

Atividade PDM

Aluno: Sérgio Diniz Correia

## Services e Broadcastreceivers

Para o desenvolvimento desta atividade, foi seguido os passos descrito no documento proposto no grupo da disciplina.

A ferramenta de desenvolvimento utilizado foi a IDE Eclipse com a API Android 8, referente ao Android 2.2.

## Projeto Services

Inicialmente criamos um projeto com as seguintes configurações:

1. Nome do projeto **Services**.
2. Nome do pacote **br.com.k19.android.cap08**
3. O nome da activity deve ser mantido o padrão, **MainActivity**.

Na pasta **res/layouts**, criamos um arquivo do tipo Android xml com o nome **main** e introduzimos o seguinte código:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical" >
6
7 <Button
8     android:id="@+id/start_button"
9     android:layout_width="match_parent"
10    android:layout_height="wrap_content"
11    android:text="@string/start_downloads"
12    android:layout_gravity="center" />
13
14 </LinearLayout>
```

Figura 1 - Código no Arquivo main.

Vamos até o arquivo **res/values/strings.xml** e inserimos os valores:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3
4     <string name="app_name">Services</string>
5     <string name="hello_world">Hello world!</string>
6     <string name="action_settings">Settings</string>
7     <string name="menu_settings">Settings</string>
8     <string name="title_activity_main">MainActivity</string>
9     <string name="start_downloads">Iniciar downloads</string>
10    <string name="download_error">Erro ao fazer download.</string>
11    <string name="download_success">Download feito com sucesso %1$s.</string>
12
13 </resources>
```

**Figura 2 - Código do arquivo strings.xml**

Agora, criamos uma classe Java na pasta **src/br.com.k19.android.cap08** com o nome **DownloadService** com o seguinte código:

```

1 package br.com.k19.android.cap08;
2
3 import java.io.File;
4 import java.io.FileOutputStream;
5 import java.io.IOException;
6 import java.io.InputStream;
7 import java.io.InputStreamReader;
8 import java.net.URL;
9 import android.app.Activity;
10 import android.app.IntentService;
11 import android.content.Intent;
12 import android.net.Uri;
13 import android.os.Bundle;
14 import android.os.Environment;
15 import android.os.Message;
16 import android.os.Messenger;
17 import android.util.Log;
18
19 public class DownloadService extends IntentService {
20     private int result = Activity.RESULT_CANCELED;
21
22     public DownloadService() {
23         super(" DownloadService ");
24     }
25
26     @Override
27     protected void onHandleIntent(Intent intent) {
28         Uri data = intent.getData();
29         String urlPath = intent.getStringExtra(" urlPath ");
30         String fileName = data.getPath();
31         File output = new File(Environment.getExternalStorageDirectory(),
32             fileName);
33         if (output.exists()) {
34             output.delete();
35         }
36         InputStream stream = null;
37         FileOutputStream fos = null;
38         try {
39             URL url = new URL(urlPath);
40             stream = url.openConnection().getInputStream();
41             InputStreamReader reader = new InputStreamReader(stream);
42             fos = new FileOutputStream(output.getPath());
43             int next = -1;
44             while ((next = reader.read()) != -1) {
45                 fos.write(next);
46             }
47             result = Activity.RESULT_OK;
48         } catch (Exception e) {
49             e.printStackTrace();
50         } finally {
51             if (stream != null) {
52                 try {
53                     stream.close();
54                 } catch (IOException e) {
55                     e.printStackTrace();
56                 }
57             }
58             if (fos != null) {
59                 try {
60                     fos.close();
61                 } catch (IOException e) {
62                     e.printStackTrace();
63                 }
64             }
65         }
66         Bundle extras = intent.getExtras();
67         if (extras != null) {
68             Messenger messenger = (Messenger) extras.get(" messenger ");
69             Message msg = Message.obtain();
70             msg.arg1 = result;
71             msg.obj = output.getAbsolutePath();
72             try {
73                 messenger.send(msg);
74             } catch (android.os.RemoteException e1) {
75                 Log.e(" DownloadService ", " Erro ao enviar mensagem ", e1);
76             }
77         }
78     }
79 }
80 }

