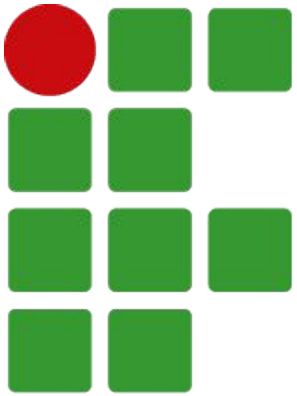


INSTITUTO FEDERAL
Paraná
Campus Paranaguá

Intent, Intent Filter e Permissões

Dispositivos Móveis

Prof. Diego Stiehl

INSTITUTO FEDERAL

Intent

- Intents (intenções)
  - A intenção de uma aplicação realizar algo
- São o coração do Android (LECHETA, 2015)
  - Na verdade estão mais para artérias (STIEHL, 2018)
- Elas são responsáveis pela troca de mensagens entre diferentes aplicações
  - Diferentes Activities de uma mesma app também

INSTITUTO FEDERAL

Intent - Fluxo

- Uma aplicação dispara uma mensagem ao sistema (*broadcast*)
  - Solicitando que algo seja realizado
- O sistema interpreta a mensagem
  - Decide o que fazer
  - Realiza a ação
    - Ou não

INSTITUTO FEDERAL

Uma Intent pode

- Enviar uma mensagem para o SO
- Abrir uma nova tela (Activity)
- Ligar para um número de celular
- Abrir o browser em determinada página
- Enviar um SMS ou e-mail
- Abrir o Google Maps em determinado endereço
- Abrir a galeria de fotos ou câmera padrão do celular
- Abrir o Google Play para instalar algum app
- …

INSTITUTO FEDERAL

Classe Intent

- Uma Intent, no Java (e Kotlin), é representada pela classe:
  - `android.content.Intent`
- Às vezes utilizaremos alguma subclasse dela

INSTITUTO FEDERAL

Voltando no Tempo

- Nas aulas passadas
  - Lembra de como iniciávamos outras Activities?

```
val intent = Intent(this, OutraActivity::class.java)
startActivity(intent)
```

- Utilizamos uma Intent para dizer ao sistema que desejamos iniciar uma instância da classe OutraActivity

INSTITUTO FEDERAL

Intent Explícita

- A Intent de abertura de uma Activity da nossa própria aplicação é explícita
  - Eu digo ao sistema que quero uma Activity específica (OutraActivity)
  - Funciona quando quero iniciar uma Activity do meu próprio aplicativo
    - Mesmo assim, a mensagem do Intent vai como um broadcast para o sistema e ele é quem trata da abertura

INSTITUTO FEDERAL

Intent Implícita

- Para trocar mensagens com outras aplicações, não sabemos exatamente o nome da classe
  - Aplicações ficam ouvindo (aguardando) por intents específicas no sistema
    - Quando alguma de interesse é disparada, uma ou mais aplicações podem responder
    - Mas não sabemos exatamente quem receberá a mensagem
- Neste caso, dispara-se uma intent implícita

INSTITUTO FEDERAL

Intent Implícita: Exemplo

- Desejo iniciar a activity responsável por fazer ligações no meu Android
  - Disparamos uma intent implícita
    - Não sei qual app/Activity vai responder por ela
    - O sistema tem aplicações registradas (ou não) esperando por determinados intents implícitos em específico
- Exemplo:

```
val uri = Uri.parse("tel:04199998888")
val intent = Intent(Intent.ACTION_CALL, uri)
startActivity(intent)
```

