

Programação Para Dispositivos Móveis I

INTENTS

2024/_25 CTeSP – Desenvolvimento para a Web e Dispositivos Móveis

Ricardo Barbosa , rmb@estg.ipp.pt

Carlos Aldeias, cfpa@estg.ipp.pt

Índice

- Intents;
- Composição do Intent;
- Casos de uso dos Intents;
- Broadcast;
- Pending Intents;
- Leitura Adicional.

Intent

Intent representa uma descrição abstrata de uma operação a ser executada na plataforma Android;

- É apenas uma **estrutura de dados** que descreve uma operação a ser efetuada;
- Também ser visto como **uma mensagem** que é enviada ao sistema com instruções sobre operações a realizar;

Três dos componentes mais importantes do Android (Activity, Service e BroadcastReceiver) são iniciados através de mensagens (Intents).

Intent (2)

- Iniciar uma Activity

```
startActivity(Intent)
```

- Iniciar um Service

```
startService(Intent)
```

- Enviar broadcast

```
sendBroadcast(Intent)
```

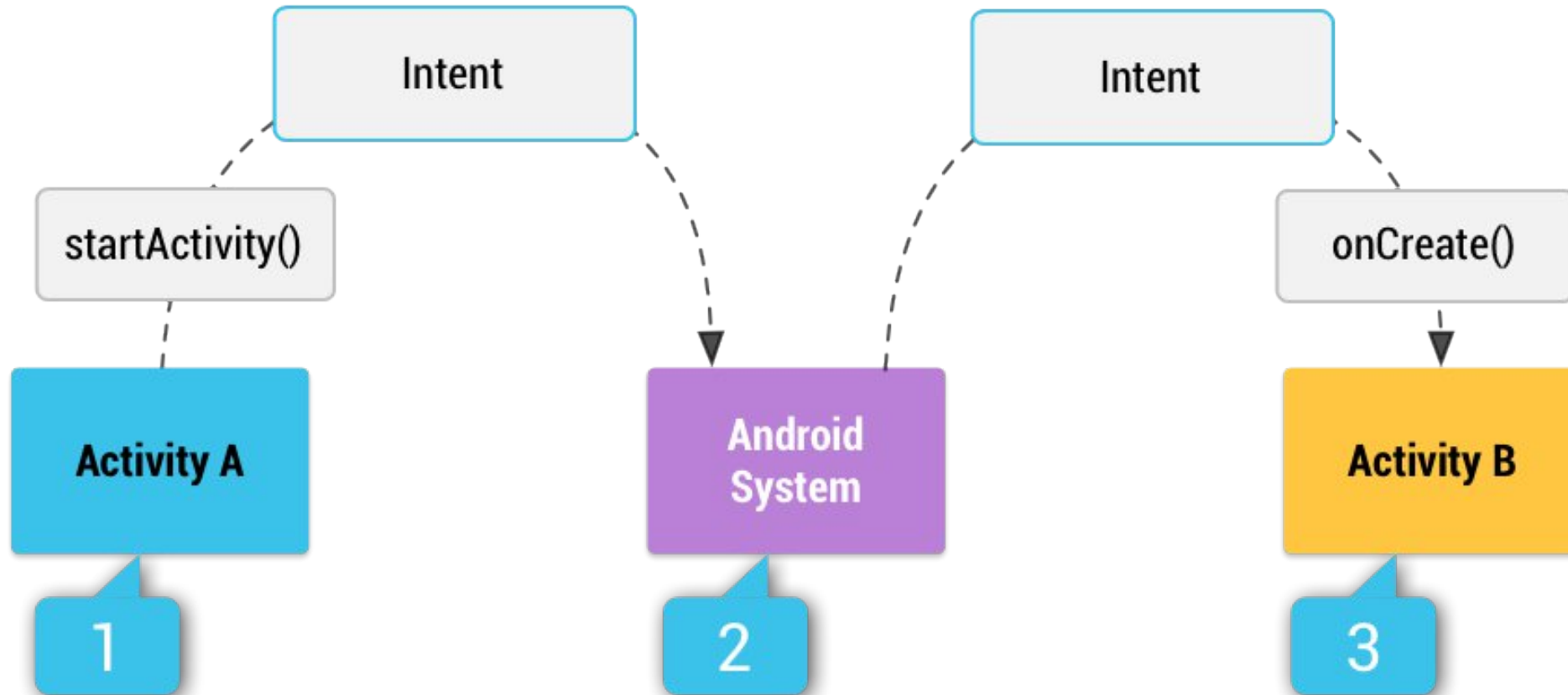
Para cada caso o Android encontra a Activity, Service ou BroadcastReceiver apropriado para receber o Intent e se necessário inicializa esse objeto.

