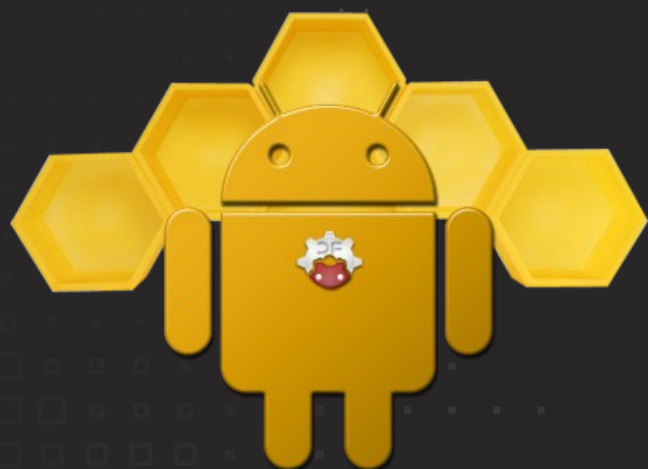




Desenvolvimento de Aplicações Android

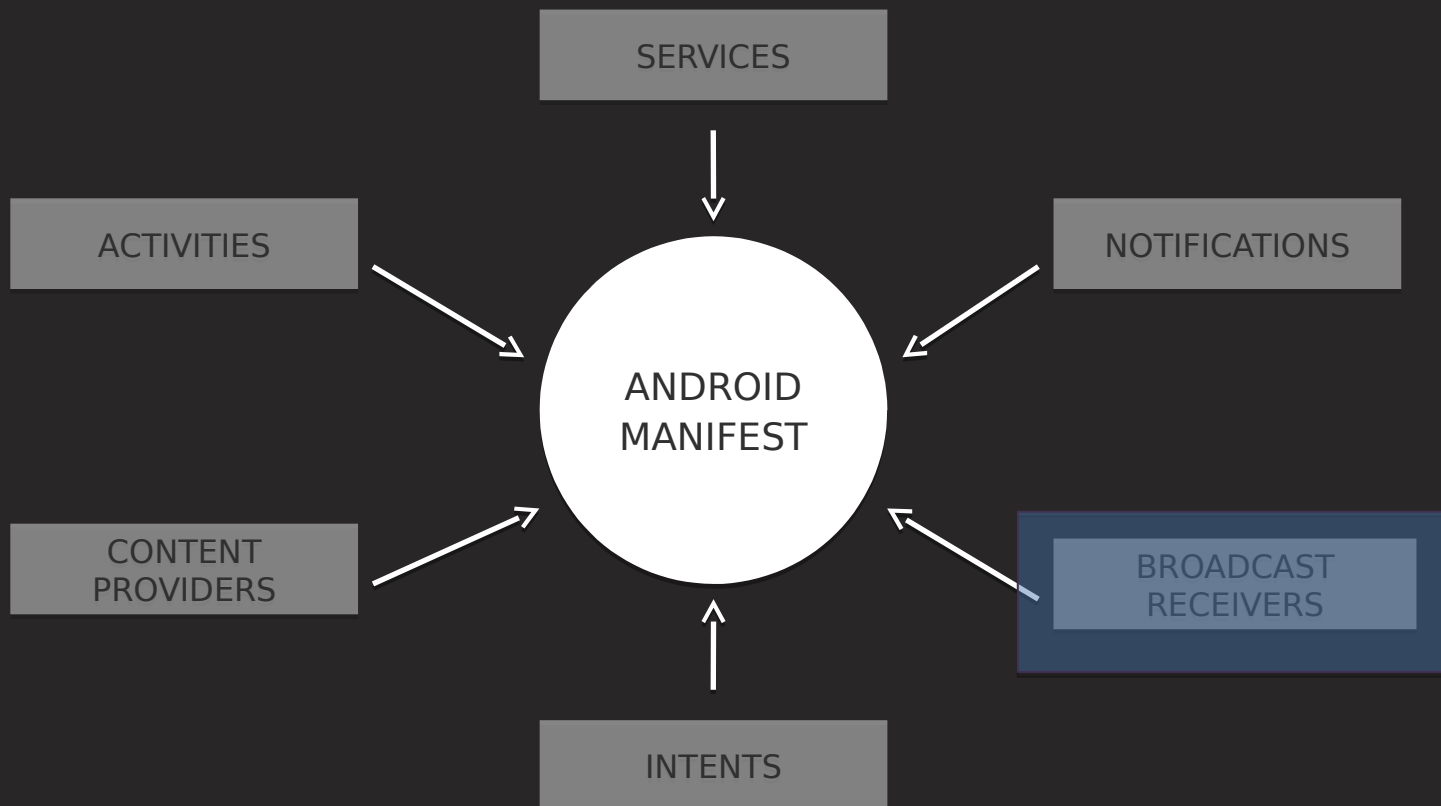


DESENVOLVIMENTO ANDROID

BROADCAST RECEIVER E NOTIFICATION

ÍNDICE

- BROADCAST RECEIVERS
- NOTIFICATIONS



Android Manifest é um arquivo XML (*AndroidManifest.xml*) que define e integra os componentes de uma aplicação vistos acima.

