
PROGRAMAÇÃO COM ANDROID NA ÁREA DA ENGENHARIA CARTOGRÁFICA: ALGUNS EXEMPLOS

GLAUCO LEITE DE FREITAS
ANA LÚCIA BEZERRA CANDEIAS

Universidade Federal de Pernambuco - UFPE
Centro de Tecnologia e Geociências - CTG
Departamento de Engenharia Cartográfica, Recife, PE
Glauco_freitas@hotmail.com, analucia@ufpe.br

RESUMO - Este trabalho mostra exemplos de como desenvolver a programação de Android para a área da Engenharia Cartográfica. Os resultados fazem parte de um estágio desenvolvido no 1º semestre de 2014 no Departamento de Engenharia Cartográfica da UFPE (Universidade Federal de Pernambuco) com a segunda autora deste artigo.

ABSTRACT – This paper shows some examples of Android development programming for Cartographic Engineering area. The results are part of a developed stage into the 1st semester of 2014 at the Cartographic Engineering Department of UFPE (Federal University of Pernambuco) with the second author of this article.

1 INTRODUÇÃO

Este trabalho apresenta exemplos de como programar em Android com aplicações voltadas para a área de da Engenharia Cartográfica. Ele é o resultado de um estágio de graduação no 1º semestre de 2014, no Departamento de Engenharia Cartográfica da UFPE (Universidade Federal de Pernambuco).

Com a programação em Android podem-se criar aplicativos para dispositivos móveis tais como *smartphones* e *tablets*. Esta plataforma foi desenvolvida pela Google™, e posteriormente pela OHA (Open Handset Alliance).

Uma das vantagens de desenvolver aplicativos para o Android é que as ferramentas e programas necessários são gratuitos e disponíveis para todos os sistemas operacionais (OS X, Windows™ e Linux).

A programação em Android é baseada em Java. Entre as ferramentas necessárias dessa programação estão: o JDK (Java Development Kit) e o Android SDK que inclui as bibliotecas e várias ferramentas, como o emulador, por exemplo, o Eclipse e o ADT *Plugin* - a IDE é utilizada e o *plugin* para facilitar o desenvolvimento de programação para Android. O uso do eclipse e do *plugin* não é obrigatório, mas ao utiliza-lo, diminui-se as tarefas repetitivas.

Um ponto importante é que na programação em Android não é necessário se ter um aparelho com esta tecnologia. O SDK inclui um emulador que permite rodar as aplicações desenvolvidas em Android. Recomenda-se porém, que na finalização da programação de um aplicativo gerado utilizando emulador, rode-se o programa em um aparelho com Android para os últimos testes, antes de liberar o aplicativo, por exemplo, para o Google Play (loja de aplicativos do Android).

Em K19 Treinamentos (2014) e Réquia (2013) têm-se um detalhamento da programação nesta plataforma.

2 METODOLOGIA

A metodologia do trabalho consistiu de pesquisa bibliografia da área, instalação dos programas, e geração de exemplos como aplicativos. Para que isso pudesse ser realizado foi usado um *notebook*, o programa Android Studio Beta v0.8.0 e o BlueStacks 0.8.11, esse último é um programa que emula o sistema operacional Android para o sistema operacional do Windows.

A idéia do trabalho é a elaboração de aplicativos exemplos que mostrem uma programação em Android: leitura de dados, processamentos e apresentação dos resultados.

De forma geral, utilizou-se a leitura dos dados a partir de um arquivo em .txt. Este arquivo serviu para a obtenção das coordenadas dos pontos.

No processamento dos dados, utilizou-se como 1º exemplo: a adição dos pontos. Isto foi feito diretamente no aplicativo e também utilizando arquivo com formato .txt.

