FOTOS – CÓDIGO DE BARRAS E WEBVIEW

Prof. Danilo Ruy Gomes

Introdução

O que veremos:

- Permissões pelo usuário e armazenado;
- Tirando fotos;
- Leitura de código de barras;
- WebView.

O que precisaremos

- Android Studio;
- App barCodScanner;
- Acesso a internet;
- Emulador ou Smartphone conectado no computador.

Permissões

A partir da versão 6.0 do Android a questão das permissões foram divididos em dois tópicos:

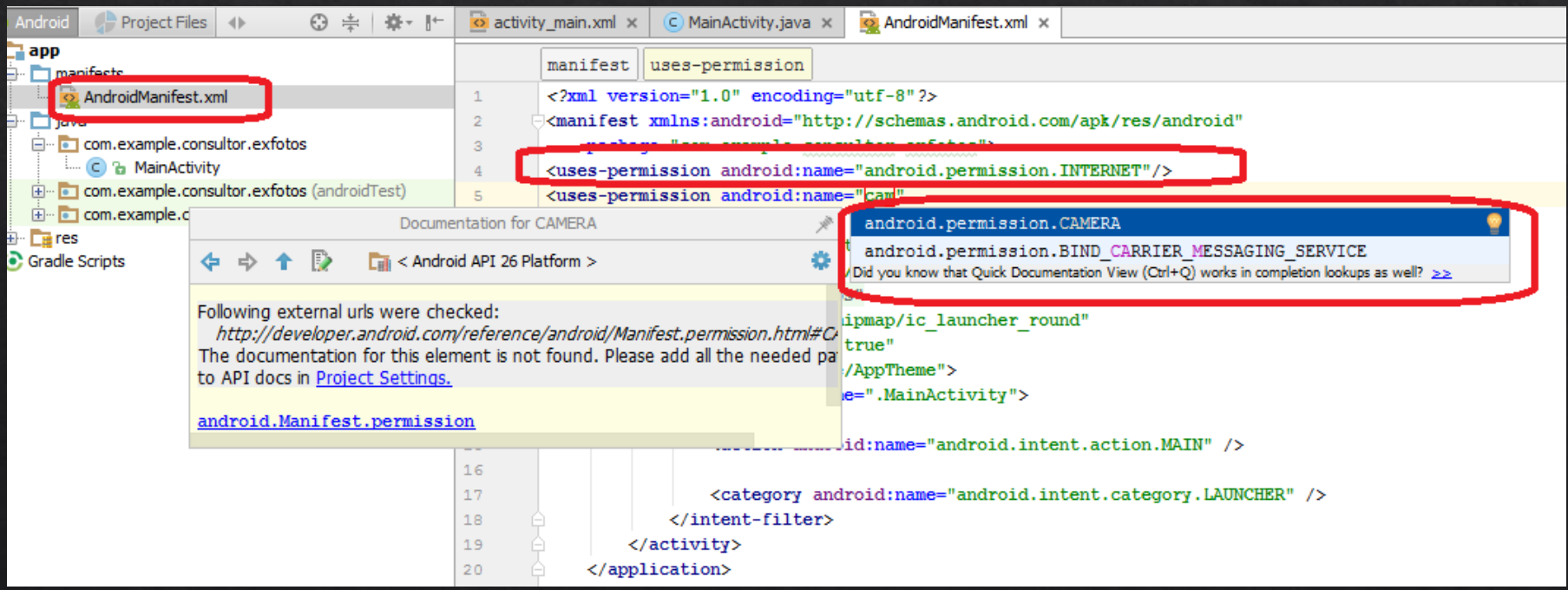
- Permissões de usuário;
- Permissões de sistema;

Permissões de sistema

- As permissões de sistema são aquelas que não utilizam recursos que precisam de autorização do usuário no momento da abertura do *"app"* e que não implica de uso de informações confidenciais, como vibração, armazenamento, etc;
- Normalmente estas permissões são declaradas com a clausula "*uses*" no arquivo "*manifest*" como no exemplo:

Permissões de sistema

Exemplo de declaração de permissão no manifest:



Permissões de usuário

- As permissões de usuário são aquelas que de acordo com o google, podem violar a privacidade, devendo assim, perguntar ao usuário no momento da abertura do "*app*", como é o caso da câmera, agenda, gps, etc.
- Para isto deveremos criar uma classe que montará uma pergunta para o usuário toda vez que tal recurso for utilizado.