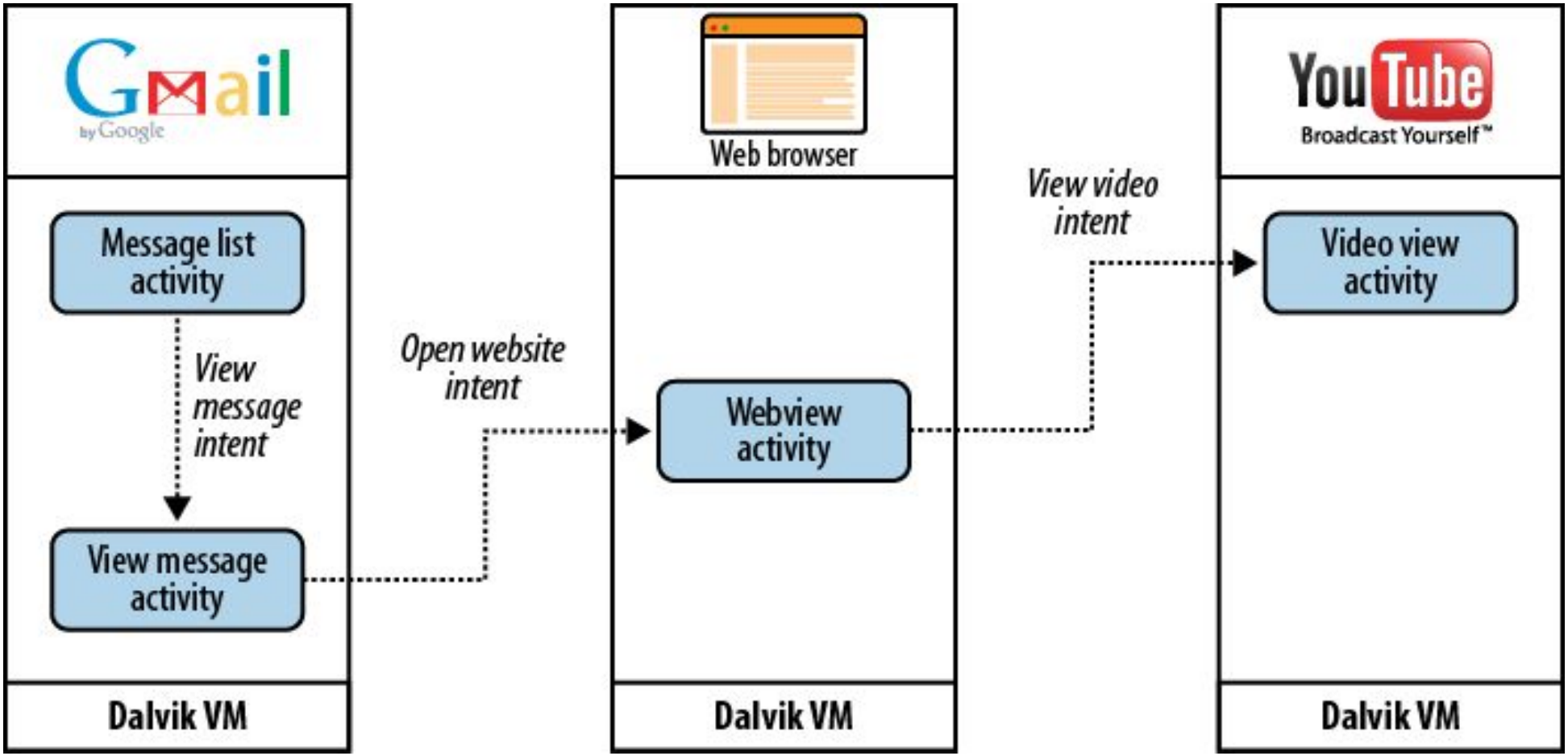
INSTITUTO FEDERAL

Constantes

- Intenções implícitas conhecidas pelo sistema têm constantes dentro da classe Intent
  - Fazer uma chamada
    - `Intent.ACTION_CALL`
  - Compartilhar (enviar) um conteúdo
    - `Intent.ACTION_SEND`
  - Ver algo dentro de determinado recurso
    - `Intent.ACTION_VIEW`
  - ...