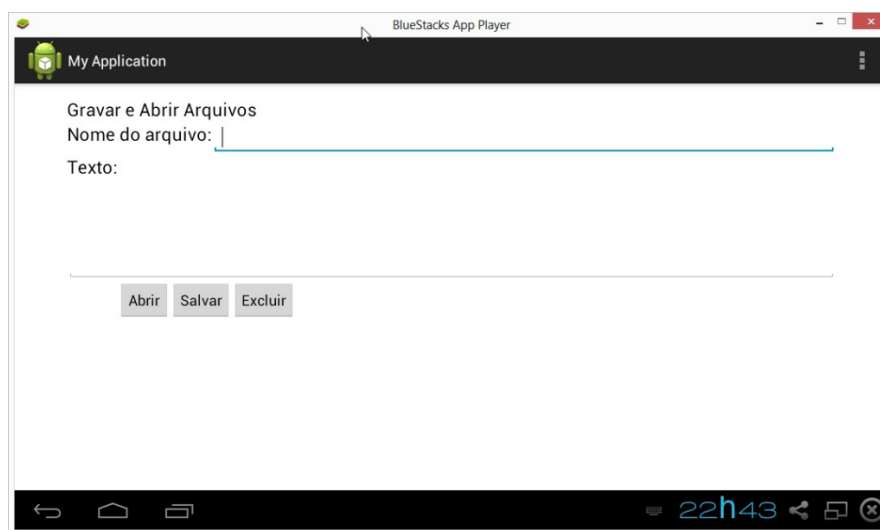
A proposta seguinte, 2º exemplo, foi desenvolver algumas operações utilizadas na cartografia tais como: a) transformação de coordenadas de SAD69 para WGS84, b) Projeção de graus decimais para UTM, c) mudança de graus, minutos e segundos para graus decimais, dentre outros. Finalmente como 3º exemplo a proposta era a visualização de coordenadas no Google Maps e salvar os resultados em arquivo também em .txt.

No desenvolvimento destes três exemplos observa-se as seguintes etapas: a) aplicativo 1: leitura, gravação e exclusão de arquivos. b) aplicativo 2: cálculos para a transformação, projeção e conversão. c) aplicativo 3: utilização de pontos coordenados calculados ou obtidos diretamente do .txt apresentados no Google Maps.

2.1 LER, GRAVAR E EXCLUIR ARQUIVOS

Esta seção mostra como ler, gravar e excluir arquivos em Android. Portanto, deve-se primeiramente fazer com que o programa faça a leitura das coordenadas dos pontos de um arquivo obtido no formato “txt” e também que grave o arquivo com o mesmo formato “txt”. A Figura 1 apresenta uma aplicação que abre, salva e exclui um arquivo de dados.

Figura 1. Janela com aplicação em Android que Abre, Salva e Exclui um arquivo em txt .



A Figura 2 apresenta o programa em Android que mostra a quantidade de colunas que será utilizada para a aquisição das coordenadas em X, Y ou E, N ou Latitude, Longitude ou na Gravação da mesma (ex://FileOutputStream arquivoGravar = openFileOutput (etNomeArquivo.getText().toString(), MODE_PRIVATE);):

