 <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p>	<p>Computação Móvel e Ubíqua</p> <p>1º Semestre</p> <p>Docente: fas</p> <p>Ficha Prática: Mapas e Sensores</p>
---	---

Tema: Mapas e Sensores

Objetivos:

- Utilização dos sensores de um dispositivo Android
- Utilização da localização geográfica e representação de informação em mapas

PARTE 1 - Sensores

Exercício 1

Utilizando o IDE Android Studio crie um novo projeto com as seguintes características:

<i>Atributo</i>	<i>Valor</i>
<i>Application Name</i>	<i>SensorList</i>
Project Location	Leave the default value
Form factor	Phone and tablet only
Minimum SDK	API 23 Marshmallow
Type of Activity	Empty
Activity Name	MainActivity (default)
Layout Name	Activity_main (default)

Neste exercício pretende-se o desenvolvimento de uma aplicação liste todos os sensores existentes no dispositivo Android num fragmento dentro de uma atividade em Android.

Deve implementar um projeto que faça uso dos seguintes componentes numa atividade:

- *ScrollView* para inserir a lista de sensores. Esta *View* permite a ver a lista completa caso a mesma necessite de mais espaço que o ecrã
- *TextView* dentro da *ScrollView* com o texto “lista de sensores”

Na atividade principal deverá apresentar a lista com todos os sensores como na Figura 1.

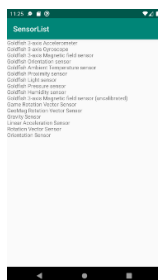



Figura 1 - Exemplo de atividade principal com todos os sensores existentes

<div data-bbox="146 159 403 248">  </div> <div data-bbox="496 181 625 284"> <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p> </div>	<div data-bbox="858 181 1225 215"> <p>Computação Móvel e Ubíqua</p> </div> <div data-bbox="978 230 1110 255"> <p>1º Semestre</p> </div> <div data-bbox="975 271 1112 295"> <p>Docente: fas</p> </div> <div data-bbox="865 300 1220 327"> <p>Ficha Prática: Mapas e Sensores</p> </div>
---	---

Exercício 2

Utilizando o exercício 1 desenvolvido anteriormente, acrescente um novo fragmento com a capacidade de demonstrar o valor real dos sensores do dispositivo Android. Na figura 2, um exemplo para valores do sensor de luminosidade e proximidade. Considere também a representação de sensores adicionais como acelerómetro e giroscópio. Verifique que a medida que o emulador ou dispositivo Android é rodado os valores dos sensores são atualizados. Caso esteja a testar num dispositivo real, considere também sensores de deteção de atividade e deteção de passos.

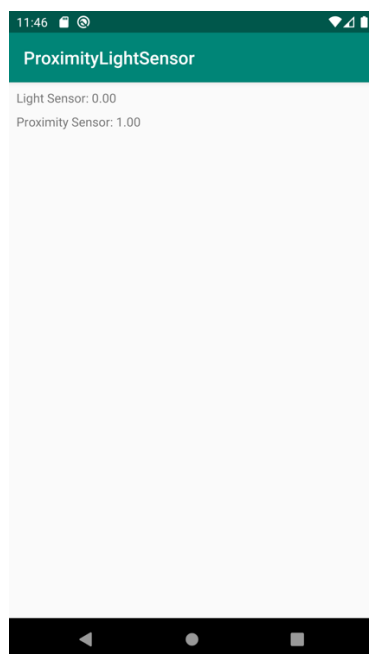


Figura 2 - Exemplo de valores captados para os sensores de luminosidade e proximidade

Adicionalmente, considere o uso da biblioteca MPAndroidChart – (<https://weeklycoding.com/mpandroidchart/>) demonstrada na ficha prática 1 e represente o valor de cada sensor usando um gráfico apropriado.

