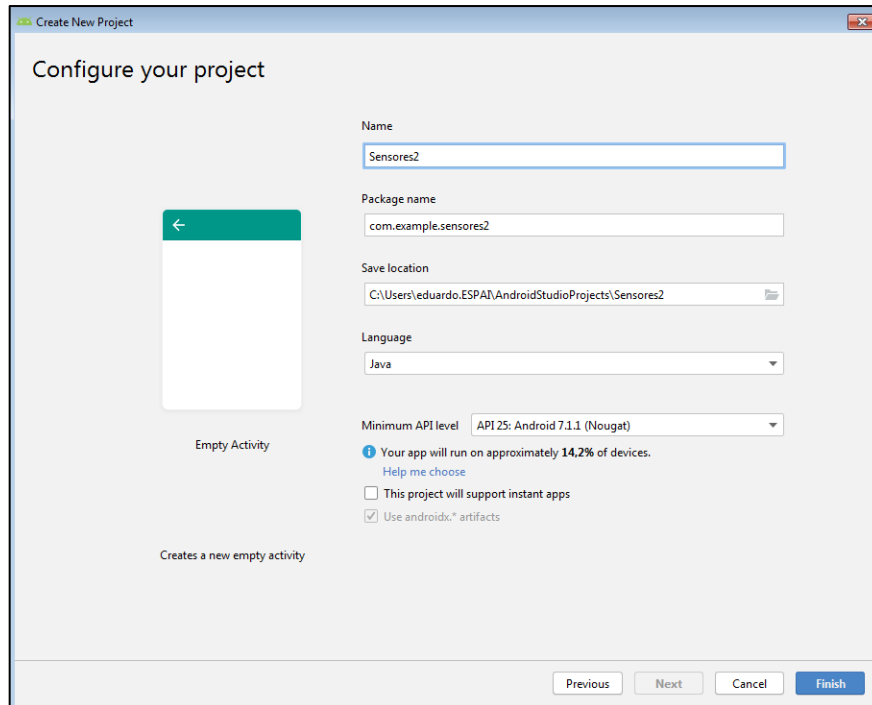


PRACTICA 8: SENSORES 2

Parte I: Creación de una vista desde código para mostrar los datos de los sensores

Paso 1. Crea un nuevo proyecto con nombre *Sensores2*.



Paso 2. Abre el Layout *main.xml* y añade al *LinearLayout* el siguiente atributo:

`<LinearLayout android:id="@+id/raiz"`



Paso 3. Reemplaza el código de la actividad por el siguiente:

```
public class MainActivity extends AppCompatActivity implements SensorEventListener {

    private List<Sensor> listaSensores;
    private TextView aTextView[][] = new TextView[20][3];

    @Override public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        LinearLayout raiz = (LinearLayout) findViewById(R.id.raiz);
        SensorManager sm = (SensorManager) getSystemService(SENSOR_SERVICE);
        listaSensores = sm.getSensorList(Sensor.TYPE_ALL);
        int n = 0;
        for (Sensor sensor : listaSensores) {
            TextView mTextView = new TextView(context: this);
            mTextView.setText(sensor.getName());
            raiz.addView(mTextView);
            LinearLayout nLinearLayout = new LinearLayout(context: this);
            raiz.addView(nLinearLayout);
            for (int i = 0; i < 3; i++) {
                aTextView[n][i] = new TextView(context: this);
                aTextView[n][i].setText("?");
                aTextView[n][i].setWidth(87);
            }
            TextView xTextView = new TextView(context: this);
            xTextView.setText(" X: ");
            nLinearLayout.addView(xTextView);
            nLinearLayout.addView(aTextView[n][0]);
            TextView yTextView = new TextView(context: this);
            yTextView.setText(" Y: ");
            nLinearLayout.addView(yTextView);
            nLinearLayout.addView(aTextView[n][1]);
            TextView zTextView = new TextView(context: this);
            zTextView.setText(" Z: ");

            nLinearLayout.addView(zTextView);
            nLinearLayout.addView(aTextView[n][2]);
            sm.registerListener(listener: this, sensor, SensorManager.SENSOR_DELAY_UI);
            n++;
        }
    }
}
```

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

@Override
public void onSensorChanged(SensorEvent event) {
    synchronized (this) {
        int n = 0;
        for (Sensor sensor: listaSensores) {
            if (event.sensor == sensor) {
                for (int i=0; i<event.values.length; i++){
                    if (n<18 && i<3)
                        aTextView[n][i].setText(String.valueOf(event.values[i]));
                }
            }
            n++;
        }
    }
}
```

Como puedes observar esta actividad utiliza el *Layout* creado por defecto que básicamente es un **LinearLayout** (en el código corresponde a la variable raíz) con un **TextView** en su interior (“*Hello Word ...*”). A raíz se le va a ir añadiendo una serie de vistas adicionales según los sensores encontrados en el dispositivo. Por cada sensor se añade: un **TextView** con el nombre del sensor, un **LinearLayout** de tipo horizontal [2] para contener a su vez un **TextView** con “X:”, un **TextView** con el valor del sensor en el eje X, un **TextView** con “Y”, un **TextView** con el valor del sensor en el eje Y, un **TextView** con “Z:” y un **TextView** con el valor del sensor en el eje Z. Las referencias a los **TextView** donde se visualizará los valores de los sensores se almacenan en el array **aTextView[][]** donde el primer índice identifica el número de sensor y el segundo la dimensión X,Y o Z.

En el método **onSensorChanged()** se hace un bucle para localizar el índice del sensor que ha cambiado y se modifican los **TextView** correspondiente al sensor con los valores leídos.

NOTA: No todos los sensores tienen tres dimensiones. Por ejemplo, en el caso del sensor de temperatura solo se cambiará en el valor de X.

Paso 4. Verifica sobre un dispositivo real que el programa funciona correctamente.

