

CAPÍTULO II

Introdução à API Google Static Maps com Android

Desenvolva aplicações com acesso a mapas, baixando as imagens deste e os utilize de forma off-line

Capítulo utiliza estudo de caso desenvolvido na IDE:

Atualmente existe uma grande procura por novos meios de localização e mapeamento digital. Seguindo essa tendência, foram lançados no mercado inúmeros sistemas e aparelhos que atendem a essa crescente demanda, dentre eles localizadores automotivos e GPS’s veiculares. Entretanto, o maior destaque do segmento é o portal de informações Google Maps, desenvolvido e mantido pela Google, o qual revolucionou a internet ao disponibilizar a busca de mapas rodoviários e de satélites, o qual provê a localização de países, estados, cidades e ruas de forma gratuita aos internautas.

Algum tempo depois foi lançado pela mesma empresa o Google Maps API, que consiste basicamente de um conjunto de bibliotecas que proporcionam ao desenvolvedor acesso a serviços disponibilizados pelo Google Maps. Através dela é possível construir aplicativos que façam uso tanto da busca quanto da visualização dos mapas.

Dado o elevado poder de processamento dos novos celulares, apresentou-se a possibilidade destes acessar mapas digitais, facilitando a vida dos usuários que costumam viajar para novas cidades e que desejam localizar endereços. Porém, sem precisar obrigatoriamente utilizar as redes de dados das operadoras, reduzindo assim o custo de utilização do aplicativo a zero.

O Google Maps tradicional é um sistema de repositório de mapas online, ou seja, é necessário estar conectado a Internet para o aplicativo acessar os mapas. Desta forma, uma alternativa interessante ao uso dos mapas é salvá-los no próprio device, enquanto este esteja conectado a uma rede livre (ex. wifi), e após, apenas fazer uso dos mapas sem acesso a Internet e sem custos para o usuário.

Uma alternativa ao uso do Google Maps é a utilização da API da Google chamada Static Maps. Esta facilita a inclusão de imagens de mapas estáticos em uma aplicação, seja ela desktop, web ou móvel, tendo esta API uma interface com o programador via requisições HTTP, o que garante a fácil portabilidade do seu uso nas mais diversas plataformas.

Na prática, através do Google Static Maps, é possível a recuperação das imagens do mapa apenas solicitando-as através de uma requisição http, acessando-as via URL. Esta pode ser incluída, por exemplo, em HTML utilizando-o na tag IMG, no caminho da imagem passando a url e seus parâmetros, ou em componentes visuais do Android como o ImageView.

Parâmetros para acesso ao Google StaticMaps

O Google Static Maps é acessado via HTTP, formatando uma requisição via URL, nos quais os parâmetros da URL são os parâmetros da requisição. O formato básico da URL que deve ser passada na requisição é:

http://maps.googleapis.com/maps/api/staticmap?parametros

Para personalizar a consulta, utiliza-se os seguintes parâmetros:

* Parâmetros de local:
  + **center** (obrigatório se não houver marcadores presentes): define o centro do mapa e equivale ao local desejado. Podem ser passados dados como String com o endereço do local, como por exemplo “Francisco, PR”, ou ponto geográfico seguido pelo par de latitude e longitude, separadas por virgula, como “-26.081185,-53.091238”.
  + **zoom** (obrigatório se não houver marcadores presentes): define o nível de zoom da imagem (podem ser passados apenas dados numéricos).
* Parâmetros de mapa:
  + **size** (obrigatório): define as dimensões de área da imagem. Pode ser passada apenas uma String formada por valores numéricos correspondentes à altura e largura e o caractere “x” que os separa, conforme exemplo: “500x400”;
  + **scale** (opcional): define a resolução da imagem (recomendado por exemplo para uso das imagens em impressões). Pode ser passado apenas o seguinte dado numérico: 1, 2 ou 4 (4 está disponível apenas com o uso de chave de API para empresas);
  + **format** (opcional): define o formato da imagem retornada. Por padrão, a API retorna o formato PNG. Pode ser passada uma string do conjunto: png8, png, png32, gif, jpg ejpg-baseline;
  + **maptype** (opcional): define o tipo de mapa a ser criado. Por padrão, a API retorna a imagem no tipo roadmap, mas se preferir pode ser informado outro tipo dentre as opções: satellite, terrain e hybrid;
  + **language** (opcional): define a linguagem a ser utilizada para exibição de rótulos nos blocos do mapa. Se a linguagem informada não for suportada, será utilizada a linguagem padrão do conjunto de blocos do mapa;
  + **region** (opcional): define as bordas apropriadas a serem exibidas com base em limites geopolíticos.
* Parâmetros de recurso:
  + **markers** (opcional): define um ou mais marcadores a serem exibidos na imagem. Esse parâmetro pode receber vários valores, e todos separados por barra vertical (|). O primeiro valor informado é o estilo do marcador, que deve ser informado da seguinte maneira: estilo:valor. Os estilos suportados são:
    - size: opcional e define o tamanho do marcador que pode ser tiny, mid ou small;
    - color: opcional e define a cor do marcador - podem ser informados valores de 24bits ou valores em String do conjunto: black, brown, green, purple, yellow, blue, gray, orange, red, white;
    - label: opcional e define um caractere alfanumérico maiúsculo para ser exibido dentro do marcador, os marcadores tiny e small não podem exibir o label.
  + **path** (opcional): define um único caminho de dois ou mais pontos a ser exibido na imagem. Esse parâmetro pode receber vários valores, e todos separados por barra vertical (|). O primeiro valor que pode ser informado é o estilo do caminho, que deve ser informado na seguinte maneira: estilo:valor. Os estilos suportados são:
    - weight: opcional e define a espessura do caminho em pixels, a largura padrão é 5 pixels;
    - color: opcional e define uma cor para o caminho - podem ser especificados valores de 24 bits “0xFFFFCC”, 32 bits “0xFFFFCCFF” ou valores em strings do conjunto: black, brown, green, purple, yellow, blue, gray, orange, red, White;
    - fillcolor: opcional e define que o caminho formará uma área a ser preenchida com uma cor;
  + **visible** (opcional): define um ou mais locais que devem permanecer visíveis na área da imagem.
  + **style** (opcional): define uma estilização da imagem. Esse parâmetro pode receber vários valores, e todos separados por barra vertical (|). Os primeiros valores que podem ser informados são o feature (indica quais elementos devem ser selecionados para mudança de estilo: all, road e landscape) e element (indica um subconjunto de recursos que deve ser selecionado: all, geometry e labels). Na sequência, regras de estilos devem ser informadas através do valor rule1, rule2...: as regras que podem ser definidas para estilo são:
    - hue: define uma cor a ser aplicada à seleção. Pode ser informada uma string RGB hexadecimal no formato 0Xrrggbb;
    - lightness: define um percentual de brilho da seleção. Pode ser informado um valor pontual variável entre -100 e 100;
    - saturation: define a saturação a ser aplicada à cor básica. Pode ser informado um valor pontual variável entre -100 e 100;
    - gamma: define a quantidade de correção gama a ser aplicada à seleção. Pode ser informado um valor pontual variável entre 0.01 e 10.0, em que 1.0 não aplica nenhuma correção;
    - inverse\_lightness: define a inversão de luminosidade. Pode ser informado o valor true;
    - visibility: define se um elemento apresentará ou não no mapa ou apresentará de forma simplificada. Pode ser informado os valores on, off ou simplified.
* Parâmetro de relatório:
  + **sensor** (obrigatório): identifica se o recurso está sendo executado a partir de dispositivos com capacidade de identificação geográfica com o uso de sensores. Pode ser passado true ou false.

Retornos e limitações do Google StaticMaps

O retorno às requisições via Google Static Maps basicamente são imagens. Elas são tratadas automaticamente pelo aplicativo chamador, ou devem ser tratadas manualmente pelo programador, isto depende da forma que é chamado.