Intent (2)

Futuramente iremos abordar os diferentes usos de Intents, mas neste momento iremos utilizar Intents apenas para:

- Iniciar atividades;
- Interagir com elementos da plataforma Android;
- Interagir com outras aplicações na plataforma Android.

Intent



Fonte: <https://developer.android.com/guide/components/intents-filters>

Composição de um Intent

Um Intent pode ser constituído pelos seguintes parâmetros:

- **Ação:** Uma String que representa a ação a ser desencadeada;
- **URI:** Identificador que representa as especificações do componente a inicializar;
- **Contexto:** Contexto da aplicação que contém a classe;
- **Classe:** Classe que o sistema terá de encontrar e instanciar.

Composição de um Intent

Passagem de Argumentos

Opcionalmente podemos com os nossos Intents passar argumentos via:

- Bundles;
- Extras.

No fundo são objetos semelhantes onde os argumentos são armazenados um a um ou todos de uma só vez.

Uso de Intents

Iniciar nova Atividade

Iniciar uma nova atividade e passar argumentos de uma atividade para outra.

`Intent(Context, Classe)`

- Contexto em que se insere a classe a inicializar;
- Classe a inicializar pelo Android;

Iniciar uma nova Activity com INTENT

```
Intent mIntent = new Intent(this, NewActivity.class);  
mIntent.putExtra("KEY", "VALUE");  
startActivity(mIntent);
```

Uso de Intents

Iniciar Aplicações -> Realizar uma chamada

Intent(Action, URI)

- A string refere-se à ação a ser realizada;
- O URI refere-se a um objeto de dados de um determinado tipo.

Realizar uma chamada através de um INTENT

```
Intent mIntent = new Intent(Intent.ACTION_CALL, Uri.parse("tel:+351255314002"));
startActivity(mIntent);
```

Uso de Intents

Iniciar Aplicações -> Visualizar um contacto existente

Intent(Action, URI)

- A string refere-se à ação a ser realizada;
- O URI refere-se a um objeto de dados de um determinado tipo.

Visualizar contacto existente através de um INTENT

```
Intent mIntent = new Intent(Intent.ACTION_VIEW, Uri.parse("content:/contacts/people/1"));
startActivity(mIntent);
```

Uso de Intents

Iniciar Aplicações -> Visualizar localização no mapa

Intent(Action, URI)

- A string refere-se à ação a ser realizada;
- O URI refere-se a um objeto de dados de um determinado tipo.

Visualizar localização no mapa através de um INTENT

```
Intent mIntent = new Intent(Intent.ACTION_VIEW, Uri.parse("geo:41.366788, -8.194751"));
startActivity(mIntent);
```

Uso de Intents

Iniciar Aplicações -> Iniciar direções no mapa

Intent(Action, URI)

- A string refere-se à ação a ser realizada;
- O URI refere-se a um objeto de dados de um determinado tipo.

Iniciar direções no mapa através de um INTENT

```
Intent mIntent = new Intent(Intent.ACTION_VIEW, Uri.parse("google.navigation:q=Felgueiras"));
startActivity(mIntent);
```

Uso de Intents

Iniciar Aplicações -> Extra

Mais casos de uso de Intents com aplicações do sistema Android podem ser encontradas em:

<https://developer.android.com/guide/components/intents-common>

Uso de Intents

Broadcast Intent

Intent(String)

- A String refere-se à ação a ser realizada.

