



Programação para Android

Aula 10: Acesso a câmera, sms e recursos do aparelho

Objetivos

- ▶ Nesta aula iremos aprender como invocar os recursos do aparelho:
 - ▶ Fazer chamadas telefônicas
 - ▶ Enviar SMS
 - ▶ Utilizar a câmera



Parte 01: Fazer uma ligação telefônica

Fazer uma ligação telefônica

- ▶ Antes de criar uma intenção (INTENT) para fazer uma ligação precisamos da permissão do Sistema Operacional e para isso no arquivo AndroidManifest.xml devemos colocar a permissão `android.permission.CALL_PHONE`, veja sua implementação completa abaixo:

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

Fazer uma ligação telefônica

- Criamos uma Intent passando no seu construtor o tipo de ligação, ACTION_DIAL ou ACTION_CALL. ACTION_DIAL abre a janela com a opção de alterar o telefone, já ACTION_CALL, faz a ligação direta, veja seus exemplos abaixo.

```
//Tipos de chamadas  
//ACTION_DIAL  
//ACTION_CALL
```

```
Intent chamada = new Intent(Intent.ACTION_DIAL);
```

```
//altera o número do telefone
```

```
Uri uri = Uri.parse("tel:"+telefone);  
chamada.setData(uri);
```

```
startActivity(chamada);
```

Parte 02: Enviar SMS

Enviar SMS

- ▶ A plataforma Android fornece métodos de envio de SMS pela aplicação.
- ▶ Podemos enviar mensagem SMS de duas formas:
 - ▶ 1ª forma: realizando uma chamada do aplicativo de mensagens com o texto que queremos passar e/ou o número;
 - ▶ 2ª forma: Enviar o SMS direto da sua aplicação
- ▶ Para usar esses métodos é necessário declarar uma permissão no AndroidManifest:

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

Enviar SMS

- Utilizando a 1ª forma, o que precisamos fazer é apenas chamar o aplicativo de mensagens com o texto que queremos passar e/ou o número:

```
Intent smsIntent = new Intent(Intent.ACTION_VIEW);  
smsIntent.setType("vnd.android-dir/mms-sms");  
smsIntent.putExtra("address", "NUMERO DO TELEFONE");  
smsIntent.putExtra("sms_body", "MENSAGEM A SER ENVIADA");  
startActivity(smsIntent);
```

- Feito isso a mensagem e número é passado para outra aplicação e não nos preocupados com o envio propriamente dito.

Enviar SMS

- ▶ A segunda forma de envio, seria encaminhar o SMS direto da aplicação. Precisamos para isto utilizar um objeto do tipo **SmsManager**.

```
SmsManager smsManager = SmsManager.getDefault();
```

- ▶ Depois utiliza o seguinte comando se você tiver certeza que sua mensagem terá menos de 160 caracteres:

```
smsManager.sendTextMessage("07380808080", null, "Mensagem  
que estou enviando", null, null);
```

- ▶ Se você não tem certeza se sua mensagem será menor que 160 caracteres use o seguinte método:

```
smsManager.sendMultipartTextMessage ("07187853344", null,  
smsManager.divideMessage("Mensagem muito grande que estou enviando"),  
null, null);
```

Parte 03: Acessar a câmera

Utilizando a câmera

- ▶ Grande parte dos dispositivos Android possui uma ou mais câmeras, que permitem tirar fotos e gravar vídeos. Podemos utilizar este recurso na programação de nossas APPs.
- ▶ A integração de uma aplicação com a câmera de um dispositivo pode ser feita de duas formas. A primeira, mais simples, utiliza a aplicação de câmera nativa que já existe no Android. E a segunda permite que o desenvolvedor tenha controle total do hardware da câmera. As duas formas serão abordadas na sequência.
- ▶ Utilizaremos a forma mais simples uma vez que a aplicação já existente no dispositivo já está pronta para utilização.

Utilizando a câmera

- ▶ A forma mais simples de integrar uma aplicação com a câmera é chamar a aplicação nativa de câmera do dispositivo. A chamada de outras aplicações no Android é feita através do disparo de uma intent para a plataforma, que fica responsável por carregar a aplicação.
- ▶ Para exemplificar a utilização da câmera iremos criar uma aplicação simples que fará uma chamada ao aplicativo da câmera e após a foto ser tirada a mesma será visualizada em uma ImageView.
- ▶ **OBS: Este exemplo só funcionará em um dispositivo físico com Android.**

Utilizando a câmera - Layout XML

- ▶ A aplicação constituirá de:
 - ▶ 1 Button para invocar a câmara física do dispositivo
 - ▶ 1 Label com o texto “Resultado”
 - ▶ 1 ImageView para receber a foto obtida pela câmera.