Figura 2. Programa em Android que Abre, Salva e Exclui um arquivo em txt .

```
package com.projeto.glauco.simgeo2014;

import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;

import java.io.FileNotFoundException;
import java.io.FileOutputStream;

public class MyActivity extends Activity {
    private final int DELAYSEGUNDOS = 4000;
```

```

        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.arquivotexto);
            Button btGravar = (Button) findViewById(R.id.btGravar);
            Button btLer = (Button) findViewById(R.id.btLer);
            Button btExcluir = (Button) findViewById(R.id.btExcluir);

            btGravar.setOnClickListener(new OnClickListener() {
                @Override
                public void onClick(View v) {

            try {
                EditText etNomeArquivo = (EditText) findViewById(R.id.etNomeArquivo);
                EditText etConteudoArquivo = (EditText) findViewById(R.id.etConteudoArquivos);
                //acrescenta ao arquivo
                FileOutputStream arquivoGravar = openFileOutput (etNomeArquivo.getText().toString(), MODE_APPEND);
                //substitui (grava por cima)
                //FileOutputStream arquivoGravar = openFileOutput (etNomeArquivo.getText().toString(),
                MODE_PRIVATE);
                String conteudoArquivo = etConteudoArquivo.getText().toString();
                arquivoGravar.write(conteudoArquivo.getBytes());
                arquivoGravar.close();
                mensagemExibir("Arquivo gravado", "Gravado com sucesso!")
            }
            catch (FileNotFoundException erro) {
                mensagemExibir("Arquivo não encontrado", ""+erro);
            }
            catch (IOException erro) {
                mensagemExibir ("Erro de entrada e Saída", ""erro);
            }
            }
        });
    }

            btLer.setOnClickListener(new OnClickListener() {

        @Override
        public void onClick(View v) {
            try {
                EditText etNomeArquivo = (EditText) findViewById(R.id.etNomeArquivo);
                EditText etConteudoArquivo = (EditText) findViewById(R.id.etConteudoArquivos);
                //acrescenta ao arquivo
                FileOutputStream arquivoGravar = openFileOutput (etNomeArquivo.getText().toString(), MODE_APPEND);
                //substitui (grava por cima)
                //FileOutputStream arquivoGravar = openFileOutput (etNomeArquivo.getText().toString(),
                MODE_PRIVATE);
                String conteudoArquivo = etConteudoArquivo.getText().toString();
                arquivoGravar.write(conteudoArquivo.getBytes());
                arquivoGravar.close();
                mensagemExibir("Arquivo gravado", "Gravado com sucesso!");
            }
            catch (FileNotFoundException erro) {
                mensagemExibir("Arquivo não encontrado", ""+erro);
            }
            catch (IOException erro) {
                mensagemExibir ("Erro de entrada e Saída", ""erro);
            }
        }
    }

```

```

    }
});

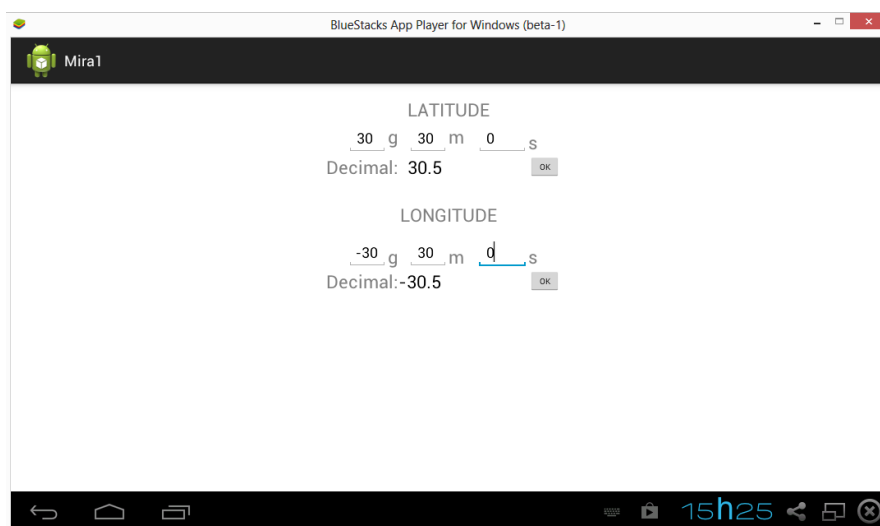
public void mensagemExibir(String titulo, String texto){
    AlertDialog.Builder mensagem = new AlertDialog.Builder (LeGravaExcluirArquivos.this);
    mensagem.setTitle(titulo);
    mensagem.setMessage(texto);
    mensagem.setNeutralButton("OK",null);
    mensagem.show();
}
}

```

2.2 UTILIZANDO O SISTEMA ANDROID PARA A EXECUÇÃO DE FÓRMULAS

Nesta etapa foi criado um exemplo baseado em Transformação de Graus Sexagesimais em Graus Decimais. A Figura 3 mostra a Janela com aplicação dessa Transformação. A entrada é latitude e longitude e a saída é o grau decimal. Este exemplo pode servir de base para se desenvolver outros programas mais sofisticados.

