Desenvolvimento de Apps para Mobile

Fabrício Tonetto Londero

Aula 08 - Camera

Projeto

- Seu app tem que ser capaz de:
  - Abrir a câmera
  - Tirar uma foto
  - Armazenar a foto na galeria
  - Mostrar a foto no ImageView

Tirar fotos com Android

- Primeiro de tudo, permissões!
  - AndroidManifest

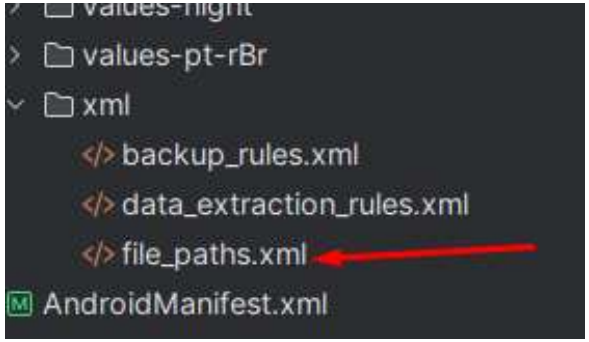
```
<uses-feature
   android:name="android.hardware.camera"
   android:required="false" />
```

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

Tirar fotos com Android e ARMAZENAR

```
<provider
   android:name="androidx.core.content.FileProvider"
   android:authorities="${applicationId}.fileprovider"
   android:exported="false"
   android:grantUriPermissions="true">
   <meta-data
      android:name="android.support.FILE_PROVIDER_PATHS"
      android:resource="@xml/file_paths" />
</provider>
```

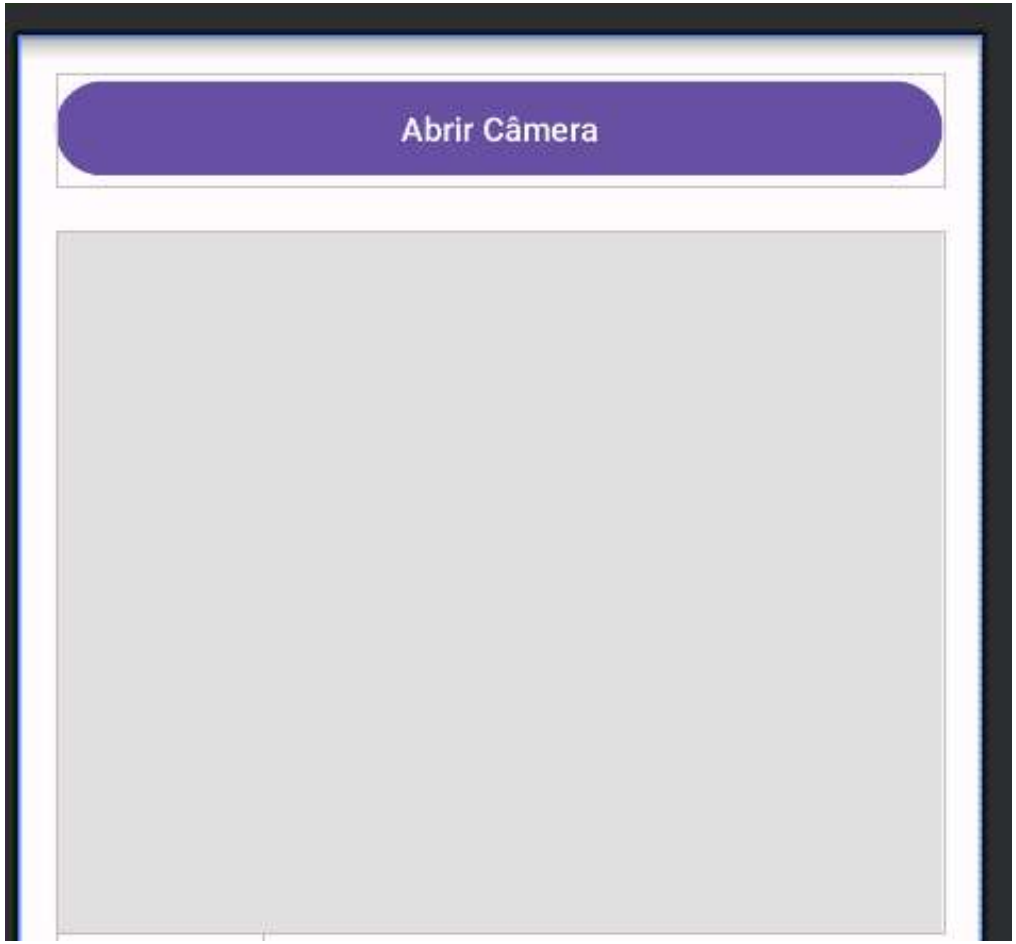
file_paths.xml




```xml
<?xml version="1.0" encoding="utf-8"?>
<paths xmlns:android="http://schemas.android.com/apk/res/android">
    <external-files-path
        name="my_images"
        path="Pictures" />
</paths>
```

Activity

- Vamos colocar:
  - um Botão
  - um ImageView



Activity

- Atributos:

```
private ImageView imageView;
private Uri fotoUri;
private File fotoArquivo;
```

