AGENDA 5

PERMISSÕES

GEEaD - Grupo de Estudos de Educação a Distância

Centro de Educação Tecnológica Paula Souza

GOVERNO DO ESTADO DE SÃO PAULO

EIXO TECNOLÓGICO DE INFORMAÇÃO E COMUNICAÇÃO

CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS

PROGRAMAÇÃO MOBILE I

Expediente

Autor:

GUILHERME HENRIQUE GIROLLI

Revisão Técnica:

Eliana Cristina Nogueira Barion

Revisão Gramatical:

Juçara Maria Montenegro Simonsen Santos

Editoração e Diagramação:

Flávio Biazim

São Paulo – SP, 2020

MERGULHANDO NO TEMA...

Com a evolução na aprendizagem do Android Studio, nos deparamos na agenda anterior com uma situação bem peculiar, onde o aplicativo desenvolvido por Serginho consumia um recurso do dispositivo Mobile, ou seja, o aplicativo tinha que conectar na Internet para adquirir as informações de um servidor JSON. Essa situação foi barrada pelo sistema operacional Android, e para que o aplicativo funcionasse de maneira correta, foi preciso efetuar uma liberação de acesso desse recurso no arquivo “AndroidManifest.xml” do projeto.

Com a evolução na aprendizagem do Android Studio, nos deparamos na agenda anterior com uma situação bem peculiar, onde o aplicativo desenvolvido por Serginho consumia um recurso do dispositivo Mobile, ou seja, o aplicativo tinha que conectar na Internet para adquirir as informações de um servidor JSON. Essa situação foi barrada pelo sistema operacional Android, e para que o aplicativo funcionasse de maneira correta, foi preciso efetuar uma liberação de acesso desse recurso no arquivo “AndroidManifest.xml” do projeto.

Você deve estar se perguntando: por qual motivo o Android barrou a utilização da Internet para o aplicativo de Serginho? E a resposta é muito simples e objetiva: - Segurança para o usuário do dispositivo Mobile!

Imagine que o aplicativo desenvolvido para consultar CEP, seja instalado em um dispositivo Mobile que possui uma franquia pequena para utilização dos recursos de Internet de uma operadora qualquer de telefonia celular. Nesse cenário, se o sistema operacional Android não se preocupar em limitar acessos a Internet para o aplicativo, alguns problemas poderiam surgir, como até mesmo cobranças financeiras para o usuário. Desta forma nada melhor que avisar, ou seja, deixar o usuário do aplicativo ciente de que ele utiliza recursos de Internet.

Esse foi um exemplo bem simples e menos complexo, porém o desenvolvimento de aplicativos Mobile lida com outros recursos mais complexos e bem mais preocupantes dependendo do projeto. Alguns exemplos são o acesso a informações particulares como fotos e vídeos do usuário, acesso a informações do cartão de memória de um dispositivo, câmera, microfone etc. Se esses recursos não forem controlados pelo sistema operacional, e sua liberação de utilização for liberada pelos usuários, podem surgir problemas graves, tornando o sistema falho e de baixa segurança, o que seria um paraíso para pessoas má intencionadas e para os criminosos digitais!

Nas últimas agendas, estudamos que o sistema operacional Android foi concebido por meio do Kernel do sistema Linux, portanto, ele traz em seu DNA uma política rigorosa de segurança que garante alta proteção aos seus usuários.

O desenvolvedor de aplicativos para esse sistema operacional obrigatoriamente tem que conhecer algumas regras básicas, à medida que vai se aprofundando na programação Mobile.

A regra primordial é que o sistema operacional possua um conjunto de permissões para que o aplicativo acesse e/ou utilize determinadas áreas do dispositivo. Por padrão, todas as permissões são bloqueadas, desta forma não corre o risco de um aplicativo prejudicar o outro ou até mesmo o sistema operacional do aparelho, sem ter que passar por uma autorização prévia.

Como todo aplicativo é executado separadamente um do outro, as permissões são exclusivas, ou seja, o que é autorizado para um aplicativo, não será autorizado, de maneira automática, para os demais.

O desenvolvedor Mobile não deve se preocupar com permissões para projetos básicos e aplicativos simples, pois os recursos básicos de processamento e execução não oferecem risco para o sistema e são liberados para o usuário, uma vez que o programa é executado dentro de uma “SandBox”.

