

**Licenciatura em Engenharia Informática e de Computadores**

**Semestre de Inverno 2014/2015**

Programação em

Dispositivos Móveis

2ª & 3ª Série de exercícios

*Trabalho elaborado por:*

* Flávio Cadete

Nº 35383

*Engenheiro:*

* Pedro Félix

Índice

[AnniversaryReminder 3](#_Toc407213692)

[AlarmStartupReceiver 3](#_Toc407213693)

[AlarmNotificationReceiver 4](#_Toc407213694)

[Notificações 5](#_Toc407213695)

[Preferências 6](#_Toc407213696)

# **AnniversaryReminder**

### AlarmStartupReceiver

Esta classe extende de ***BroadcastReceiver*** e tem como objectivo detectar quando foi feito o ***BOOT*** do dispositivo ou quando o número de semanas a filtrar encontrado preferências foi alterado.

Depois de feita a confirmação irá requisitado um serviço de ***AlarmManager*** que já é intrínseco á plataforma Android.

É adicionado um *PendingIntent* ao **AlarmManager** criado e que contém um *Intent* para fazer broadcast à classe **AlarmNotificiationReceiver** (que é o receiver responsável do envio das notificações)*.*

Neste momento só falta recolher das ***DefaultSharedPreferences*** a hora defenida nas preferências (***Figura 6***), e passar ao setRepeat do ***AlarmManager*** criado.

*AlarmManager.setRepeat*

*PendingIntent*

*AlarmNotificationReceiver*

**BroadcastReceiver**

*BOOT\_COMPLETED*

*PREFS\_UPDATE*

ALARM\_SERVICE

**Service***(Started Service)*

Figura 1

**Alterações necessárias no Manifesto para que o componente BroadcastReceiver em cima descrito tenha funcionalidade pretendida:**

<receiver android:name=".broadcastreceivers.AlarmStartupReceiver"

android:enabled="true"

android:permission="android.permission.RECEIVE\_BOOT\_COMPLETED">

<intent-filter>

<action android:name="android.intent.action.BOOT\_COMPLETED" />

<action android:name="com.starlon.froyvisuals.PREFS\_UPDATE"/>

<category android:name="android.intent.category.DEFAULT" />

</intent-filter>

</receiver>

<uses-permission android:name="android.permission.RECEIVE\_BOOT\_COMPLETED" />

### AlarmNotificationReceiver

Como o próprio nome deixa induzir, esta classe extende do componente ***BroadcastReceiver***.

Tem como objectivo receber os broadcasts do ***AlarmManager*** criado em ***AlarmStartupReceiver*** e enviar as notificações com a informação necessária para que o utilizador consiga distinguir os aniversariantes, além de poder abrir a aplicação de contactos com o contacto aniversariante selecionado na notificação.

Foram adicionados às notificações pequenas funcionalidade como vibração, luzes led e som ao Notification.Builder, sendo o mesmo passado ao método *notify* ***NotificationManager***, além do ID da notificação a enviar (é enviada um ID diferente por cada contacto aniversariante).

O método ***notify*** enviará assim a notificação activando as funcionalidades passadas.

Figura 2

ALARM\_SERVICE

**Service***(Started Service)*

*NotificationManager.notify*

**BroadcastReceiver**

**Alterações necessárias no Manifesto para que o componente BroadcastReceiver em cima descrito tenha funcionalidade pretendida:**

