



Desenvolvimento de Software para Dispositivos Móveis

Aula 14 - Sensores



Professor: Anderson Almada

Introdução

The screenshot shows a web browser with multiple tabs. The active tab is 'Sensors Overview | Android'. The address bar shows the URL 'developer.android.com/guide/topics/sensors/sensors_overview'. The page header includes the 'developers' logo and navigation links: 'Plataforma', 'Android Studio', 'Google Play', 'Documentos' (highlighted), and 'Mais'. There is a search bar with the text 'Pesquisa' and a 'LANGUAGE' dropdown menu. A 'FAZER LOGIN' link is also present.

The main content area is titled 'Documentação' and has sub-tabs: 'VISÃO GERAL', 'GUIAS' (highlighted), 'REFERÊNCIA', 'AMOSTRAS', and 'DESIGN E QUALIDADE'. On the left, a sidebar lists various topics, with 'Sensores' expanded to show sub-topics like 'Visão geral', 'Visão geral dos sensores', 'Sensores de movimento', 'Sensores de posição', 'Sensores de ambiente', 'Medições GNSS brutas', 'Conectividade', 'RenderScript', 'Conteúdo na Web', and 'Android App Bundles'.

The main content area features a light blue banner for the 'Android Dev Summit, October 23-24: two days of technical content, directly from the Android team. [Sign-up for livestream updates.](#)'. Below this, the breadcrumb trail is 'Android Developers > Documentos > Guias'. The title 'Sensors Overview' is displayed with a five-star rating. The text explains that most Android-powered devices have built-in sensors for motion, orientation, and environmental conditions, providing raw data with high precision and accuracy. It gives examples of how these sensors are used in games, weather applications, and travel apps. At the bottom, it says 'Refer to the following related resources:' followed by a link to 'Sensors'.

On the right side, there is an 'Índice' (Index) section with a list of links: 'Introduction to Sensors', 'Sensor Framework', 'Sensor Availability', 'Identifying Sensors and Sensor Capabilities', 'Monitoring Sensor Events', 'Handling Different Sensor Configurations', 'Sensor Coordinate System', and 'Best Practices for Accessing and Using Sensors'.

Introdução

aulas - Google Drive x Aula 14 - Android - Sensores x Sensors Overview | Android x +

← → ↺ developer.android.com/guide/topics/sensors/sensors_overview ⓘ ☆ 🔴 🟢 🟡 f? 🛠️ 📄 👤 ⋮

developers Plataforma Android Studio Google Play Documentos Mais 🔍 Pesquisa LANGUAGE FAZER LOGIN

- Áudio e vídeo
- Tarefas em segundo plano
- Dados e arquivos do app
- Dados e identidade do usuário
- Localização do usuário
- Toque e entrada
- CameraX
- Câmera
- Sensores
 - Visão geral
 - Visão geral dos sensores**
 - Sensores de movimento
 - Sensores de posição
 - Sensores de ambiente
 - Medições GNSS brutas
- Conectividade
- RenderScript
- Conteúdo na Web
- Android App Bundles
- Google Play Instant
- Ações no app
- Slices
- Jogos
- Desenvolvimento de jogos para Android

Sensor	Type	Description	Common Uses
TYPE_ACCELEROMETER	Hardware	Measures the acceleration force in 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.).
TYPE_AMBIENT_TEMPERATURE	Hardware	Measures the ambient room temperature in degrees Celsius ($^{\circ}C$). See note below.	Monitoring air temperatures.
TYPE_GRAVITY	Software or Hardware	Measures the force of gravity in m/s^2 that is applied to a device on all three physical axes (x, y, z).	Motion detection (shake, tilt, etc.).
TYPE_GYROSCOPE	Hardware	Measures a device's rate of rotation in rad/s around each of the three physical axes (x, y, and z).	Rotation detection (spin, turn, etc.).
TYPE_LIGHT	Hardware	Measures the ambient light level (illumination) in lx.	Controlling screen brightness.
TYPE_LINEAR_ACCELERATION	Software or Hardware	Measures the acceleration force in 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.
TYPE_MAGNETIC_FIELD	Hardware	Measures the ambient geomagnetic field for all three physical axes (x, y, z) in μT .	Creating a compass.
TYPE_ORIENTATION	Software	Measures degrees of rotation that a device makes around all three physical axes (x, y, z). As of API level 3 you can	Determining device position

Índice

- [Introduction to Sensors](#)
- [Sensor Framework](#)
- [Sensor Availability](#)
- [Identifying Sensors and Sensor Capabilities](#)
- [Monitoring Sensor Events](#)
- [Handling Different Sensor Configurations](#)
- [Sensor Coordinate System](#)
- [Best Practices for Accessing and Using Sensors](#)