No Android, por exemplo, a imagem é recuperada através de um Stream (um array de bytes com os dados da imagem), podendo este ser passado por parâmetro no método correspondente do componente ImageView. Neste caso deve ser feito manualmente. Já em html, este pode ser usado como parâmetro da tag img, e seu conteúdo já é apresentado automaticamente na página, ou seja, sua conversão e apresentação é feita de forma automática.

Quanto às limitações, o Google Static Maps, por padrão, possui um limite diário de 25.000 solicitações gratuitas por aplicativo. Caso queira adquirir uma cota superior a este limite é necessário a geração de uma chave de API que pode ser solicitada através do Console de APIs da Google.

Se o usuário ultrapassar o limite diário, o servidor retorna status HTTP 403 à requisição http e apresenta a **Figura 2.1** como indicativo de que a cota foi ultrapassada.



Figura 2.1. Indicativo de cota ultrapassada.

Além do limite diário, outra limitação do Google Static Maps está no tamanho da requisição já que o mesmo faz uso do HTTP. Se a requisição tiver um tamanho superior a 2048 bytes, ou seja, o usuário parametrizou muito a consulta e o tamanho da String ultrapassou 2KB, a figura retornada será idêntica à **Figura 2.2**.

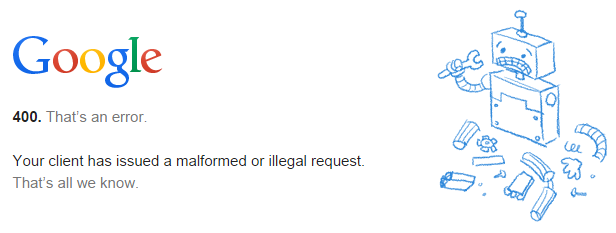


Figura 2.2. Indicativo de requisição ilegal

Exemplos práticos de uso dos parâmetros

Para facilitar o entendimento do uso dos parâmetros, alguns exemplos práticos de sua utilização serão apresentados na sequência.

Parâmetro Centro (center)

O parâmetro Center aceita dois tipos de informações, um conjunto numérico representando a latitude e longitude, como o valor de “-26.081185,-53.091238” ou uma string contendo o endereço desejado, como o valor de “UTFPR – Campus Francisco Beltrão”, lembrando que valores para URL precisam estar devidamente codificados para URI, sendo o valor exemplo transformado para “UTFPR+Campus+Francisco+Beltr%C3%A3o”.

Desta forma, para obter a imagem em questão, a consulta pode ser feita de duas maneiras:

http://maps.googleapis.com/maps/api/staticmap?center=-26.081185,-53.091238&zoom=15&size=400x400&sensor=false

ou

http://maps.googleapis.com/maps/api/staticmap?center=UTFPR+Campus+Francisco+Beltr%C3%A3o&zoom=15&size=400x400&sensor=false

Para a requisição, foi utilizado o mínimo de parâmetros possível. Não se preocupem por hora com os demais parâmetros, apenas com o conteúdo em negrito na URL. Em ambas as requisições, o resultado será uma imagem semelhante à apresentada na **Figura 2.3**.

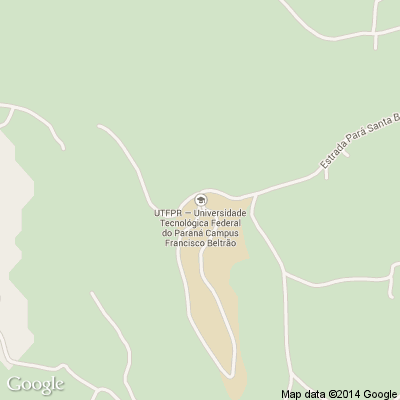


Figura 2.3. Figura demonstra o mesmo resultado para parâmetro center com latitude e longitude ou por endereço.

Parâmetro Zoom (zoom)

O parâmetro zoom permite aproximar ou distanciar uma imagem, fazendo com que o centro da imagem tenha uma área menor quando aproximado, mas com mais detalhes de visualização, e uma visão mais ampla, com uma área maior no mesmo quadro quando é distanciado.

O valor mínimo do zoom é zero, e neste caso, será apresentado para o usuário o mapa mundi. O valor máximo para o zoom é 22. A **Figura 2.4** apresenta o resultado de uma consulta utilizando zoom 15 (esquerda) e 17 (direita).