Em contrapartida, para utilizar alguns recursos fora da “SandBox”, o programador deve solicitar a permissão no arquivo “AndroidManifest.xml” do projeto e solicitar a permissão de acesso para o usuário, dependendo do tipo do recurso a ser utilizado.

Como estudamos na agenda 1, o arquivo “AndroidManifest.xml” é responsável pelas principais configurações do aplicativo, inclusive pela solicitação de acesso a recursos específicos do dispositivo.

Os recursos são divididos basicamente em **“Permissões Normais”** e **“Permissões Perigosas”.**

Permissões Normais

As permissões normais são aquelas que o aplicativo utiliza e que oferecem um pequeno risco à privacidade do usuário ou ao sistema operacional. Essas permissões são declaradas no arquivo “AndroidManifest.xml” e durante a execução o sistema operacional concede a devida autorização automaticamente, sem a necessidade de informar o usuário. Um exemplo é a utilização da Internet por um aplicativo de rede social.

Para conhecer as principais “Permissões Normais” verifique a figura 1, retirada do site oficial do sistema Android.

- ACCESS_LOCATION_EXTRA_COMMANDS
- ACCESS_NETWORK_STATE
- ACCESS_NOTIFICATION_POLICY
- ACCESS_WIFI_STATE
- BLUETOOTH
- BLUETOOTH_ADMIN
- BROADCAST_STICKY
- CHANGE_NETWORK_STATE
- CHANGE_WIFI_MULTICAST_STATE
- CHANGE_WIFI_STATE
- DISABLE_KEYGUARD
- EXPAND_STATUS_BAR
- FOREGROUND_SERVICE
- GET_PACKAGE_SIZE
- INSTALL_SHORTCUT
- INTERNET
- KILL_BACKGROUND_PROCESSES
- MANAGE_OWN_CALLS
- MODIFY_AUDIO_SETTINGS
- NFC
- READ_SYNC_SETTINGS
- READ_SYNC_STATS
- RECEIVE_BOOT_COMPLETED
- REORDER_TASKS
- REQUEST_COMPANION_RUN_IN_BACKGROUND
- REQUEST_COMPANION_USE_DATA_IN_BACKGROUND
- REQUEST_DELETE_PACKAGES
- REQUEST_IGNORE_BATTERY_OPTIMIZATIONS
- SET_ALARM
- SET_WALLPAPER
- SET_WALLPAPER_HINTS
- TRANSMIT_IR
- USE_FINGERPRINT
- VIBRATE
- WAKE_LOCK
- WRITE_SYNC_SETTINGS

Figura 1 – Permissões Normais. Site: https://developer.android.com/guide/topics/permissions/overview?hl=pt-br#normal-dangerous

A imagem anterior apresenta alguns recursos utilizados atualmente pelos principais aplicativos disponíveis no mercado. Recursos simples que fazem parte da nossa vida, como o alerta por vibração do aparelho, que é executado quando chega uma mensagem em nosso smartphone. Esse alerta utiliza o recurso “VIBRATE” do dispositivo, que não apresenta perigo e muito menos exponha a segurança do usuário do aparelho. Sendo assim classificada no grupo de “Permissões Normais”.

Entre os principais recursos desse grupo, destacamos além do “VIBRATE”, o uso do “BLUETOOTH”, “INTERNET” e “NFC”.

Permissões Perigosas

As permissões perigosas são aquelas que comprometem a privacidade do usuário e podem comprometer o sistema operacional e demais aplicativos. Para utilizar essas permissões, o aplicativo necessita informar no arquivo “AndroidManifest.xml” a intenção da utilização, e durante a execução, o usuário necessita conceder a sua autorização expressa para a liberação da permissão, por meio de uma mensagem exibida em tela.

Um exemplo clássico é encontrado nos programas de edição de fotos, onde o aplicativo solicita a permissão de acesso ao usuário as pastas de galeria de imagens do dispositivo.

Para conhecer as principais “Permissões Perigosas” verifique a figura 2, retirada do site oficial do sistema Android.