São classes receptoras que podem ser acionadas via Intent;

Para criar um Broadcast Receiver basta estender a classe **BroadcastReceiver**;

Ao ser acionado um Broadcast Receiver executa o método **onReceive()**;

Já o método **sendBroadcast()** serve para acionar os Broadcast Receivers;

É necessário também registrar o Broadcast Receiver junto ao **AndroidManifest**;

android.intent.action.**ACTION_BOOT_COMPLETED**:

indica que o telefone acabou de ser ligado

android.intent.action.**ACTION_POWER_CONNECTED**

indica que o carregador foi conectado

android.intent.action.**ACTION_BATTERY_LOW**

indica que o nível de bateria está baixo.

BROADCAST RECEIVER

Exemplo:

```
public class MeuBroadcast extends BroadcastReceiver{  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        Log.i("MeuBroadcast", "Recebido...");  
    }  
}
```

Para acioná-lo:

```
Intent i = new Intent(this, MeuBroadcast.class);  
sendBroadcast(i);
```

BROADCAST RECEIVER

Um Broadcast Receiver pode ser configurado para executar quando determinado evento ocorrer no dispositivo, por exemplo, o recebimento de uma chamada telefônica ou mensagem;

Para tanto basta criar um **intent-filter** para o Receiver no **AndroidManifest**:

Exemplo para chamada telefônica e mensagem:

```
<receiver android:label="Meu Broadcast" android:name="broadcast.MeuBroadcast">
    <intent-filter>
        <action android:name="android.intent.action.PHONE_STATE"/>
        <action android:name="android.provider.Telephony.SMS_RECEIVED"/>
    </intent-filter>
</receiver>
```

Obs.: necessário somente até o Marshmallow

Também é necessário incluir uma autorização para ler os estados do telefone /recebimento de SMS:

```
<uses-permission  
android:name="android.permission.READ_PHONE_STATE" />  
  
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
```

BROADCAST RECEIVER

O exemplo abaixo mostra como obter uma mensagem SMS enviada e o número do remetente:

```
Bundle myBundle = intent.getExtras();
SmsMessage [] messages = null;
String strMessage = "";

if (myBundle != null) {
    Object[] pdus = (Object[]) myBundle.get("pdus");

    messages = new SmsMessage[pdus.length];

    for (int i = 0; i < messages.length; i++) {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            String format = myBundle.getString("format");
            messages[i] = SmsMessage.createFromPdu((byte[]) pdus[i], format);
        } else {
            messages[i] = SmsMessage.createFromPdu((byte[]) pdus[i]);
        }
        strMessage += "SMS From: " + messages[i].getOriginatingAddress();
        strMessage += " : ";
        strMessage += messages[i].getMessageBody();
        strMessage += "\n";
    }

    Toast.makeText(context, strMessage, Toast.LENGTH_SHORT).show();
}
```

BROADCAST RECEIVER

Você pode configurar um Broadcast Receiver para ser executado na inicialização (BOOT) do dispositivo;

Para tanto, basta registrá-lo como um intent-filter **BOOT_COMPLETED**;