Permissões de usuário

Crie a classe permissionUtilis com as seguintes instruções:

```
public class PermiissionUtils {
    public static boolean validate (Activity activity, int requestCode, String ... permissions)
    {
        List<String> list = new ArrayList<String>();
        for (String permission : permissions) {
            // Valida permissao
            //A instrucao abaixo eh responsavel por fazer a pergunta da pemissao passando a activity (classe
            //que chamou) assim como o parametro da permissao
            boolean ok = ContextCompat.checkSelfPermission(activity, permission)
                    == PackageManager.PERMISSION_GRANTED;
            if (! ok ) {
                //se reposta nao eh falsa entao .. adiciona permissao
                list.add(permission);
            }
        }
        if (list.isEmpty()) {
            // Tudo ok, retorna true
            return true;
        }
        // Lista de permissoees que faltam acesso.
        String[] newPermissions = new String[list.size()];
        list.toArray(newPermissions);
        // Solicita permissao, metodo provida pela classe app compat
        ActivityCompat.requestPermissions(activity, newPermissions, 1);

        return false;
    }
}
```

Permissões de usuário

Volte ao "*MainActivity*" e declare um *"array"* com as permissões desejadas e uma variável que marcará o estado da permissão:

```
public class MainActivity extends AppCompatActivity {

    String[] permissoes = new String[]{
            Manifest.permission.CAMERA,
            Manifest.permission.ACCESS_FINE_LOCATION,
    };
    Boolean permissaook=false;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

No método *"create"* chame o método *"validate"* da classe criada:

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        PermissionUtils.validate(MainActivity.this, 0, permissoes);
    }
```

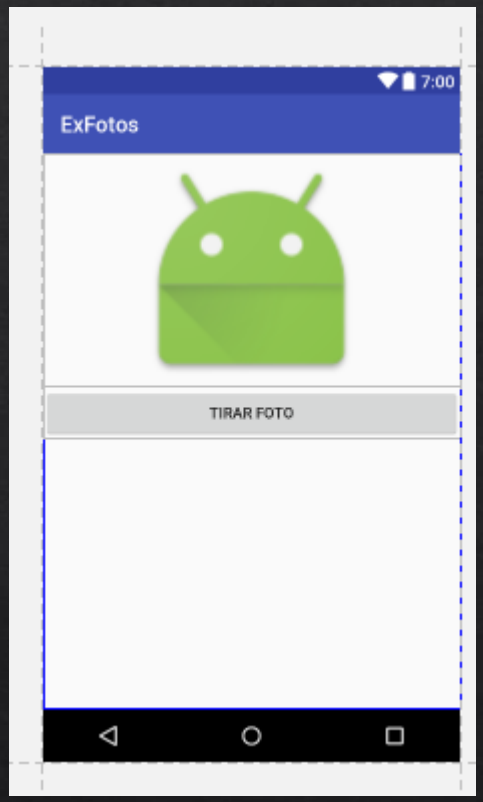
Permissões de usuário

Crie o método *"onRequestPermissionResult"*, que ficará responsável por perguntar ao usuário se ele permite ou não utilizar tais permissões

```
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);

        for (int result : grantResults) {
            if (result == PackageManager.PERMISSION_DENIED) {
                Toast.makeText(getApplicationContext(), "Permissão negada", Toast.LENGTH_LONG).show();
                return;
            } else {
                permissaook=true;
                Toast.makeText(getApplicationContext(), "Permissão concedida", Toast.LENGTH_LONG).show();
            }
        }

    }
}
```

Tirando fotos

Adicione um imageview e um button como no modelo ao lado:




```
<LinearLayout
    xmlns:android="http://schemas.android.co
    xmlns:app="http://schemas.android.com/ap
    xmlns:tools="http://schemas.android.com/
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.consultor.exf
    android:weightSum="1">

    <ImageView
        android:id="@+id/imagem"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:srcCompat="@mipmap/ic_launcher"
        android:layout_weight="0.40" />

    <Button
        android:id="@+id/btnTiraFoto"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Tirar foto" />
</LinearLayout>
```