Introdução

aulas - Google Drive x Aula 14 - Android - Sensores - x Sensors Overview | Android | x +

developer.android.com/guide/topics/sensors/sensors_overview

developers Plataforma Android Studio Google Play Documentos Mais

Pesquisa LANGUAGE FAZER LOGIN

VISÃO GERAL **GUIAS** REFERÊNCIA AMOSTRAS DESIGN E QUALIDADE

- Áudio e vídeo
- Tarefas em segundo plano
- Dados e arquivos do app
- Dados e identidade do usuário
- Localização do usuário
- Toque e entrada
- CameraX
- Câmera
- Sensores
 - Visão geral
 - Visão geral dos sensores**
 - Sensores de movimento
 - Sensores de posição
 - Sensores de ambiente
 - Medições GNSS brutas
- Conectividade
- RenderScript
- Conteúdo na Web
- Android App Bundles
- Google Play Instant
- Ações no app
- Slices

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 ¹	n/a ¹
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 ²	Yes ²	Yes ²	Yes
TYPE_PRESSURE	Yes	Yes	n/a ¹	n/a ¹
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 ²	Yes	Yes	Yes

Índice

- Introduction to Sensors
 - Sensor Framework
 - Sensor Availability**
- Identifying Sensors and Sensor Capabilities
- Monitoring Sensor Events
- Handling Different Sensor Configurations
- Sensor Coordinate System
- Best Practices for Accessing and Using Sensors

Vamos para prática

- Vamos utilizar um gerenciador de sensores
 - `SensorManager`
- Dentro do `onCreate`

```
sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
```

Vamos para prática

- Para listar todos os sensores do dispositivo, basta chamar o método `getSensorList`.

```
List<Sensor> listSensors = sensorManager.getSensorList(Sensor.TYPE_ALL);
```

- Usando um `foreach`, liste todos:

```
for (Sensor s : listSensors) {  
    Log.i("Main", s.getStringType());  
}
```

Vamos para prática

- Utilizando o sensor de luminosidade

```
sensorLight = sensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);
```

- Para iniciar a coleta de dados, implemente a interface de escuta do sensor
 - SensorEventListener

Vamos para prática

- Depois de adicionar os métodos, registre o escutador no onResume e desregistre o escutador no onPause

onResume ()

```
sensorManager.registerListener (this, sensorLight, SensorManager.SENSOR_DELAY_NORMAL) ;
```

onPause ()

```
sensorManager.unregisterListener (this) ;
```


Vamos para prática

- O método `onSensorChanged` sempre é chamado quando os dados são capturados pelos sensores. Logo, ele é o método a ser explorado

```
@Override  
  
public void onSensorChanged(SensorEvent event) {  
    Log.i("Main", ""+event.values[0]);  
}
```

Exercício

- Sempre que a luminosidade for menor que 5, ligue o flash do dispositivo, caso contrário deixe desligado.

Localização

- Realize a importação do dexter
- Build.gradle
 - `implementation 'com.karumi:dexter:6.0.0'`
- Clique em sync

Localização

- Adicione as permissões no manifest

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

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

Localização

- No onCreate

```
Dexter.withActivity(this)

    .withPermissions(

        Manifest.permission.ACCESS_COARSE_LOCATION,

        Manifest.permission.ACCESS_FINE_LOCATION

    ).withListener(new MultiplePermissionsListener() {

@Override public void onPermissionsChecked(MultiplePermissionsReport report) {

}

@Override public void onPermissionRationaleShouldBeShown(List<PermissionRequest> permissions,

PermissionToken token) { /* ... */}

}).check();
```

Localização

- Crie um método para recuperar a última localização

```
location = LocationServices.getFusedLocationProviderClient(this).getLastLocation();  
if (location != null) {  
    location.addOnSuccessListener(new OnSuccessListener<Location>() {  
        @Override  
        public void onSuccess(Location location) {  
            Log.i("Main", location.getLatitude() + " " + location.getLongitude());  
        }  
    });  
}
```

Localização

- Crie um método para ser notificado quando a localização mudar. Para isso, configure como deseja ser notificado

```
private void startLocationUpdates() {  
    // config  
    locationRequest = new LocationRequest();  
    locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);  
    locationRequest.setInterval(UPDATE_INTERVAL);  
    locationRequest.setFastestInterval(FATEST_INTERVAL);  
    // update
```

Localização

```
// update
LocationServices.getFusedLocationProviderClient(this).requestLocationUpdates
(locationRequest, new LocationCallback() {
    @Override
    public void onLocationResult(LocationResult locationResult) {
        if (locationResult == null) { return; }
        for (Location location : locationResult.getLocations()) {
            if (location != null) { }
        }
    }
}, Looper.myLooper());
} // fim do método startLocationUpdates
```


API Maps

- <https://www.google.com/maps/search/google+maps/@-3.8063156,-38.6036227,13z/data=!3m1!4b1>
- <https://nominatim.openstreetmap.org/reverse?&format=json&lat=-3.806317&lon=-38.6036227>

Link importante

- https://developer.android.com/guide/topics/sensors/sensors_overview



Dúvidas??

E-mail: almada@crateus.ufc.br