Grupo de Permissão	Permissões
CALENDAR	• READ_CALENDAR • WRITE_CALENDAR
CALL_LOG	• READ_CALL_LOG • WRITE_CALL_LOG • PROCESS_OUTGOING_CALLS
CAMERA	• CAMERA
CONTACTS	• READ_CONTACTS • WRITE_CONTACTS • GET_ACCOUNTS
LOCATION	• ACCESS_FINE_LOCATION • ACCESS_COARSE_LOCATION
MICROPHONE	• RECORD_AUDIO
PHONE	• READ_PHONE_STATE • READ_PHONE_NUMBERS • CALL_PHONE • ANSWER_PHONE_CALLS • ADD_VOICEMAIL • USE_SIP
SENSORS	• BODY_SENSORS
SMS	• SEND_SMS • RECEIVE_SMS • READ_SMS • RECEIVE_WAP_PUSH • RECEIVE_MMS
STORAGE	• READ_EXTERNAL_STORAGE • WRITE_EXTERNAL_STORAGE

Figura 2 – Permissões Perigosas. Site: https://developer.android.com/guide/topics/permissions/overview?hl=pt-br#permission-groups

A figura 2 apresenta alguns recursos dos dispositivos que se utilizados de forma indevida, podem resultar em sérios problemas ao usuário e ao próprio aparelho. Imagine um aplicativo malicioso, que tenha acesso completo aos recursos de “SMS” do dispositivo. Esse aplicativo consegue enviar mensagens para números aleatórios ou até mesmo números da sua agenda, como se as mensagens fossem escritas por você!

Por esse e outros motivos o sistema operacional Android informa ao usuário, e solicita dele uma aprovação. Todos os recursos da lista de “Permissões Perigosas” são pertinentes, e destacamos o uso da localização por GPS, utilização de chamadas de voz, utilização de SMS, calendário, câmera e até mesmo acesso aos arquivos e contatos do dispositivo.

Uso de permissões de acesso aos recursos de Software

Os projetos executados nos sistemas operacionais inferiores ao Android 5.1, solicitavam a autorização do usuário para as permissões utilizadas no momento da instalação. A partir da versão 6.0, as permissões são autorizadas na primeira execução do aplicativo. A figura 3 mostra um exemplo de solicitação da versão 5.1 e a figura 4 mostra a solicitação das versões 6.0 ou superiores.

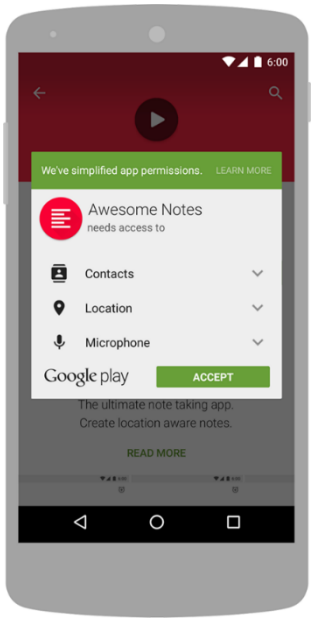


Figura 3 – Permissão nas versões 5.1 ou inferiores.

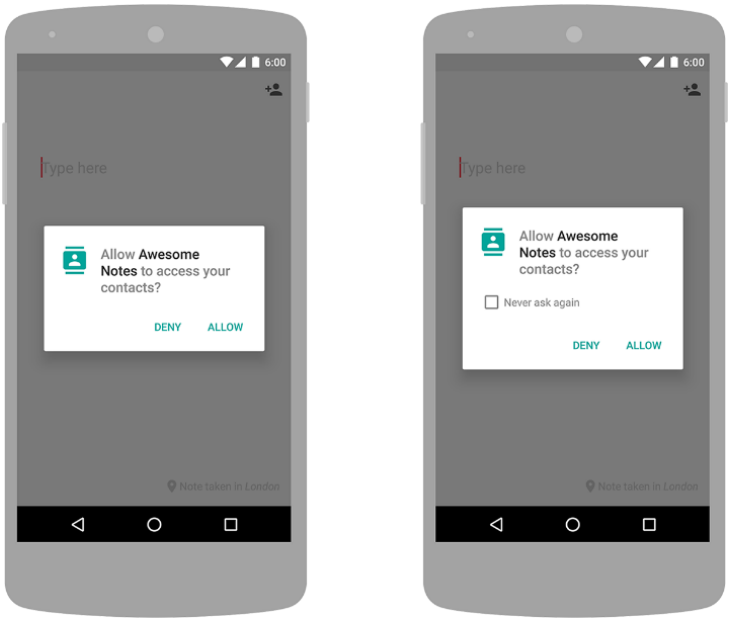


Figura 4 – Permissão nas versões 6.0 ou superiores.

As permissões devem ser declaradas como mencionado anteriormente no arquivo "AndroidManifest.xml" do seu projeto, para que durante a sua execução o sistema operacional verifique as suas intenções de utilização. A seguir, verifique o código de exemplo, onde o desenvolvedor solicita permissão ao usuário/sistema para enviar SMS.