PARTE 1 – Localização e Mapas

Exercício 1


Utilizando o IDE Android Studio crie um novo projeto com as seguintes características:

<i>Atributo</i>	<i>Valor</i>
<i>Application Name</i>	<i>Location</i>
Project Location	Leave the default value
Form factor	Phone and tablet only
Minimum SDK	API 23 Marshmallow
Type of Activity	Empty
Activity Name	MainActivity (default)
Layout Name	Activity_main (default)

Neste exercício pretende-se o desenvolvimento de uma aplicação que utilize os serviços de localização. Deve implementar um projeto que faça uso dos seguintes componentes:

- Uma *MainActivity* com um fragmento (idêntico à Figura 1) que possua
 - Um botão *Obter última localização* que invoque o método *getLastLocation()* da biblioteca de localização dos serviços *Google Play*
 - Uma *TextView* para apresentar ao utilizador a localização no formato latitude, longitude e precisão
 - Um botão *Começar localizações periódicas* que invoque o método *requestLocationUpdates()* da biblioteca de serviços do *Google Play*. As localizações obtidas deverão ser apresentadas ao utilizador através da *TextView* do ponto anterior
 - Um botão *Parar localizações periódicas* que invoque o método *removeLocationUpdates()* da biblioteca de serviços do *Google Play*

Note que a utilização dos serviços de localização necessita de permissões específicas. É necessário garantir que as permissões não foram desativadas pelo utilizador, devendo requerer a ativação dos serviços (Figura 2).

<div data-bbox="146 159 403 248">  </div> <div data-bbox="496 181 625 286"> <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p> </div>	<div data-bbox="858 181 1225 215"> <p>Computação Móvel e Ubíqua</p> </div> <div data-bbox="978 230 1109 257"> <p>1º Semestre</p> </div> <div data-bbox="975 273 1112 300"> <p>Docente: fas</p> </div> <div data-bbox="865 300 1220 327"> <p>Ficha Prática: Mapas e Sensores</p> </div>
---	--

Com base na localização obtida a aplicação deve usar a classe *GeoCoder* (disponível no Android) para obter o endereço no formato rua, cidade e país. A utilização desta classe é um ponto extra na resolução do exercício.



Figura 3 – MainActivity da aplicação de localização

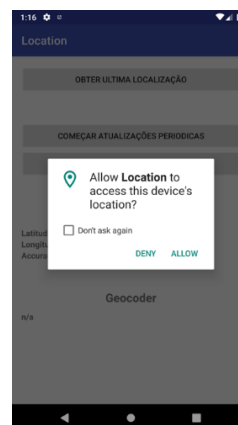


Figura 4 – Dialog para gestão de permissões

Exercício 2

Utilizando o exercício 1 desenvolvido anteriormente, pretende-se que adicione um novo fragmento que permita a visualização da localização actual num mapa idêntico à figura 5. Lembre-se de registar o projeto na Google APIs e ativar a API do google maps.

A localização deve ser demonstrada através de um marcador (*Marker*) posicionado nas coordenadas detetadas pelo dispositivo e atualizado sempre que a localização se alterar.

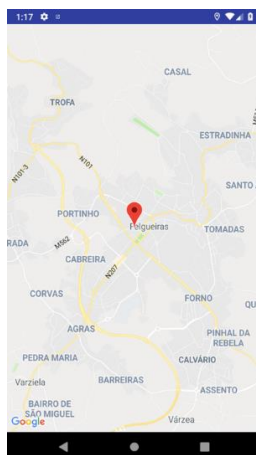


Figura 5 – MainActivity com apresentação do mapa

Exercício 3

Utilizando o IDE Android Studio crie um novo projeto com as seguintes características:

<i>Atributo</i>	<i>Valor</i>
<i>Application Name</i>	<i>Retrofit</i>
Project Location	Leave the default value
Form factor	Phone and tablet only
Minimum SDK	API 23 Marshmallow
Type of Activity	Empty
Activity Name	MainActivity (default)
Layout Name	Activity_main (default)

Neste exercício pretende-se o desenvolvimento de uma aplicação que utilize a *API TourPedia* disponível em <http://tour-pedia.org/api>.

Deve implementar um projeto que faça uso dos seguintes componentes:

- Uma *MainActivity* (idêntica à Figura 4) que possua
 - Um *SupportMapFragment*
 - Uma *Toolbar*
- Uma *Interface* para declaração da *API* definida por *TourDataAPI*
- Uma classe que replique o modelo de dados que é devolvido pela *API*
- Uma classe que proceda à criação dos objetos *Retrofit* e *TourPediaAPI*

De forma a demonstrar a realização de pedidos à *API* e obtenção de resultados, implemente no método *onResume()* da *MainActivity* um pedido pelos pontos de interesse numa cidade baseada numa localização (ex: Londres). Após receção da lista de pontos de interesse, coloque-os no mapa com base na localização de cada ponto recebido.

