# **Verificando se o SMS foi enviado por um aluno**

Acabamos de fazer o recebimento do SMS e adicionamos um Toast que avisa sobre a chegada da mensagem. Agora, queremos que quando a mensagem for de um aluno da agenda, além do Toast, queremos receber um aviso sonoro diferente.

Teremos que encontrar uma forma de descobrir o telefone do remetente, adicionaremos um String telefone. Quando quisermos saber qual foi o SMS recebido, ele irá usar uma Intent. Iremos usar um objeto que chamaremos de sms.getPhone - e estará dentro da intent.

public class SMSReceiver extends BroadcastReceiver {  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 String telefone = sms.getPhone();  
  
 Toast.makeText(context, "Chegou um SMS!", Toast.LENGTH\_SHORT).show();  
  
 }  
}

Podemos usar o objeto SmsMessage sms, para representar uma mensagem de SMS. Porém, o que recebemos na Intent não é um objeto serializado. Para que um SMS consiga trafegar entre celulares mais novos e antigos, as mensagens possuem um formato específico, chamado de PDU(*protocol data unit*).

Então, teremos que pegar a pdu dentro da Intent, convertê-lo para um SmsMessage. Iremos adicionar o método createFromPdu e passaremos dois parâmetros: pdu e formato. Ambos virão da Intent.

public class SMSReceiver extends BroadcastReceiver {  
 @Override  
 public void onReceive(Context context, Intent intent) {  
  
 SmsMessage sms = SmsMessage.createFromPdu(pdu, formato);  
  
 String telefone = sms.getPhone();  
  
 Toast.makeText(context, "Chegou um SMS!", Toast.LENGTH\_SHORT).show();  
  
 }  
}

Após importarmos a classe SmsMessage, precisaremos desempacotar o pdu da Intent. Usaremos o método getSerializableExtra, que irá nos pedir a chave do que queremos recuperar. A documentação do Android irá nos informar que a chave é pdus. Iremos guardá-lo em um *array* de *objects*. Cada pdu será um Object.

Quando escrevemos um SMS, cada mensagem deve conter 160 caracteres. Em alguns celulares, quando escrevemos uma quantidade superior a essa, a mensagem será quebrada em duas mensagens ou mais - cada uma terá um pdu. E por isso, teremos um *array* com vários pdus.

No cabeçalho de cada pdu estará um número de telefone. Se não for uma uma "multimensagem", podemos aproveitar o pdu do primeiro SMS no *array*, ou seja, aquela que ocupar a posição 0.

pdu = pdus[0];

Depois, iremos transformá-la em um *array* de bytes.

byte[] pdu = (byte[]) pdus[0];

Agora, só falta pegarmos o formato. Criaremos uma String para isto.

String formato = (String) intent.getSerializableExtra("format");

Nosso código ficou assim:

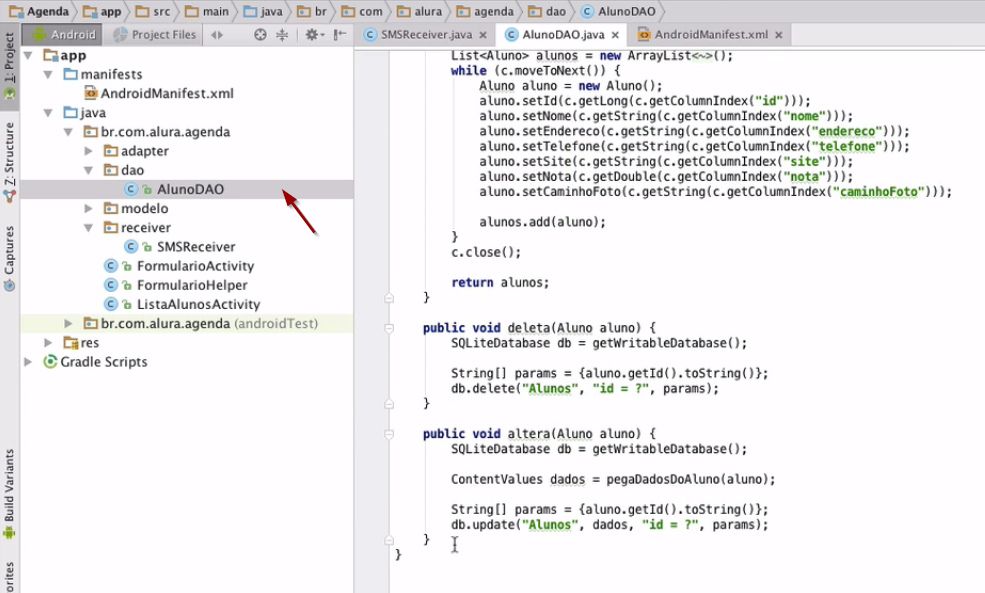
public class SMSReceiver extends BroadcastReceiver {  
 @Override  
 public void onReceive(Context context, Intent intent) {  
  