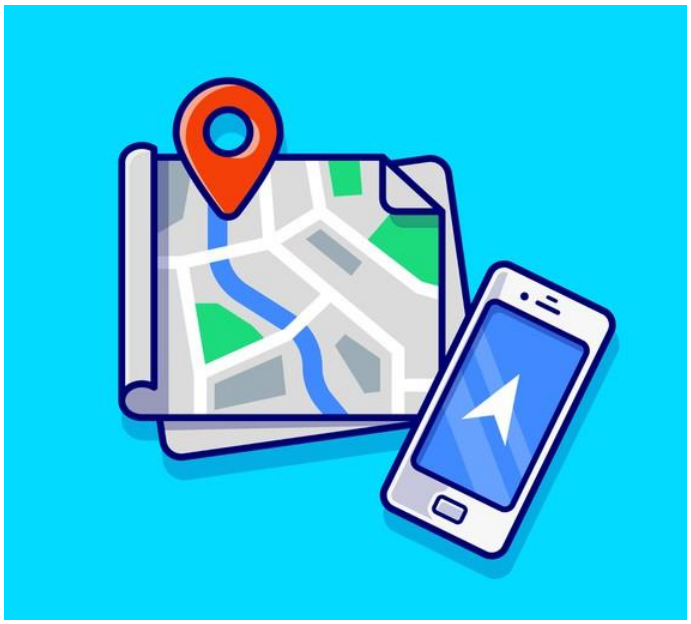
```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
          package="com.example.snazzyapp">

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

    <application ...>
        ...
    </application>
</manifest>
```

Uso de permissões de acesso aos recursos de Hardware

O sistema operacional Android também solicita permissões de uso de Hardware, e sabemos que nem todos os dispositivos são iguais, e muito menos oferecem todos os recursos como câmeras, GPS, Bluetooth etc. Isso pode implicar negativamente em um projeto. Desta maneira, é importante o desenvolvedor declarar as permissões de maneira correta, evitando assim aborrecimentos do usuário.



Imagine um jogo que utiliza o sistema de GPS e o usuário instala esse aplicativo em um dispositivo que não possui este recurso. Caso o desenvolvedor tenha esse cuidado com o código, podemos bloquear o funcionamento deste jogo neste aparelho.

Veja a seguir, um exemplo de código de solicitação de permissão para uso do Hardware da Câmera do dispositivo.

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

O complemento **android:required="false"** indica que o aplicativo poderá ser instalado em dispositivos que não tenham o hardware de câmera. Se o complemento for alterado para "**true",** automaticamente a instalação será barrada em aplicativos que não tenham o hardware de câmera.

O sistema operacional conta com outros tipos de permissão, desta forma é fundamental acompanhar na documentação oficial os outros tipos, e sempre que surgir alguma dúvida recorrer a esta documentação, disponível em:
https://developer.android.com/guide/topics/permissions/overview?hl=pt-br#normal-dangerous

Exemplo de utilização de permissões em um projeto

Este exemplo contempla apenas a solicitação de permissão ao sistema operacional Android, desta forma o vídeo a seguir demostra como efetuar as devidas configurações no arquivo "AndroidManifest.xml" e realiza o teste com o dispositivo virtual para exemplificar o processo de solicitação de permissão para o usuário.

https://www.youtube.com/watch?v=7cpkzblZBXo

Na sequência, verifique o arquivo “AndroidManifest.xml” utilizado no projeto.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.guilherme.permissao">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-feature android:name="android.hardware.camera" />
</manifest>
```

A seguir, verifique o arquivo Java “MainActivity.java” utilizado no projeto.

```
package com.example.guilherme.permissao;

import android.Manifest;
import android.content.pm.PackageManager;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ActivityCompat.requestPermissions(MainActivity.this, new String[]
{Manifest.permission.CAMERA}, 100);
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String[] permissions,
int[] grantResults) {
        switch( requestCode ){
            case 100 :
                if (grantResults.length > 0  &&  grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {

                    Toast.makeText(MainActivity.this, "Permissão Aprovada!",
                            Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(MainActivity.this, "Permissão Negada!",
                            Toast.LENGTH_SHORT).show();
                }
                break;
        }
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    }
}
```

Agora que já aprendemos com a Gabriela tudo sobre as permissões do sistema operacional Android, vamos praticar um pouco com a “Atividade Online”.