<div data-bbox="146 185 403 248"> <p>P.PORTO</p> </div> <div data-bbox="501 185 625 286"> <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p> </div>	<div data-bbox="860 185 1225 331"> <p>Computação Móvel e Ubíqua</p> <p>1º Semestre</p> <p>Docente: fas</p> <p>Ficha Prática: Mapas e Sensores</p> </div>
--	--

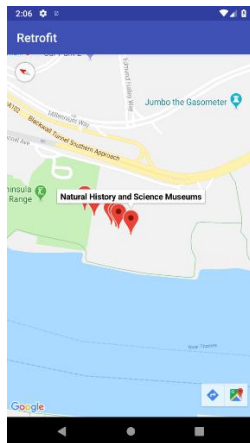



Figura 6 - MainActivity após inserção dos pontos recebidos no mapa

 <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p>	<p>Computação Móvel e Ubíqua</p> <p>1º Semestre</p> <p>Docente: fas</p> <p>Ficha Prática: Mapas e Sensores</p>
---	---

PARTE 3

Questão 1

Qual das seguintes características são fornecidas pela classe *SensorManager*? (selecione todas as que se apliquem)

- Métodos para registar e remover *sensor listeners*.
- Métodos para detetar a orientação do dispositivo.
- Constantes que representam os tipos de sensores.
- Constantes que representam a precisão do sensor.
- Métodos para indicar se o sensor é um *wake-up sensor*.

Questão 2


Em que momento do ciclo de vida da atividade se devem registar os *sensor listeners*?

- `onResume()`
- `onCreate()`
- `onStart()`
- `onRestart()`

Questão 3

Quais a melhores práticas para utilização de sensores na sua aplicação? (Selecione todas as que se aplicam)

- Registar *listeners* apenas para os sensores que são necessários.
- Verificar se o sensor está disponível no dispositivo antes da sua utilização.
- Verificar as permissões para utilização do sensor.
- Registar o *sensor listener* para a menor taxa de atualização.
- Não bloquear `onSensorChanged()` para filtrar ou transformar os dados recebidos.

 <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p>	<p>Computação Móvel e Ubíqua</p> <p>1º Semestre</p> <p>Docente: fas</p> <p>Ficha Prática: Mapas e Sensores</p>
---	---

Questão 4

Qual a API utilizada para obter a última localização conhecida do dispositivo?

- Método *getLastKnownLocation()* existente na classe *FusedLocationProviderApi*
- Método *getLastKnownLocation()* existente na classe *LocationServices*
- Método *getLastLocation()* existente na classe *FusedLocationProviderApi*
- Método *getLastLocation()* existente na classe *LocationServices*

Questão 5

Que método utilizaria para realizar *updates* periódicos à localização do dispositivo?

- Método *requestPeriodicUpdates()* presente na classe *FusedLocationClient*
- Método *requestLocationUpdates()* presente na classe *FusedLocationProviderClient*
- Método *requestUpdates()* presente na classe *FusedLocationProviderClient*
- Método *requestLocationUpdates()* presente na classe *FusedLocationProvider*

Questão 6


Que método é chamado quando o mapa é carregado e se encontra preparado para ser utilizado pela aplicação?

- *onMapReady(GoogleMap googleMap)*
- *onMapLoaded(GoogleMap googleMap)*
- *onMapCreate(GoogleMap googleMap)*
- *onMapInitialize(GoogleMap googleMap)*

Questão 7

Que componentes do Android utiliza para incluir os mapas do Google na sua aplicação?

- *MapView MapFragment*
- *MapFragment e MapActivity*
- *MapView MapActivity*
- Apenas o *MapFragment*

 <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p>	<p>Computação Móvel e Ubíqua</p> <p>1º Semestre</p> <p>Docente: fas</p> <p>Ficha Prática: Mapas e Sensores</p>
---	---

Questão 8

Que tipos de mapas a *API Android* dos mapas da *Google* disponibiliza?

- Normal, híbrido, terreno, satélite e estradas
- Normal, híbrido, terreno, satélite e 'nada'
- Híbrido, terreno, satélite, estradas e 'nada'
- Normal, terreno, satélite, mapa de imagens e 'nada'