 Object[] pdus = (Object[]) intent.getSerializableExtra("pdus");  
 byte[] pdu = (byte[]) pdus[0];  
 String formato = (String) intent.getSerializableExtra("format");  
  
 SmsMessage sms = SmsMessage.createFromPdu(pdu, formato);  
  
 String telefone = sms.getPhone();  
  
 Toast.makeText(context, "Chegou um SMS!", Toast.LENGTH\_SHORT).show();  
  
 }  
}

Iremos agora substituir getPhone pelo método getDisplayOriginatingAddress(), que guardaremos na String Telefone.

String telefone = sms.getDisplayOriginatingAddress();

Com a alteração, o trecho do SMSReceiver.java ficou assim:

public class SMSReceiver extends BroadcastReceiver {  
 @Override  
 public void onReceive(Context context, Intent intent) {  
  
 Object[] pdus = (Object[]) intent.getSerializableExtra("pdus");  
 byte[] pdu = (byte[]) pdus[0];  
 String formato = (String) intent.getSerializableExtra("format");  
  
 SmsMessage sms = SmsMessage.createFromPdu(pdu, formato);  
  
 String telefone = sms.getDisplayOriginatingAddress();  
  
 Toast.makeText(context, "Chegou um SMS!", Toast.LENGTH\_SHORT).show();  
  
 }  
}

Agora, como faremos para descobrir que o telefone da mensagem é de um aluno? Iremos trabalhar com o AlunoDAO, porque ele tem acesso ao banco e pode checar este tipo de informação. 

Criaremos um novo método, que chamaremos de ehAluno. Dentro, passaremos uma String telefone. Em seguida iremos pegar uma instância do banco de dados, que seja apenas de leitura. Para fazermos uma busca, usaremos o rawQuery, e especificaremos que será na tabela Alunos.

public boolean ehAluno(String telefone) {  
 SQLiteDatabase db = getReadableDatabase();  
 db.rawQuery("SELECT \* FROM ALunos")  
}

Se quisermos pesquisar por um aluno específico, adicionaremos WHERE telefone = ?. Para termos mais segurança, usaremos ? e assim estaremos marcando qual o ponto será substituído. No parâmetro seguinte da rawQuery, diremos que String será o substituto.

public boolean ehAluno(String telefone) {  
 SQLiteDatabase db = getReadableDatabase();  
 db.rawQuery("SELECT \* FROM ALunos WHERE telefone = ?", new String[]{telefone});  
}

Se o nosso cursor trouxer um resultado da busca, significa que o telefone é de um aluno. Se não trouxer nenhum resultado, significa que não é o telefone de um aluno. Guardaremos o resultadocom c.getCount, que irá indicar a quantidade de resultados. E fecharemos o Cursor com c.close. E fecharemos com return.

public boolean ehAluno(String telefone) {  
 SQLiteDatabase db = getReadableDatabase();  
 Cursor c = db.rawQuery("SELECT + FROM ALunos WHERE telefone = ?", new String[]{telefone});  
 int resultados = c.getCount();  
 c.close();  
 return resultados > 0;  
}

Se o resultado for maior que 0, devolvemos um true. Caso o resultado seja igual a 0, o telefone não é de um aluno.

De volta ao SMSReceiver, iremos adicionar um if. Se o telefone for de um aluno, iremos mostrar o Toast que havíamos criado.

