

## Android: O que são Intents?

Componentes de aplicativos Android podem se conectar a outros aplicativos Android. Essa conexão é baseada em uma descrição de tarefa representada por um objeto Intent.

Intents são mensagens assíncronas que permitem que os componentes do aplicativo solicitem funcionalidades de outros componentes do Android. Intents permitem que você interaja com componentes dos mesmos aplicativos, bem como com componentes contribuídos por outros aplicativos. Por exemplo, uma atividade pode iniciar uma atividade externa para tirar uma foto.

“Intent é um objeto de mensagem que pode ser usado para solicitar uma ação de outro componente de aplicativo.

Em outras palavras, conforme estudamos (Unidade 1, Seção 1.2,) uma Activity é um componente de aplicativo Android.

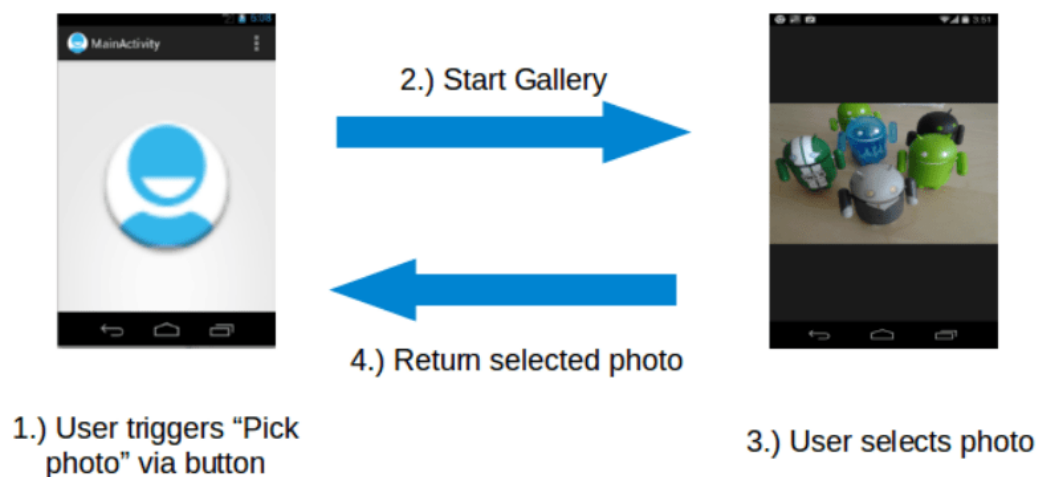
Uma Activity representa uma tela para o usuário interagir e, para que uma Activity possa trocar dados com outra e até mesmo iniciar outra, é necessário utilizar um objeto Intent.

As intenções são objetos do tipo `android.content.Intent`. Seu código pode enviá-los para o sistema Android, definindo os componentes que você está segmentando. Por exemplo, através do método `startActivity()` você pode definir que a intenção deve ser usada para iniciar uma atividade.

Uma intenção pode conter dados por meio de um Bundle. Esses dados podem ser usados pelo componente de recebimento.

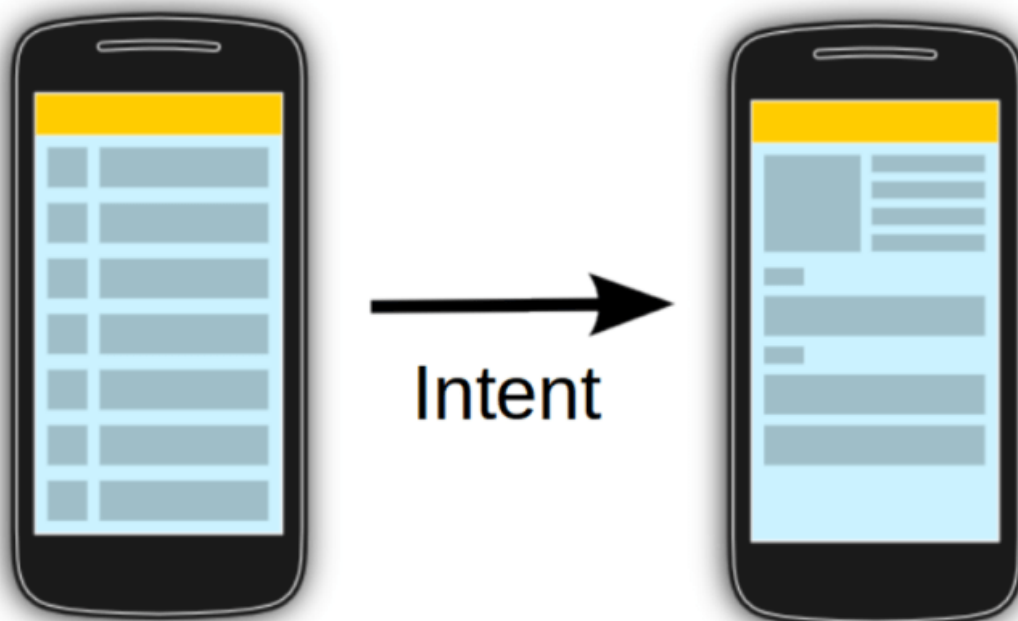
No Android, a reutilização de outros componentes de aplicativos é um conceito conhecido como tarefa. Um aplicativo pode acessar outros componentes do Android para realizar uma tarefa. Por exemplo, de um componente do seu aplicativo, você pode acionar outro componente no sistema Android, que gerencia fotos, mesmo que esse componente não faça parte do seu aplicativo. Neste componente, você seleciona uma foto e retorna à sua aplicação para usar a foto selecionada.