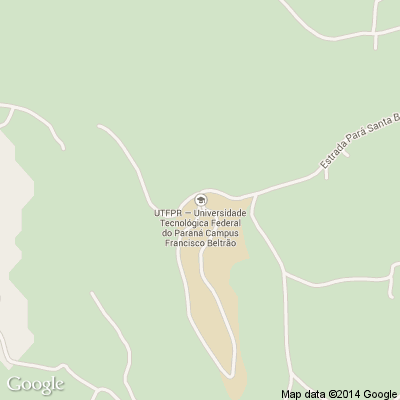
 

Figura 2.4. Figura com parâmetro zoom com valor 15 e 17

Parâmetro de Tipo de mapa (maptype)

O Google Static Maps permite a geração de imagens de mapas estáticos com vários tipos. Podemos ter a visão de satélite, estradas, vegetação, terreno, relevo e mapas combinados entre visão de satélite e de estradas. Os tipos mais utilizados são o roadmap, satellite, terrain e hybrid, apresentados na **Figura 2.5**.

|  |  |
| --- | --- |
| staticmap (400×400)  roadmap | staticmap (400×400)  satellite |
| staticmap (400×400)  terrain | staticmap (400×400)  hybrid |

Figura 2.5. Tipos de mapas mais utilizados

Suas principais características são:

* roadmap: caso não especificado, esse será o tipo de mapa para o resultado;
* satellite: define o tipo de mapa como imagem de satellite;
* terrain: define o tipo da imagem para apresentar o relevo físico, mostrando terreno e a vegetação;
* hybrid: define a combinação de imagens entre a visão de satélite e a visão de ruas e estradas.

Para alternar entre um mapa e outro, basta mudar o parâmetro na URL, conforme apresentado no campo em negrito:

http://maps.googleapis.com/maps/api/staticmap?center=Dois%20Vizinhos&zoom=15&size=400x400&maptype=satellite&sensor=false

Parâmetro de Marcadores (markers)

Os marcadores são aqueles ícones, pingo invertido, que apontam para locais. No caso do Google Static Maps é possível inserir vários marcadores a fim de indicar posições geográficas distintas no mapa.

Os marcadores podem ser estilizados, alternando seu tamanho, sua cor e inserindo ícones ou caractere identificador. A **Figura 2.6** representa a utilização de dois marcadores.

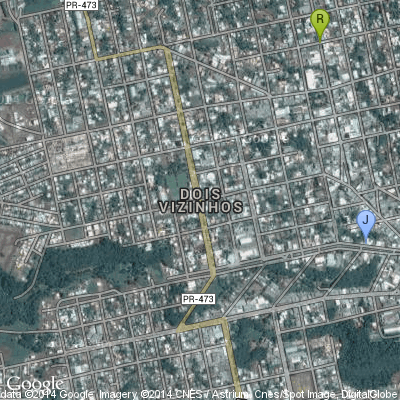


Figura 2.6. Marcadores estilizados.

Para adicionar ao mapa os marcadores, os parâmetros de consulta devem ser alterados, conforme a URL que segue:

http://maps.googleapis.com/maps/api/staticmap?center=Dois%20Vizinhos&zoom=15&size=400x400&maptype=hybrid&markers=color:blue|label:J|Avenida%20Prefeito%20Dedi%20Barrichelo%20Montagner%20190,%20Dois%20Vizinhos&markers=color:green|label:R|Rua%207%20de%20setembro,%20Dois%20Vizinhos&sensor=false

Observe que os dois marcadores iniciam com o parâmetro color, que concatena com o próximo parâmetro (neste exemplo o label) através do caractere | e assim segue com o parâmetro que define o local do marcador. Para adicionar um outro parâmetro, basta usar o caractere & e repetir os parâmetros, personalizando-o conforme a necessidade.

Parâmetro de Caminhos (path)

Esse parâmetro define um caminho a ser sobreposto na imagem, conectado por um ou mais locais. Esse parâmetro de caminho não traça rotas, podendo o caminho ser livremente apresentado sobre o mapa. Para representar uma rota é necessário ter o conhecimento dos locais e incluir todos eles detalhadamente no parâmetro path.