<receiver android:name=".broadcastreceivers.AlarmNotificationReceiver"

android:enabled="true">

<intent-filter>

<action android:name="android.intent.action.SCREEN\_ON"/>

<action android:name="android.intent.action.VIEW" />

</intent-filter>

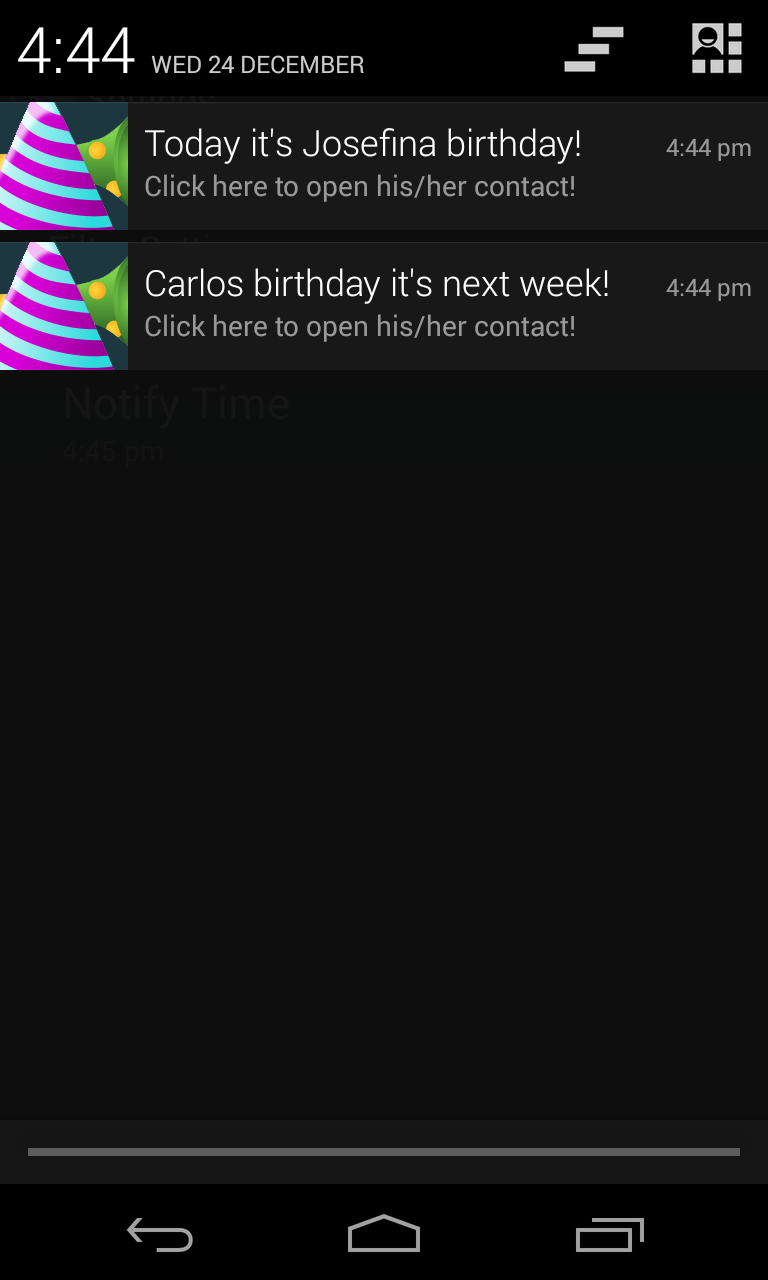
</receiver>

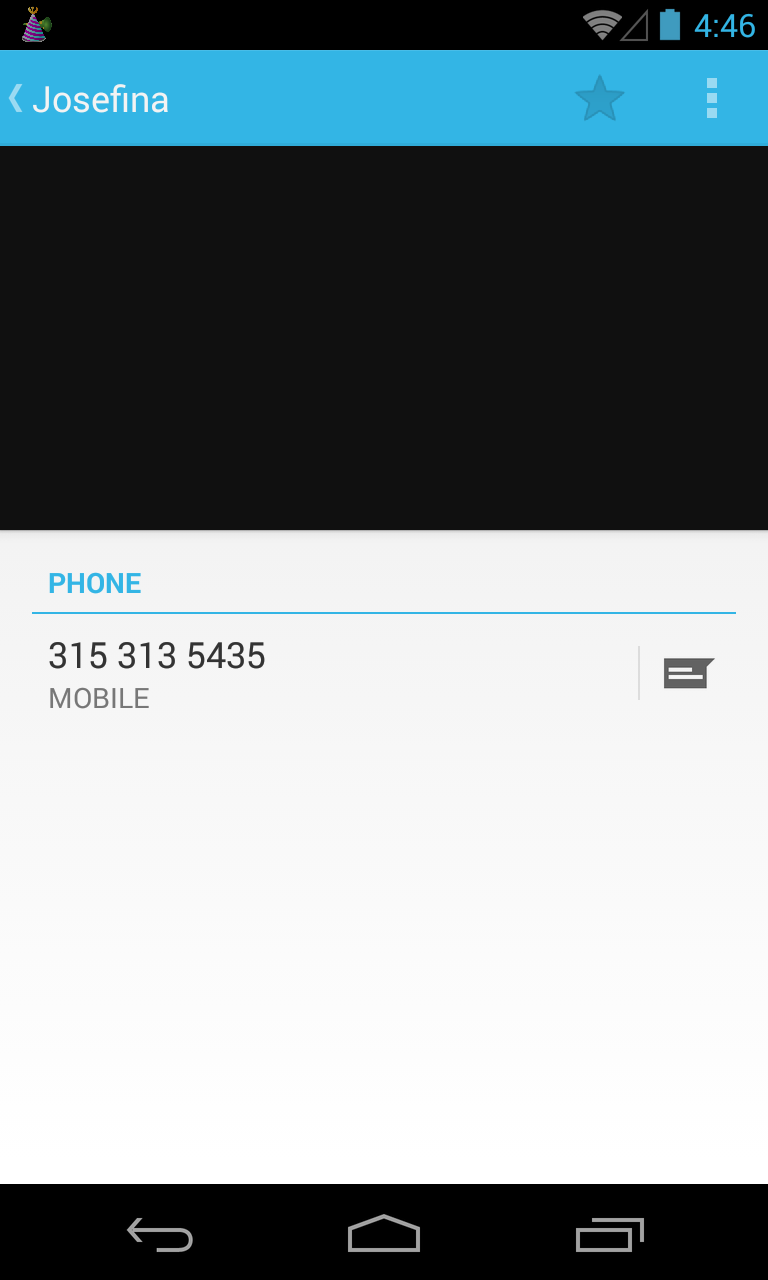
<uses-permission android:name="android.permission.VIBRATE" />

### Notificações

Na ***Figura 3*** é possível verificar as ***notificações*** recebidas. Cada contacto tem a sua respectiva notificação. Ao clicar na notificação irá abrir a aplicação responsável pela gestão dos contactos, com a vista do contacto selecionado na notificação e exemplificado na ***Figura 4***, podendo aí ligar/enviar SMS ao contacto aniversariante.

Também é possível com um ***long* *click*** num contacto da lista de contactos filtrados pela nossa aplicação, abrir a vista responsável pela informação do contacto e presente na ***Figura 4***.





