



BROADCAST RECEIVERS

Professor Emerson Alencar
emerson@imd.ufrn.br

BROADCAST RECEIVERS

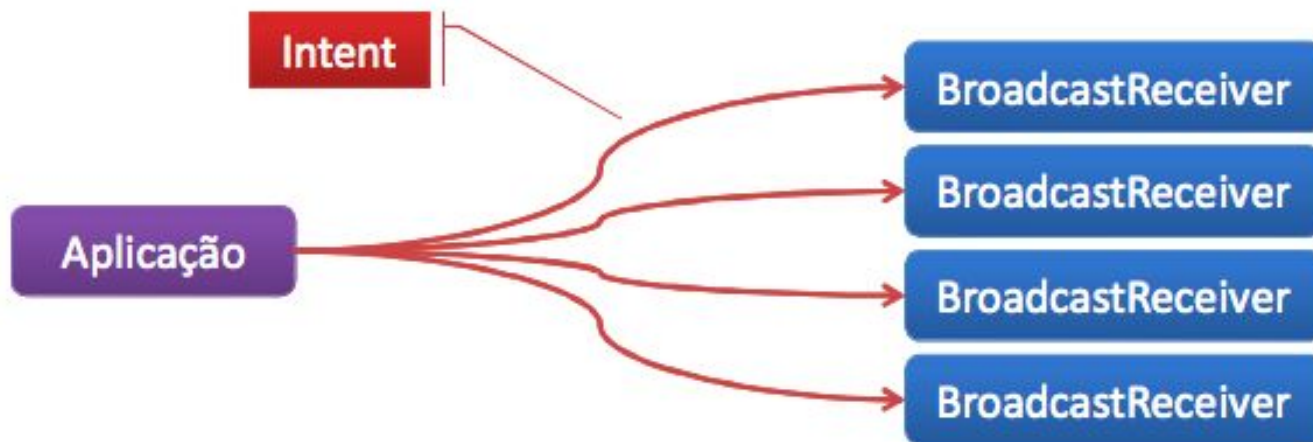
- ▶ No Android é possível detectar quando eventos do sistema operacional acontecem por meio de **Broadcast Receivers**
- ▶ Eventos customizados também são possíveis de dispará-los semelhante aos eventos de sistema

BROADCAST RECEIVERS

- ▶ Um componente broadcast receiver funciona de forma análoga a uma activity
 - Iniciados através de intents
 - Usam intent filters para decidir quais intents devem receber
- ▶ O Broadcast Receiver não trabalha atrelado à interface gráfica
 - Ele executa em segundo plano, sem o conhecimento do usuário

BROADCAST RECEIVERS

- ▷ Os broadcast receivers reagem a eventos disparados em forma de intents
 - Estes eventos podem ser das suas próprias aplicações ou de aplicações nativas do Android



A CLASSE BroadcastReceivers

- ▶ Um broadcast receiver é representado pela classe **BroadcastReceiver**
- ▶ Basta estender esta classe e implementar o método **onReceive()**

```
public class MyBroadcastReceiver extends BroadcastReceiver {  
    public void onReceive(Context context, Intent intent) {  
        //...  
    }  
}
```

Contexto de
invocação

Intent utilizada

CONFIGURANDO BROADCASTRECEIVERS

- ▶ A configuração pode ser feita de duas formas
 - Estática (no arquivo AndroidManifest.xml)
 - Dinâmica (no código)
- ▶ Na configuração estática, o broadcast receiver pode receber a intent mesmo que a aplicação não esteja sendo executada
- ▶ Na configuração dinâmica, a aplicação deve estar executando

Por questões de desempenho, declarar receivers no AndroidManifest.xml é um recurso que foi removido parcialmente no Android 7.0 (API LEVEL 24) e quase completamente a partir do Android 8.0 (API LEVEL 26), restando apenas alguns poucos eventos que ainda podem ser capturados