Tirando fotos

Volte ao método "*create*" e adicione a chamada para o método "*retiraFoto*" e depois faça a implementação:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    PermissionUtils.validate(MainActivity.this, 0, permissoes);

    if (permissaook==true)
    {
        retiraFoto();
    }
}
```

Tirando fotos

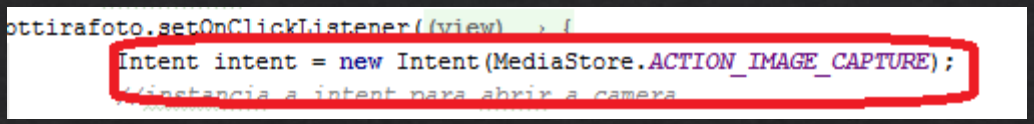
Método retiraFoto:

```
private void retiraFoto()
{

    img = (ImageView)findViewById(R.id.imagem);
    Button bttirafoto = (Button)findViewById(R.id.btnTiraFoto);
    bttirafoto.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
            //instancia a intent para abrir a camera
            startActivityForResult(intent, 0);
        }
    });

}
```

Salve e execute. Repare que irá tirar a foto e armazenar no banco de imagens do seu smartphone.

Recuperando a foto

Repare que nós abrimos a câmera através da instancia de uma *"intent"*.



Ao instanciarmos, chamamos esta "*activity*" através do método "*startActivityForResult*", que inicia uma "*activity*", porém dá a possibilidade de recebermos alguma coisa da "*intent*" que abrimos (a câmera", quando esta é finalizada.

Recuperando a foto

Assim teremos que implementar o método "*onActivityForResult*", assim, se houver dados ele retorna para o nosso imagem "*view*" através do "*getExtras*".

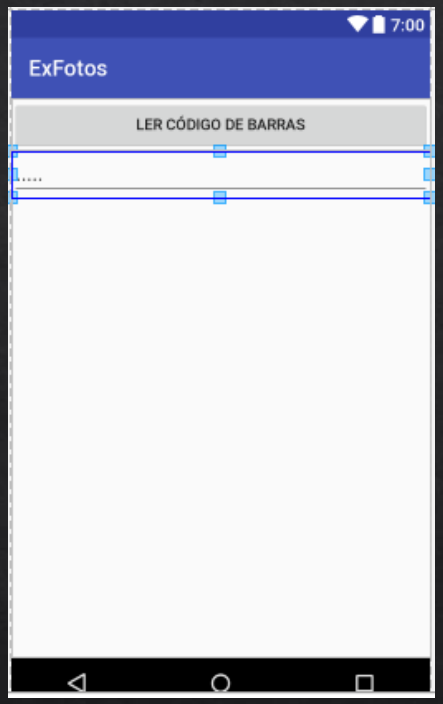
```
}
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
    super.onActivityResult(requestCode, resultCode, data);
    Bundle bundle = data.getExtras();
    //O Tipo bundle permite que se tranfira dados entre a intent e a aplicacao
    //
    if (data != null)
    {
        Bitmap bitmap = (Bitmap)bundle.get("data");
        //pega o bitmap do bundle
        img.setImageBitmap(bitmap);
        //seta o bitmap no imageview
    }
}
```

Leitura de código de barras

- Semelhante a abertura da câmera, realizaremos a chamada do código de barras através de uma *"intent"*, mas ao invés de chamarmos um recurso do próprio dispositivo, iremos abrir um *"app"* construído por uma outra pessoa. Neste caso o *"barCodeScanner"*.
- Para isto instale este *"app"* através da Google Play.

Leitura de código de barras

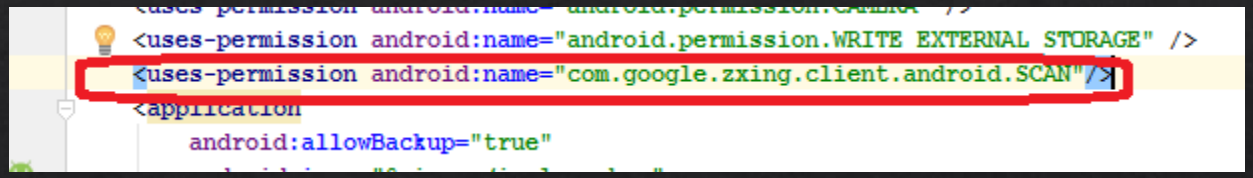
Agora, no mesmo projeto, crie uma nova *"activity"* e coloque um *"button"* e um *"text"* para exibir o código lido, conforme a imagem:




```
<LinearLayout
    xmlns:android="http://schemas.android.com/ap
    xmlns:app="http://schemas.android.com/apk/re
    xmlns:tools="http://schemas.android.com/tool
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.consultor.exfotos

    <Button
        android:id="@+id/btnLerCodigo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Ler código de barras" />

    <EditText
        android:id="@+id/edtCodigo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName"
        android:text="....." />
</LinearLayout>
```

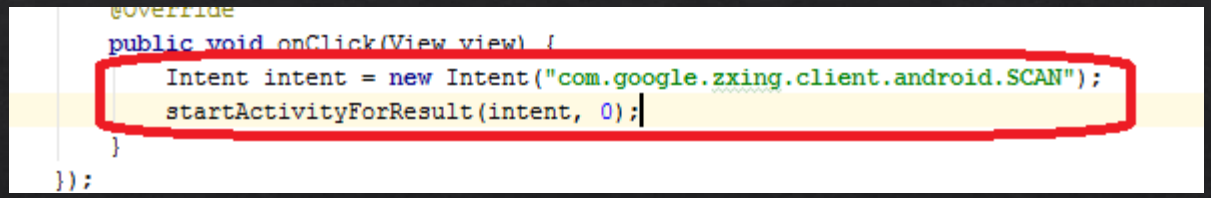
Leitura de código de barras

Insira no arquivo *"manifest"* a permissão para acesso do seu *"app"* ao aplicativo *"barCodeScanner"*.



Leitura de código de barras

No método *"create"* adicione as chamadas para o *"edit"* e o *"button"* e implemente as seguintes instruções:



Leitura de código de barras

Por fim, crie o método *"onActivityResult"* e implemente a instrução e faça o teste:

```
public void onActivityResult(int requestCode, int resultCode, Intent intent)
{
    if (requestCode ==0)
    {
        edCoodigo.setText(intent.getStringExtra("SCAN_RESULT"));
    }
}
```

WebView

O *"webView"* é o *"widget"* que possibilita abrir um navegador a partir de seu "*app*";
Pode ser útil, por exemplo para exibir um link a partir de seu "*app*" ou quem sabe fazer um filtro de sites.

WebView

Para isto, crie uma nova *"activity"*, inserindo um *"webView"*, um *"text"* e um *"button"*.


```
<WebView
    android:layout_width="234dp"
    android:layout_height="234dp"
    android:id="@+id/wvbusca"/>

<EditText
    android:id="@+id/edtUrl"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:text="Name" />

<Button
    android:id="@+id/btnUrl"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Carregar" />
</LinearLayout>
```

WebView

Adicione agora as instruções para carregar o site.

```
public class WbVActivity extends AppCompatActivity {
private Button btUrl;
private EditText edUrl;
private WebView wb;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_wb_v);

        edUrl = (EditText)findViewById(R.id.edtUrl);
        wb = (WebView)findViewById(R.id.wvbusca);
        btUrl = (Button)findViewById(R.id.btnUrl);
        btUrl.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                wb.loadUrl("http:\\"+edUrl.getText().toString());
            }
        });
```

Dúvidas ???

Referências

Google Android – 4° Edição – Ricardo R. Lecheta. Novatech São Paulo. Capítulos 6 e 7.

Ricardo R. Lecheta – Livros Android
http://ricardolecheta.com.br/?p=560
Acesso em 29/04/2017.

Google Developer:
https://developer.android.com/training/camera/photobasics.html
Acesso em 29/04/2017.

Google Developer:
https://developer.android.com/reference/android/webkit/WebView.html
29/04/2017.