

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: 1IL143

Contador de Pasos (Step Counter)

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 conteo de pasos.

```
<uses-permission  
android:name="android.permission.ACTIVITY_RECOGNITION"/>
```

Creación de archivo de recurso drawable

Creamos un archivo drawable llamado circle.xml en donde estamos haciendo un trazo circular para usarlo, en ImageView en el archivo XML principal de la aplicación.

```
<?xml version="1.0" encoding="utf-8"?>  
<selector xmlns:android="http://schemas.android.com/apk/res/android">  
  <item>  
    <shape android:shape="oval">  
      <stroke android:color="#0F9D58" android:width="3dp"/>  
      <size android:width="120dp" android:height="120dp"/>  
    </shape>  
  </item>  
</selector>
```

Código de MainActivity.kt

Nota: para esta aplicación se utilizó el lenguaje Kotlin, esto debido a que los tutoriales utilizados en java no nos permitían correr la aplicación adecuadamente.

```
package com.example.step_counter  
  
import android.content.Context  
import android.hardware.Sensor  
import android.hardware.SensorEvent  
import android.hardware.SensorEventListener  
import android.hardware.SensorManager  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.util.Log  
import android.widget.TextView  
import android.widget.Toast
```

```

class MainActivity : AppCompatActivity(), SensorEventListener {

    // hemos asignado sensorManger a null
    private var sensorManager: SensorManager? = null
    // Creando una variable que dará el estado de ejecución
    private var running = false

    // Creando una variable que contará los pasos totales
    private var totalSteps = 0f

    // Creando una variable que cuenta los pasos totales anteriores
    private var previousTotalSteps = 0f

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        loadData()
        resetSteps()

        // Agregando un contexto de SENSOR_SERVICE como Administrador de
        // sensores
        sensorManager = getSystemService(Context.SENSOR_SERVICE) as
        SensorManager
    }

    override fun onResume() {
        super.onResume()
        running = true

        // Devuelve el número de pasos dados por el usuario desde el último
        // reinicio mientras estaba activado
        val stepSensor =
        sensorManager?.getDefaultSensor(Sensor.TYPE_STEP_COUNTER)

        if (stepSensor == null) {
            // Esto le dará un toast message al usuario si no hay un sensor
            // en el dispositivo
            Toast.makeText(this, "No sensor detected on this device",
            Toast.LENGTH_SHORT).show()
        } else {
            sensorManager?.registerListener(this, stepSensor,
            SensorManager.SENSOR_DELAY_UI)
        }
    }

    override fun onSensorChanged(event: SensorEvent?) {

        // Llamando al TextView que hicimos en activity_main.xml
        var tv_stepsTaken = findViewById<TextView>(R.id.tv_stepsTaken)

        if (running) {
            totalSteps = event!!.values[0]

            // Los pasos actuales se calculan tomando la diferencia de los

```

```

pasos totales y los pasos anteriores
        val currentSteps = totalSteps.toInt() -
previousTotalSteps.toInt()

        // Mostrará los pasos actuales al usuario
        tv_stepsTaken.text = ("{$currentSteps}")
    }
}

fun resetSteps() {
    var tv_stepsTaken = findViewById<TextView>(R.id.tv_stepsTaken)
    tv_stepsTaken.setOnClickListener {
        // Esto dará un toast menssage si el usuario desea restablecer
los pasos
        Toast.makeText(this, "Long tap to reset steps",
Toast.LENGTH_SHORT).show()
    }

    tv_stepsTaken.setOnLongClickListener {

        previousTotalSteps = totalSteps

        // Cuando el usuario haga un toque largo en la pantalla, los
pasos se restablecerán a 0
        tv_stepsTaken.text = 0.toString()

        saveData()

        true
    }
}

private fun saveData() {

    // Shared Preferences nos permitirá guardar y recuperar datos en
forma de clave, par de valores.
    // En esta función guardamos la data.
    val sharedPreferences = getSharedPreferences("myPrefs",
Context.MODE_PRIVATE)

    val editor = sharedPreferences.edit()
    editor.putFloat("key1", previousTotalSteps)
    editor.apply()
}

private fun loadData() {

    // En esta función recibimos la data
    val sharedPreferences = getSharedPreferences("myPrefs",
Context.MODE_PRIVATE)
    val savedNumber = sharedPreferences.getFloat("key1", 0f)

    // Log.d se utiliza para depurar
    Log.d("MainActivity", "{$savedNumber}")

    previousTotalSteps = savedNumber
}

```

```

        override fun onAccuracyChanged(sensor: Sensor?, accuracy: Int) {

        }
    }
}

```

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/steps"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="290dp"
        android:layout_marginEnd="8dp"
        android:text="Steps"
        android:textColor="#E66E12"
        android:textSize="45sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tv_stepsTaken"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="30dp"
        android:layout_marginEnd="8dp"
        android:text="0"
        android:textColor="#E66E12"
        android:textSize="37sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/steps" />

    <ImageView
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:layout_marginTop="264dp"
        android:background="@drawable/circle"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.497"

```

```
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />  
</androidx.constraintlayout.widget.ConstraintLayout>
```

Capturas de Funcionamiento

Captura luego de realizar algunos pasos.



Captura luego de resetear los pasos.