Tal fluxo de eventos é descrito no gráfico a seguir.



## Iniciando uma activitie ou serviço

Para iniciar uma atividade, use o método `startActivity (intent)`. Este método é definido no objeto `Contexto` que a `Activity` estende.



O código a seguir demonstra como você pode iniciar outra atividade por meio de uma intenção.

```
# Start the activity connect to the  
# specified class
```

```
Intent i = new Intent(this, ActivityTwo.class);
startActivity(i);
```

As atividades que são iniciadas por outras atividades do Android são chamadas de subatividades. Esta redação facilita a descrição de qual atividade é destinada.

Para iniciar um serviço por meio de intenções, use a chamada do método `startService (Intent)`

Existem dois tipos de objetos `Intent`. Para que possamos compreendê-los, analise as definições e os cenários abaixo.

i) `Intent Explícito`: conforme o nome sugere, `Intent Explícito` é utilizado quando o nome do componente Android que desejamos carregar está mencionado no objeto `Intent`.

Com o arquivo `AndroidManifest.xml` configurado desta forma, o sistema Android saberá que a `MainActivity` deverá ser iniciada ao executar o aplicativo.

```
<action android:name="android.intent.action.MAIN"/>
<category android:name="android.intent.category.LAUNCHER" />.
```

Observe o código abaixo do arquivo `AndroidManifest.xml`. Neste código são declaradas duas `Activities`, porém apenas uma delas possui o `Intent-filter` citado.

```
12. <activity android:name=".MainActivity">
13. <intent-filter>
14. <action android:name="android.intent.
    action.MAIN" />
15. <category android:name="android.intent.
    category.LAUNCHER" />
16. </intent-filter>
17. </activity>
18.
19. <activity android:name=".MenuActivity"></
    activity>
```

116 U2 - Armazenamento Key-Value e aplicações com Android™

ii) Intent Implícito: declara uma ação que deverá ser executada pelo sistema Android, porém o nome do componente que executará esta ação não está mencionado.

Cenário 1: você está desenvolvendo um aplicativo que terá duas Activities. A primeira refere-se à tela de login e a segunda refere-se à tela principal do aplicativo. Ao iniciar o aplicativo, a tela de login será exibida para o usuário. Após ele preencher os dados de login e pressionar o botão “Entrar”, os dados serão verificados pelo aplicativo. Se estiverem corretos, a segunda Activity será iniciada e deverá receber o usuário de login informado.

Neste cenário, vamos trabalhar com o Intent Explícito, pois nos permite informar explicitamente qual Activity será carregada. Ao declararmos um novo Intent, devemos informar dois parâmetros para o construtor:

- i) O primeiro refere-se a um objeto Context.
- ii) O segundo refere-se à Class do componente que será iniciado.

Observe o método abaixo que exemplifica o botão “Entrar” e apresenta os códigos para carregar a Activity MenuActivity.java.

```
31. private void realizarLogin(String usuario, String
senha) {
32. if ((usuario.equals("Aluno")) && (senha.equals("123")) ){
33. Intent intent = new Intent(this, MenuActivity.
class);
34. intent.putExtra("login", usuario);
35. startActivity(intent);
36. }
37. }
```

Linha 33: é criado um objeto Intent que recebe dois parâmetros:

i) o primeiro refere-se a um Context. Conforme estudamos na seção anterior, uma Activity é filha da classe Context, desta forma, ao informarmos o valor this para o parâmetro, estamos fazendo referência à Activity e à classe Context. Porém, caso esteja trabalhando com uma Inner Class (por exemplo, ao utilizar um Listener), chame o método getBaseContext() para recuperar a instância de Context. ii) o segundo parâmetro refere-se à Class do

componente, ou seja, devemos informar o nome da Activity que desejamos iniciar com o sufixo `.class`.

Linha 34: o método `putExtra()` trabalha com a estrutura key-value.

Este método nos permite adicionar informações extras para enviar com o Intent. No nosso exemplo enviaremos a chave “login” com o valor correspondente ao usuário.

Linha 35: utilizamos o método `startActivity(intent)` para iniciar uma Activity e passamos como parâmetro o Intent criado.