

Broadcast Receiver, Notification e Alarm Manager

Bruno Lopes Alcantara Batista

Prazer em conhecer,

Bruno Lopes A Batista

Me. Ciência da Computação - UECE

Esp. em Desenvolvimento Web com JavaEE - FJN

Bel. em Sistemas de Informação - FJN

Professor Pesquisador - LARCES

Gerente de Projetos - LARCES

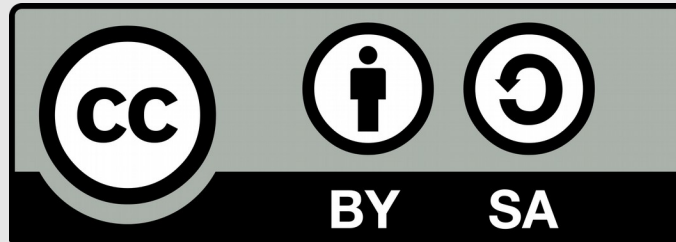
Gerente de Projetos - Athomustec

bruno@larces.uece.br

<http://www.larces.uece.br/bruno>



Copyleft



*Esta obra está licenciado com uma Licença
[Creative Commons Atribuição-CompartilhaIgual 4.0 Internacional](https://creativecommons.org/licenses/by-sa/4.0/)*

Agenda

- ☐ Broadcast Receiver
- ☐ Notification
- ☐ Alarm Manager
- ☐ Let's try it!

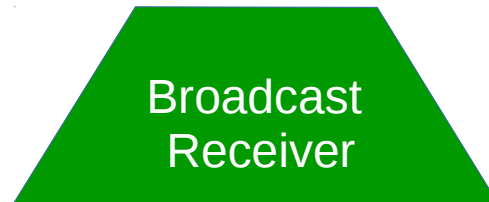
Broadcast Receiver

Broadcast Receiver

- ❑ Broadcast Receiver é a implementação Android do padrão *publish/subscribe messaging pattern*.
- ❑ Mais precisamente o padrão **Observer**.
- ❑ Uma aplicação, denominada *publisher*, pode gerar eventos via broadcast.
- ❑ E aplicações interessadas, denominadas *subscribers*, recebem esses eventos baseado em filtros (Intent-Filters).
- ❑ Podem ser declarados estaticamente, via AndroidManifest ou dinamicamente, via código Java.

Broadcast Receiver

- Durante esse curso, representaremos um broadcast receiver visualmente da seguinte maneira:



Broadcast Receiver

- ❑ Broadcast Receiver é uma classe Java que herda da classe BroadcastReceiver.
- ❑ Essa classe deve implementar o método onReceive()

```
10 public class MyBroadcastReceiver extends BroadcastReceiver {  
11  
12     @Override  
13     public void onReceive(Context context, Intent intent) {  
14  
15         // TODO: Tratar o evento disparado via Broadcast Receiver  
16  
17     }  
18  
19 }  
20
```


Broadcast Receiver

- Para registrar estaticamente o Broadcast Receiver via AndroidManifest, fazemos:

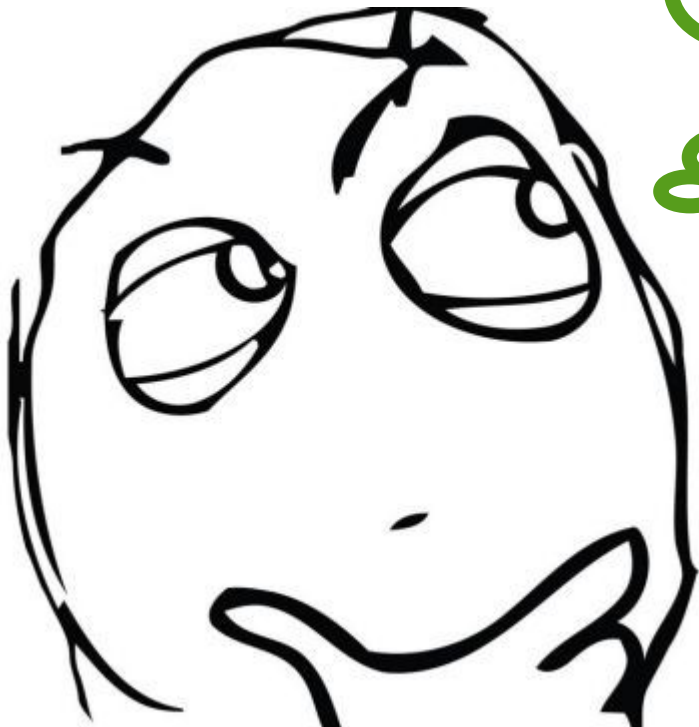
```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3          package="br.edu.fa7.welcome" >
4
5      <application
6          android:allowBackup="true"
7          android:icon="@mipmap/ic_launcher"
8          android:label="Welcome"
9          android:theme="@style/AppTheme" >
10
11          <!-- Declaração das Activities -->
12
13          <receiver android:name=".MyBroadcastReceiver">
14              <intent-filter>
15                  <action android:name="android.intent.action.BOOT_COMPLETED"/>
16                  <category android:name="android.intent.category.DEFAULT"/>
17              </intent-filter>
18          </receiver>
19
20      </application>
21
22  </manifest>
23
```

Broadcast Receiver

- Para registrar dinamicamente o Broadcast Receiver via Java, fazemos:

```
11 public class NewActivity extends Activity {
12
13     private MyBroadcastReceiver myBroadcastReceiver;
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_new);
19
20         this.myBroadcastReceiver = new MyBroadcastReceiver();
21     }
22
23     @Override
24     protected void onStart() {
25         super.onStart();
26         registerReceiver(myBroadcastReceiver, new IntentFilter(Intent.ACTION_BOOT_COMPLETED));
27     }
28
29     @Override
30     protected void onStop() {
31         super.onStop();
32         unregisterReceiver(myBroadcastReceiver);
33     }
34
35 }
36
```

Broadcast Receiver



Certo, e onde eu uso isso?

Broadcast Receiver

- ❑ Exemplos de usos:
 - ❑ Término do boot de dispositivo;
 - ❑ Saber a respeito do estado da bateria;
 - ❑ Saber quando um dispositivo entra/sai do modo avião;
 - ❑ Notificações do processo de discovery do bluetooth;
 - ❑ Eventos próprios;
 - ❑ E etc.