Utilizando a câmera - Layout XML

```
<Button
    android:id="@+id/btAbrirCamera"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/lbChamar" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/lbResultado"/>

<ImageView
    android:id="@+id/ivFoto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
```

Utilizando a câmera - Programação Java

- ▶ A programação Java consistirá de 2 classes:
 - ▶ MainActivity: classe relacionada ao layout XML
 - ▶ usarCamera: classe criada para codificar os métodos de manipulação da Câmera
- ▶ A classe usarCamera consiste de 2 métodos:
 - ▶ **public void carregarFoto(String nomeArquivoFoto, ImageView ivFoto):** Esta classe carrega um arquivo de imagem em uma ImageView
 - ▶ **public Intent capturarFoto(String nomeArquivoFoto) :** Esta classe captura uma imagem retirada na câmera do dispositivo e retorna a imagem retirada com o nome passado no parâmetro do método(nomeArquivoFoto)

Utilizando a câmera - Classe usarCamera: carregaFoto

```
public void carregarFoto(String nomeArquivoFoto, ImageView ivFoto)
{

    //Local onde será buscado a foto a ser carregada na ImageView
    //Environment.DIRECTORY_PICTURES pega o diretório Pictures do Android
    String local = Environment.getExternalStoragePublicDirectory(
        Environment.DIRECTORY_PICTURES) + "/" + nomeArquivoFoto;

    //Cria uma imagem a partir de um caminho onde se encontrar um arquivo
    Bitmap imagem = BitmapFactory.decodeFile(local);

    //Altera um imageView para uma nova imagem, neste caso a imagem com o caminho
    especificado acima
    ivFoto.setImageBitmap(imagem);
}
```


Utilizando a câmera - Classe usarCamera: capturarFoto

```
public Intent capturarFoto(String nomeArquivoFoto)
{
    //Cria uma intenção de Capturar de Imagem ou seja, usar a camera
    Intent i = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);

    // Environment.DIRECTORY_PICTURES pega o diretório Pictures do Android
    //Usaremos este local para armazenar as imagens
    File picsDir = Environment.getExternalStoragePublicDirectory(
        Environment.DIRECTORY_PICTURES);
    File imageFile = new File(picsDir, nomeArquivoFoto);

    i.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(imageFile)); // Arquivo a ser
armezado
    // Inicia a captura da imagem
    return i;
}
```

Utilizando a câmera - MainActivity

- ▶ Na MainActivity além das declarações de objetos, associações entre objetos Java e layout Xml (findViewById), teremos:
 - ▶ 1 Listener para chamar a câmera ao tocar no botão CHAMAR
 - ▶ 1 Método onActivityResult para para pegar o resultado gerado após ser tirado uma foto, ou seja, carregar a foto em uma ImageView.

Utilizando a câmera - OnClickListener

```
View.OnClickListener abrirCamera = new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
  
        //Cria a classe usarCamera  
        usarCamera c = new usarCamera();  
        //Chamada do método CapturarFoto com o nome do arquivo usado quando for gravado a  
        foto  
        //Este método retornar uma intenção para que depois seja iniciada a captura da  
        imagem  
        Intent i = c.capturarFoto("regilan.jpg");  
        //Inicia a intenção ou seja, captura a imagem  
        startActivity(i);  
        //Toast.makeText(getContext(), "Mensagem = " + mensagem,  
        Toast.LENGTH_LONG).show();  
    }  
};
```

Utilizando a câmera - onActivityResult

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    super.onActivityResult(requestCode, resultCode, data);  
  
    Toast.makeText(getBaseContext(), "requestCode = " + requestCode + " resultCode =  
    " + resultCode, Toast.LENGTH_LONG).show();  
  
    if (requestCode == 100) { //requestCode = 100 para Imagem  
        if (resultCode == RESULT_OK) { //resultCode = OK se tudo deu certo  
            //Cria a classe usarCamera  
            usarCamera c = new usarCamera();  
            //Chama o método para carregar uma foto no ImageView  
            c.carregarFoto("regilan.jpg", ivFoto);  
        } else if (resultCode == RESULT_CANCELED) {  
            Toast.makeText(getBaseContext(), "Erro ao carregar a foto!",  
            Toast.LENGTH_LONG).show();  
        } else {}  
    }  
}
```

Utilizando a câmera

- No método onCreate da MainActivity ainda deve conter a chamada ao método OnClickListener. Também será feita uma chamada do método CarregarFoto para ao ser iniciada, uma foto será carregada na ImageView.

```
//Relacionamento dos objetos no Java com o Arquivo de Layout
ivFoto = (ImageView)findViewById(R.id.ivFoto);
btAbrirCamera = (Button)findViewById(R.id.btAbrirCamera);

//Associar o Click ao botão de Abrir Camera
btAbrirCamera.setOnClickListener(abrirCamera);

//Cria a classe usarCamera
usarCamera c = new usarCamera();
//Chama o método para carregar uma foto no ImageView
c.carregarFoto("regilan.jpg", ivFoto);
```

Na próxima aula...

- ▶ Acessando Mapas e GPS

