[**Desenvolvimento para Android: Utilizando GPS em um aplicativo Android**](http://www.escolacriatividade.com/desenvolvimento-para-android-utilizando-gps-em-um-aplicativo-android/)

19-11-2012, em [Tutoriais](http://www.escolacriatividade.com/categoria/tutoriais/), por [Jair Rebello](http://www.escolacriatividade.com/author/jair-rebello/)

* 3  
  [inShare](javascript:void(0);)
* [Email](mailto:?subject=http://www.escolacriatividade.com/desenvolvimento-para-android-utilizando-gps-em-um-aplicativo-android/)

Continuando com os artigos de introdução ao desenvolvimento Android, hoje iremos aprender a utilizar o recurso de GPS em nossas aplicações para Android, já aprendemos a construir [pequenas aplicações android](http://www.escolacriatividade.com/introducao-ao-desenvolvimento-para-android-parte-iv/) e também a se [conectar a um servidor FTP](http://www.escolacriatividade.com/desenvolvimento-para-android-conexao-e-envio-via-ftp/), realizando também upload e mais este artigo fará com que os passos para se desenvolver para dispositivos com este sistema operacional sejam dados.

O ideal é que você tenha um dispositivo Android para realizar um teste real, caso não tenha, no final do artigo darei a solução para testar diretamente em uma AVD (Android Virtual Device).

Caso queira acompanhar a série de introdução ao desenvolvimento em Android:

* [Introdução ao Desenvolvimento para Android (Parte I)](http://www.escolacriatividade.com/introducao-ao-desenvolvimento-para-android-parte-i/)
* [Introdução ao Desenvolvimento para Android (Parte II)](http://www.escolacriatividade.com/introducao-ao-desenvolvimento-para-android-parte-ii/)
* [Introdução ao Desenvolvimento para Android (Parte III)](http://www.escolacriatividade.com/introducao-ao-desenvolvimento-para-android-parte-iii/)
* [Introdução ao Desenvolvimento para Android (Parte IV)](http://www.escolacriatividade.com/introducao-ao-desenvolvimento-para-android-parte-iv/)
* [Introdução ao Desenvolvimento para Android (Parte V)](http://www.escolacriatividade.com/desenvolvimento-para-android-conexao-e-envio-via-ftp/)



**PORQUE UTILIZAR GPS EM UMA APLICAÇÃO MÓVEL?**

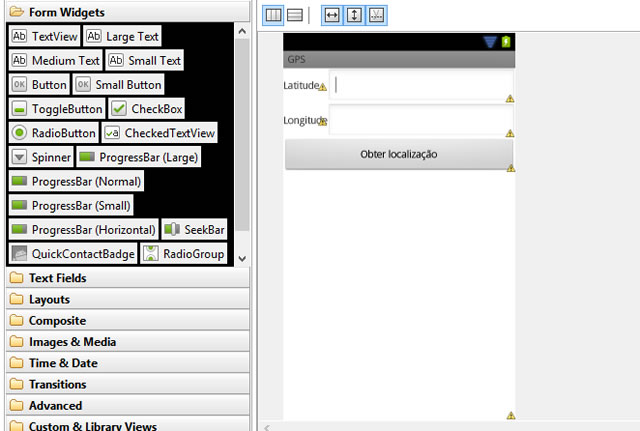
Certamente um dos recursos mais incríveis dos celulares mais modernos é o GPS. GPS é o acrônimo para **sistema de posicionamento global,** é um sistema que permite ao dispositivo determinar a posição exata do aparelho no globo terrestre segundo coodernadas passadas por um dos satélites mundiais. É utilizado largamente em veículos automotivos para determinar rotas mas ultimamente é bem comum ver dispositivos móveis como celulares e tablets com GPS. Este sistema é muito utilizado por aplicativos sociais para realizar checkin, determinar onde a foto foi tirada ou mesmo montar um mapa de fotos em aplicativos sociais como o Facebook, Instagram, ou mesmo o Foursquare.

Um exemplo bem prático para aplicar esta funcionalidade em um aplicativo é determinar a posição exata de uma ocorrência de trânsito, este é apenas um de muitos exemplos. E você pode estar se perguntando, não existem já no mercado diversos aplicativos de GPS bem completos, para que aprender a desenvolver uma funcionalidade destas? Primeiramente possui uma razão didática, é sempre bom aprender a utilizar a API de sistema amplamente usado como o Android e a outra razão é que você pode necessitar agregar ao seu sistema esta funcionalidade sem que dependa de outras ferramentas para realizar tal tarefa.