Intent utilizado para enviar por Broadcast

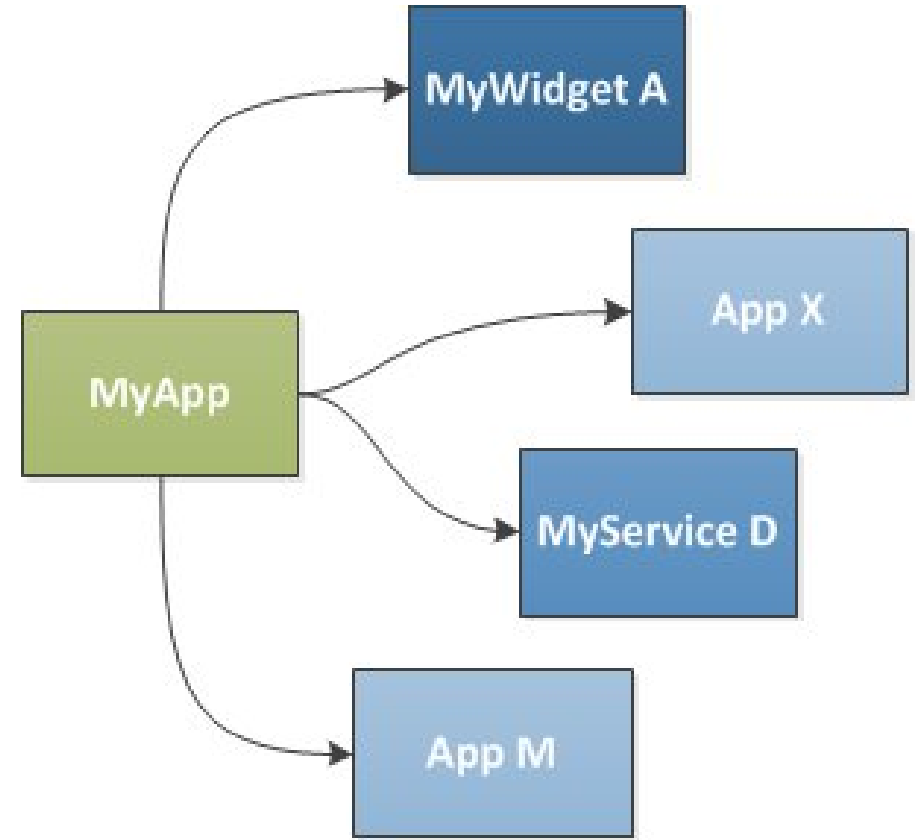
```
Intent mIntent = new Intent("pt.ipp.estg.pdm.DOWNLOAD_FINISHED");  
sendBroadcast(mIntent);
```

Broadcast

Utilizado para comunicar informações entre aplicações.

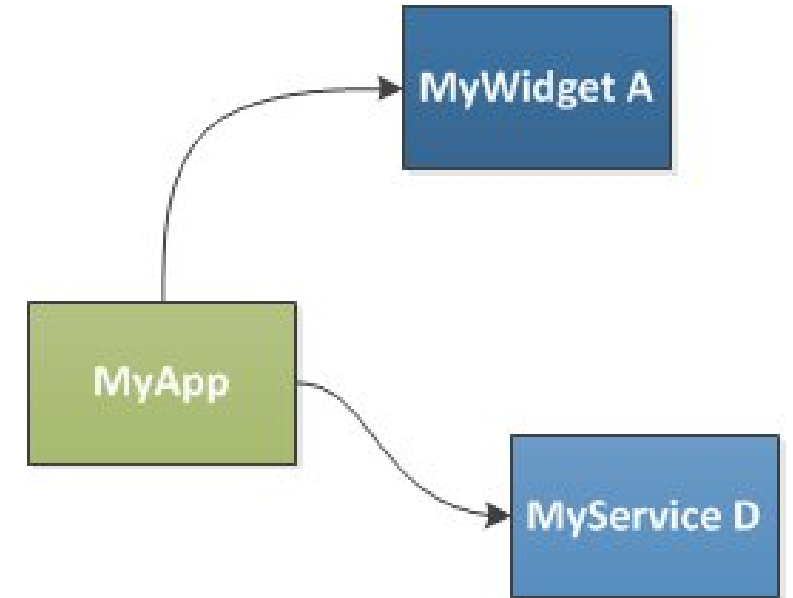
O método `Context.sendBroadcast()` envia um Intent para os BroadcastReceivers registados no sistema (que recebem esse mesmo Intent)

- `sendBroadcast(Intent intent)`



Broadcast (2)

Caso não seja necessário enviar Intents para o sistema, mas sim apenas entre os vários componentes da aplicação deve-se utilizar o método



`LocalBroadcastManager.getInstance(Context).sendBroadcast(Intent)`

Broadcast

Broadcast Receiver

Componente que é utilizado para receber Intents enviados pelo método `sendBroadcast()`.

Estes têm de ser registados no `AndroidManifest.xml` ou dinamicamente no código Java utilizando o método `Context.registerReceiver(BroadcastReceiver, IntentFilter)`

ou

`LocalBroadcastManager.getInstance(Context).registerReceiver(BroadcastReceiver, IntentFilter)`

Estes receivers devem ser registados no método `onResume()` da Activity ou Fragment.

No método `onPause()`, da Activity ou Fragment, deve-se remover esse registo
`unregisterReceiver(BroadcastReceiver receiver)`

Broadcast

Broadcast Receiver -> Intent Filter

Componente que é utilizado para informar o sistema operativo quais os Intents que o BroadcastReceiver vai tratar.

