 <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: Services</p>
---	---

Tema: *Services e Media*

Objetivos:

- Utilização de *Services* para processos em *background*
- Criação de *services*
- Reprodução de ficheiros áudio

PARTE 1 - Services

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>	AlarmApplication
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 permita a implementação de *Android Services* de acordo com a informação disponibilizada nos *slides* teóricos.

Deve implementar um projeto que use uma actividade e um foreground service que permita :

- Ter uma atividade que permita acionar o modo alarme ou desativá-lo;
- Quando activo o modo de alarme, deve existir um serviço *START_STICKY*, que monitorize os valores do acelerómetro e sempre que a norma do vector for superior a 11 ou inferior a 7 o serviço deve activar um alarme sonoro. Na tabela 1 encontra-se o código para iniciar a reprodução de um ficheiro áudio em loop em android.

$$normaVector = \sqrt{x^2 + y^2 + z^2}$$

	ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO	Computação Móvel e Ubíqua 1º Semestre Docente: fas Ficha Prática: Services
---	---	--

- O alarmeService deve funcionar mesmo com a activity desligada depois de ativo.

O objetivo da aplicação passa por permitir pousar o dispositivo com o alarm ativo e sempre que alguém insuspeito o tentar utilizar este reproduza um som alertando o utilizador.

Tabela 1- Excerto para reprodução de áudio em Android

```
myPlayer = MediaPlayer.create(this, R.raw.sound_file);
myPlayer.setLooping(true);
myPlayer.start();
```

Exercício 2

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

<i>Atributo</i>	<i>Valor</i>
<i>Application Name</i>	StepLogger
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)

Utilizando novamente um foreground service deve criar uma aplicação que use um android service para efetuar o log de passos que um utilizador dá durante o dia.

Deve implementar um projeto que use uma actividade e um foreground service que permita :

- Deteção de passos a partir de um android service;
- Guardar os passos reconhecidos pelo sensor numa base de dados do tipo Room;
- Permitir a consulta do histórico de passos numa activity com uma recycler view. A recycler view deve fornecer em cada entrada os passos dados e a data e hora em que foram detetados.

Nota: O sensor de passos através do android apenas pode ser testado em dispositivos reais. Caso esteja apenas a usar o emulador, não será possível testar a aplicação. Neste caso use a aplicação para guardar as localizações do utilizador

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>SoundService</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 cujo objetivo é a reprodução de um ficheiro de áudio. Este ficheiro poderá ser obtido através de um *link* num repositório público ou poderá utilizar um *asset* do tipo *raw* em *Android*.

Deve implementar um *player* que implemente as seguintes funcionalidades:

- Iniciar a reprodução da música após receção do comando *onStartComand* no serviço *Android*
- Criação de um botão que permita terminar a execução do *media player*

A utilização de um ficheiro de áudio implica a criação de uma pasta *raw* dentro da pasta *res* no *Android Studio*. O ficheiro de áudio deve ser colocado dentro da pasta referida (Figura 2). Considere o código fornecido na Tabela 2 no desenvolvimento da aplicação.

 <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: Services</p>
---	---

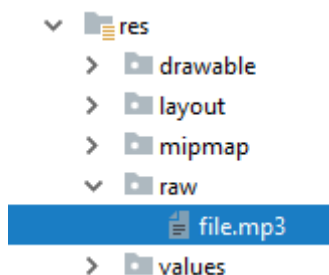


Figura 1 - Inserção de ficheiro de áudio no Android Studio

Tabela 2 - Exemplo de utilização do Media Player em Android

```
public class MyService extends Service {
    MediaPlayer myPlayer;
    @Nullable
    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }
    @Override
    public void onCreate() {
        Toast.makeText(this, "Service Created", Toast.LENGTH_LONG).show();

        myPlayer = MediaPlayer.create(this, R.raw.sun);
        myPlayer.setLooping(false); // Set looping
    }
    @Override
    public void onStart(Intent intent, int startid) {
        Toast.makeText(this, "Service Started", Toast.LENGTH_LONG).show();
        myPlayer.start();
    }
    @Override
    public void onDestroy() {
        Toast.makeText(this, "Service Stopped", Toast.LENGTH_LONG).show();
        myPlayer.stop();
    }
}
```

Adicione o serviço musical a aplicação que implementa o jogo de perguntas e respostas da ficha 2 exercício 3 ou na ficha 4 exercício 3. Neste contexto a música deve tocar apenas durante o período de resposta à questão.