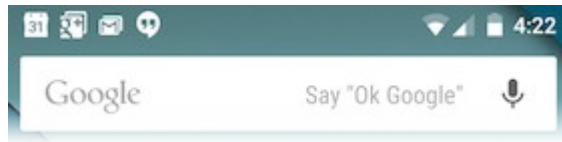


Notification

Notification

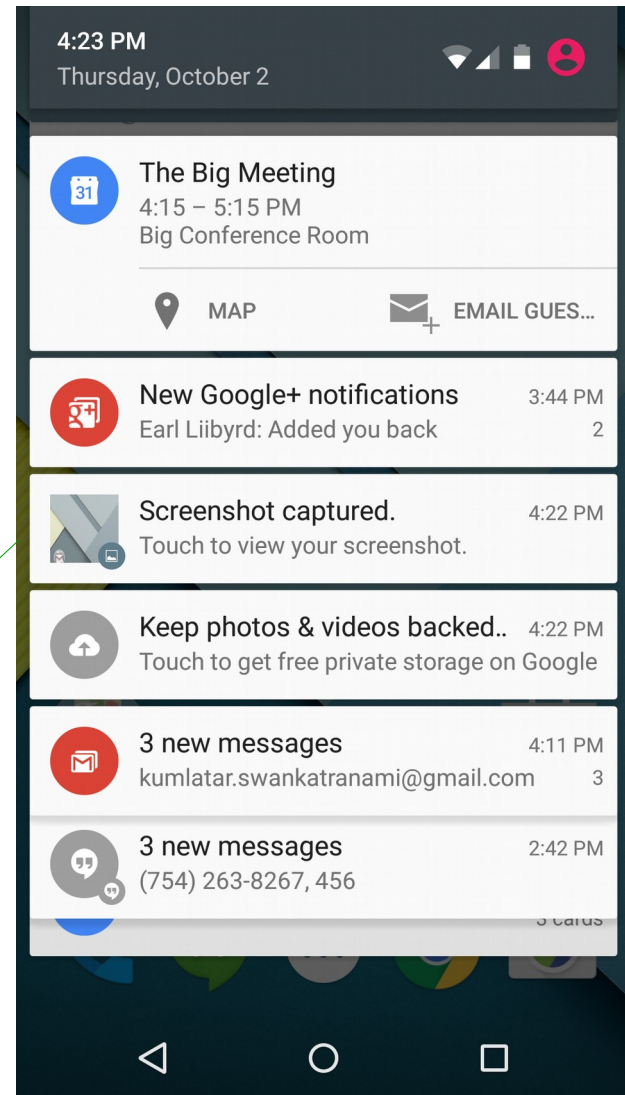
- ❑ São mensagens que são ser exibidas ao usuário fora da aplicação.
- ❑ Quando uma aplicação cria uma Notification, a mesma é exibida na *notification area*.
- ❑ Para visualizar os detalhes de uma notificação, o usuário abre o *notification drawer*.
- ❑ Tanto a *notification area* e o *notification drawer* são controlados pelo Android.

Notification



Notification Area

Notification Drawer



Notification

- Durante esse curso, representaremos uma Notification visualmente da seguinte maneira:



Notification

□ Para criar uma Notification simples:

```
25 public class MainActivity extends Activity {
26
27     private static final short PENDING_INTENT_ID = 1;
28     private static final short NOTIFICATION_ID = 1;
29
30     @Override
31     protected void onCreate(Bundle savedInstanceState) {
32         super.onCreate(savedInstanceState);
33
34         Notification.Builder mBuilder = new Notification.Builder(this)
35             .setSmallIcon(R.mipmap.ic_launcher)
36             .setContentTitle("MY Notification")
37             .setContentText("Hello World")
38             .setAutoCancel(true);
39
40         Intent resultIntent = new Intent(this, NewActivity.class);
41
42         TaskStackBuilder stackBuilder = TaskStackBuilder.create(this);
43         stackBuilder.addParentStack(NewActivity.class);
44         stackBuilder.addNextIntent(resultIntent);
45
46         PendingIntent resultPendingIntent = stackBuilder.getPendingIntent(PENDING_INTENT_ID, PendingIntent.FLAG_UPDATE_CURRENT);
47         mBuilder.setContentIntent(resultPendingIntent);
48
49         NotificationManager mNotificationManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
50         mNotificationManager.notify(NOTIFICATION_ID, mBuilder.build());
51     }
52 }
53
54
```

Notification

□ Para criar uma Notification BigTextStyle:

```
25 public class MainActivity extends Activity {
26
27     private static final short PENDING_INTENT_ID = 1;
28     private static final short NOTIFICATION_ID = 1;
29
30     @Override
31     protected void onCreate(Bundle savedInstanceState) {
32         super.onCreate(savedInstanceState);
33
34         Notification.Builder mBuilder = new Notification.Builder(this)
35             .setSmallIcon(R.mipmap.ic_launcher)
36             .setContentTitle("MY Notification")
37             .setContentText("Hello World")
38             .setAutoCancel(true)
39             .setStyle(new Notification.BigTextStyle().bigText("Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam v
40
41         Intent resultIntent = new Intent(this, NewActivity.class);
42
43         TaskStackBuilder stackBuilder = TaskStackBuilder.create(this);
44         stackBuilder.addParentStack(NewActivity.class);
45         stackBuilder.addNextIntent(resultIntent);
46
47         PendingIntent resultPendingIntent = stackBuilder.getPendingIntent(PENDING_INTENT_ID, PendingIntent.FLAG_UPDATE_CURRENT);
48         mBuilder.setContentIntent(resultPendingIntent);
49
50         NotificationManager mNotificationManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
51         mNotificationManager.notify(NOTIFICATION_ID, mBuilder.build());
52     }
53 }
54
55
```