**Figura 4**: Informação do Contacto (Genymotion Contacts Application)

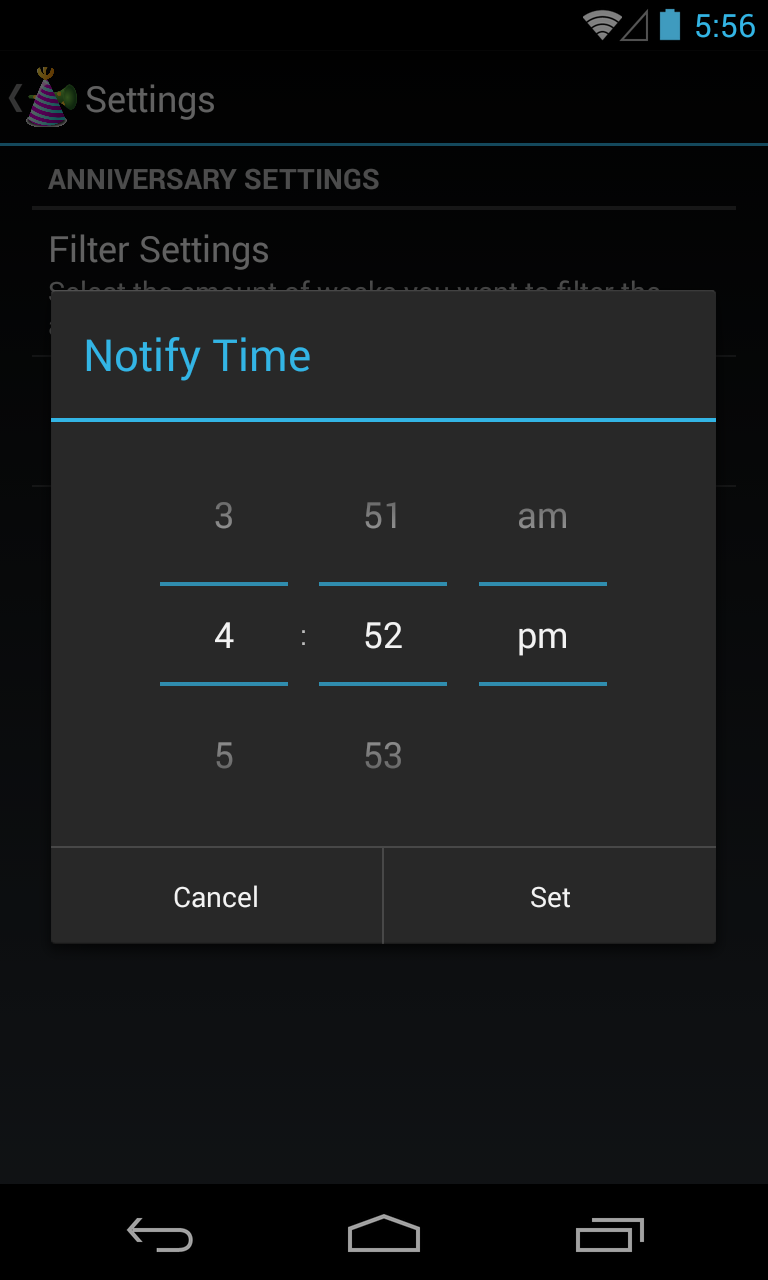
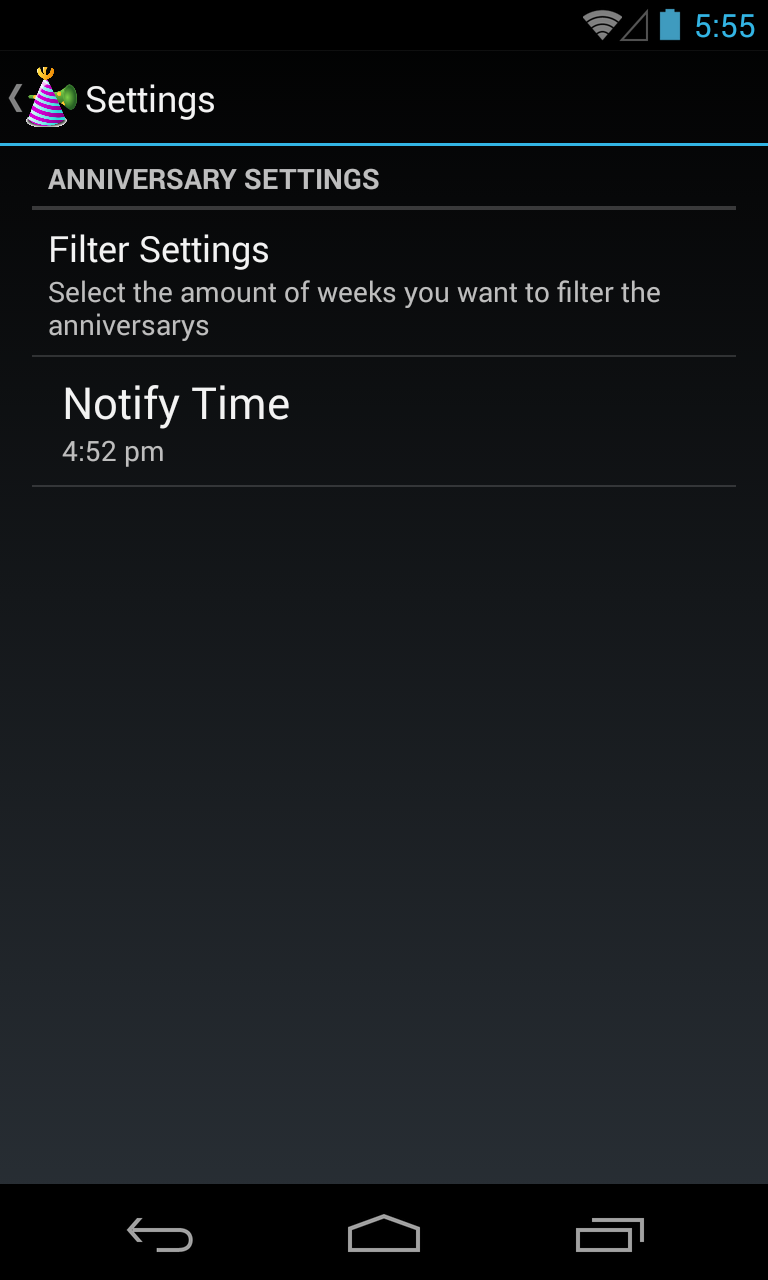
**Figura 3**: Notificações (própria dia e dentro de uma semana)

