

 03
Exercício 1

Vamos criar um novo especialista que ficará atento a chegada de mensagens do tipo SMS. Para isso crie a classe `SMSReceiver` no pacote `br.com.caelum.cadastro.receiver` e faça-a estender `BroadCastReceiver`.

Para lidar com a chegada de uma mensagem interna no Android vamos sobrepor o método `onReceive`:

```
@Override
public void onReceive(Context context, Intent intent) {
    Toast.makeText(context, "Chegou um SMS!!!", Toast.LENGTH_LONG).show();
}
```

Precisamos configurar nosso `SMSReceiver` para que ele esteja atento à chegada de mensagens do tipo `SMS`. Configure as seguintes linhas no `AndroidManifest` dentro da tag `application`:

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

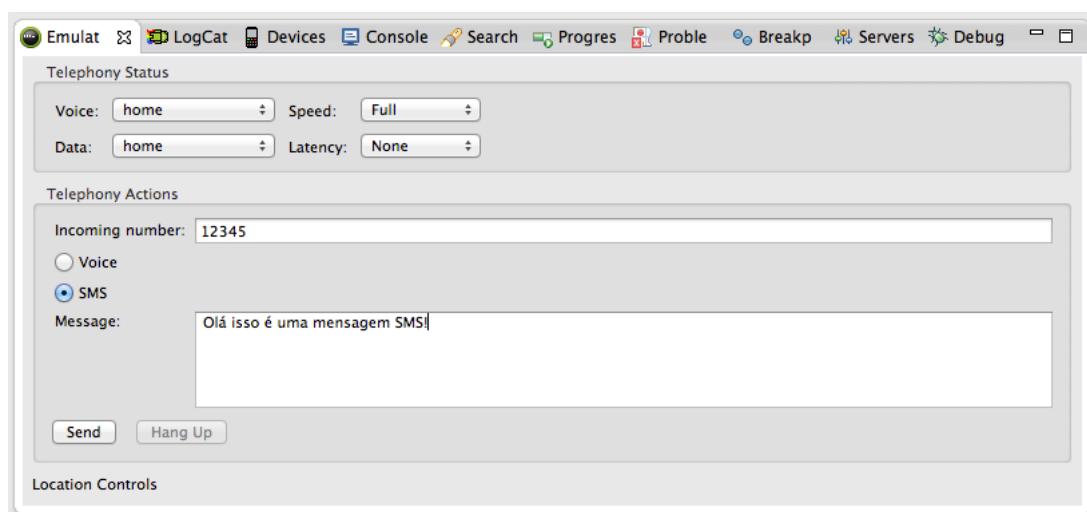
Precisamos também que o usuário autorize que nossa aplicação intercepte a chegada de `SMS`'s portanto devemos cadastrar uma nova permissão no `AndroidManifest`:

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

Rode a aplicação com essas alterações.

Para testarmos no emulador o recebimento de um `SMS` precisamos da aba **Emulator Control**. Para acessá-la use o atalho `CTRL+3` e depois digite **Emulator Control**.

Com um emulador rodando, preencha o campo `Incoming number` da aba **Emulator Control** com algum telefone. Clique na opção `SMS` e preencha uma mensagem, depois clique em `Send`.



O que aconteceu quando você rodou a aplicação e fez o envio de um `SMS` para o emulador?

Como ficou sua implementação da classe `SMSReceiver`?

Responda

INserir CÓDIGO		FORMATaÇÃO
<div style="height: 100px;"></div>		