

## **Apartado 15:** **Sensores**

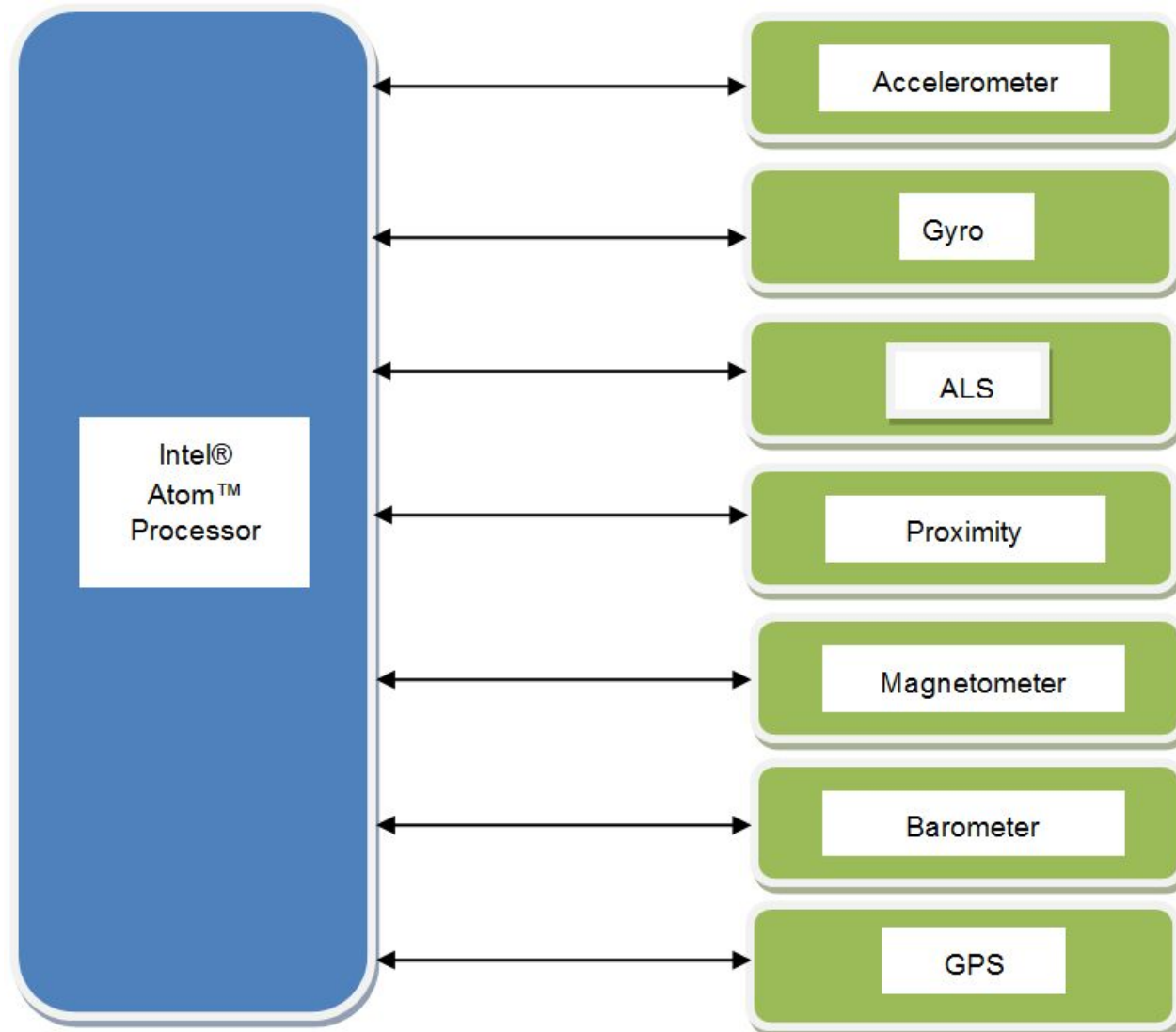
**Autor:**

Miguel Ángel García Cumberras

# Introducción a los sensores

- Sensores: todos los dispositivos con los podremos obtener información con el mundo exterior (no consideraremos la cámara ni el micrófono ni el GPS).
- Android permite acceder a los sensores internos del dispositivo a través de las clases `Sensor`, `SensorEvent`, `SensorManager`, y la interfaz `SensorEventListener`, del paquete `android.hardware`.
- La clase `Sensor` acepta ocho tipos de sensores. Aunque, los sensores disponibles varían en función del dispositivo utilizado.

