

Sensores

- Computação Sensível ao Contexto (*Context Aware*)
- Uso de Sensores

Referências

- DEITEL, Harvey, DEITEL, Paul, DEITEL, Abbey, MORGANO, Michael. Android para programadores: uma abordagem baseada em aplicativos. Porto Alegre: Bookman. 2013.
- Programming Handheld Systems – Prof. Adam Porter of University of Maryland (<https://www.coursera.org/course/android>)
- <http://developer.android.com>
- <https://developers.google.com/university/courses/mobile>
- <http://oreilly.com/training/androidapps>

Sensores

- Dispositivos (hardwares) que medem alguma variável de ambiente:
 - Movimento
 - Localização (latitude e longitude)
 - Ambiente

Alguns Exemplos de Sensores

- Movimento: Acelerômetro 3D
- Localização: Campo Magnético 3D
- Ambiente: Pressão através de um Barômetro

SensorManager

- Serviço do Sistema que gerencia os sensores
- Para obter as instâncias dos sensores

```
getSystemService(  
    Context.SENSOR_SERVICE)
```

- Para acessar um sensor específico

```
SensorManager.  
    getDefaultSensor(int type)
```

Alguns Tipos de Constantes de Sensores

- **Acelerômetro:**

`Sensor.TYPE_ACCELEROMETER`

- **Campo Magnético:**

`Sensor.TYPE_MAGNETIC_FIELD`

- **Pressão:**

`Sensor.TYPE_PRESSURE`

Veja mais em:

http://developer.android.com/guide/topics/sensors/sensors_overview.html

SensorEventListener

- Interface para tratamiento de retorno de eventos relacionados a sensores

SensorEventListener

- É chamado quando a precisão de um sensor é alterada

```
void onAccuracyChanged(  
    Sensor sensor, int accuracy)
```

- Ou quando algum valor do sensor é modificado

```
void onSensorChanged(  
    SensorEvent event)
```


Registro de Eventos de Sensores

- Utiliza o `SensorManagement` para registrar e liberar eventos de sensores (`SensorEvent`)

Registro de Eventos de Sensores

- Para registrar um `SensorEventListener` para um dado sensor

```
public boolean registerListener(  
    SensorEventListener listener,  
    Sensor sensor, int rate)
```

Liberação de Registro de Eventos de Sensores

- Para liberar um `Listener` de sensores que estão registrados

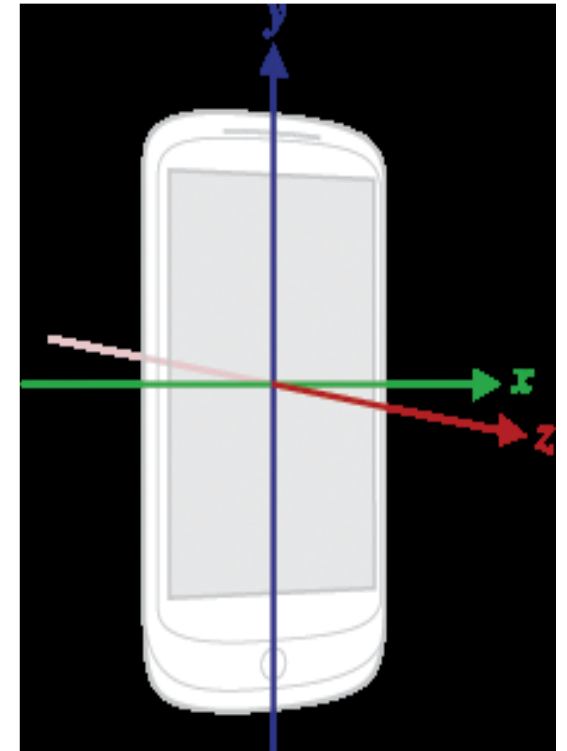
```
public void unregisterListener(  
    SensorEventListener listener,  
    Sensor sensor)
```

SensorEvent

- Dados específicos de cada sensor
 - Sensor Type
 - Time-Stamp
 - Accuracy
 - Measurement Data

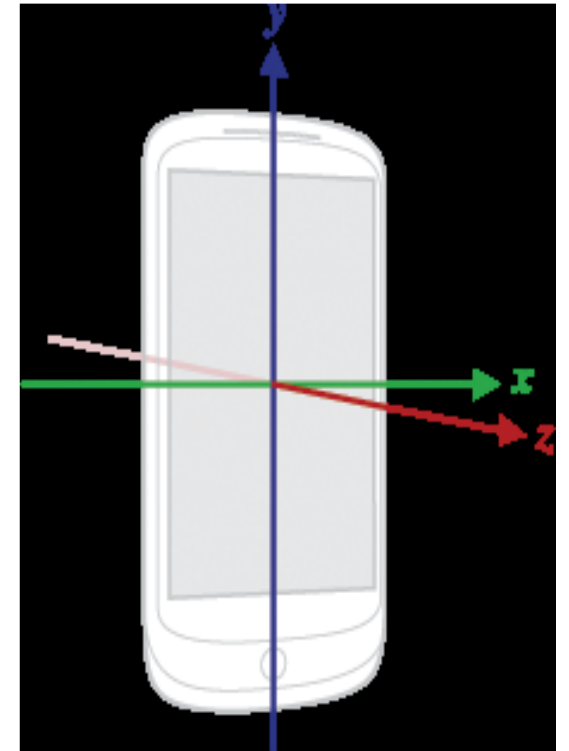
Sistema de Coordenadas de Sensores

- Quando a orientação padrão é retrato e o dispositivo está deitado, virado para cima sobre uma mesa, os eixos são:
 - X – direita para esquerda
 - Y – do topo para a base
 - Z – de baixo para cima



