



Alarms

Anderson Rodrigues



Intent

- O que são **Intents**?
- Tipos de **Intent**
- Utilizando **Intents**
- Como são instanciadas?



Intent

```
val intent = Intent(packageContext: this, FromNotificationActivity::class.java)
```

```
val intentActivity = Intent(App1BroadcastReceiver.ACTION)
```

```
val intent = Intent(context, CountdownService::class.java)  
intent.putExtra(EXTRA_COUNTDOWN_TIME, countDownTime)  
context.startService(intent)
```



PendingIntent

- O que é?
 - Possui uma ação associada
 - Permite a execução de uma ação
 - Activity, BroadcastReceiver, Service



Intent-filter

- Especifica os tipos de **Intents** que um componente pode responder
- Componentes:
 - Activity
 - Service
 - BroadcastReceiver
- Elementos:
 - Action
 - Category
 - Data



System Services

- Serviços a nível de sistema
- Obtidos através de um *Context*

Currently available classes are: `WindowManager`, `LayoutInflater`, `ActivityManager`, `PowerManager`, `AlarmManager`, `NotificationManager`, `KeyguardManager`, `LocationManager`, `SearchManager`, `Vibrator`, `ConnectivityManager`, `WifiManager`, `AudioManager`, `MediaRouter`, `TelephonyManager`, `SubscriptionManager`, `InputMethodManager`, `UiModeManager`, `DownloadManager`, `BatteryManager`, `JobScheduler`, `NetworkStatsManager`, `DomainVerificationManager`.

```
getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager
```



BroadcastReceiver

- Estático
- Dinâmico
- Execução em background
- System broadcasts



Alarms

- Agendar algo para um ponto futuro
- Férias/Agenda/Reuniões/Médico/Futebol/Novela/Lançamentos



Alarms

- Executar ações fora do tempo de vida da aplicação
 - Iniciar um serviço para sincronizar com o servidor
- Com ou sem repetição
- Intervalos de tempo ou data/hora específica



Alarms

```
private fun scheduleElapsedTime() {  
    val intent = Intent( packageContext: this, AlarmBroadcastReceiver::class.java)  
    val pendingIntent = PendingIntent.getBroadcast( context: this, requestCode: 0, intent, PendingIntent.FLAG_ONE_SHOT)  
  
    val manager = getAlarmManager()  
    manager.set(AlarmManager.ELAPSED_REALTIME, triggerAtMillis: SystemClock.elapsedRealtime() + 30000, pendingIntent)  
}
```



Alarms

- Atividade:
 - Criar um Alarm que, ao ser acionado, exibe um log utilizando um BroadcastReceiver
 - Comentários?



Alarms

! **Note:** Beginning with API 19 ([Build.VERSION_CODES.KITKAT](#)) alarm delivery is inexact: the OS will shift alarms in order to minimize wakeups and battery use. There are new APIs to support applications which need strict delivery guarantees; see [setWindow\(int, long, long, android.app.PendingIntent\)](#) and [setExact\(int, long, android.app.PendingIntent\)](#). Applications whose **targetSdkVersion** is earlier than API 19 will continue to see the previous behavior in which all alarms are delivered exactly when requested.



Alarms

- Performance e uso de bateria
- Inexatidão
- setExact



Alarms

- Atividade:
 - Criar um Alarm utilizando setExact



Alarms

- Tipos de Alarm
 - ELAPSED_REALTIME -> tempo desde a inicialização
 - ELAPSED_REALTIME_WAKEUP
 - RTC -> Tempo específico
 - RTC_WAKEUP



Alarms

- ELAPSED_REALTIME:

```
SystemClock.elapsedRealtime()
```

- RTC:

```
System.currentTimeMillis()
```

```
Epoch time: 1626450303037
```

```
System time: 56747265
```




Alarm

- `setRepeating`
- `setInexactRepeating`
- `Cancel` - `FLAG_ONE_SHOT`



Alarms

```
private fun cancel() {  
    val intent = Intent( packageContext: this, AlarmBroadcastReceiver::class.java)  
    val cancelPending = PendingIntent.getBroadcast( context: this, requestCode: 0, intent, flags: 0)  
  
    getAlarmManager().cancel(cancelPending)  
}
```



Alarms

- Aplicação encerra a execução tão logo o método onReceive seja executado
- Não é permitido mais que 10s de execução
- WakeLock
- WorkManager
 - <https://developer.android.com/topic/libraries/architecture/workmanager>



Alarms

- Alarms são excluídos quando o dispositivo é desligado/reiniciado
- Qual abordagem utilizar?



Alarms

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

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