Notification

□ Para criar uma Notification BigPictureStyle:

```
26 public class MainActivity extends Activity {
27
28     private static final short PENDING_INTENT_ID = 1;
29     private static final short NOTIFICATION_ID = 1;
30
31     @Override
32     protected void onCreate(Bundle savedInstanceState) {
33         super.onCreate(savedInstanceState);
34
35         Notification.Builder mBuilder = new Notification.Builder(this)
36             .setSmallIcon(R.mipmap.ic_launcher)
37             .setContentTitle("MY Notification")
38             .setContentText("Hello World")
39             .setAutoCancel(true)
40             .setStyle(new Notification.BigPictureStyle().bigPicture(BitmapFactory.decodeResource(getResources(), R.mipmap.ic_launcher)));
41
42         Intent resultIntent = new Intent(this, NewActivity.class);
43
44         TaskStackBuilder stackBuilder = TaskStackBuilder.create(this);
45         stackBuilder.addParentStack(NewActivity.class);
46         stackBuilder.addNextIntent(resultIntent);
47
48         PendingIntent resultPendingIntent = stackBuilder.getPendingIntent(PENDING_INTENT_ID, PendingIntent.FLAG_UPDATE_CURRENT);
49         mBuilder.setContentIntent(resultPendingIntent);
50
51         NotificationManager mNotificationManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
52         mNotificationManager.notify(NOTIFICATION_ID, mBuilder.build());
53     }
54 }
55
56
```

Notification

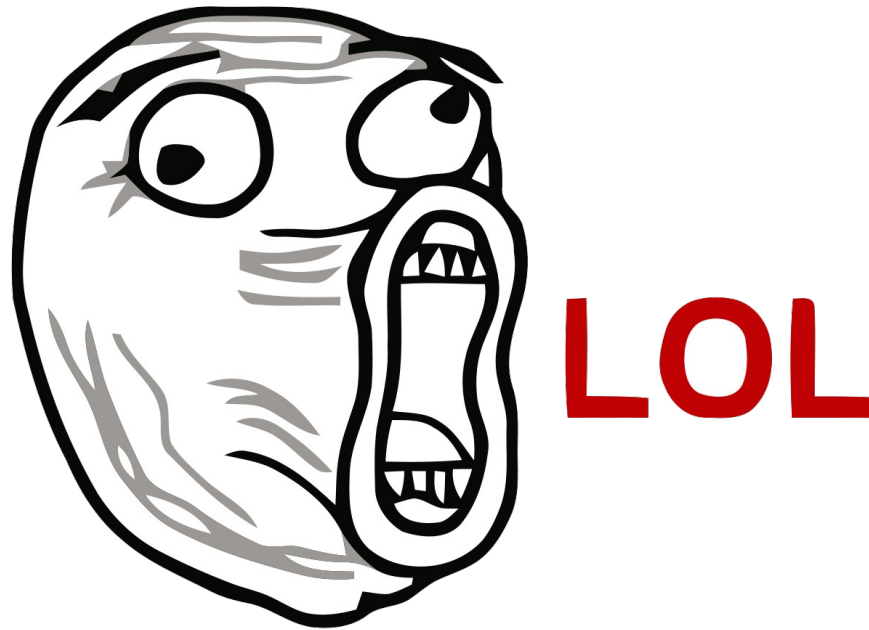
□ Para criar uma Notification InboxStyle:

```
26 public class MainActivity extends Activity {
27
28     private static final short PENDING_INTENT_ID = 1;
29     private static final short NOTIFICATION_ID = 1;
30
31     @Override
32     protected void onCreate(Bundle savedInstanceState) {
33         super.onCreate(savedInstanceState);
34
35         Notification.Builder mBuilder = new Notification.Builder(this)
36             .setSmallIcon(R.mipmap.ic_launcher)
37             .setContentTitle("MY Notification")
38             .setContentText("Hello World")
39             .setAutoCancel(true);
40
41         Notification.InboxStyle style = new Notification.InboxStyle();
42         style.addLine("This is a line. Here you will put your content");
43         style.setBigContentTitle("This is the title of big content");
44         style.setSummaryText("This is a summary text!");
45
46         mBuilder.setStyle(style);
47
48         Intent resultIntent = new Intent(this, NewActivity.class);
49
50         TaskStackBuilder stackBuilder = TaskStackBuilder.create(this);
51         stackBuilder.addParentStack(NewActivity.class);
52         stackBuilder.addNextIntent(resultIntent);
53
54         PendingIntent resultPendingIntent = stackBuilder.getPendingIntent(PENDING_INTENT_ID, PendingIntent.FLAG_UPDATE_CURRENT);
55         mBuilder.setContentIntent(resultPendingIntent);
56
57         NotificationManager mNotificationManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
58         mNotificationManager.notify(NOTIFICATION_ID, mBuilder.build());
59     }
60 }
61
62
```