Os caminhos podem ser estilizados, alternando a espessura, cor e preenchimento. A **Figura 2.7** apresenta um exemplo de utilização de caminho.



Figura 2.7. Caminho ligando dois pontos no mapa.

Para utilizar o recurso do caminho, este parâmetro deve ser formatado na requisição conforme exemplo que segue:

http://maps.googleapis.com/maps/api/staticmap?center=Dois%20Vizinhos&zoom=15&size=400x400&maptype=hybrid&path=color:blue|weight:10|Avenida%20Prefeito%20Dedi%20Barrichelo%20Montagner%20190,%20Dois%20Vizinhos|Rua%207%20de%20setembro,%20Dois%20Vizinhos&sensor=false

Observe que após o parâmetro path, é definida a cor do caminho, seguido pelo caractere | e dos demais campos, como espessura da linha, local da origem da linha e o local para o fim da linha.

Estudo de caso do uso do Google StaticMaps no Android

Para exemplificar o uso da API na plataforma Android, vamos desenvolver um aplicativo com a interface apresentada na Figura 2.8.

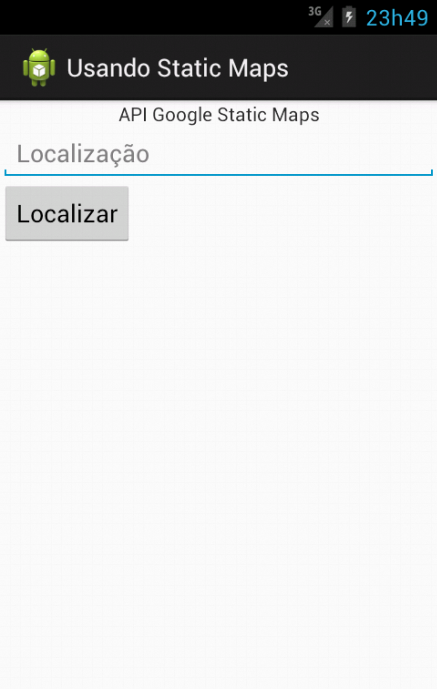


Figura 2.8. Interface do aplicativo.

Nesta aplicação, o usuário poderá informar em um campo EditText a localização a ser pesquisada no mapa, podendo esta ser um texto ou uma posição geográfica (longitude e latitude). Na sequência, o usuário selecionará o botão Localizar, o qual apresentará em um componente ImageView, abaixo do botão, a imagem correspondente ao ponto localizado conforme **Figura 2.9**.

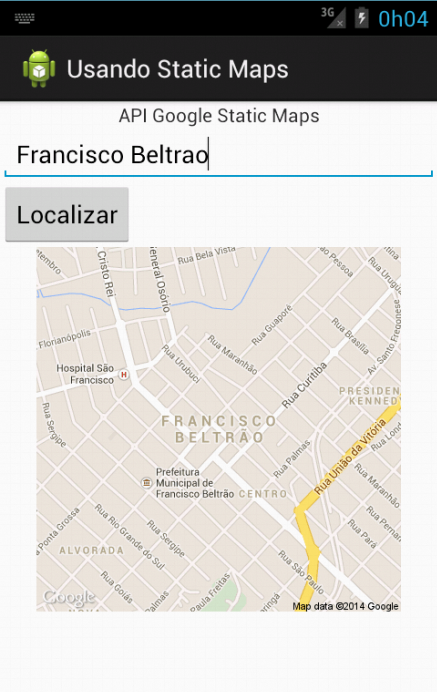


Figura 2.9. Apresentando o mapa da localização.

Para este exemplo, deve-se criar um aplicativo Android tradicional, utilizando a IDE de sua preferência. Este aplicativo deve ter uma classe principal – PrincipalActivity.java, bem como uma interface gráfica - activity\_principal.xml. Além disso, deve-se alterar o arquivo de manifesto – AndroidManifest.xml, adicionando à aplicação a permissão de acesso a Internet, uma vez que os mapas serão baixados em tempo de execução.

