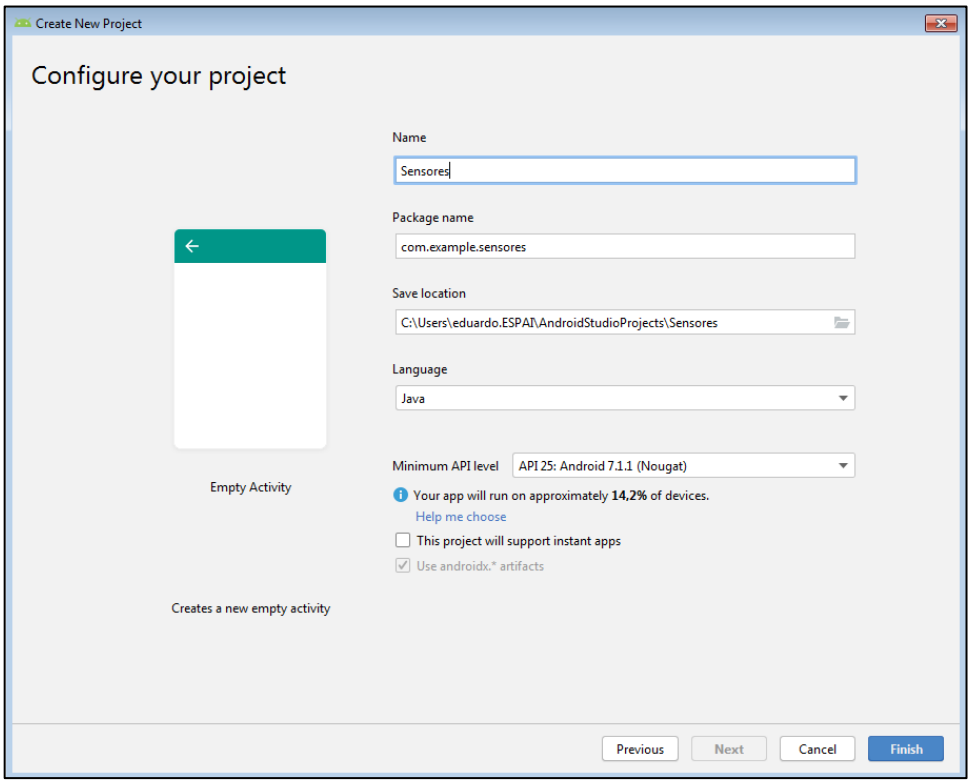
PRACTICA 7: SENSORES

Parte I: Listar los sensores del dispositivo

No todos los dispositivos disponen de los mismos sensores. Por lo tanto, la primera tarea consiste en averiguar los sensores disponibles.

Paso 1. Crea un nuevo proyecto con nombre Sensores.



Paso 2. Añade la siguiente propiedad al TextView de *res/layout/main.xml*:

android:id="@+id/salida"

Paso 3. Inserta este código en la actividad principal:

```
public class MainActivity extends AppCompatActivity {

    private TextView salida;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        salida = (TextView) findViewById(R.id.salida);
        SensorManager sensorManager = (SensorManager)getSystemService(SENSOR_SERVICE);
        List<Sensor> listaSensores = sensorManager.getSensorList(Sensor.TYPE_ALL);
        for(Sensor sensor: listaSensores) {
            log(sensor.getName());
        }
    }

    private void log(String string) {
        salida.append(string + "\n");
    }
}
```

Paso 4. El método comienza indicando el *Layout* de la actividad y obteniendo el TextView salida, donde mostraremos los resultados. A continuación vamos a vamos a utilizar el método getSystemService para solicitar al sistema servicios específicos. Este método pertenece a la clase Context (Como somos Activity también somos Context) y será muy utilizados para acceder a gran cantidad de servicios del sistema. Al indicar como parámetro SENSOR_SERVICE, indicamos que queremos utilizar los sensores. Lo haremos a través del objeto sensorManager. En primer lugar llamamos al método getSensorList() del objeto para que nos de listaSensores, una lista de objetos Sensor. La siguiente línea recorre todos los elementos de esta lista parar llamar a su método getName() para mostrar el nombre de sensor.