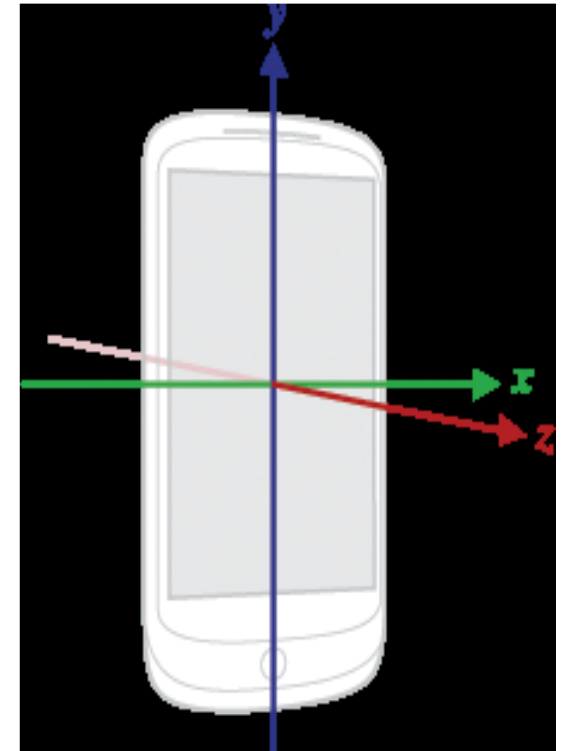
Sistema de Coordenadas de Sensores

- O sistema de coordenadas não muda quando o dispositivo muda de orientação



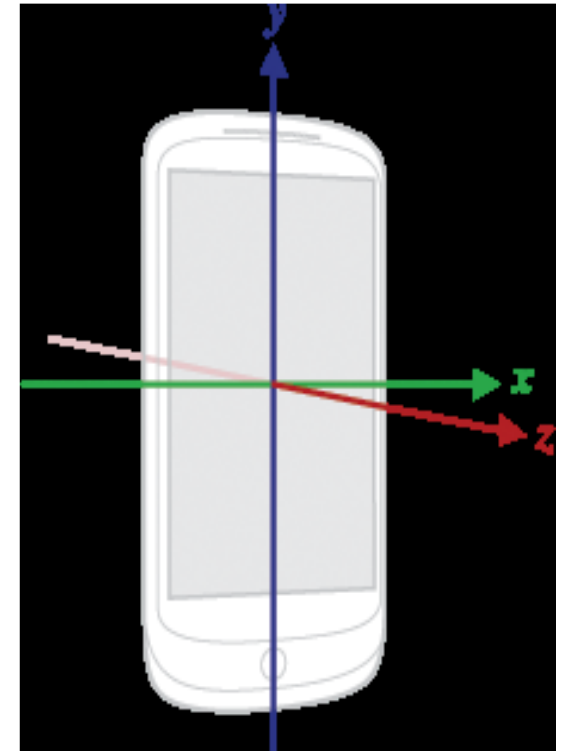
Valores do Acelerômetro

- Se o dispositivo estiver de pé, o acelerômetro deveria informar:
 - $X \approx 0\text{m/s}^2$
 - $Y \approx 9,8\text{m/s}^2$
 - $Z \approx 0\text{m/s}^2$



Valores do Acelerômetro

- Mas esses valores podem variar devido a movimentos naturais, superfícies não planas, ruído, etc.



Exercício de aula (em dupla)

1. Fazer uma app que use acelerômetro identificar se o usuário que está com o dispositivo caiu

Obs.:

Tente usar o `SensorSimulator` do `code.google.com`

Exercício de aula (em dupla)

1. Fazer um app que capture um e dois movimentos rápidos de cima para baixo, obtenha a latitude e longitude do usuários e que verifique se é dia (“*Bom dia!*”) ou se é noite (“*Boa noite!*”)
2. O código da app deve estar bem documentado
3. A app deve usar as melhores práticas de interface com o usuário

TP2 (Parte 3/3) – Desafio!

1. Acrescentar a funcionalidade de digitalizar um documento e salva-lo como um PDF
2. O código da app deve estar bem documentado
3. A app deve usar as melhores práticas de interface com o usuário

Esta parte não é obrigatória. Valerá 5 pontos extras se estiver funcionando!