**CONSTRUINDO A INTERFACE**

Como vimos no artigo [Introdução ao Desenvolvimento para Android (Parte II)](http://www.escolacriatividade.com/introducao-ao-desenvolvimento-para-android-parte-ii/) toda aplicação Android necessita de um arquivo XML que definirá a visualização do aplicativo, ou seja, será aquilo que o usuário verá quando iniciar a aplicação, estes arquivos XML não realizam operações, apenas são responsáveis por definir cores, tamanhos de elementos e outros fatos, para este exemplo iremos construir um XML bem simples, apenas iremos inserir um botão de ação, 2 textView e 2 EditText, que serão carregados com as informações de GPS. Caso tenha dificuldades para criar projetos, arquivos XML, acesse o link acima que você terá todas as informações necessárias para tal.

A visualização de nosso aplicativo deve ser semelhante a esta:



Para que você consiga este resultado, seu arquivo XML deve estar semelhante a isto:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60 | <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"          android:layout\_width="fill\_parent"          android:layout\_height="fill\_parent" >        <LinearLayout          android:layout\_width="match\_parent"          android:layout\_height="match\_parent"          android:orientation="vertical" >            <LinearLayout              android:layout\_width="match\_parent"              android:layout\_height="wrap\_content" >                <TextView                  android:id="@+id/textView1"                  android:layout\_width="wrap\_content"                  android:layout\_height="wrap\_content"                  android:layout\_weight="0.22"                  android:text="Latitude" />                <EditText                  android:id="@+id/edLatitude"                  android:layout\_width="wrap\_content"                  android:layout\_height="wrap\_content"                  android:layout\_weight="0.73"                  android:ems="10" >                    <requestFocus />              </EditText>            </LinearLayout>            <LinearLayout              android:layout\_width="match\_parent"              android:layout\_height="wrap\_content" >                <TextView                  android:id="@+id/textView2"                  android:layout\_width="wrap\_content"                  android:layout\_height="wrap\_content"                  android:text="Longitude" />                <EditText                  android:id="@+id/edLongitude"                  android:layout\_width="wrap\_content"                  android:layout\_height="wrap\_content"                  android:layout\_weight="1"                  android:ems="10" />            </LinearLayout>            <Button              android:id="@+id/btLocalizar"              android:layout\_width="match\_parent"              android:layout\_height="wrap\_content"              android:text="Obter localização" />        </LinearLayout>    </LinearLayout> |

Note que nas **linhas 22, 44 e 53** definimos os nomes do dois EditText e também do botão de ação, sem a definição correta dos nomes destes elementos teremos problemas quando passarmos para o código Java necessário para obter a localização e mostrar para o usuário o resultado, atente para isso.

**CONSTRUINDO A LÓGICA NECESSÁRIA**

A lógica necessária para que o nosso aplicativo funcione de fato consiste basicamente em três passos:

* Importar os elementos do arquivo XML, ou seja, pegar os elementos da tela através da classe R, gerada automaticamente pelo plugin do Android.
* Ativar o GPS e depois receber os parâmetros de latitude e longitude
* Por fim, precisamos apresentar ao usuário estas informações.

Irei inserir o código inteiro da classe Main.java e depois explicar as linhas mais importantes do código, vamos lá:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62 | //nome do pacote, pode mudar de acordo com a configuração do projeto  package br.com.gps;    //importação das classes necessárias para o funcionamento do aplicativo  import android.location.Location;  import android.location.LocationListener;  import android.location.LocationManager;  import android.os.Bundle;  import android.app.Activity;  import android.content.Context;  import android.view.View;  import android.widget.Button;  import android.widget.EditText;    public class Main extends Activity {        //variáveis que usaremos durante o processo      private EditText edLatitude;      private EditText edLongitude;      private Button btLocalizar;        @Override // Método onCreate, iniciada quando o aplicativo é aberto      public void onCreate(Bundle savedInstanceState) {          super.onCreate(savedInstanceState);          setContentView(R.layout.main);          setupElements();      }        // Método usado para importar os elementos da classe R      public void setupElements(){          edLatitude = (EditText) findViewById(R.id.edLatitude);          edLongitude = (EditText) findViewById(R.id.edLongitude);            btLocalizar = (Button) findViewById(R.id.btLocalizar);          btLocalizar.setOnClickListener(new Button.OnClickListener() {              public void onClick(View v){                  startGPS();          }});      }      //Método que faz a leitura de fato dos valores recebidos do GPS      public void startGPS(){          LocationManager lManager = (LocationManager)getSystemService(Context.LOCATION\_SERVICE);          LocationListener lListener = new LocationListener() {              public void onLocationChanged(Location locat) {                  updateView(locat);              }              public void onStatusChanged(String provider, int status, Bundle extras) {}              public void onProviderEnabled(String provider) {}              public void onProviderDisabled(String provider) {}          };          lManager.requestLocationUpdates(LocationManager.GPS\_PROVIDER, 0, 0, lListener);      }        //  Método que faz a atualização da tela para o usuário.      public void updateView(Location locat){          Double latitude = locat.getLatitude();          Double longitude = locat.getLongitude();            edLatitude.setText(latitude.toString());          edLongitude.setText(longitude.toString());      }  } |

O primeiro grupo de código que vai desde a linha 4 até a linha 13 são importações de classes que serão usadas em nosso aplicativo, provavelmente o Eclipse já incluirá ou dará assistência para que sejam colocadas todas estas importações, caso você tenha algum problema copie todo o trecho que está acima.