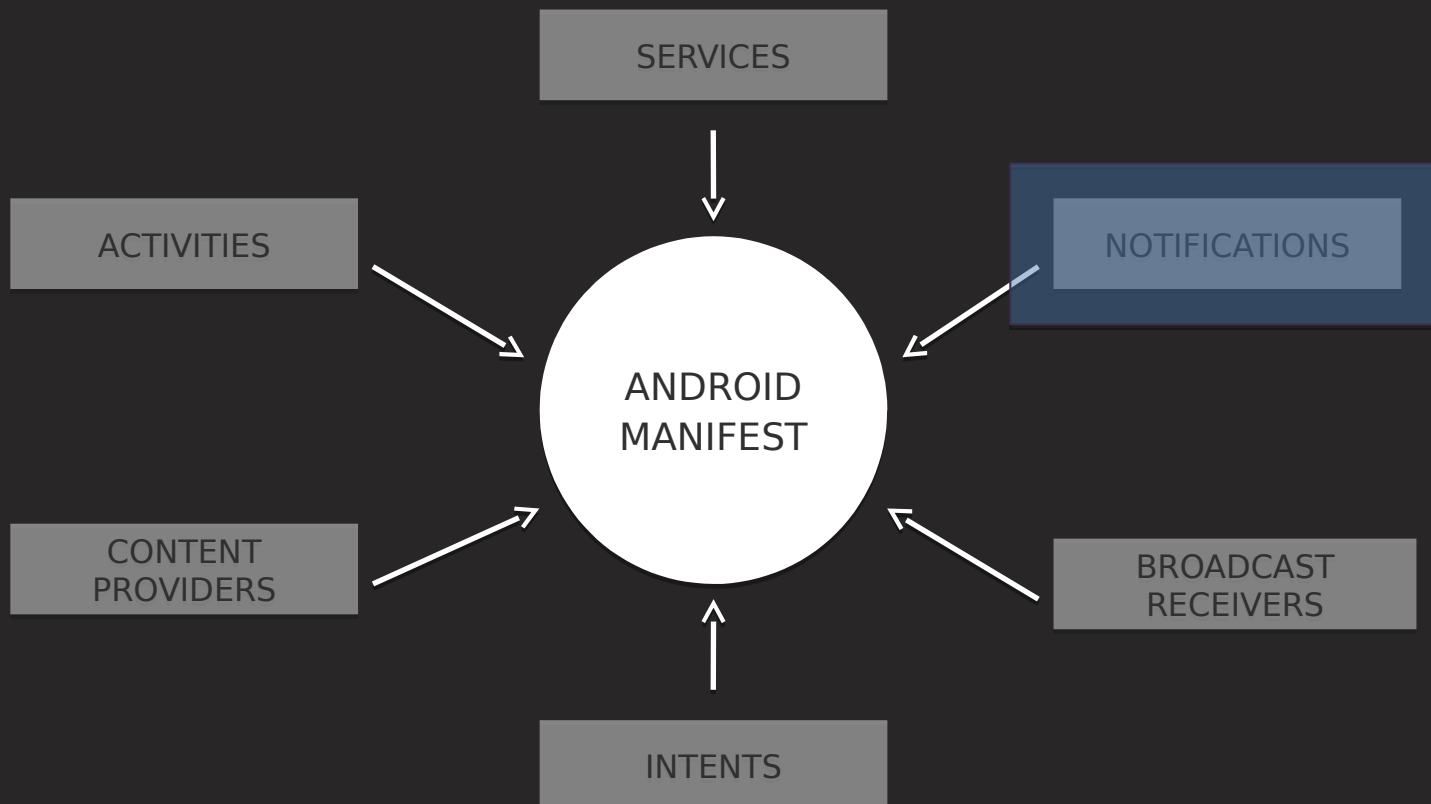
Também é necessário dar a permissão de **RECEIVE_BOOT_COMPLETED** no AndroidManifest;

```
<receiver android:name="broadcast.BoaNoiteBroadcast">
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED"/>
    </intent-filter>
</receiver>

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

</application>
</manifest>
```

Manifest [A] Application [P] Permissions [I] Instrumentation [AndroidManifest.xml]



Android Manifest é um arquivo XML (*AndroidManifest.xml*) que define e integra os componentes de uma aplicação vistos acima.

As notificações oferecem um recurso interessante para comunicar ao usuário sobre a execução ou resultado de determinada operação (principalmente quando associados a serviços em background);

As notificações são gerenciadas por meio do **NotificationManager** que deve ser obtido dos serviços padrão do sistema (**NOTIFICATION_SERVICE**):

```
mNotificationManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
```

As notificações podem ser criadas por meio da classe Notification.Builder:

```
NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(this);  
  
mBuilder.setContentTitle("New Message");  
mBuilder.setContentText("You've received new message.");  
mBuilder.setTicker("New Message Alert!");  
mBuilder.setSmallIcon(R.drawable.leandro);
```

E acionadas/canceladas através dos métodos **notify** e **cancel**:

```
mNotificationManager.notify(100, mBuilder.build());  
mNotificationManager.cancel(100);
```

Ações com **PendingIntent**

```
nb.setContentIntent(PendingIntent.getActivity(this, 0,  
new Intent(this, LoginActivity.class), PendingIntent.FLAG_UPDATE_CURRENT));
```

Vibração na notificação

```
Notification n = nb.build();  
n.vibrate = new long[]{150, 300, 150, 600}; // 150 = espera, 300 e 600 = vibração
```



Copyright © 2016 - Profs. Me. Leandro Rubim, Prof. Me. Thiago T. I. Yamamoto e Prof. Me. Edson Sensato

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Autor.

