

Prof. Menezes	PROGRAMAÇÃO MOBILE Estudo de Caso	Data 27/09/2016
------------------	---	--------------------

UTILIZANDO A CLASSE *Intent* PARA CHAMAR AS TELAS

“nativas” E USAR OS RECURSOS DO ANDROID

A classe Intent possui inúmeras funções no Android, podendo ser utilizada para abrir novas janelas (associadas a Activities), passar parâmetros, não passar parâmetros ou ainda receber os parâmetros de outras Activities. Porém, um recurso muito interessante da classe Intent é a chamada de telas e recursos nativos do device, como, por exemplo, chamar a tela responsável por ligações, enviar mensagens, ou até mesmo ligar/desligar recursos como, Bluetooth e GPS. Na sequência, são mostrados alguns códigos interessantes utilizando a classe Intent.

O primeiro deles, um dos mais interessantes, é o código para fazer com que sua aplicação Android realize uma ligação. Aparentemente, a utilização de tal recurso parece algo muito complexo, mas, na verdade, resume-se a algumas poucas linhas de código.

Para tal, na tela activity_menu_principal.xml, crie um botão de nome btLigar com texto “Fazer ligações”.

Na MenuPrincipalActivity.java crie a variável pública, btLigar.

```
public class MenuPrincipalActivity extends Activity {
    private Button btLancamento;
    private Button btSair;
    private Button btLigar;
```

Declare no método onCreate

```
btLigar = (Button) findViewById(R.id.btLigar);

btLigar.setOnClickListener( new View.OnClickListener() {
    @Override
    public void onClick(View arg0) {
        btLigarOnClick();
    }
});
```

Crie abaixo do método onCreate a função btLigarOnClick():

```
public void btLigarOnClick() {
    Intent i = new Intent (this, LigarActivity.class);
    startActivity(i);
}
```

Crie agora uma Activity de nome LigarActivity.java com o XML abaixo:

Prof. Menezes	PROGRAMAÇÃO MOBILE Estudo de Caso	Data 27/09/2016
------------------	---	--------------------

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".LigarActivity" >
```

```
<ImageButton
```

```
    android:id="@+id/imLigar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:layout_marginLeft="103dp"
    android:src="@drawable/ic_launcher" />
```

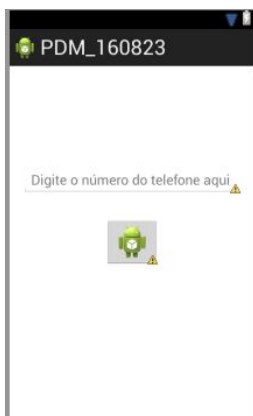
```
<EditText
```

```
    android:id="@+id/edTelefone"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/imLigar"
    android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"
    android:layout_marginBottom="30dp"
    android:ems="10"
    android:hint="Digite o número do telefone aqui"
    android:inputType="number" />
```

```
</RelativeLayout>
```

Observe que o atributo `inputType="number"` no `EditText edTelefone`, ativa apenas o teclado numérico.

A tela ficaria assim:



Na classe `LigarActivity.java` temos:

```
package com.examp.e.pdm_160823;
```

Prof. Menezes	PROGRAMAÇÃO MOBILE Estudo de Caso	Data 27/09/2016
------------------	---	--------------------

```

import android.net.Uri;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;

public class LigarActivity extends Activity {
    private ImageButton iMLigar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ligar);
        iMLigar = (ImageButton) findViewById(R.id.iMLigar);

        iMLigar.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                iMLigarOnClick();
            }
        });
    }

    public void iMLigarOnClick() {
        EditText campoTelefone = (EditText) findViewById(R.id.edTelefone);
        String telefone = campoTelefone.getText().toString();
        Uri uri = Uri.parse("tel:"+telefone);
        Intent intent = new Intent(Intent.ACTION_DIAL, uri);
        startActivity(intent);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is
present.
        getMenuInflater().inflate(R.menu.ligar, menu);
        return true;
    }
}

```

Este método apresentado possui a função de recuperar o número para o qual se deseja ligar da tela, a partir de um campo EditText. Na sequência, o texto digitado no campo é armazenado em uma string para, então, instanciar um objeto do tipo Uri, o qual deve conter o número iniciando com "tel:".

Após, o objeto Intent, já conhecido por nós, é instanciado, porém, recebe por parâmetro o valor Intent.ACTION_DIAL, o que identifica a tela de ligação, assim como o uri com o número de telefone.

Prof. Menezes	PROGRAMAÇÃO MOBILE Estudo de Caso	Data 27/09/2016
------------------	---	--------------------