# Introducción a los sensores



# Categorías de sensores

- Sensores de movimiento: acelerómetro, gravedad, giroscopio o vector rotacional
- Sensores de ambiente: temperatura y presión exterior, iluminación, humedad (barómetros, termómetros...)
- Sensores de posición (ya los hemos visto)

[http://developer.android.com/guide/topics/sensors/sensors\\_overview.html](http://developer.android.com/guide/topics/sensors/sensors_overview.html)

# Categorías de sensores

Sensores de movimiento	Acelerómetro (TYPE_ACCELEROMETER)	Mide las aceleraciones de un dispositivo en $m/s^2$	Detección de movimiento
	Giroscopio (TYPE_GYROSCOPE)	Mide las velocidades de rotación de un dispositivo	Detección de rotación
Sensores de posición	Magnetómetro (TYPE_MAGNETIC_FIELD)	Mide la intensidad de los campos geomagnéticos de la tierra en $\mu T$	Brújula
	Proximidad (TYPE_PROXIMITY)	Mide la proximidad de un objeto en cm	Detección de objeto a corta distancia
	GPS (no es un tipo de android.hardware.Sensor)	Obtiene ubicaciones geográficas precisas del dispositivo	Detección de ubicaciones geográficas precisas
Sensores del entorno	ALS (TYPE_LIGHT)	Mide el nivel de luz ambiental en lx	Control automático de brillo en pantalla
	Barómetro	Mide la presión del aire ambiental en mbar	Detección de altitud

# Categorías de sensores

Sensor	Type	Description	Common Uses
<code>TYPE_ACCELEROMETER</code>	Hardware	Measures the acceleration force in $\text{m/s}^2$ that is applied to a device on all three physical axes (x, y, and z), including the force of gravity.	Motion detection (shake, tilt, etc.).
<code>TYPE_AMBIENT_TEMPERATURE</code>	Hardware	Measures the ambient room temperature in degrees Celsius ( $^{\circ}\text{C}$ ). See note below.	Monitoring air temperatures.
<code>TYPE_GRAVITY</code>	Software or Hardware	Measures the force of gravity in $\text{m/s}^2$ that is applied to a device on all three physical axes (x, y, z).	Motion detection (shake, tilt, etc.).
<code>TYPE_GYROSCOPE</code>	Hardware	Measures a device's rate of rotation in $\text{rad/s}$ around each of the three physical axes (x, y, and z).	Rotation detection (spin, turn, etc.).
<code>TYPE_LIGHT</code>	Hardware	Measures the ambient light level (illumination) in lx.	Controlling screen brightness.
<code>TYPE_LINEAR_ACCELERATION</code>	Software or Hardware	Measures the acceleration force in $\text{m/s}^2$ that is applied to a device on all three physical axes (x, y, and z), excluding the force of gravity.	Monitoring acceleration along a single axis.
<code>TYPE_MAGNETIC_FIELD</code>	Hardware	Measures the ambient geomagnetic field for all three physical axes (x, y, z) in $\mu\text{T}$ .	Creating a compass.

# Categorías de sensores

<code>TYPE_ORIENTATION</code>	Software	Measures degrees of rotation that a device makes around all three physical axes (x, y, z). As of API level 3 you can obtain the inclination matrix and rotation matrix for a device by using the gravity sensor and the geomagnetic field sensor in conjunction with the <code>getRotationMatrix()</code> method.	Determining device position.
<code>TYPE_PRESSURE</code>	Hardware	Measures the ambient air pressure in hPa or mbar.	Monitoring air pressure changes.
<code>TYPE_PROXIMITY</code>	Hardware	Measures the proximity of an object in cm relative to the view screen of a device. This sensor is typically used to determine whether a handset is being held up to a person's ear.	Phone position during a call.
<code>TYPE_RELATIVE_HUMIDITY</code>	Hardware	Measures the relative ambient humidity in percent (%).	Monitoring dewpoint, absolute, and relative humidity.
<code>TYPE_ROTATION_VECTOR</code>	Software or Hardware	Measures the orientation of a device by providing the three elements of the device's rotation vector.	Motion detection and rotation detection.
<code>TYPE_TEMPERATURE</code>	Hardware	Measures the temperature of the device in degrees Celsius (°C). This sensor implementation varies across devices and this sensor was replaced with the <code>TYPE_AMBIENT_TEMPERATURE</code> sensor in API Level 14	Monitoring temperatures.

