Trabalhando com outras informações da agenda

Nós já conseguimos colocar a funcionalidade de dar um clique longo sobre o nome do aluno e ter a opção de visitar o site cadastrado no perfil. Conseguimos aproveitar uma informação que tínhamos disponível na agenda. Mas se observarmos os perfis novamente, veremos que existem duas informações não aproveitadas: **Endereço** e **Telefone**. Vamos aproveitá-las agora.

Ao darmos um clique longo sobre o nome do aluno, queremos que apareça um menu com a opção de enviarmos um SMS para ao aluno. Também queremos uma opção, que a partir do endereço, nos indique a localização do aluno.

Voltando ao código, no menu de contexto, vamos incluir mais dois itens. Iremos usar a mesma ideia da Intent implícita, para aproveitar os apps já instalados no nosso celular e o usuário possa fazer a escolha mais interessante.

Antes do "Visitar site", vou incluir um novo menu.add("Enviar SMS"). Vou dar um ALT + ENTER para que ele se torne o itemSMS. Depois, criaremos nossa Intent que nos fará mudar de tela. Como é uma Intent implícita, diremos para o Android qual ação executar. Neste caso, nós também queremos visualizar e usaremos a ACTION\_VIEW. Queremos que ele abra uma aplicação que visualize SMS, e nos permita digitar, corrigir os dados, além de enviar a mensagem.

MenuItem itemSMS = menu.add("Enviar SMS");  
Intent intentSMS = new Intent(Intent.ACTION\_VIEW) ;

Após criarmos a intentSMS, faltará associar as duas linhas:

itemSMS.setIntent(intentSMS);

Elas estão associadas, mas agora, precisamos especificar o que queremos visualizar. Precisamos passar a uri e informar qual recurso queremos utilizar. Será o mesmo processo que fizemos com o site: vamos adicionar intentSMS, depois o método setData, e dentro, iremos usar a uri. Lembrando que não podemos usar diretamente uma *string*, temos que acrescentar a classe Uri.parse e em seguida a URI.

Precisaremos conhecer a documentação do Android, para descobrir qual o protocolo utilizado para reconhecer o SMS. Neste caso, quando falamos de SMS, usamos o sms:. Logo adiante, podemos adicionar um número de telefone, por exemplo.

intentSMS.setData(Uri.parse("sms:12345678"));

O trecho do código correspondente ficará assim:

MenuItem itemSMS = menu.add("Enviar SMS");  
Intent intentSMS = new Intent(Intent.ACTION\_VIEW) ;  
intentSMS.setData(Uri.parse("sms:12345678"));  
itemSMS.setIntent(intentSMS);

Após adicionarmos as novas linhas, podemos fazer um teste e clicar no nome do aluno, selecionaremos a opção de enviar SMS. O app de mensagem já estará preenchido com o número cadastrado. Nós iremos concatenar no código o número do aluno.

intentSMS.setData(Uri.parse("sms:" + aluno.getTelefone()));

O código ficou assim:

MenuItem itemSMS = menu.add("Enviar SMS");  
Intent intentSMS = new Intent(Intent.ACTION\_VIEW) ;  
intentSMS.setData(Uri.parse("sms:" + aluno.getTelefone()));  
itemSMS.setIntent(intentSMS);

Vamos aproveitar e adicionar o segundo item, que seria o de visualizar no mapa. Faremos o mesmo procedimento: incluímos o menuadd e o nome que será exibido no menu Visualizar no mapa. Depois de atribuir a variável, adicionaremos o itemMapa.

MenuItem itemMapa = menu.add("Visualizar no mapa");

Criaremos outra Intent implícita e usaremos o ACTION\_VIEW novamente. Depois, adicionamos intentMapa e associamos com intentMapa.

MenuItem itemMapa = menu.add("Visualizar no mapa");  
Intent intentMapa = new Intent(Intent.ACTION\_VIEW);  
itemMapa.setIntent(intentMapa);

