="1,2"
                  android:background="#B3E5FC"
62 ■
                  android:padding="12dp">
```

```
<TableRow>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textStyle="bold"
        android:padding="8dp"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Mínimo"
        android:textStyle="bold"
        android:padding="8dp"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Máximo"
        android:textStyle="bold"
        android:padding="8dp"/>
</TableRow>
```

```
<TableRow>
        android:text="X="
        android:padding="8dp"/>
   <TextView
        android:id="@+id/minX"
        android:text="0.0"
        android:padding="8dp"/>
   <TextView
        android:id="@+id/maxX"
        android:text="0.0"
        android:padding="8dp"/>
</TableRow>
<TableRow>
        android:padding="8dp"/>
        android:padding="8dp"/>
        android:text="0.0"
        android:padding="8dp"/>
</TableRow>
```

MainActivity.java

```
package com.example.acelerometro;

import androidx.appcompat.app.AppCompatActivity;

import androidx.ontent.pm.ActivityInfo;

import android.hardware.SensorEvent;

import android.hardware.SensorEvent;

import android.hardware.SensorEventListener;

import android.hardware.SensorManager;

import android.os.Bundle;

import android.widget.TextView;

import java.util.List;

public class MainActivity extends AppCompatActivity implements SensorEventListener {

2 usages

private TextView txtCoordenadaX, txtCoordenadaY, txtCoordenadaZ;
2 usages

private TextView minX, minY, minZ, maxX, maxY, maxZ;
6 usages

private float actualX = 0, actualY = 0, actualZ = 0;
3 usages

private float minValorX = Float.MAX_VALUE, minValorY = Float.MAX_VALUE, minValorZ = Float.MAX_VALUE;
3 usages

private float maxValorX = Float.MIN_VALUE, maxValorY = Float.MIN_VALUE, maxValorZ = Float.MIN_VALUE;

private float maxValorX = Float.MIN_VALUE, maxValorY = Float.MIN_VALUE, maxValorZ = Float.MIN_VALUE;
```

```
@Override
protected void onCreate(Bundle savedInstanceState){
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
this.setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

txtCoordenadaX = findViewById(R.id.txtCoordenadaX);
txtCoordenadaY = findViewById(R.id.txtCoordenadaY);
txtCoordenadaZ = findViewById(R.id.txtCoordenadaZ);

minX = findViewById(R.id.minX);
minZ = findViewById(R.id.minX);
minZ = findViewById(R.id.minY);
maxX = findViewById(R.id.minY);
maxY = findViewById(R.id.maxX);
maxY = findViewById(R.id.maxZ);

maxX = findViewById(R.id.maxZ);

}

@Override
protected void onResume() {
super.onResume();
SensorNanager sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
List<Sensor> sensors = sensorManager.getSensorList(Sensor.TYPE_ACCELEROMETER);

if (sensors.size() > 0) {
sensorManager.registerListener( Mistener, this, sensors.get(0), SensorManager.SENSOR_DELAY_GAME);
}
}
```

```
@Override
protected void onStop() {

SensorManager sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
sensorManager.unregisterListener(this);
super.onStop();
}

3 usages

Qoverride
public void onAccuracyChanged(Sensor sensor, int accuracy) {}
```

```
@Uverride
public void onSensorChanged(SensorEvent event){
synchronized (this) {
    actualX = event.values[0];
    actualY = event.values[2];

    txtCoordenadaX.setText("Valor de X: " + String.format("%.2f", actualX));
    txtCoordenadaY.setText("Valor de Y: " + String.format("%.2f", actualX));
    txtCoordenadaZ.setText("Valor de Y: " + String.format("%.2f", actualY));
    txtCoordenadaZ.setText("Valor de Z: " + String.format("%.2f", actualZ));

// Actualizar minimos
    if (actualX < minValorX) minValorX = actualX;
    if (actualY < minValorY) minValorY = actualY;
    if (actualY < minValorY) minValorZ = actualZ;

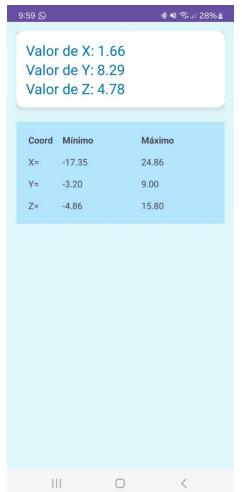
// Actualizar méximos
    if (actualX > maxValorX) maxValorX = actualZ;

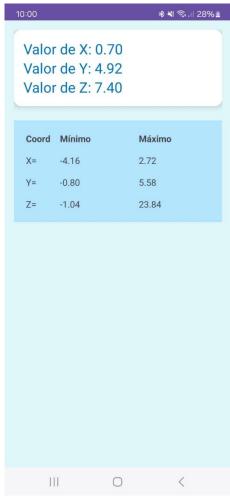
// Actualizar méximos
    if (actualX > maxValorX) maxValorX = actualZ;

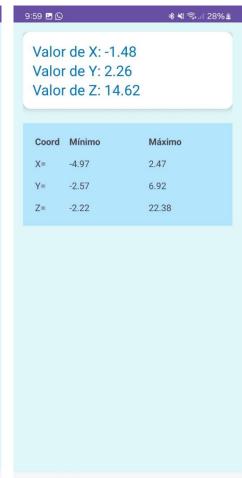
if (actualY > maxValorY) maxValorY = actualZ;

// Mostrar en la tabla
    minX.setText(String.format("%.2f", minValorX));
    minY.setText(String.format("%.2f", minValorY));
    maxX.setText(String.format("%.2f", minValorY));
    maxX.setText(String.format("%.2f", maxValorY));
    maxX.setText(String.format("%.2f", maxValorY));
    maxX.setText(String.format("%.2f", maxValorY));
    maxX.setText(String.format("%.2f", maxValorY));
}
```

Resultado







<