# Disponibilidad por plataforma

Sensor	Android 4.0 (API Level 14)	Android 2.3 (API Level 9)	Android 2.2 (API Level 8)	Android 1.5 (API Level 3)
TYPE_ACCELEROMETER	Yes	Yes	Yes	Yes
TYPE_AMBIENT_TEMPERATURE	Yes	n/a	n/a	n/a
TYPE_GRAVITY	Yes	Yes	n/a	n/a
TYPE_GYROSCOPE	Yes	Yes	n/a <sup>1</sup>	n/a <sup>1</sup>
TYPE_LIGHT	Yes	Yes	Yes	Yes
TYPE_LINEAR_ACCELERATION	Yes	Yes	n/a	n/a
TYPE_MAGNETIC_FIELD	Yes	Yes	Yes	Yes
TYPE_ORIENTATION	Yes <sup>2</sup>	Yes <sup>2</sup>	Yes <sup>2</sup>	Yes
TYPE_PRESSURE	Yes	Yes	n/a <sup>1</sup>	n/a <sup>1</sup>
TYPE_PROXIMITY	Yes	Yes	Yes	Yes
TYPE_RELATIVE_HUMIDITY	Yes	n/a	n/a	n/a
TYPE_ROTATION_VECTOR	Yes	Yes	n/a	n/a
TYPE_TEMPERATURE	Yes <sup>2</sup>	Yes	Yes	Yes



# Programación: clases

Nombre	Tipo	Descripción
SensorManager	Clase	Se usa para crear una instancia del servicio de sensores. Proporciona varios métodos para el acceso a sensores, el registro y la eliminación de registros de las escuchas de eventos de sensores, etc.
Sensor	Clase	Se usa para crear la instancia de un sensor específico.
SensorEvent	Clase	El sistema lo usa para publicar datos del sensor. Incluye los valores de datos de sensores sin procesar, el tipo de sensor, la precisión de los datos y una marca de hora.
SensorEventListener	Interfaz	Proporciona métodos de llamada de regreso para recibir avisos del SensorManager cuando los datos o la precisión del sensor han cambiado.

# Programación

- Exigiendo en manifest un determinado sensor, filtraremos en Google Play quién puede ver y descargar nuestra app

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

# Programación

- Con el `SensorManager` de Android podemos conocer qué sensores tiene nuestro dispositivo, sus capacidades, conseguiremos datos y pondremos listeners

```
private SensorManager mSensorManager;  
...  
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
if (mSensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD) != null){  
    // Success! There's a magnetometer.  
}  
else {  
    // Failure! No magnetometer.  
}
```

# Programación

- Incluso podemos consultar el proveedor del sensor y la versión para hacer una cosa u otra

```
private SensorManager mSensorManager;
private Sensor mSensor;

...

mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);

if (mSensorManager.getDefaultSensor(Sensor.TYPE_GRAVITY) != null){
    List<Sensor> gravSensors = mSensorManager.getSensorList(Sensor.TYPE_GRAVITY);
    for(int i=0; i<gravSensors.size(); i++) {
        if ((gravSensors.get(i).getVendor().contains("Google Inc. ")) &&
            (gravSensors.get(i).getVersion() == 3)){
            // Use the version 3 gravity sensor.
            mSensor = gravSensors.get(i);
        }
    }
}
else{
    // Use the accelerometer.
    if (mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER) != null){
        mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
    }
    else{
        // Sorry, there are no accelerometers on your device.
        // You can't play this game.
    }
}
```

# Programación

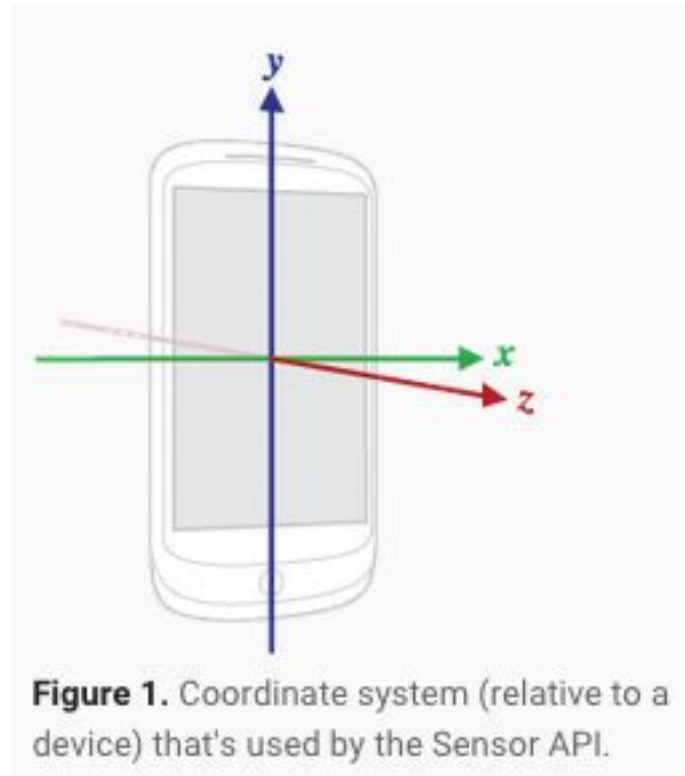
- También podemos comprobar el intervalo de tiempo mínimo (en microsegundos) que cualquier sensor necesita para adquirir datos: con el método `getMinDelay()`.
- Para la gestión de eventos tenemos que programar dos métodos callback de la interfaz `SensorEventListener`, que son `onAccuracyChanged()` y `onSensorChanged()`.
  - `onAccuracyChanged()` se invoca cuando hay un cambio en la precisión de un sensor
  - `onSensorChanged()` se invoca cuando cambia el valor que recoge el sensor

