

 04  
**Exercício 2**

Agora, vamos tratar o retorno no método de callback `onActivityResult`. Faça o `override` do método na `FormularioActivity`. Repare que o método `carregaImagem` ainda não existe no `FormularioHelper`, vamos criá-lo!

```
public class FormularioActivity extends Activity{
    private FormularioHelper helper;
    private String localArquivoFoto;
    private static final int TIRA_FOTO = 123;

    // outros métodos...

    @Override
    protected void onActivityResult(int requestCode,
        int resultCode, Intent data) {

        if (requestCode == TIRA_FOTO) {
            if (resultCode == Activity.RESULT_OK) {
                helper.carregaImagem(this.localArquivoFoto);
            } else {
                this.localArquivoFoto = null;
            }
        }
    }
}
```

Para implementarmos o método `carregaImagem` no `FormularioHelper` precisaremos carregar a foto, para isso vamos usar a classe `BitmapFactory`, que vai mostrar a imagem e gerar seu thumbnail:

```
public class FormularioHelper {
    //...

    public void carregaImagem(String localArquivoFoto) {
        // carregar no formulario a imagem do aluno...
    }
}
```

Para obter a imagem a partir da String:

```
Bitmap imagemFoto = BitmapFactory.decodeFile("String com local do arquivo");
```

Nesse momento a imagem se encontra grande demais, vamos trabalhar com uma versão reduzida dessa imagem, usando a própria classe `Bitmap` e o método `createScaledBitmap`:

```
Bitmap imagemFoto = //...
Bitmap imagemFotoReduzida = Bitmap.createScaledBitmap(imagemFoto, 100, 100, true);
```

Agora basta colocar a imagem no atributo `foto` do `FormularioHelper`.

Além disso acrescente a seguinte permissão no seu `AndroidManifest`:

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

Rode a aplicação clique na foto do formulario, tire uma fotografia e aprove a criação do arquivo.

**Não** esqueça de clicar no botão **Alterar** ou **Salvar** para que a foto seja armazenada no SQLite!

Como ficou a sua implementação do método `carregaImagem` do `FormularioHelper`? E o método `onActivityResult` do `FormularioActivity`?

## Responda