Definição em XML

```
<action android:name="pt.ipp.estg.pdm1.exemplo.mybroadcast" />
```

Definição em Java

```
IntentFilter mIF = new IntentFilter();  
mIF.addAction("pt.ipp.estg.pdm1.exemplo.mybroadcast");  
mIF.addAction("pt.ipp.estg.pdm1.exemplo.ourbroadcast");
```

Broadcast

Broadcast Receiver

É necessário criar uma nova classe do tipo `BroadcastReceiver` onde será também criado o método `onReceive()`

O método `onReceive()` **não foi desenhado** para realizar operações **assíncronas e pesadas**.

Considera-se como um processo em **foreground** e é mantido com uma **prioridade elevada**.

```
public class MyReceiver extends BroadcastReceiver {  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        // Código a implementar  
    }  
}
```

PendingIntent

Objeto para ser **utilizado por outras aplicações** para executarem uma ação descrita por nós;

- Ao passar o PendingIntent à outra aplicação, estamos-lhe a **garantir** o direito de executar operações conforme fosse a nossa aplicação (com as mesmas permissões e identidade);

O PendingIntent é simplesmente uma **referência para um token** que é mantido pelo sistema para descrever os dados que devem ser retornados;

- Mesmo que o processo da aplicação termine, o PendingIntent **é mantido no sistema para ser utilizado por outras aplicações.**

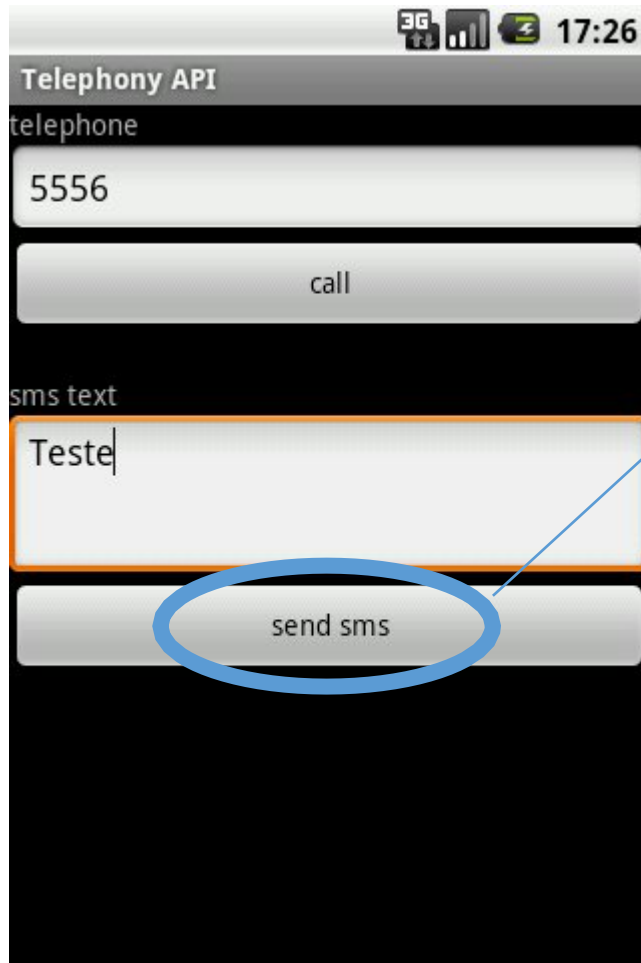
PendingIntent

O PendingIntent pode ser inicializado como

- `getActivity(Context, int [não utilizado], Intent, int);`
- `getService(Context, int [não utilizado], Intent, int);`
- `getBroadcast(Context, int [não utilizado], Intent, int);`

```
Intent mIntent = new Intent("STRING");  
mIntent.putExtra("KEY", "VALUE");  
PendingIntent mPI = getBroadcast(getApplicationContext(), 0, mIntent, 0);
```

PendingIntent



```
SmsManager.sendMessage(  
    NUMERO,  
    null,  
    TEXTO,  
    PendingIntent sent,  
    PendingIntent delivery)
```

Quando a SMS for
enviada envia-me
o Intent mIntent

sendBroadcast(mIntent)

Leitura Adicional

- Intents:

<https://developer.android.com/guide/components/intents-filters#Receiving>

- Common Intents:

<https://developer.android.com/guide/components/intents-common>

- Intents Google Maps:

<https://developers.google.com/maps/documentation/urls/android-intents>

Programação Para Dispositivos Móveis I

INTENTS

2024/_25 CTeSP – Desenvolvimento para a Web e Dispositivos Móveis

Ricardo Barbosa , rmb@estg.ipp.pt

Carlos Aldeias, cfpa@estg.ipp.pt

Adaptação do conteúdo dos slides de João Ramos jrmr@estg.ipp.pt e Fábio Silva fas@estg.ipp.pt