

## Trabalhando com serviços em background

Services é a maneira de rodar tarefas em background de uma maneira mais alto nível que programaticamente. O exemplo mais famoso é um tocador de MP3, onde você quer ficar ouvindo a música enquanto navega na internet ou usa o GPS.

### Recebendo SMS

Para receber um SMS, vamos declarar um receptor de broadcast no AndroidManifest.xml, e então faremos o código. Nossa intenção é fazer um código que filtre quando um aluno enviar uma mensagem consequentemente reclamando da nota.

Coloque este código dentro da tag `application`

```
<receiver android:name=".SMSReceiver" android:enabled="true">
  <intent-filter>
    <action android:name="android.provider.Telephony.SMS_RECEIVED" />
  </intent-filter>
</receiver>
```

Vamos agora criar uma classe a `SMSReceiver`, ela estenderá de `BroadcastReceiver`. Esta classe funciona do mesmo modo dos *listeners* que temos trabalhado até agora. Vamos implementar um método que será chamado sempre que um novo *sms* chegar no celular. O método a ser implementado é o `onReceive`.

```
Bundle bundle = intent.getExtras();

Object messages[] = (Object[]) bundle.get("pdus");
SmsMessage smsMessage[] = new SmsMessage[messages.length];

for (int n = 0; n < messages.length; n++) {
  smsMessage[n] = SmsMessage.createFromPdu((byte[]) messages[n]);
}

AlunoDao dao = new AlunoDao (context) ;
if (dao.isAluno (smsMessage[0].getDisplayOriginatingAddress())) {
  Toast.makeText(context, "SMS de Aluno: "
    + smsMessage[0].getMessageBody(), Toast.LENGTH_LONG).show();
}
dao.close();
```

Lembre-se que como qualquer função avançada do Android, segue a permissão para o `AndroidManifest.xml`.

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

### Tocando MP3



Se o SMS recebido for realmente de um aluno, optamos por tocar um pedaço do "hino da vitória". É um MP3 pequeno que ficará dentro do seu aplicativo na subpasta `raw` de `res`, se não existir, pode criar. Para tocar, basta criar o `MediaPlayer` e dar um `start`:

```
MediaPlayer mp = MediaPlayer.create(context, R.raw.msg);  
mp.start();
```

Coloque este código dentro de SMSReceiver quando o telefone que enviou o SMS for de um aluno cadastrado.

## Lendo a bateria (Opcional)

Para conseguir ler a bateria, vamos registrar programaticamente um *BroadcastReceiver*. Ele será responsável por "ouvir" quando a bateria lança seus status e reportá-los ao código.

```
private BroadcastReceiver bateria = new BroadcastReceiver(){  
    @Override  
    public void onReceive(Context ctx, Intent intent) {  
        int valor = intent.getIntExtra("level", 0);  
        Toast.makeText(ctx, valor + "%", Toast.LENGTH_SHORT).show() ;  
    }  
};
```

Uma vez que temos o broadcast, devemos dispará-lo com o devido register:

```
registerReceiver(bateria, new IntentFilter(Intent.ACTION_BATTERY_CHANGED));
```

Agora é só esperar que as chamadas se darão automaticamente. Isso ocorre pois o Android avisa os broadcasts registrados evitando assim, o alto consumo de recursos do sistema.