CONFIGURAÇÃO ESTÁTICA

- ▷ Broadcast receivers são configurados no arquivo AndroidManifest.xml

```
<receiver android:name=".MyReceiver">  
  <intent-filter>  
    <action android:name="company.android.action.COMPLETED" />  
  </intent-filter>  
</receiver>
```

Classe do broadcast receiver

Intent filters que determinam quais intents podem ser recebidas

CONFIGURAÇÃO DINÂMICA

- ▷ Broadcast receivers são configurados pelo código
- ▷ São usados dois métodos
 - `registerReceiver()`
 - `unregisterReceiver()`

CONFIGURAÇÃO DINÂMICA

- ▷ Broadcast receivers são configurados pelo código
- ▷ São usados dois métodos
 - `registerReceiver()`
 - `unregisterReceiver()`

```
MyToastReceiver receiver;  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    receiver = new MyToastReceiver();  
    IntentFilter intentFilter = new IntentFilter();  
    intentFilter.addAction("com.emersonalencar.android.broadcast.TOAST");  
    registerReceiver(receiver, intentFilter);  
}
```

ENVIANDO MENSAGEM

- ▶ Para que os broadcast receivers registrados na plataforma recebam as intents, elas devem ser disparadas
- ▶ Isto é feito através do método **sendBroadcast()**

```
Intent intent = new Intent( action: "com.emersonalencar.android.broadcast.TOAST");  
intent.putExtra( name: "msg", value: "Mensagem do BroadcastReceiver");  
  
sendBroadcast(intent);
```

ENVIANDO MENSAGEM

- ▷ O método `sendBroadcast()` é assíncrono
- ▷ A intent é enviada a todos os broadcast receivers ao mesmo tempo
- ▷ Não existe garantia de ordem na entrega das intents

CICLO DE VIDA

- ▶ O objeto que representa o broadcast receiver fica ativo apenas durante a execução do método `onReceive()`
 - Depois disso o Android pode destruí-lo
- ▶ Devido a isso, operações assíncronas não são permitidas
- ▶ O método `onReceive()` deve completar em até 10 segundos
 - Não é possível realizar processamento "pesado" na entrega das intents
 - Se não concluir em até 10 segundo, o Android exibirá uma mensagem ANR (*Application Not Responding*)

RECOMENDAÇÕES

- ▶ Broadcast receivers não devem interferir no que o usuário está fazendo
 - Exibir caixas de diálogo, abrir activities, etc.
 - A forma correta é utilizar o serviço de notificação para avisar o usuário
- ▶ Apesar de executar em segundo plano, os broadcast receivers não devem executar código que demanda tempo
 - Limite de 10 segundos para execução – Services são recomendados nestes casos

EVENTOS BROADCAST NATIVOS

- ▶ O Android faz o broadcast de intents para avisar sobre eventos internos do sistema
- ▶ As aplicações podem registrar broadcast receivers para serem notificadas a respeito destes eventos
 - ACTION_POWER_CONNECTED
 - ACTION_POWER_DISCONNECTED
 - ACTION_BOOT_COMPLETED
 - ACTION_BATTERY_CHANGED
 - etc.

EVENTOS BROADCAST NATIVOS

- ▶ A partir do Android 8 (API Level 26), broadcasts nativos não podem ser interceptados por broadcast receivers configurados de forma estática
 - A configuração precisa ser dinâmica

EXPORTANTO BROADCAST

- ▷ O atributo **exported** pode ser utilizado para indicar se um broadcast receiver é exportado ou não

```
<receiver ...  
    exported = "true" />
```

Um broadcast receiver exportado pode receber broadcasts de outros aplicativos

```
<receiver ...  
    exported = "false" />
```

Um broadcast receiver não exportado pode receber apenas broadcasts do próprio aplicativo

- ▷ Se o atributo for omitido
 - Assume false se não houver um intent filter
 - Assume true se houver um intent filter

Obrigado!
Dúvidas?

Professor Emerson Alencar
emerson@imd.ufrn.br