Notification

- Main sobre notificações, visite a documentação oficial em:

`http://developer.android.com/guide/topics/ui/notifiers/notifications.html`





Alarm Manager

AlarmManager

- ❑ Utilizado para executar “**algo**” em determinado horário ou de tempos em tempos.
- ❑ Esse “**algo**” pode ser uma Activity, um Service ou um Broadcast Receiver.
- ❑ Podemos usar as seguintes contantes:
 - ❑ **RTC**: Executa o alarme após X segundos baseado no relógio, caso o dispositivo não esta hibernando
 - ❑ **RTC_WAKEUP**: Idem ao anterior, mas “acorda” o dispositivo caso o mesmo esteja hibernando
 - ❑ **ELAPSED_REALTIME**: Executa o alarme após X segundos após o boot do dispositivo, caso o dispositivo não esta hibernando
 - ❑ **ELAPSED_REALTIME_WAKEUP**: Idem ao anterior, mas “acorda” o dispositivo caso o mesmo esteja hibernando

AlarmManager

- Durante esse curso, representaremos uma AlarmManager visualmente da seguinte maneira:



AlarmManager

```
16 public class NewActivity extends Activity {
17
18     private MyBroadcastReceiver myBroadcastReceiver;
19
20     @Override
21     protected void onCreate(Bundle savedInstanceState) {
22         super.onCreate(savedInstanceState);
23         setContentView(R.layout.activity_new);
24
25         Intent it = new Intent("TESTE_ALARM");
26         PendingIntent alarmIntent = PendingIntent.getBroadcast(this, 0, it, 0);
27
28         Calendar cal = Calendar.getInstance();
29         cal.setTimeInMillis(System.currentTimeMillis());
30         cal.add(Calendar.SECOND, 10);
31
32         AlarmManager alarmManager = (AlarmManager) getSystemService(Context.ALARM_SERVICE);
33         alarmManager.set(AlarmManager.RTC_WAKEUP, cal.getTimeInMillis(), alarmIntent);
34     }
35 }
36
37
38
```