Paso 5. Ejecuta el programa.

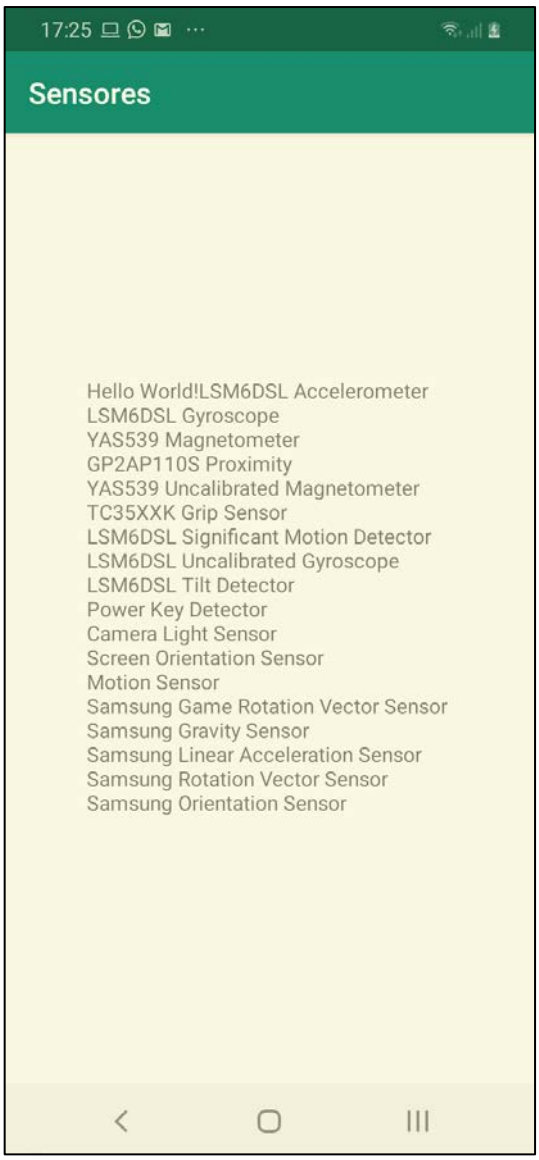
Esta es una lista de los valores devueltos por el código anterior ejecutándose en el HTC Magic:

AK8976A 3-axis Accelerometer

AK8976A 3-axis Magnetic field sensor

AK8976A Orientation sensor

AK8976A Temperature sensor



Paso 6. El AK8976A es una combinación de acelerómetro de tres ejes y magnetómetro de tres ejes. Combinando la lectura de los campos gravitatorio y magnético terrestres proporciona también información de orientación. Incluye además un sensor interno de temperatura, útil para comprobar si el móvil se está calentado demasiado.

Parte II: Acceso a los datos del sensor

Veamos ahora como obtener la lectura de cada uno de los sensores.

Paso 1. Copia el siguiente código al final de onCreate.

```
public class MainActivity extends AppCompatActivity {

    private TextView salida;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        salida = (TextView) findViewById(R.id.salida);
        SensorManager sensorManager = (SensorManager)getSystemService(SENSOR_SERVICE);
        List<Sensor> listaSensores = sensorManager.getSensorList(Sensor.TYPE_ALL);
        for(Sensor sensor: listaSensores) {
            log(sensor.getName());
        }

        listaSensores = sensorManager.getSensorList(Sensor.TYPE_ORIENTATION);

        if (!listaSensores.isEmpty()) {
            Sensor orientationSensor = listaSensores.get(0);
            sensorManager.registerListener(this, orientationSensor, SensorManager.SENSOR_DELAY_UI);}

        listaSensores = sensorManager.getSensorList(Sensor.TYPE_ACCELEROMETER);

        if (!listaSensores.isEmpty()) {
            Sensor acelerometerSensor = listaSensores.get(0);
            sensorManager.registerListener(this, acelerometerSensor, SensorManager.SENSOR_DELAY_UI);}

        listaSensores = sensorManager.getSensorList(Sensor.TYPE_MAGNETIC_FIELD);

        if (!listaSensores.isEmpty()) {
            Sensor magneticSensor = listaSensores.get(0);
            sensorManager.registerListener(this, magneticSensor, SensorManager.SENSOR_DELAY_UI);}

        listaSensores = sensorManager.getSensorList(Sensor.TYPE_TEMPERATURE);

        if (!listaSensores.isEmpty()) {
            Sensor temperatureSensor = listaSensores.get(0);
            sensorManager.registerListener(this, temperatureSensor, SensorManager.SENSOR_DELAY_UI);}
    }
```