```

Figura 3 - Código Classe DownloadService

Ainda no mesmo pacote, vamos abrir o arquivo **MainActivity.java** e vamos modificar para o seguinte estado:

```
1 package br.com.k19.android.cap08;
2 import android.app.Activity;
3 import android.content.Intent;
4 import android.net.Uri;
5 import android.os.Bundle;
6 import android.os.Handler;
7 import android.os.Message;
8 import android.os.Messenger;
9 import android.support.v7.app.ActionBarActivity;
10 import android.view.Menu;
11 import android.view.MenuItem;
12 import android.view.View;
13 import android.view.View.OnClickListener;
14 import android.widget.Button;
15 import android.widget.Toast;
16 public class MainActivity extends ActionBarActivity {
17
18     private Handler handler = new Handler() {
19
20         public void handleMessage(Message message) {
21             Object path = message.obj;
22             if (message.arg1 == RESULT_OK && path != null) {
23                 Toast.makeText(MainActivity.this,
24                     getString(R.string.download_success, path.toString()),
25                     Toast.LENGTH_LONG).show();
26             } else {
27                 Toast.makeText(MainActivity.this,
28                     getString(R.string.download_error), Toast.LENGTH_LONG)
29                     .show();
30             }
31         };
32     };
33
34     @Override
35     public void onCreate(Bundle savedInstanceState) {
36         super.onCreate(savedInstanceState);
37         setContentView(R.layout.main);
38         Button startButton = (Button) findViewById(R.id.start_button);
39         startButton.setOnClickListener(new OnClickListener() {
40
41             @Override
42             public void onClick(View v) {
43                 Intent intent = new Intent(MainActivity.this,
44                     DownloadService.class);
45                 Messenger messenger = new Messenger(handler);
46                 intent.putExtra(" messenger ", messenger);
47                 intent.setData(Uri.parse(" cursos . html "));
48                 intent.putExtra(" urlPath ", " http :// k19.com .br/ cursos ");
49                 startService(intent);
50             }
51         });
52     };
53
54 }
55
56 @Override
57 public boolean onCreateOptionsMenu(Menu menu) {
58     // Inflate the menu; this adds items to the action bar if it is present.
59     getMenuInflater().inflate(R.menu.main, menu);
60     return true;
61 }
62
63 @Override
64 public boolean onOptionsItemSelected(MenuItem item) {
65     // Handle action bar item clicks here. The action bar will
66     // automatically handle clicks on the Home/Up button, so long
67     // as you specify a parent activity in AndroidManifest.xml.
68     int id = item.getItemId();
69     if (id == R.id.action_settings) {
70         return true;
71     }
72     return super.onOptionsItemSelected(item);
73 }
74 }
75 }
```

Figura 4 - Código MainActivity

Precisamos agora fazer uma configuração no emulador do android para que ele possa guarda o arquivo a ser baixado, para isso vamos abrir o AVD Manager e aditar o emulador:

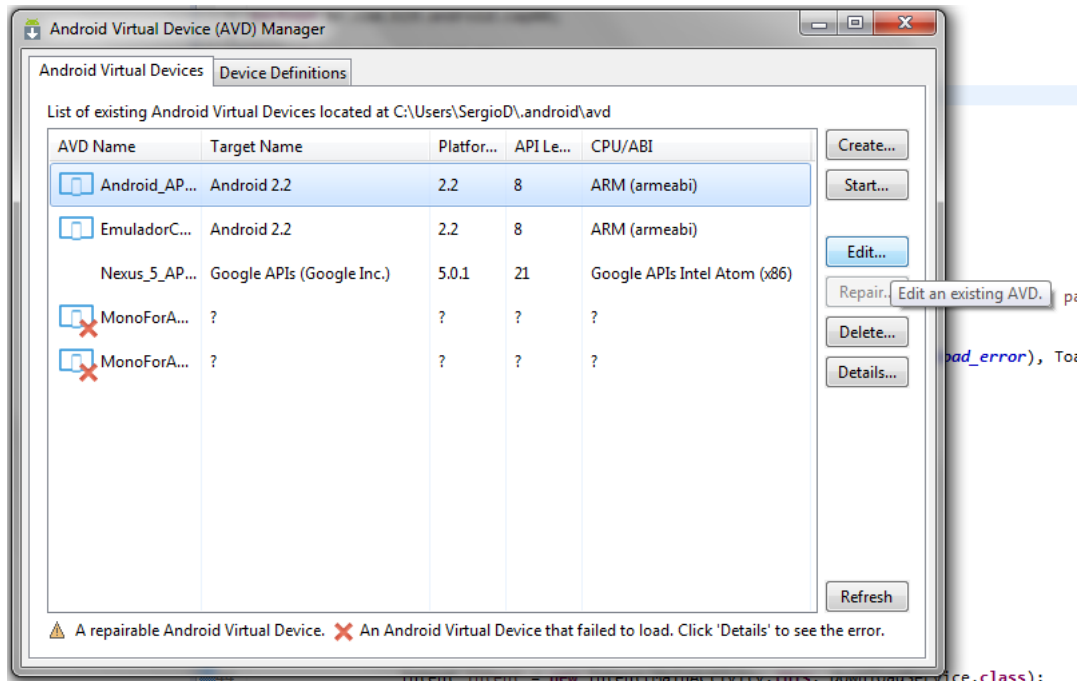


Figura 5 - AVD Manager

Agora em SD Card vamos colocar um valor que seja possível guardo o arquivo a ser baixado:

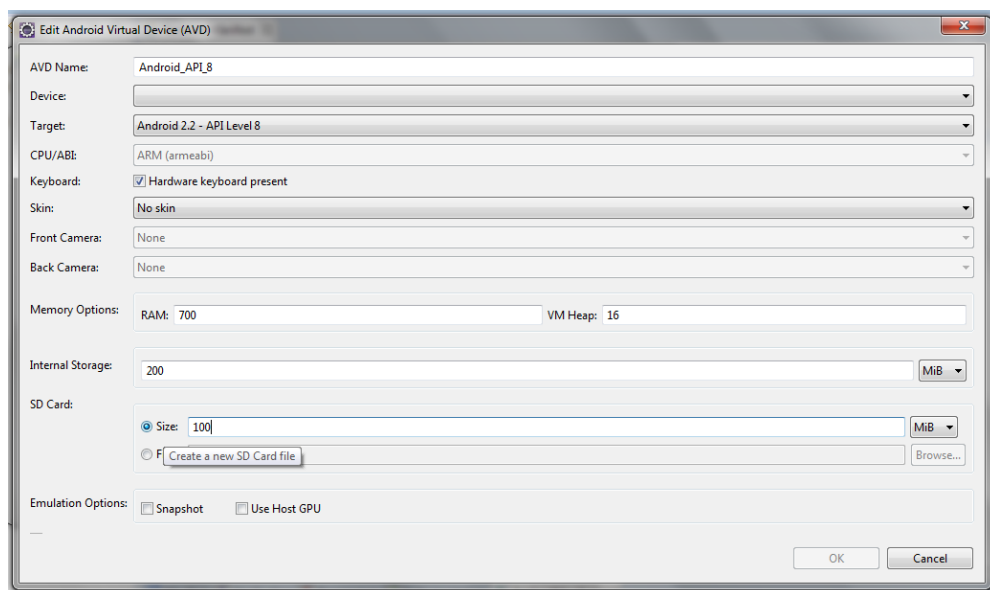


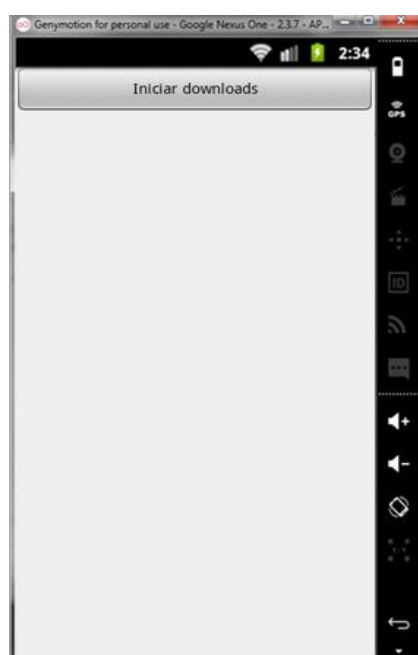
Figura 6 - SD Card

Agora vamos dar permissão para que nossa aplicação possa acessar a internet e gravar o nosso download no SD Card virtual:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="br.com.k19.android.cap08"
4     android:versionCode="1"
5     android:versionName="1.0" >
6
7     <!-- Dando Permissão para Acesso a Internet e Escrita -->
8     <uses-permission android:name="android.permission.INTERNET" />
9     <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
10    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
11
12    <uses-sdk
13        android:minSdkVersion="8"
14        android:targetSdkVersion="21" />
15
16    <application
17        android:allowBackup="true"
18        android:icon="@drawable/ic_launcher"
19        android:label="@string/app_name"
20        android:theme="@style/AppTheme" >
21        <activity
22            android:name=".MainActivity"
23            android:label="@string/app_name" >
24            <intent-filter>
25                <action android:name="android.intent.action.MAIN" />
26
27                <category android:name="android.intent.category.LAUNCHER" />
28            </intent-filter>
29        </activity>
30    </application>
31
32 </manifest>
```

**Figura 7 - Permissão de Internet e Escrita**

Agora tudo pronto, podemos salvar tudo e executar o projeto, será apresentado uma tela com um botão. Clique no botão para realizar o download.



