



ÁREA DEPARTAMENTAL DE ENGENHARIA
DE ELETRÓNICA E TELECOMUNICAÇÕES
E DE COMPUTADORES - ADEETC

Licenciatura em Engenharia Informática e de Computadores
Semestre de Inverno 2014/20145

1ª Série de exercícios

P rogramação em D ispositivos M óveis

Eng.º Pedro Félix

Trabalho elaborado por:

- Flávio Cadete nº 35383
- Tiago Martins nº 36903

Índice

Exercício 1 - ThothNews	3
Activities	3
PreferencesActivity	3
ClassesActivity	4
NewsActivity	5
SingleNewActivity	5
Extractors	7
Exercício 2 – Anniversary Reminder	8
Activities	8
MainActivity	8
AddAnniversaryActivity	9
PreferencesActivity	11
Bibliografia	12

Exercício 1 - ThothNews

Objectivo:

Seleccção de um conjunto de turmas do Thoth e visualização das notícias associadas às turmas seleccionadas. Para isso foi necessário utilizar a API Thoth disponível em thoth.cc.isel.ipl.pt/api/.

Activities

PreferencesActivity

Esta actividade tem o objectivo de mostrar as definições disponíveis da aplicação ao utilizador, além da responsabilidade de guardar convenientemente essas definições (*SharedPreferences*). Se não existir nenhuma turma seleccionada (*Figura 2*), impossibilitamos ao utilizador o acesso á vista da lista de turmas (*Figura 4*) e ao retornar é limpa o stack de chamadas (actividades lógicas).

- **Figura 1:** Vista das definições
- **Figura 2:** Seleccção das turmas pretendidas e disponibilizadas no Thoth
- **Figura 3:** Limpar turmas seleccionadas