# Programación

- Como valores de DELAY tenemos los siguientes:
  - `SENSOR_DELAY_NORMAL` 200.000 microsegundos
  - `SENSOR_DELAY_GAME` 20.000 microsegundos
  - `SENSOR_DELAY_UI` 60.000 microsegundos
  - `SENSOR_DELAY_FASTEST` 0 microsegundos
  - Desde Android 3.0 (API Level 11) podemos especificar el valor absoluto de delay (en microsegundos).

# Sensores de movimiento

- Acelerómetro
- Sensor de gravedad
- Giroscopio
- Sensor de aceleración lineal
- Sensor de campos geomagnéticos



# Acelerómetro

Aceleración aplicada al dispositivo, incluyendo la fuerza de la gravedad

Si movemos el dispositivo hacia la derecha, la aceleración en x será positiva. Si lo alejamos de nosotros, la aceleración en y será positiva.

```
private SensorManager mSensorManager;  
private Sensor mSensor;  
...  
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);  
...  
public void onSensorChanged(SensorEvent event){  
}
```



# Sensor de gravedad

Devuelve un vector 3d con la información de dirección y magnitud de la gravedad.

```
private SensorManager mSensorManager;  
private Sensor mSensor;  
...  
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_GRAVITY);
```

# Giroscopio

Devuelve el valor de rotación respecto a un eje. La rotación es positiva en el sentido de las agujas del reloj.

```
private SensorManager mSensorManager;  
private Sensor mSensor;  
...  
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_GYROSCOPE);
```

# Ej: sensor que cuenta los pasos dados

```
private SensorManager mSensorManager;  
private Sensor mSensor;  
...  
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_STEP_COUNTER);
```

# Sensores de posición

- Game Rotation Vector Sensor
- Geomagnetic Rotation Vector Sensor
- Sensor de orientación
- Sensor del campo geomagnético
- Sensor de proximidad

# Sensor de proximidad

Determina cómo de lejos estamos respecto de un objeto. Normalmente se utiliza para determinar la distancia entre la cara de una persona y el micrófono del dispositivo. Puede devolver la distancia en valor absoluto o los valores near y far.

```
private SensorManager mSensorManager;  
private Sensor mSensor;  
...  
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
mSensor = mSensorManager.getDefaultSensor(Sensor.TYPE_PROXIMITY);
```

# Sensores de ambiente o entorno

Android cuenta con 4 sensores que permiten monitorizar propiedades del entorno: humedad, iluminación, presión y temperatura.

Sensor	Sensor event data	Units of measure	Data description
<code>TYPE_AMBIENT_TEMPERATURE</code>	<code>event.values[0]</code>	°C	Ambient air temperature.
<code>TYPE_LIGHT</code>	<code>event.values[0]</code>	lx	Illuminance.
<code>TYPE_PRESSURE</code>	<code>event.values[0]</code>	hPa or mbar	Ambient air pressure.
<code>TYPE_RELATIVE_HUMIDITY</code>	<code>event.values[0]</code>	%	Ambient relative humidity.
<code>TYPE_TEMPERATURE</code>	<code>event.values[0]</code>	°C	Device temperature. <sup>1</sup>

# Sensores de ambiente o entorno

Estos sensores no requieren calibración, y sólo hay que crear una instancia de `SensorManager`, registrar un listener con `onResume()`, y tomar los datos con `onSensorChanged()`.

# Sensors

MTV: Practicar el uso de gestión de sensores

- Desarrolla una aplicación que muestre los sensores disponibles en el dispositivo, y tras la selección de uno de ellos muestra los valores medidos.
- Sube a la plataforma un documento PDF en el que se muestre una captura de pantalla con el sensor seleccionado y los datos obtenidos.