O código a seguir mostra o arquivo de manifesto da aplicação referente à permissão de acesso à internet, a qual é essencial para a aplicação funcionar.

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

Na sequência, deve-se codificar a interface gráfica do aplicativo. Para isso, altere o arquivo activity\_principal.xml conforme **Listagem 2.1**.

Listagem 2.1. activity\_principal.xml – Interface gráfica do aplicativo.

01. <LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"

02. xmlns:tools="http://schemas.android.com/tools"

03. android:layout\_width="match\_parent"

04. android:layout\_height="match\_parent"

05. android:orientation="vertical"

06. tools:context=".PrincipalActivity" >

07.

08. <TextView

09. android:layout\_width="match\_parent"

10. android:layout\_height="wrap\_content"

11. android:gravity="center"

12. android:text="API Google Static Maps" />

13.

14. <EditText

15. android:id="@+id/etCenter"

16. android:layout\_width="match\_parent"

17. android:layout\_height="wrap\_content"

18. android:ems="10"

19. android:hint="Localização" />

20.

21. <Button

22. android:layout\_width="wrap\_content"

23. android:layout\_height="wrap\_content"

24. android:onClick="btBuscarOnClick"

25. android:text="Localizar" />

26.

27. <LinearLayout

28. android:layout\_width="match\_parent"

29. android:layout\_height="match\_parent"

30. android:gravity="top|center"

31. android:orientation="vertical" >

32.

33. <ImageView

34. android:id="@+id/ivMapa"

35. android:layout\_width="wrap\_content"

36. android:layout\_height="wrap\_content"

37. android:background="#cacaca"

38. android:contentDescription="Mapa Google" />

39.

40. </LinearLayout>

41.

42. </LinearLayout>

Esta tela é composta por componentes visuais simples. O primeiro componente da tela é um TextView (linha 08), com o objetivo de apresentar uma informação textual na tela do dispositivo. Na sequência, temos um EditText (linha 14) para digitação da localização. Este componente possui nome etCenter. O próximo componente será o botão Localizar (linha 21). Ao ser pressionado, este chamará o método btBuscarOnClick (linha 24) que deve ser codificado com este nome na ActivityPrincipal.

Por fim, temos o componente ImageView que receberá a imagem do mapa baixada pela API Google StaticMaps. Este componente possui nome ivMapa.

O passo seguinte é a codificação da ActivityPrincipal.java. Esta tem a função de tratar o clique de botão, o qual recuperará o mapa referente à localização digitada, apresentando-o na tela via componente ImageView conforme apresentado na **Listagem 2.2**.

Listagem 2.2.PrincipalActivity – Classe da activity principal.

01. package com.br.livro.usandostaticmaps;

02.

03. import java.io.InputStream;

04. import java.io.UnsupportedEncodingException;

05. import java.net.URLEncoder;

06.

07. import org.apache.http.client.HttpClient;

08. import org.apache.http.client.methods.HttpGet;

09. import org.apache.http.impl.client.DefaultHttpClient;

10.

11. import android.app.Activity;

12. import android.graphics.Bitmap;

13. import android.graphics.BitmapFactory;

14. import android.os.AsyncTask;

15. import android.os.Bundle;

16. import android.view.View;

17. import android.widget.EditText;

18. import android.widget.ImageView;

19.

20. public class PrincipalActivity extends Activity {

21.

22. private EditText etCenter;

23.

24. private ImageView ivMapa;

25.

26. @Override

27. protected void onCreate(Bundle savedInstanceState) {

28. super.onCreate(savedInstanceState);

29. setContentView(R.layout.activity\_principal);

30.

31. etCenter = (EditText) findViewById(R.id.etCenter);

32. ivMapa = (ImageView) findViewById(R.id.ivMapa);

33. }

34.

35. public void btBuscarOnClick(View view) {

36. HttpThread httpThread = new HttpThread();

37.

38. httpThread.execute(etCenter.getText().toString());

39. }

40.

41. class HttpThread extends AsyncTask<String, Void, Bitmap> {

42.