INSTITUTO FEDERAL

Exemplo



Fonte: GARGENTA (2011)

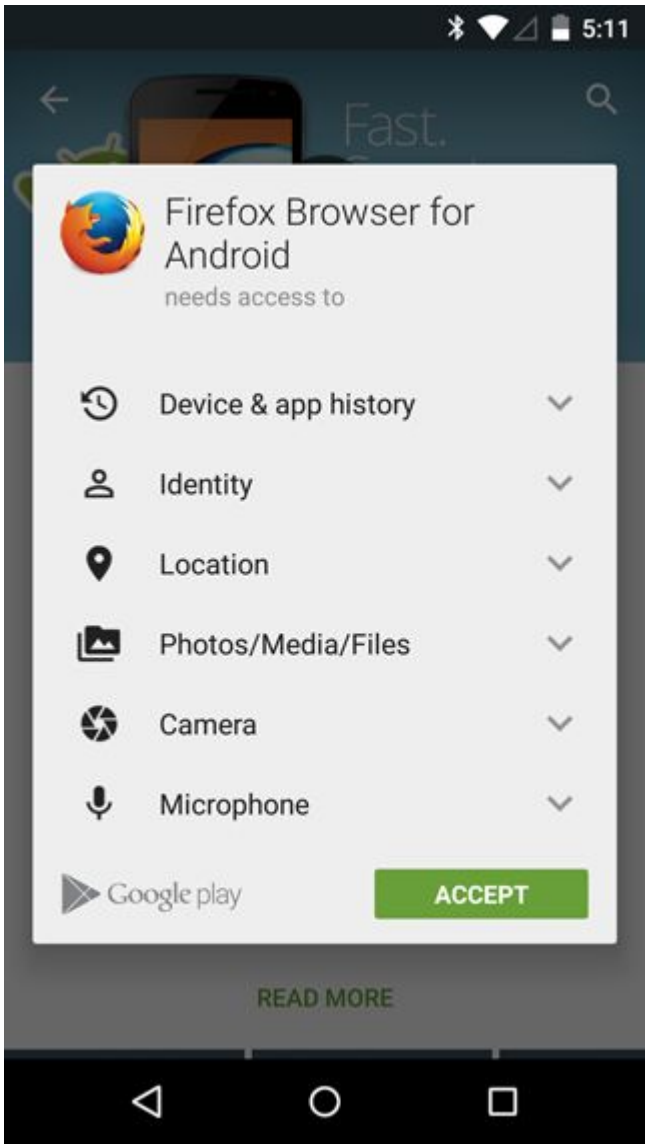
INSTITUTO FEDERAL

Permissões

- Algumas ações padrão suportadas pelo Android necessitam de permissão do usuário
  - Fazer uma ligação
  - Enviar um SMS
  - Abrir a câmera para tirar uma foto
  - Abrir a galeria para selecionar uma foto
  - Acessar os dados do cartão SD
  - …

Permissões – Antes

- Até antes do Android 6.0 (API Level < 23):
  - Todas as permissões solicitadas pelo aplicativo eram mostradas na instalação na Play Store
  - Usuário tinha que aceitar ou recusar todas