### Preferências

Alteração das preferências face á série anterior . Foi adicionado um ***DialogPreference*** (***Figura 6***) que dá a possibilidade ao utilizador de indicar a que hora pretende receber as notificações. Esta alteração será depois visivel ao utilizador nas preferências (***Figura 5***).

Quando é seleccionado o botão ***Set*** exemplificado na ***Figura 6***, é guardado em milisegundos a hora escolhida nas ***DefaultSharedPreferences*** da aplicação e enviado um broadcast com um Intent com a acção de ***PREFS\_UPDATE***, fazendo com que a classe ***AlarmStartupReceiver*** possa criar um novo ***AlarmManager*** com a nova a hora que irá fazer os brodcasts.

*(Na 2ªSérie ainda não tinha sido leccionado os Fragmentos, por isso tenho consiência de que irei perder o dialog quando são feitas rotações com o dispositivo e que face ao pedido no momento esta requisito não era obrigatório)*



**Figura 5**: Preferências

**Figura 6**: DialogPreference para escolher quando pretende receber as notificações

# ThothNews

1. Adicione à aplicaçãao de notícias do Thoth as seguintes características:

* Existência dum content provider para armazenamento e gestão da seguinte informação:

1. Conjunto de turmas disponíveis no Thoth.
2. Conjunto de turmas selecionadas.
3. Conjunto de notícias das turmas selecionadas, incluindo o estado de visualização.

Este content provider deve usar uma base de dado SQLite para persistência da informação.

* Existência dum serviço para actualização periódica da informação, apenas realizada quando existir conectividade via WiFi.

1. Altere a aplicação de notícias do Thoth para usar fragmentos na implementação da user interface. Nomeadamente, acrescente suporte para:

* Interface master-details com dois painels.
* Navegação entre notícias através de swipe.

2. Adicione à aplicação do Thoth a capacidade de apresentar os participantes de uma turma. Esta apresentação deve incluir a fotografia do participante. Realize o necessário para minimizar o consumo de recursos necessários para esta tarefa.