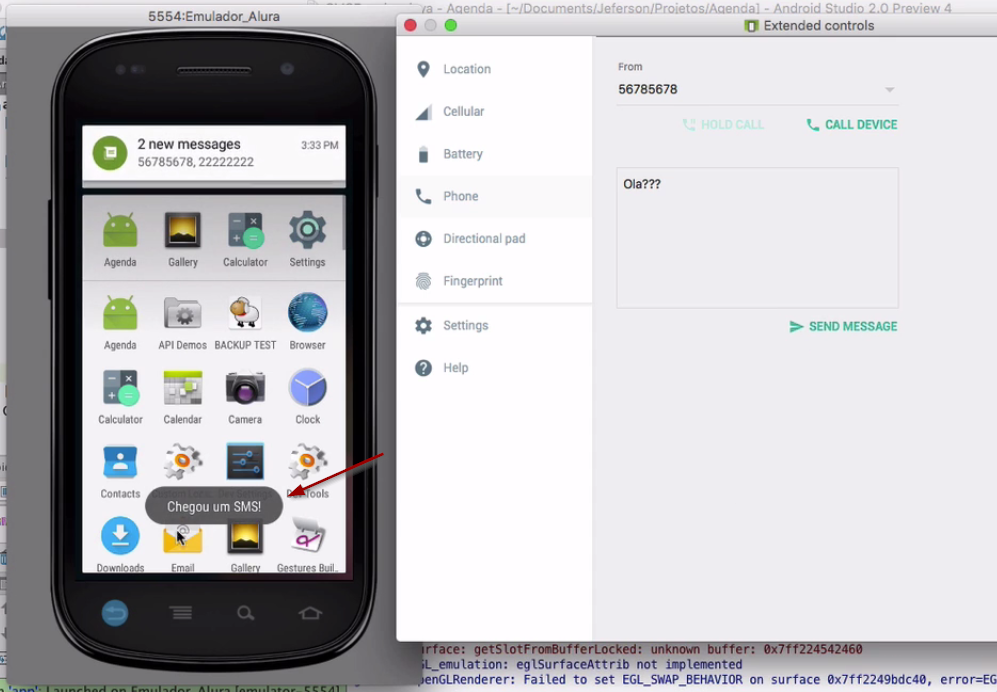
String telefone = sms.getDisplayOriginatingAddress();  
if(dao.ehAluno(telefone)) {  
 Toast.makeText(context, "Chegou um SMS!", Toast.LENGTH\_SHORT).show();  
}

Vamos depois instanciar o AlunoDAO e passar context como parâmetro.

String telefone = sms.getDisplayOriginatingAddress();  
AlunoDAO dao = new AlunoDAO(context);  
if(dao.ehAluno(telefone)) {  
 Toast.makeText(context, "Chegou um SMS!", Toast.LENGTH\_SHORT).show();  
 }  
 dao.close();

Faremos agora alguns testes no emulador. Enviaremos uma mensagem de um número desconhecido e outra de um telefone cadastrado no nosso aplicativo.

O resultado será que no primeiro envio, não veremos o Toast aparecer. No caso em que a mensagem foi enviada por um aluno, veremos o Toast aparecer e um novo sinal sonoro - que era usado no ICQ.



Iremos fazer uma pequena alteração na mensagem, que agora irá exibir Chegou um SMS de Aluno!. Também iremos mudar o sinal sonoro, usando a classe MediaPlayer e o método MediaPlayer.create. Como parâmetro, passaremos context e iremos referenciar um arquivo de áudio que temos salvo na pasta raw.

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

O nosso SMSReceiver.java ficou assim:

public class SMSReceiver extends BroadcastReceiver {  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 Object[] pdus = (Object[]) intent.getSerializableExtra("pdus");  
 byte[] pdu = (byte[] pdus[0];  
 String formato = (String) intent.getSerializableExtra("format");  
  
 SmsMessage sms = SmsMessage.createFromPdu(pdu, formato);  
  
 String telefone = sms.getDisplayOriginatingAddress();  
 AlunoDAO dao = new AlunoDAO(context);  
 if (dao.ehAluno(telefone)) {  
 Toast.makeText(context, "Chegou um SMS de Aluno!", Toast.LENGTH\_SHORT).show();  
 MediaPlayer mp = MediaPlayer.create(context, R.raw.msg);  
 mp.start();  
 }  
 dao.close();  
 }  
}

Com isto, encerramos a parte de tratamentos de eventos do Android.

# 