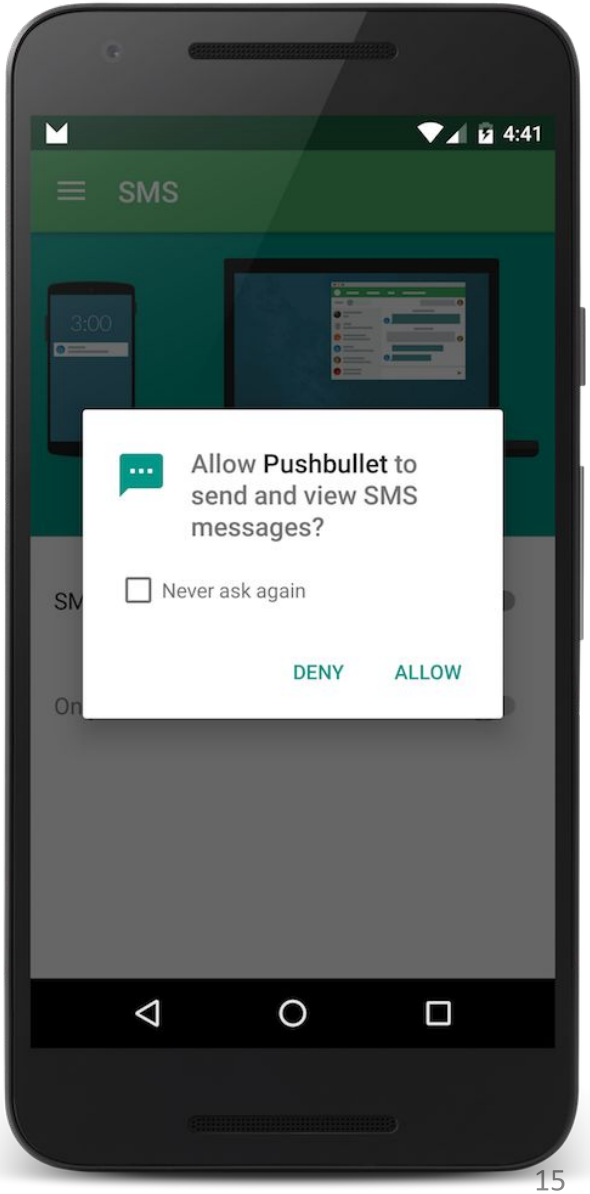
INSTITUTO FEDERAL

Permissões - Antes

- Era relativamente fácil declarar o uso de permissões em nossas aplicações
- No arquivo AndroidManifest.xml, adicionar
  - `<uses-permission android:name="android.permission.PERMISSAO" />`
- Exemplos
  - `<uses-permission android:name="android.permission.CALL_PHONE" />`
  - `<uses-permission android:name="android.permission.READ_CONTACTS" />`
  - `<uses-permission android:name="android.permission.CAMERA" />`

