

Exercício 1

Existe um atributo na classe `FormularioHelper` do tipo `ImageView` chamado `foto`. Vamos criar um `getter` para o novo atributo para que possamos deixar a `Activity` decidir o comportamento do clique.

```
public class FormularioHelper {
    // outros atributos...
    private ImageView foto;

    public FormularioHelper(FormularioActivity activity) {
        //...
        foto = (ImageView) activity.findViewById(R.id.foto);
    }

    public ImageView getFoto() {
        return foto;
    }
}
```

Vamos guardar a possível localização da cópia da imagem que a câmera tirar.

Para chamarmos a câmera, vamos aguardar um clique nesse `ImageView`. Se ainda não há foto do aluno, veremos uma imagem padrão. Quando clicarmos sobre ela, vamos chamar a intent da câmera. No `onCreate` do formulário, fazemos um listener e passamos um arquivo do tipo `.jpg` para a intent:

```
public class FormularioActivity extends Activity{
    private FormularioHelper helper;

    // guardando local no qual queremos gravar a foto
    private String localArquivoFoto;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // ...

        ImageView foto = helper.getBotaoImagem();
        foto.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                localArquivoFoto = getExternalFilesDir(null) + "/" + System.currentTimeMillis() + ".jpg";

                // AQUI VAMOS CHAMAR A CÂMERA DO ANDROID!
            }
        })

        //...
    }

    //...
}
```

O "apelido" para chamarmos a `Intent` de câmera do `Android` é uma constante encontrada na classe `MediaStore`:

```
Intent irParaCamera = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
```

Para solicitarmos que uma cópia da foto seja gravada no local que desejarmos podemos passar um `Extra` na `Intent` da câmera do tipo **URI**:

```
irParaCamera.putExtra(MediaStore.EXTRA_OUTPUT, ?? URI para um local no SD ?? );
```

Para converter nossa `String` com o local desejado para um objeto do tipo **URI**

```
String localArquivoFoto = //...
Uri localFoto = Uri.fromFile( new File(localArquivoFoto) );
```

Com a Intent de chamada da câmera devidamente configurada podemos fazer a chamada ao método que colocará essa intenção em prática, mas dessa vez esse método não será o `startActivity` . Como esperamos um retorno da Activity de câmera (se o usuário clicou em **OK** para salvar a imagem ou em **Cancelar**) vamos invocar o método `startActivityForResult` , passando nossa Intent e um código qualquer nosso, por exemplo `123`.

```
startActivityForResult(irParaCamera, 123);
```

Preferencialmente extraia o código que você criar para a chamada a `startActivityForResult` em uma constante na classe `FormularioActivity` .

Rode a aplicação e veja se o clique na imagem do formulário nos leva para a tela da captura de imagem do Android.

Como ficou a implementação do `onCreate` de seu `FormularioActivity` ?

Responda

INSERIR CÓDIGO	FORMATAÇÃO