Figura 3. Janela com aplicação de Transformação de Graus Sexagesimais para Graus Decimais



É importante observar que na programação, todas as variáveis devem ser declaradas e também se existe mudança de sinais (negativo e positivo), se o valor é em graus decimais ou se este valor é do tipo inteiro. Isto deve ser declarado na criação do textView. A seguir tem-se isto mostrado em dois exemplos:

Exemplo1: textView sinal – mudança de sinal se é positivo ou negativo.

Exemplo2: textView decimal – para que o resultado seja em graus decimais, dentre outros exemplos.

Figura 4. Programa em Android para Transformação de Graus Sexagesimais para Graus Decimais

```

public class MainActivity extends ActionBarActivity {

    public double LATDEC_G, LATDEC_M, LATDEC_S, LONDEC_G, LONDEC_M, LONDEC_S, LONDEC,
    LATDEC;
    public EditText qLatG,qLatM,qLatS,qLonG,qLonM,qLonS;
    public TextView DEC_LAT, DEC_LON;
}

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    qLatG = (EditText) findViewById(R.id.editText);
    qLatM = (EditText) findViewById(R.id.editText2);
    qLatS = (EditText) findViewById(R.id.editText7);
    qLonG = (EditText) findViewById(R.id.editText4);
    qLonM = (EditText) findViewById(R.id.editText5);
    qLonS = (EditText) findViewById(R.id.editText3);
    DEC_LAT = (TextView) findViewById(R.id.textView9);
    DEC_LON = (TextView) findViewById(R.id.textView12);
}

public void calcular (View view) {
    LATDEC_G = Double.parseDouble(qLatG.getText().toString());
    LATDEC_M = Double.parseDouble(qLatM.getText().toString());
    LATDEC_S = Double.parseDouble(qLatS.getText().toString());
    if(LATDEC_G < 0)
        LATDEC=(LATDEC_G) - (LATDEC_M/60) - (LATDEC_S/3600);
    if(LATDEC_G >= 0)
        LATDEC=(LATDEC_G) + (LATDEC_M/60) + (LATDEC_S/3600);
    DEC_LAT.setText(Double.toString(LATDEC));
}

public void calcular2 (View view) {
    LONDEC_G = Double.parseDouble(qLonG.getText().toString());
    LONDEC_M = Double.parseDouble(qLonM.getText().toString());
    LONDEC_S = Double.parseDouble(qLonS.getText().toString());
    if (LONDEC_G < 0)
        LONDEC=(LONDEC_G) - (LONDEC_M/60) - (LONDEC_S/3600);
    if (LONDEC_G >= 0)
        LONDEC=(LONDEC_G) + (LONDEC_M/60) + (LONDEC_S/3600);
    DEC_LON.setText(Double.toString(LONDEC));
}
}

```

2.3 CAMINHOS PARA UTILIZAÇÃO DO GOOGLE MAPS

Para a criação de um aplicativo que visualize coordenadas em Google Maps, é necessário informar a versão do sistema operacional e que neste sistema possa utilizar o Google APIS, pois com ele tem-se a disponibilidade da biblioteca do Google.

No desenvolvimento da programação utilizando GOOGLE MAPS é necessário se obter uma chave da seguinte forma:

i) Deve-se ir às variáveis de ambiente e colocar o caminho do java nos path da variável:
No meu computador este era o caminho “C:\Program Files\Java\jdk1.8.0_05\bin”;

ii) Em seguida deve ir no cmd do Windows e digitar o seguinte comando:

keytool -list -alias androiddebug -keystore “C:/Users/<user>/<pc>/android/debug.keystore” -storepass android -keypass android

Caso haja algum erro do tipo: “erro de keytool: java.lang ...”. A solução é alterar o idioma do windows para Inglês (Estados Unidos).

iii) Logo após ser instalado o idioma, caso não tenha mudado após reiniciar seu computador, vá no Iniciar do windows e na caixa de busca de programas e arquivos, digite: language;

iv) Após alterar o idioma, abra novamente o cmd e coloque apenas até esta parte: `keytool -list -alias androiddebug -keystore "C:/Users/<user>/<pc>/.android/debug.keystore"`