43. @Override

44. protected Bitmap doInBackground(String... params) {

45. Bitmap bmp = null;

46. try {

47. String url = "http://maps.googleapis.com/maps/api/staticmap?center="

48. + URLEncoder.encode(params[0], "UTF-8")

49. + "&zoom=15&size=400x400&sensor=false";

50. HttpClient httpclient = new DefaultHttpClient();

51. HttpGet request = new HttpGet(url);

52. InputStream in = httpclient.execute(request).getEntity()

53. .getContent();

54. bmp = BitmapFactory.decodeStream(in);

55. in.close();

56. } catch (UnsupportedEncodingException e) {

57. e.printStackTrace();

58. } catch (Exception e) {

59. e.printStackTrace();

60. }

61. return bmp;

62. }

63.

64. @Override

65. protected void onPostExecute(Bitmap bmp) {

66. ivMapa.setImageBitmap(bmp);

67. }

68.

69. }

70.}

As primeiras linhas do código (linha 1 a linha 18) são responsáveis pela declaração do pacote e pela importação das classes. Na linha 20 é declarada a classe principal e nas linhas 22 e 24 são declarados os dois componentes visuais: EditText para digitação do local e ImageView para apresentação do mapa.

No método onCreate (linha 27) são realizadas as rotinas de inicialização do aplicativo (linhas 28 e 29), assim como a recuperação dos componentes visuais (linhas 31 e 32).

Na linha 35 é codificado o método btBuscaOnClick(), o qual tratará a recuperação dos dados da API Google Static Maps. Como se trata de uma rotina que fará acesso à Internet, é necessário colocar esta lógica em uma Thread separada. Para isso, na linha 36 é instanciada a Thread HttpThread, esta codificada a partir da linha 41, assim como executada ela na linha 38 passando por parâmetro a localização digitada no componente EditText.

A classe HttpThread é uma subclasse de AsyncTask, sendo sobrescritos nesta classe dois métodos: doInBackground() e onPostExecute(). O método doInBackground(linha 44) é responsável pelo acesso à API do Google StaticMaps. Neste é declarado um componente Bitmap (linha 45) o qual armazenará o conteúdo recuperado da API.

Após isso, na linha 47, é definida a url para conexão com a API. Observem que este recebe por parâmetro o texto digitado no EditText (linha 48) já codificado no formato correto de URL através da classe URLEncoder.

Em seguida, um objeto do tipo HttpClient é instanciado (linha 50), assim como um objeto HttpGet (linha 51), responsável pela comunicação utilizando o método GET do HTTP. A resposta deste acesso será uma imagem, e sendo um dado binário, é necessário armazená-lo inicialmente em um objeto do tipo InputStream (linha 52), sendo este convertido na linha 53 em um componente Bitmap.

Por fim, no método onPostExecute (linha 65), este objeto Bitmap é adicionado ao componente ImageView como apresentado na linha 66.

Desta forma, o aplicativo está pronto para ser executado. O mesmo pode ser executado no emulador ou em um dispositivo real. O único cuidado é observar se este emulador/dispositivo está com acesso à Internet, pois muitas vezes servidores de proxy ou problemas de configuração impedem que estes devices acessem a Internet, o que prejudica a execução do aplicativo desenvolvido.

**Concluindo....**

Este artigo apresentou conceitos de estrutura e funcionamento da API do Google Static Maps, que pode ser utilizado em qualquer plataforma que disponha de recursos de recuperação de informações por requisições HTTP.

Apesar da facilidade de integração e uso da mesma, a API oferece excelentes resultados. Esse recurso é muito útil para quem deseja anexar imagens de mapas e localizações em seus sistemas, principalmente pela vantagem da velocidade que as imagens são retornadas já que as mesmas são estáticas.

Sua utilização em aplicativos móveis é estratégica, pois dependendo da estrutura do aplicativo desenvolvido, estes mapas podem ser armazenados localmente, sendo recuperados da rede apenas uma vez, o que permite uma navegação no mapa off-line eliminando trafego de rede, o que torna o aplicativo mais rápido e econômico para o usuário.