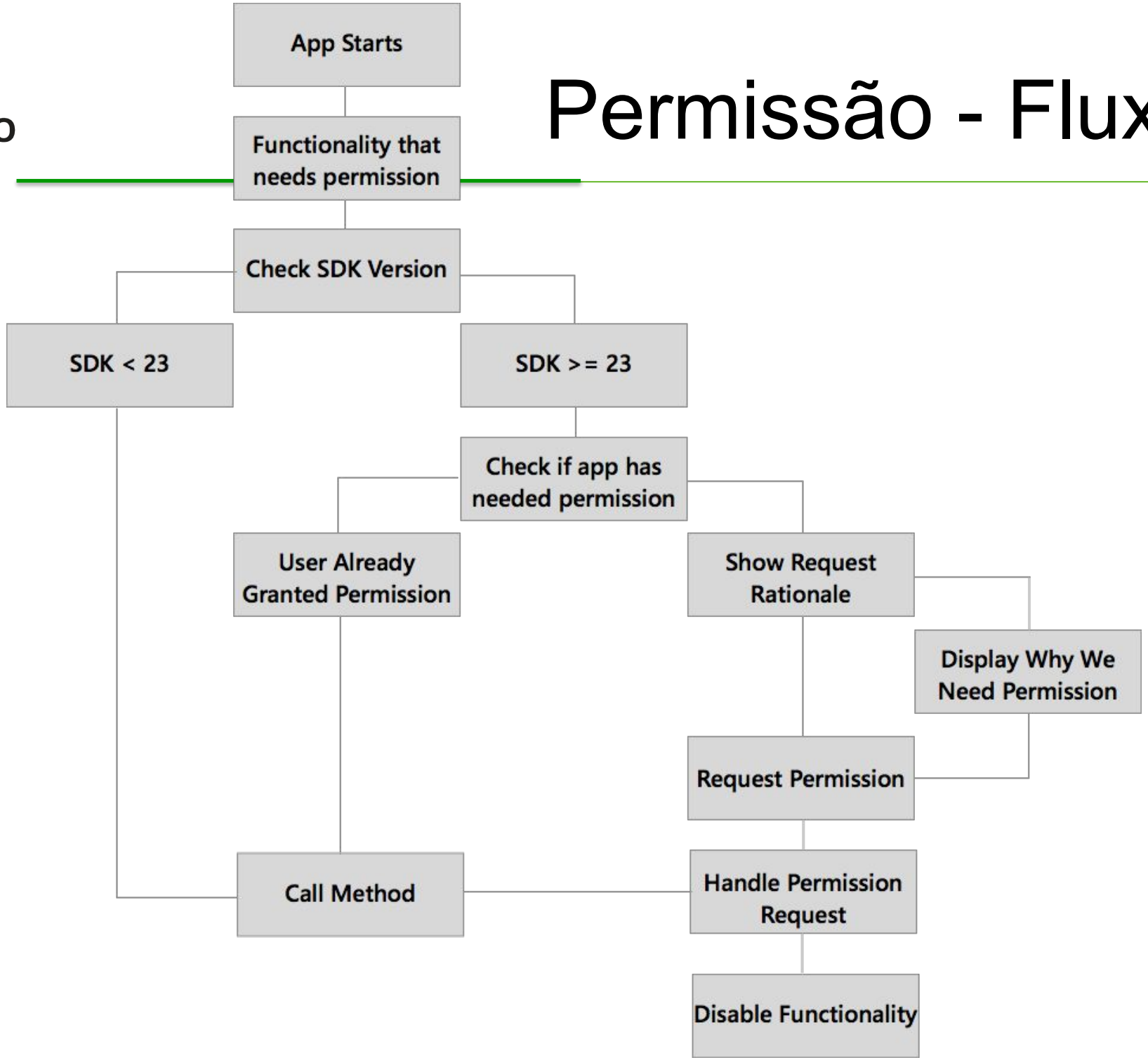
Permissões - Agora

- A partir do Android 6.0 (API Level >= 23):
  - As permissões são solicitadas dentro do aplicativo
    - Sob demanda
    - Uma solicitação para cada permissão necessária
  - Adicionou segurança e customização para o usuário
  - Dificultou a vida do programador



INSTITUTO FEDERAL

Permissão - Fluxo



https://blog.xamarin.com/requesting-runtime-permissions-in-android-marshmallow

Exemplo

```
private fun checkPermission() {
  if (ActivityCompat.checkSelfPermission(
        this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED)
    ActivityCompat.requestPermissions(this,
        arrayOf(Manifest.permission.ACCESS_FINE_LOCATION), 1)
  else
    executaAcaoQuePrecisaDePermissao()
}

override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<String>,
                                        grantResults: IntArray) {
  if (requestCode == 1)
    if (grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION_GRANTED)
      executaAcaoQuePrecisaDePermissao()
    else
      Toast.makeText(this, "Precisamos da sua localização!", Toast.LENGTH_SHORT).show()
}
```

Mais explicações: https://developer.android.com/training/permissions/requesting.html?hl=pt-br

INSTITUTO FEDERAL

Permissões – Android 6.0

- Mesmo com o código anterior:
  - Adicionar entrada no AndroidManifest.xml
- Documentação com todas as permissões:
  - https://developer.android.com/reference/android/Manifest.permission.html
- Somente as perigosas (dangerous) precisam ter a verificação reforçada no nosso código

INSTITUTO FEDERAL

Voltando aos Intents

- Enviar um e-mail

```
val emailIntent = Intent(Intent.ACTION_SEND)
with(emailIntent) {
    putExtra(Intent.EXTRA_SUBJECT, "Título do email")
    putExtra(Intent.EXTRA_TEXT, "Olá")
    putExtra(Intent.EXTRA_EMAIL, "diego.stiehl@ifpr.edu.br")
    type = "message/rfc822"
}
startActivity(emailIntent)
```

- Enviar um SMS

```
val uri = Uri.parse("sms:041999994444")
val smsIntent = Intent(Intent.ACTION_SENDTO, uri)
smsIntent.putExtra("sms_body", "Olha a mensagem!")
startActivity(smsIntent)
```