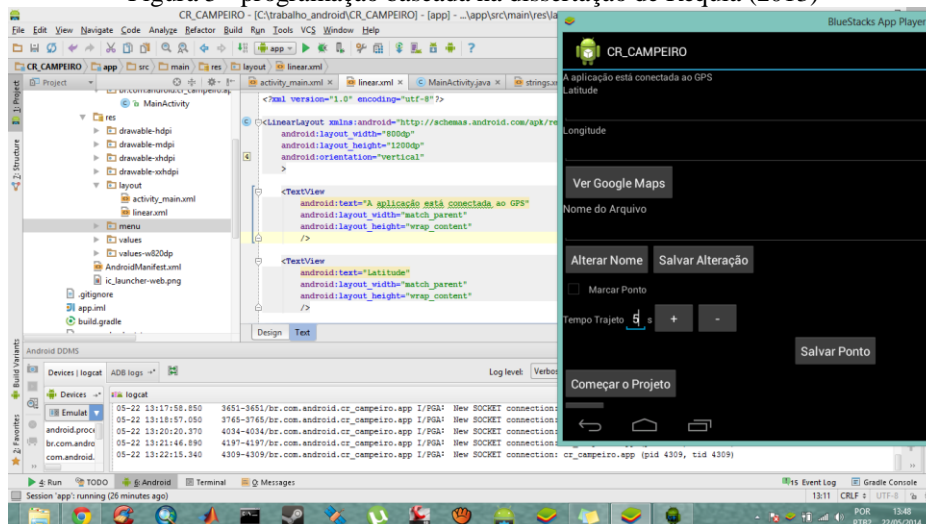
v) Ele irá pedir a senha, e você coloca: android. Pronto foi gerada a sua chave MD5 do certificado digital, em seguida acesse <http://code.google.com/intl/pt-BR/android/maps-api-signup.html> coloque sua chave MD5 e pressione o botão "Generate API Key" que apresentará o resultado com sua chave.

Depois de seguir as etapas anteriores é só trabalhar utilizando a biblioteca Android e gerar a programação utilizando também os recursos do GOOGLE MAPS

3 EXEMPLO DE PROGRAMAÇÃO EM ANDROID MAIS ELABORADO

No desenvolvimento do estágio. Foi estudado o conteúdo visto em Réquia (2013) onde é apresentado aplicativos para agricultura de precisão usando Android. A Figura 5 mostra a janela de saída que desenvolvida em seu trabalho.

Figura 5 - programação baseada na dissertação de Réquia (2013)



4 CONSIDERAÇÕES FINAIS

Este trabalho apresentou os resultados dos passos para programação para Android. Este tipo de plataforma pode auxiliar na Engenharia Cartográfica no desenvolvimento de aplicativos desta área, pois possibilita a elaboração de diversas atividades no campo e no escritório. Ele é o resultado de um estágio de graduação no 1º semestre de 2014, no Departamento de Engenharia Cartográfica da UFPE (Universidade Federal de Pernambuco). Os exemplos são simples mas despertam ao leitor alguns caminhos a serem trilhados nesta linha de programação.

Com a programação em Android podem-se criar aplicativos para dispositivos móveis tais como *smartphones* e *tablets*. Uma das vantagens de desenvolver aplicativos para o Android é que as ferramentas e programas necessários são gratuitos e disponíveis para todos os sistemas operacionais (OS X, Windows™ e Linux). Para o desenvolvimento existem emuladores. Isto facilita a programação e mesmo não tendo um equipamento em Android é possível desenvolver ferramentas que posteriormente podem ser utilizadas em *smartphones* e *tablets*.

REFERÊNCIAS

Réquia, H. G. **Desenvolvimento de Aplicativos CR Campeiro Móble - Caso de Teste: Sistema Operacional Android**. Dissertação (Mestrado) – Universidade Federal de Santa Maria, Colégio Politécnico da UFSM, Programa de Pós-Graduação em Agricultura de Precisão, RS, 2013.

K19 Treinamentos. Desenvolvimento Mobile com Android. URL: <http://www.k19.com.br/downloads/apostilas>. Acessado: em 22 de julho de 2014.