Nas linhas 22 até a linha 27 é criado um método chamado **onCreate**, este método é implementado por nós para que seja executado sempre que o aplicativo iniciar, ou seja, tão logo o aplicativo seja mostrado para nós, o método é chamado. Neste método é feito o carregamento do XML para ser a visualização do aplicativo, também é chamado o método **setupElements,**que explicarei logo abaixo seu objetivo**.**

Entre as **linhas 30 e 39**temos o método**setupElements,**ele é responsável por importar para o código java, os elementos gerados pelo arquivo XML que configuramos no início do artigo, dentro deste elemento além da instanciação dos elementos também é criado um método que possui o objetivo de executar o método **startGPS()**tão logo o usuário clicar no botão **btSalvar**.

As linhas de 41 e 52 são as linhas mais importantes do aplicativo, são elas que fazem de fato a comunicação com o GPS do celular e faz ele requisitar as coordenadas atuais de sua localização, note que existe um método chamado **onLocationChanged,**ou seja, quando a localização do celular mudar, será chamada um outro método chamado **updateView,**que nada mais faz do que atualiza os dados em tela para que o usuário enfim veja as coordenadas em seu dispositivo.

**MODULARIZAÇÃO**

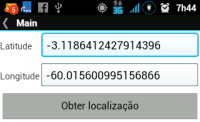
Ainda não falamos deste assunto nestes tutoriais sobre desenvolvimento Android, perceba que nosso aplicativo apesar de possuir poucas linhas, ele possui diversos métodos, isso acontece porque utilizamos um dos conceitos mais importantes da programação moderna, a modularizarão. Este padrão faz com que nosso aplicativo seja dividido em trechos de código menores chamados métodos e cada um deste método tenha um único objetivo, isso faz com que nosso código fique mais claro e também seja necessário um esforço menor quando for necessário fazer alguma manutenção no código, pois sei onde encontrar cada funcionalidade do aplicativo, se precisar mudar o modo com que o usuário visualiza o resultado da busca pelas coordenadas (Exibir em uma caixa de diálogo, por exemplo), mexeremos no método que possui este objetivo, ou seja, o **updateView**.

**PERMISSÃO PARA ACESSO AO GPS DO CELULAR**

Existe um arquivo chamado AndroidManifest.xml que guarda todas as informações do aplicativo, lá podemos encontrar que permissões aquele aplicativo precisará, como acesso a internet, publicar algo em meu nome e lá também encontramos a opção de permitir o acesso ao GPS do celular, para que isso aconteça, insira estas linhas no AndroidManifest.xml:

|  |  |
| --- | --- |
| 1  2 | <uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"/>  <uses-permission android:name="android.permission.ACCESS\_MOCK\_LOCATION"/> |

**COMO TESTAR O APLICATIVO SEM POSSUIR UM DISPOSITIVO COM GPS?**



Muitos desenvolvedores quando iniciam suas aventuras pelo desenvolvimento Android não possuem um dispositivo para testar suas aplicações, a imagem ao lado mostra como ficou o aplicativo criado neste artigo em um Galaxy Ace da Samsumg, que por sinal é um ótimo aparelho, seja para testar os aplicativos, seja para utilizar no dia-a-dia. Caso você não tenha um dispositivo físico, pode testar mesmo em uma AVD (Android Virtual Device), precisaremos apenas de algumas configurações para isto.

Dentro do Eclipse vá até o menu Window > AVD Manager, caso já tenha criado alguma AVD, clique em Editar, caso contrário crie uma nova. As últimas opções de configurações são as opções de Hardware, clique em New e depois em **GPS Support**colocando o valor yes. Após isso rode a aplicação no AVD normalemnte. Para alterar os parâmentos de localização, você pode acessar Emulator Control a alterar os valores que o GPS usará para localização.

Mesmo sendo possível realizar testes no AVD Manager, nada se compara a um dispositivo real, caso seja possível adquira um celular com Android para testar  suas aplicações em um ambiente de produção, como todos os recursos reais.

**CONCLUSÃO**

Neste artigo aprendemos a trabalhar o básico do GPS que é conseguir as informações de longitude e latitude de sua localização atual, em artigos futuros iremos ensinar como integrar estes valores com o Google Maps e ver no mapa s localização atual bem como buscar informações de estabelecimentos ou lugares que estejam próximos de nossa localização atual, qualquer dúvida que apareça durante o processo de contrução da aplicação é só comentar que estaremos disponíveis para tentar solucionar o problema.

Abraços!