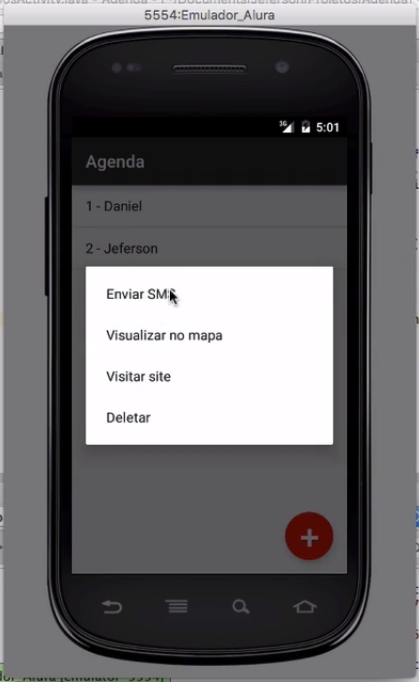
Agora, precisamos específicar neste item o que queremos visualizar. Vamos adicionar a classe Uri.parse e descobrir qual protocolo será utilizado. Na documentação, iremos descobrir que é o geo:. Como iremos trabalhar com coordenadas geográficas. Vamos ter que especificar a latitude e longitude da posição que queremos mostrar no mapa. Por exemplo, 0,0''. Ele irá abrir o GoogleMaps e irá indicar com um pininho a posição específica do aluno. Após o geo: deve aparecer as coordenadas daquele endereço do aluno selecionado no menu de contexto. Como é feita conversão do endereço? Se fôssemos fazer isto para o mundo inteiro, deveríamos ter uma tabela gigantesca. Seria um banco de dados enorme, que geralmente não fica armazenado no celular. Nós podemos consultar o Google Maps sobre a coordenada do endereço e ele irá devolver a geolocalização. Para isto, iremos usar geo:0,0?q= e depois vamos concatenar com aluno.getEndereco().

intentMapa.setData(Uri.parse("geo:0,0?q=" + aluno.getEndereco()));

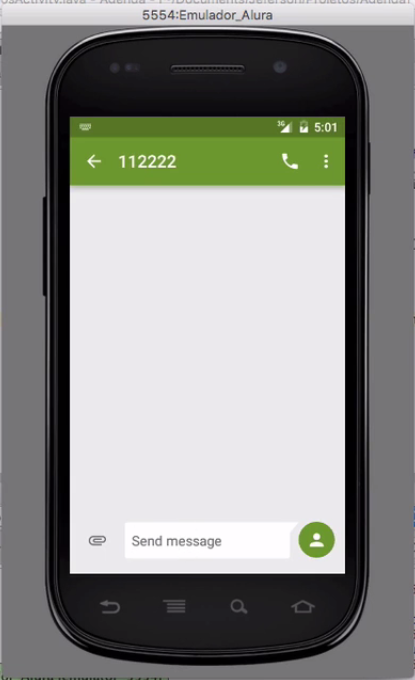
Assim, ao clicarmos sobre o nome do aluno, nós iremos enviar o endereço para o Google, ele irá retornar automaticamente a coordenada e ele abrirá o Google Maps com a coordenada específica.

Vale lembrar, que quando quisermos que o app execute uma tarefa que outro aplicativo no nosso celular já faz, podemos usar uma Intent implícita. A única diferença será o protocolo usado para representar o recurso solicitado.

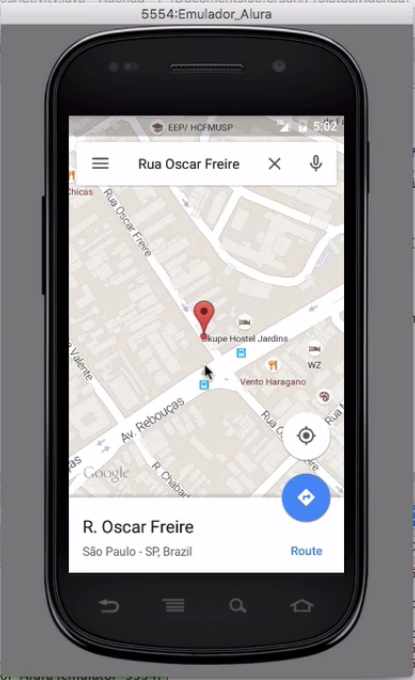
Vamos rodar o app e ver se os novos itens funcionam bem. Quando o aplicativo estiver aberto, daremos um clique longo sobre um dos nomes e aparecerá um menu com as novas opções.



Selecionaremos "Enviar SMS" e veremos será aberto o app de mensagem.



De volta a nossa aplicação, vamos testar a opção "Visualizar no mapa". O app do Google Maps será aberto com sucesso, já com o pininho característico sobre a localização.



Se observarmos o código das duas funcionalidades, perceberemos a semelhança. A única diferença foi o protocolo.

MenuItem itemSMS = menu.add("Enviar SMS");  
Intent intentSMS = new Intent(Intent.ACTION\_VIEW) ;  
intentSMS.setData(Uri.parse("sms:" + aluno.getTelefone()));  
itemSMS.setIntent(intentSMS);  
  
MenuItem itemMapa = menu.add("Visualizar no mapa");  
Intent intentMapa = new Intent(Intent.ACTION\_VIEW);  
intentMapa.setData(Uri.parse("geo:0,0?q=" + aluno.getEndereco()));  
itemMapa.setIntent(intentMapa);

Podemos criar outros recursos com o nosso código, basta pesquisarmos um pouco na documentação e analisarmos quais aplicativos temos no celular. Às vezes, temos disponível bibliotecas externas. Temos muitas opções para incrementar nossas aplicações.