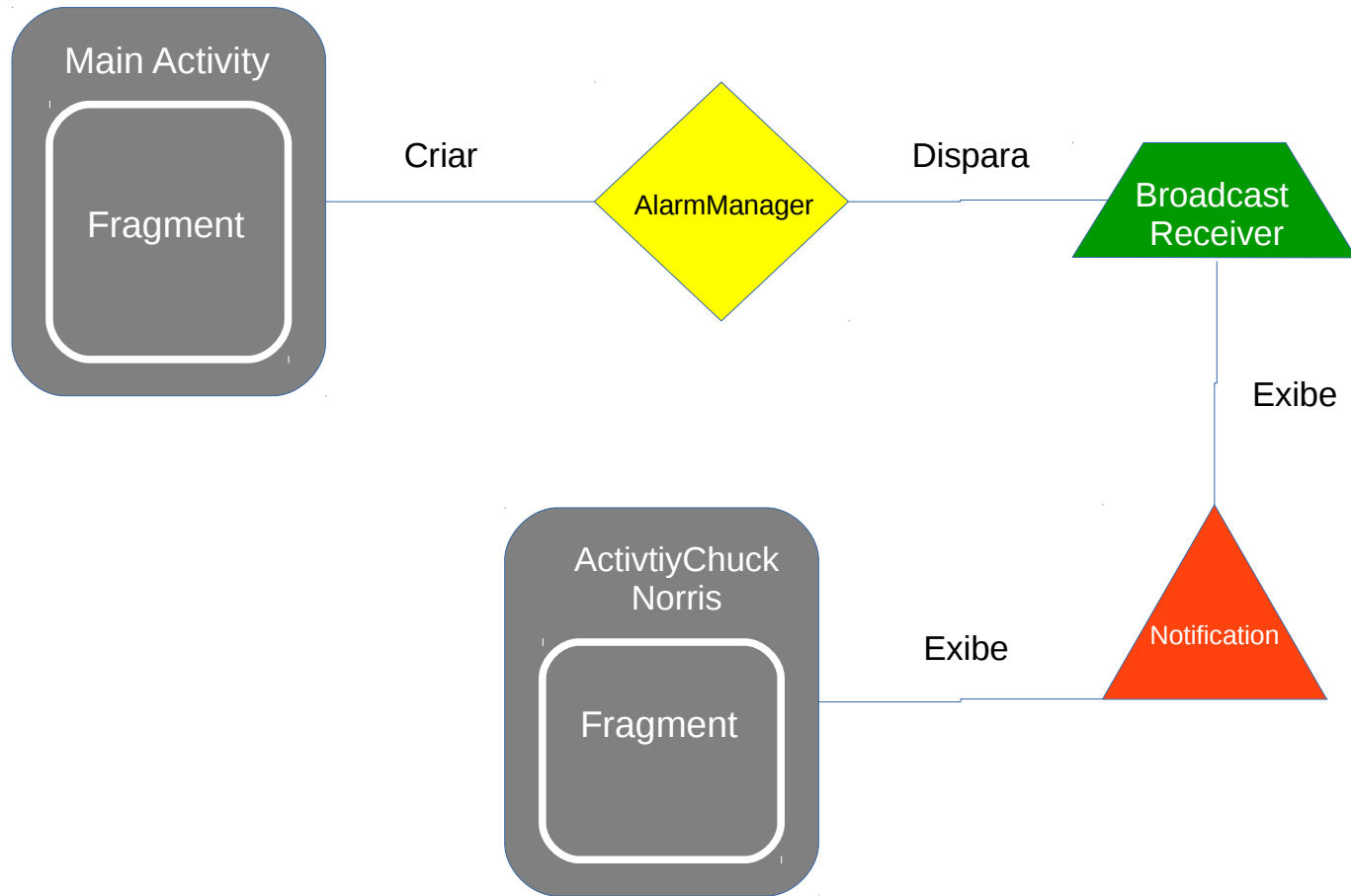


Let's try it!

Let's try it!

- ❑ Criar um novo projeto Android.
- ❑ Nesse projeto deve-se criar:
 - ❑ Uma tela que possui um botão.
 - ❑ Ao pressionar o botão, criar um alarme que irá disparar após 5 segundos.
 - ❑ O alarme irá disparar uma Intent implícita que será recuperada via BroadcastReceiver
 - ❑ Ao capturar o evento via BroadcastReceiver disparar uma notificação para o usuário.
 - ❑ Ao clicar na notificação deverá ser aberto uma Activity com a frase “**Well Done!**” e uma imagem do Chuck Norris

Let's try it!



Obrigado!