INSTITUTO FEDERAL

Mais Exemplos

- Abrir o navegador em um site

```
val uri = Uri.parse("http://google.com")
intent = Intent(Intent.ACTION_VIEW, uri)
startActivity(intent)
```

- Abrir o mapa em uma localização específica

```
val GEO_URI = "geo:-25.4089185,-49.3222402"
intent = Intent(Intent.ACTION_VIEW, Uri.parse(GEO_URI))
startActivity(intent)
```

INSTITUTO FEDERAL

Mais Exemplos

- Compartilhar

```
val shareIntent = Intent(Intent.ACTION_SEND)
with(shareIntent) {
    type = "text/plain"
    putExtra(Intent.EXTRA_SUBJECT, "Compartilhar")
    putExtra(Intent.EXTRA_TEXT, "Conteúdo!")
}
startActivity(shareIntent)
```

Mais Exemplos

- Tirar uma foto

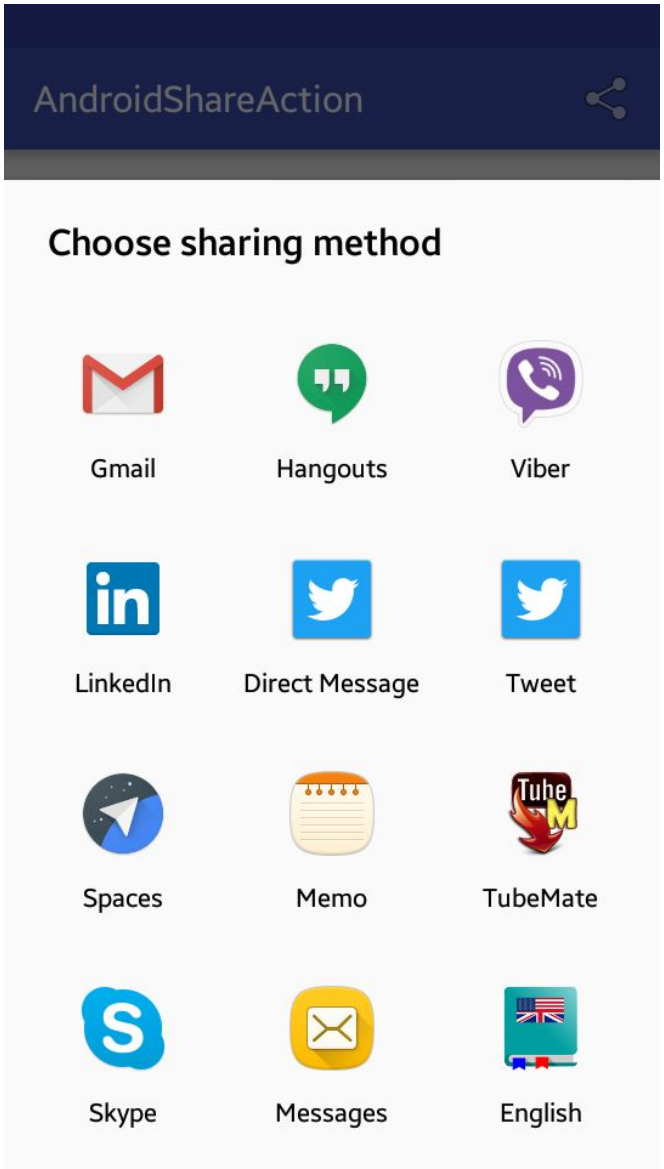
```
Intent fotoIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
startActivityForResult(fotoIntent, 9);
```

```
override fun onActivityResult(codigo: Int,
                              resultado: Int, it: Intent) {
    if (codigo == 9 && resultado == Activity.RESULT_OK) {
        val bundle = it.extras
        if (bundle != null) {
            val bitmap = bundle.get("data") as Bitmap
            mostrar(bitmap)
        }
    }
}
```

INSTITUTO FEDERAL

Quem vai abrir?