**Figura 8 - Tela Inicial Projeto Services**

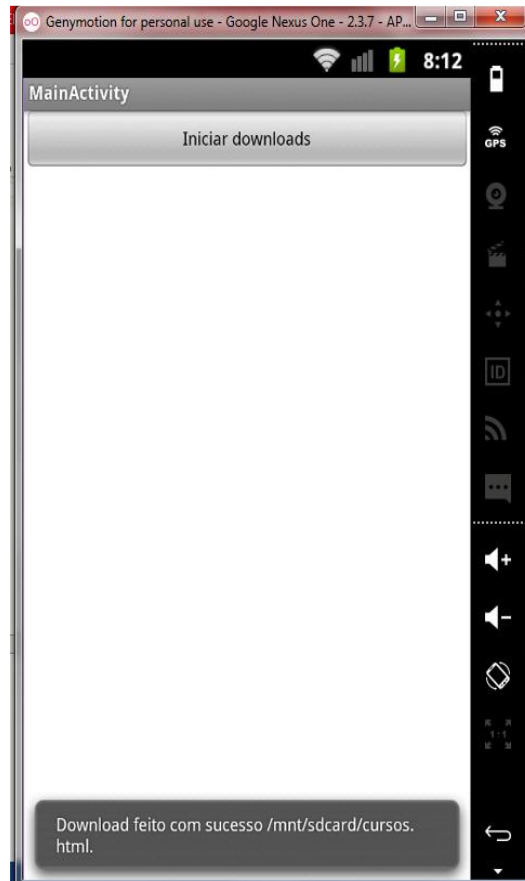


Figura 9 - Realizando Download

## Projeto Reveiver

Inicialmente criamos um projeto com as seguintes configurações:

1. Nome do projeto **Reveiver**.
2. Nome do pacote **br.com.k19.android.cap08\_02**.
3. O nome da activity deve ser mantido o padrão, **MainActivity**.

Agora, criamos uma classe Java na pasta **src/br.com.k19.android.cap08\_02** com o nome **PhoneReceiver** com o seguinte código:

```

1 package br.com.k19.android.cap08_2;
2
3 import android.content.BroadcastReceiver;
4
5
6
7
8
9
10 public class PhoneReceiver extends BroadcastReceiver {
11
12     private static final String TAG = " PhoneReceiver ";
13
14     @Override
15     public void onReceive(Context context, Intent intent) {
16
17         Bundle extras = intent.getExtras();
18         if (extras != null) {
19             String state = extras.getString(TelephonyManager.EXTRA_STATE);
20             Log.w(TAG, state);
21             if (state.equals(TelephonyManager.EXTRA_STATE_RINGING)) {
22                 String phoneNumber = extras
23                     .getString(TelephonyManager.EXTRA_INCOMING_NUMBER);
24                 Log.w(TAG, phoneNumber);
25             }
26         }
27     }
28 }
29

```

Figura 10 - Código classe PhoneReceiver

Vamos editar o arquivo **AndroidManifest.xml**:



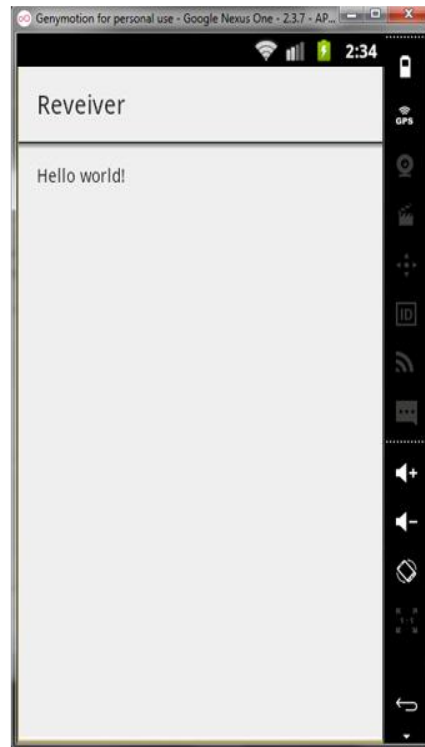
```