Activity – Criação de um Launcher

```
private final ActivityResultLauncher<Intent> cameraLauncher =
        registerForActivityResult(new ActivityResultContracts.StartActivityForResult(), result -> {
            if (result.getResultCode() == RESULT_OK) {
                if (fotoArquivo != null && fotoArquivo.exists()) {
                    /*Bitmap bitmap = BitmapFactory.decodeFile(fotoArquivo.getAbsolutePath());
                    imageView.setImageBitmap(bitmap);*/
                    Bitmap bitmap = BitmapFactory.decodeFile(fotoArquivo.getAbsolutePath());
                    bitmap = corrigirRotacao(fotoArquivo.getAbsolutePath(), bitmap);
                    imageView.setImageBitmap(bitmap);

                    try {
                        saveImageToStorage(bitmap);
                        Toast.makeText(this, "Foto salva com sucesso!", Toast.LENGTH_SHORT).show();
                    } catch (IOException e) {
                        Toast.makeText(this, "Erro ao salvar a foto: " + e.getMessage(), Toast.LENGTH_SHORT).show();
                    }
                }
            }
        });
```

Activity - Corrigir Rotação

```
private Bitmap corrigirRotacao(String caminho, Bitmap bitmap) {
   try {
      ExifInterface exif = new ExifInterface(caminho);
      int orientacao = exif.getAttributeInt(ExifInterface.TAG_ORIENTATION, ExifInterface.ORIENTATION_NORMAL);
      int rotacao = 0;
      switch (orientacao) {
         case ExifInterface.ORIENTATION_ROTATE_90:
            rotacao = 90;
            break;
         case ExifInterface.ORIENTATION_ROTATE_180:
            rotacao = 180;
            break;
         case ExifInterface.ORIENTATION_ROTATE_270:
            rotacao = 270;
            break;
      }
      if (rotacao != 0) {
         Matrix matrix = new Matrix();
         matrix.postRotate(rotacao);
         bitmap = Bitmap.createBitmap(bitmap, 0, 0, bitmap.getWidth(), bitmap.getHeight(), matrix, true);
      }
      return bitmap;
   } catch (IOException e) {
      e.printStackTrace();
      return bitmap; // original se der erro
   } }
```

Activity – Salvar imagem – Parte 1

```
private void saveImageToStorage(Bitmap bitmap) throws IOException {
    String filename = "IMG_" + UUID.randomUUID().toString() + ".jpg";

    if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.Q) {
        ContentValues values = new ContentValues();
        values.put(MediaStore.Images.Media.RELATIVE_PATH, "Pictures/MyApp");
        values.put(MediaStore.Images.Media.IS_PENDING, true);
        values.put(MediaStore.Images.Media.MIME_TYPE, "image/jpeg");
        values.put(MediaStore.Images.Media.DISPLAY_NAME, filename);
```

Activity – Salvar imagem – Parte 2

```
 Uri uri = getContentResolver().insert(MediaStore.Images.Media.EXTERNAL_CONTENT_URI,
values);
    if (uri != null) {
        try (FileOutputStream out = (FileOutputStream)
getContentResolver().openOutputStream(uri)) {
            bitmap.compress(Bitmap.CompressFormat.JPEG, 100, out);
        }
        values.put(MediaStore.Images.Media.IS_PENDING, false);
        getContentResolver().update(uri, values, null, null);
    }
} else {
```

Activity – Salvar imagem – Parte 3

```
// Android 9 ou inferior
        File directory = new File(Environment.getExternalStoragePublicDirectory(
                Environment.DIRECTORY_PICTURES), "MyApp");
        if (!directory.exists()) {
            directory.mkdirs();
        }

        File file = new File(directory, filename);
        try (FileOutputStream out = new FileOutputStream(file)) {
            bitmap.compress(Bitmap.CompressFormat.JPEG, 100, out);
        }
    }
}
```

Activity – Abrir Câmera – Parte 1

OBS: Chamar no Click do Botão

```
public void abrirCamera(View view)
{
   if (ContextCompat.checkSelfPermission(this, Manifest.permission.CAMERA) !=
PackageManager.PERMISSION_GRANTED)
   {
      ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.CAMERA}, 100);
      return;
   }
```

Activity – Abrir Câmera – Parte 2

```
try
{
    File dir = getExternalFilesDir(Environment.DIRECTORY_PICTURES);
    String nomeArquivo = "foto_" + UUID.randomUUID().toString() + ".jpg";
    fotoArquivo = new File(dir, nomeArquivo);
    fotoUri = androidx.core.content.FileProvider.getUriForFile(this, getPackageName() + ".fileprovider", fotoArquivo);

    Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    cameraIntent.putExtra(MediaStore.EXTRA_OUTPUT, fotoUri);
    cameraIntent.addFlags(Intent.FLAG_GRANT_WRITE_URI_PERMISSION);
    cameraLauncher.launch(cameraIntent);
}
catch (Exception e)
{
    Toast.makeText(this, "Erro ao abrir a câmera: " + e.getMessage(), Toast.LENGTH_SHORT).show();
}
```

Exercício

- Crie uma tela de cadastro de usuário que persista os dados
  - Um dos dados coletados é uma imagem, que pode ser tirada na hora ou selecionada da galeria