- Se houver somente uma aplicação/Activity que responda por uma intenção implícita
  - A Activity se abrirá
- Se houver mais:
  - Será mostrada uma tela de seleção
- Se nenhuma
  - Exception



Retorno

- Algumas intenções podem retornar dados de volta para minha aplicação
  - Exemplo – Selecionar um contato:

```
val uri = Uri.parse("content://com.android.contacts/contacts")
intent = Intent(Intent.ACTION_PICK, uri)
startActivityForResult(intent, 13)
```

  - O método `startActivityForResult()` dispara o listener `onActivityResult()`

```
override fun onActivityResult(codigo: Int,
                              resultado: Int, it: Intent) {
    if (codigo == 13 && resultado == Activity.RESULT_OK) {
        val uri = it.data
        Log.d("Contato", "Contato Uri: " + uri!!)
        val contato = Agenda.getContato(this, uri)
        mostraImagemContato(contato.getFoto(this))
    }
}
```

INSTITUTO FEDERAL

Intent Filter

- Intent Filter (Filtros de Intenção)
  - São o “outro lado” dos Intents
- Quando uma Intent é disparada para o sistema
  - O sistema verifica quais aplicações estão registradas para os parâmetros informados
    - A aplicação se registra via um Intent Filter
- Ou seja:
  - Uma aplicação dispara uma intenção
  - Outra aplicação está filtrando (esperando) por intenções de determinado tipo

INSTITUTO FEDERAL

Intents Customizadas

- Pode-se criar suas próprias intents customizadas
  - Outros aplicativos precisam conhecê-las
- Por enquanto vamos falar das intents relativas à ações padrão do sistema (*nativas*)
  - Compartilhar
  - Abrir câmera
  - Acessar cartão SD
  - …

Declarando Intent Filter

- Para dizer que uma Activity pode receber dados de outra aplicação via Intents
  - É só declarar um Intent Filter dentro da tag dela no arquivo AndroidManifest.xml
  - Isto que acontece com a Activity principal do app

```
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

INSTITUTO FEDERAL

Action/Category

- Todo Intent Filter é registrado com, pelo menos, uma ação e uma categoria
  - Ação é a mensagem específica que está sendo mandada
    - Normalmente é um nome único composto com o pacote
      - <action android:name="android.intent.action.MAIN" />
        - Indica que a aplicação é iniciável pelo usuário
  - Categoria contém informações adicionais sobre o tipo de componente que deve processar a intent
    - <category android:name="android.intent.category.LAUNCHER" />
      - Deixa presente na tela inicial do Android

Compartilhar Texto

- No AndroidManifest.xml

```
<activity android:name=".SomeActivity">
    <intent-filter>
        <action android:name="android.intent.action.SEND" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="text/plain" />
    </intent-filter>
</activity>
```

Intent Filter
Ação Nativa do Sistema

- Na Activity
  - Capturar parâmetro no onCreate() e fazer algo

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    if (Intent.ACTION_SEND == intent.action && intent.type != null) {
        if ("text/plain" == intent.type) {
            val sharedText = intent.getStringExtra(Intent.EXTRA_TEXT)
            if (sharedText != null)
                Toast.makeText(this, sharedText, Toast.LENGTH_SHORT).show()
        }
    }
}
```

INSTITUTO FEDERAL

Outros Exemplos

- Capturar pressionamento da tecla “Home” do dispositivo

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.HOME" />
    <category android:name="android.intent.category.DEFAULT" />
</intent-filter>
```

- Interceptar pedidos por câmera
  - Criar meu próprio aplicativo de câmera customizado

```
<intent-filter>
    <action android:name="android.intent.action.IMAGE_CAPTURE" />
    <category android:name="android.intent.category.DEFAULT" />
</intent-filter>
```

INSTITUTO FEDERAL

Implementação

- Cada Intent nativa que eu deseje interceptar com Intent Filters necessita de uma implementação específica na Activity
  - Câmera
  - Compartilhamento
  - Home Button
  - Ligação
  - Ver:
    - https://developer.android.com/guide/components/intents-common.html?hl=pt-br