1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="br.com.k19.android.cap08_2"
4      android:versionCode="1"
5      android:versionName="1.0" >
6
7      <uses-sdk
8          android:minSdkVersion="8"
9          android:targetSdkVersion="21" />
10
11     <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
12
13     <application
14         android:icon="@drawable/ic_launcher"
15         android:label="@string/app_name"
16         android:theme="@style/AppTheme" >
17         <activity
18             android:name=".MainActivity"
19             android:label="@string/app_name" >
20             <intent-filter>
21                 <action android:name="android.intent.action.MAIN" />
22
23                 <category android:name="android.intent.category.LAUNCHER" />
24             </intent-filter>
25         </activity>
26
27         <receiver android:name=".PhoneReceiver">
28             <intent-filter>
29                 <action android:name="android.intent.action.PHONE_STATE"></action >
30             </intent-filter >
31
32     </receiver >
33
34     </application>
35
36 </manifest>

```

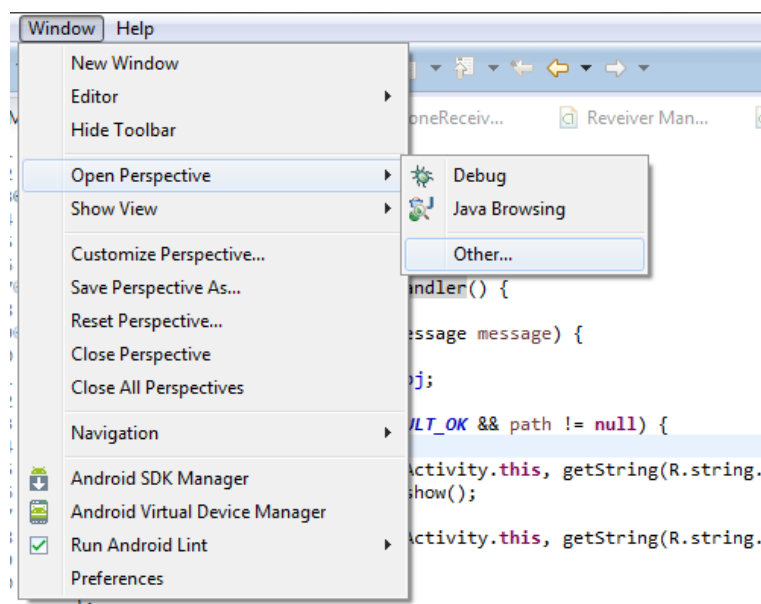
Figura 11 - Arquivo AndroidManifest.xml

Com tudo pronto, executamos nosso projeto Reveiver.



**Figura 12 - Tela Inicial Projeto Reveiver**

Para teste, devemos fazer uma ligação no emulador do android, para fazer isso, vamos mudar a perspectiva para DDMS:



**Figura 13 - DDMS**

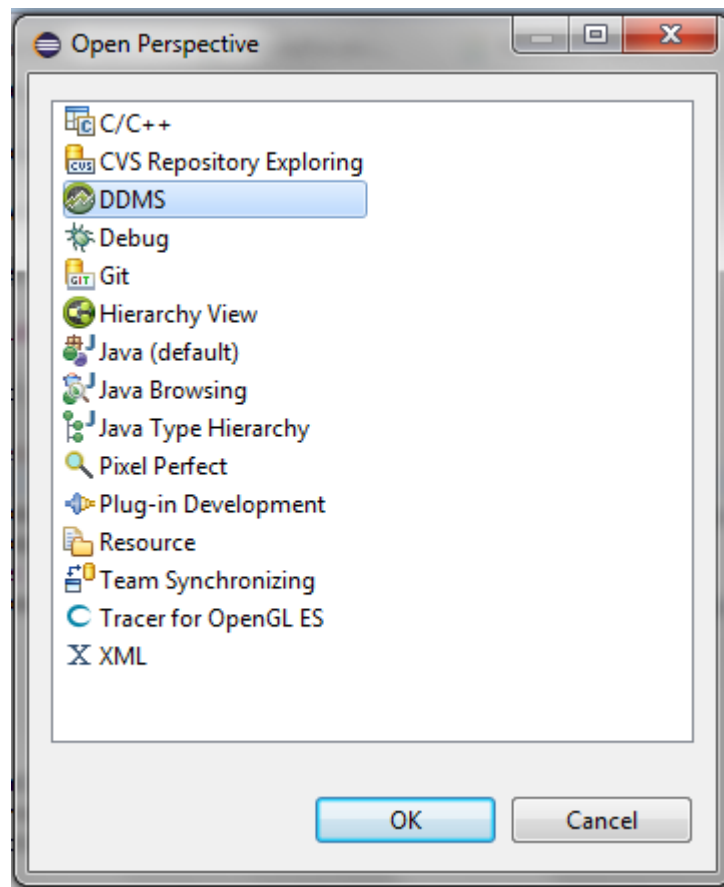


Figura 14 – DDMS

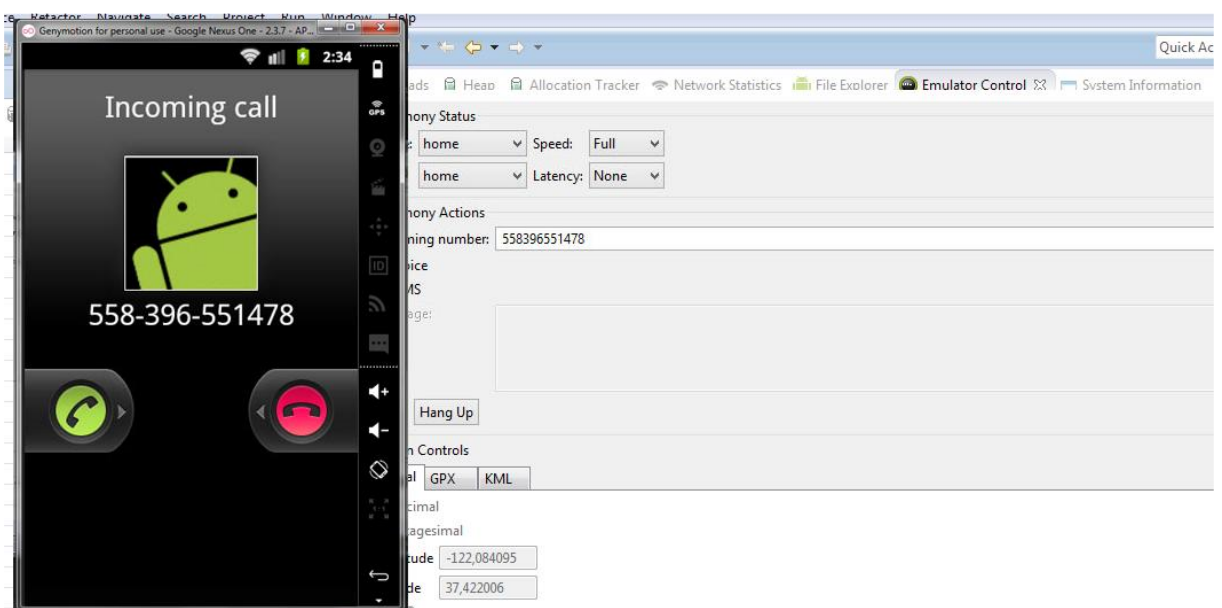


Figura 15 - Testando Ligação