Comenzamos consultando si disponemos de un sensor de orientación. Para ello preguntamos al sistema que nos de todos los sensores de este tipo llamando a getSensorList(). Si la lista no está vacía obtenemos el primer elemento (el 0). Es necesario registrar cada tipo de sensor por separado para poder obtener información de él. El método registerListener() toma como primer parámetro un objeto que implemente el interface SensorEventListener, veremos a continuación cómo se implementa esta interfaz (se indica this porque la clase que estamos definiendo implementará este interfaz para recoger eventos de sensores). El segundo parámetro es el sensor que estamos registrando. Y el tercero indica al sistema con qué frecuencia nos gustaría recibir actualizaciones del sensor. Acepta cuatro posibles valores, de menor a mayor frecuencia tenemos:

SENSOR_DELAY_NORMAL, SENSOR_DELAY_UI, SENSOR_DELAY_GAME y SENSOR_DELAY_FASTEST. Esta indicación sirve para que el sistema estime cuánta atención necesitan los sensores, pero no garantiza una frecuencia concreta.

Paso 2. Para que nuestra clase implemente el interface que hemos comentado añade a la declaración de la clase:

implements SensorEventListener

```
public class MainActivity extends AppCompatActivity implements SensorEventListener {

    private TextView salida;

    @Override
    public void onSensorChanged(SensorEvent sensorEvent) {

    }

    @Override
    public void onAccuracyChanged(Sensor sensor, int i) {

    }
```

Paso 3. Para recibir los datos de los sensores tenemos que implementar dos métodos de la interfaz SensorEventListener:

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


@Override
public void onSensorChanged(SensorEvent evento) {
    //Cada sensor puede provocar que un thread principal pase por aquí
    //así que sincronizamos el acceso (se verá más adelante)
    synchronized (this) {
        switch(evento.sensor.getType()) {
            case Sensor.TYPE_ORIENTATION:
                for (int i=0 ; i<3 ; i++) {
                    log( string: "Orientación "+i+": "+evento.values[i]);
                }
                break;
            case Sensor.TYPE_ACCELEROMETER:
                for (int i=0 ; i<3 ; i++) {
                    log( string: "Acelerómetro "+i+": "+evento.values[i]);
                }
                break;
            case Sensor.TYPE_MAGNETIC_FIELD:
                for (int i=0 ; i<3 ; i++) {
                    log( string: "Magnetismo "+i+": "+evento.values[i]);
                }
                break;
            default:
                for (int i=0 ; i<evento.values.length ; i++) {
                    log( string: "Temperatura "+i+": "+evento.values[i]);
                }
        }
    }
}
```

Paso 4. Cuando implementamos un interface estamos obligados a implementar todos sus métodos. En este caso son dos. Para onAccuracyChanged no queremos ninguna acción específica, pero lo tenemos que incluir.Cuando un sensor cambie se llamará al método onSensorChanged, aquí comprobamos qué sensor ha causado la llamada y leeremos los datos.

Paso 5. Verifica que el programa funciona correctamente.

Cuando el evento se dispara en el método onSensorChanged comprobamos qué sensor lo ha causado y leemos los datos. Los posibles valores devueltos se indican en la documentación de la clase SensorEvent.