Exercício 4

Primeira Parte

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

<i>Atributo</i>	<i>Valor</i>
<i>Application Name</i>	<i>WordService</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)

Deve implementar um projeto que faça uso de uma atividade e de um serviço de modo a obter a definição (significado) de palavras introduzidas na atividade principal. A definição da palavra fará uso do *webservice* disponibilizado pelo Oxford Dictionaries¹. **A utilização deste *webservice* implica o registo e obtenção de uma chave (API key).**

A consulta do significado da palavra introduzida pelo utilizador deverá ser realizada através de um serviço *Android* que:

- Execute as chamadas ao *webservice*
- Notifique a atividade principal assim que o resultado for obtido. Recorra a *intents* para realizar este processo
- A definição deverá ser apresentada no ecrã da atividade principal como demonstrado na Figura 1

De forma a facilitar o processo de obtenção de definições e diminuir o número de pedidos repetidos deve implementar ainda as seguintes funcionalidades:

¹ <https://developer.oxforddictionaries.com/>

<div data-bbox="146 159 403 248">P.PORTO</div> <div data-bbox="496 181 625 284">ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</div>	<div data-bbox="858 181 1225 215">Computação Móvel e Ubíqua</div> <div data-bbox="976 230 1109 255">1º Semestre</div> <div data-bbox="973 273 1166 324">Docente: fas Ficha Prática: Services</div>
--	--

- Utilização da biblioteca *JsonSimple* ou *Gson* para processar o resultado do *webservice*
- Armazenamento das palavras consultadas numa base de dados (ex. *Room*) ou em memória para evitar replicação de pedidos.

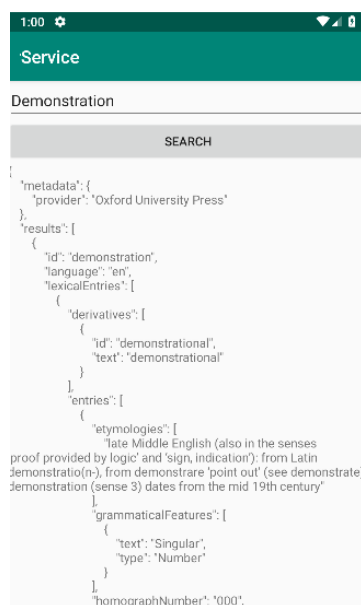


Figura 2 - Exemplo da atividade principal


A Tabela 1 demonstra um exemplo de invocação do *webservice*.

Tabela 3 - Exemplo de invocação do *webservice* do Oxford Dictionaries

```
final String language = "en";
final String word = "Demonstration";
final String word_id = word.toLowerCase();
//word_id is case sensitive and lowercase is required
final String params "https://od-api.oxforddictionaries.com:443/api/v1/entries/" +
language + "/" + word_id;

protected String invokeWebService(String params) {

    //TODO: replace with your own app id and app key
    final String app_id = "YOUR_APP_ID";
    final String app_key = "YOUR_APP_KEY";
    try {
        URL url = new URL(params);
```

 <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: Services</p>
---	---

```

        HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();
        urlConnection.setRequestProperty("Accept", "application/json");
        urlConnection.setRequestProperty("app_id", app_id);
        urlConnection.setRequestProperty("app_key", app_key);

        // read the output from the server
        BufferedReader reader = new BufferedReader(new
InputStreamReader(urlConnection.getInputStream()));
        StringBuilder stringBuilder = new StringBuilder();

        String line = null;
        while ((line = reader.readLine()) != null) {
            stringBuilder.append(line + "\n");
        }

        return stringBuilder.toString();


    }
    catch (Exception e) {
        e.printStackTrace();
        return e.toString();
    }
}

```

Segunda Parte

Reescreva a aplicação recorrendo não a services mas à biblioteca retrofit lecionada anteriormente. Neste nova versão é necessário o recurso a services?

Não

 <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: Services</p>
---	---

PARTE 2

Questão 1

Responda de entre as afirmações disponíveis as que se adequam a Started Services?

- Necessitam de estar sempre associados a uma activity;
- Não necessitam de estar associados a uma activity;
- A resposta ao método *onBind* é um valor nulo;
- ☒ Podem ser do tipo background ou foreground.

Questão 2

Bounded Services referem-se a serviços que:

- ☒ Permite obter acesso direto ao service e invocar os seus métodos;
- Podem executar tarefas mesmo quando não associados a nenhuma activity
- Não são recomendados no desenvolvimento de aplicações Android
-