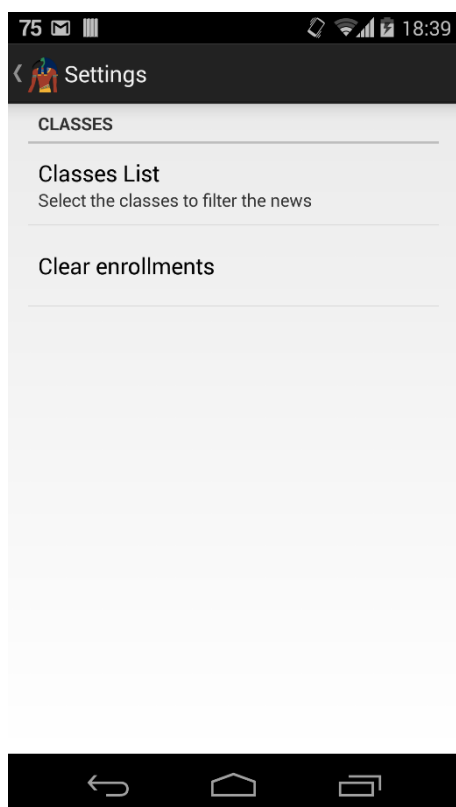


Figura 1

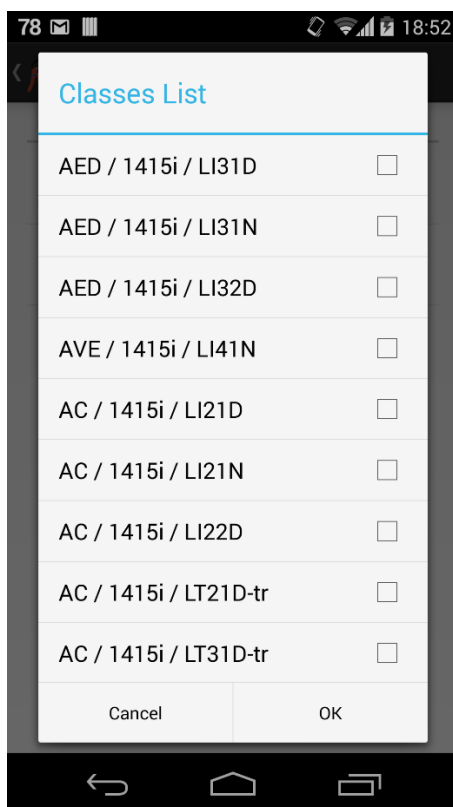


Figura 2

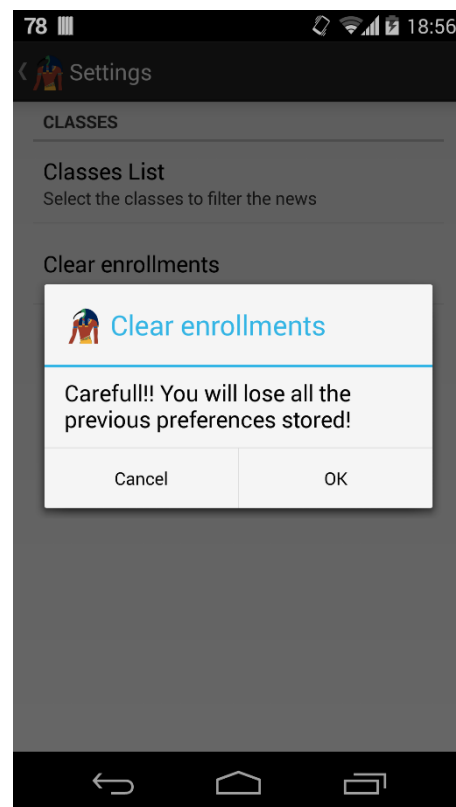


Figura 3

ClassesActivity

Actividade com a função de mostrar a lista das turmas seleccionadas nas Settings. Essa informação provém de *SharedPreferences* que é obtida a partir da chave passada em **PreferencesActivity**, que nos permite obter os ids das turmas guardadas **ExtractorClasses** que tem a responsabilidade de fazer um pedido *HTTP* á *API Thoth* singular por cada turma (id) e retornando uma lista com toda a informação necessária.

É utilizada a **ClassesListAdapter** que estende de **ArrayAdapter**, contendo assim as funções necessárias para a visualização dos items na lista de classes, a lista em cru da informação de todas as classes, para além da responsabilidade de gestão dessa lista de items (reutilização dessas views).

Cada item da lista de adapter contém um evento **onClickListener** que iniciará uma nova actividade (**NewsActivity**) passando um **Extra** ao intent a iniciar com o id da turma seleccionada.

- **Figura 4:** é possível verificar que foram seleccionadas na **PreferencesActivity**, as turmas SI e PDM de 1415i / LI51N
- **Figura 5:** menu **Settings (PreferencesActivity)** acedido através desta actividade

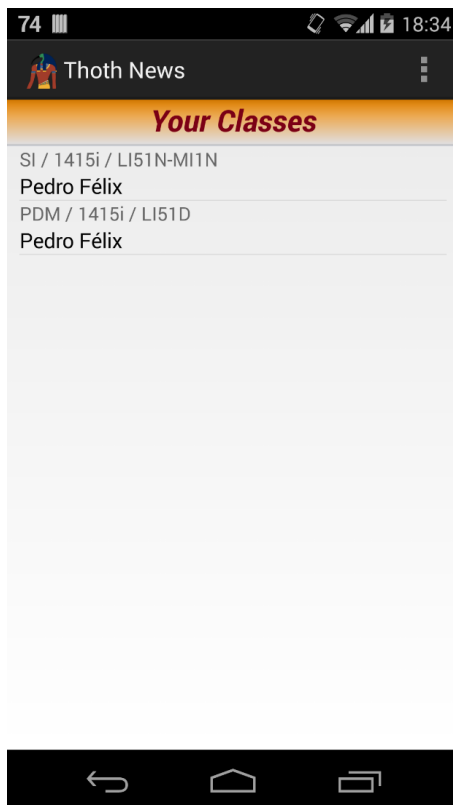


Figura 4

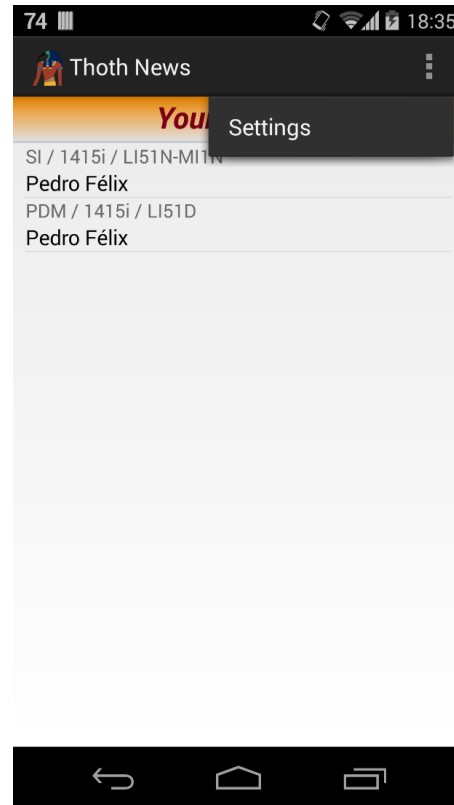


Figura 5

NewsActivity

Actividade com a função de mostrar a lista de notícias da turma seleccionada em **ClassesActivity** por ordem cronológica (a última inserida será a primeira na lista) e que obtém essa informação através de um pedido *HTTP* á *API Thoth* ao campo *newsitem* da classe id seleccionada e que se encontra implementado em **ExtractorMultipleNews**.

É feita a utilização de **NewsListAdapter** que estende de **ArrayAdapter**, contendo as funções necessárias para a visualização de um item na lista de notícias, a lista em cru da informação de todas as notícias da turma seleccionada, para além da responsabilidade de gestão dessa lista.

Nesta actividade é também criado um ficheiro com as notícias da classe, guardando também o estado da notícia (**READ** | **NOT READ**).

Cada item da lista de adapter contém um evento **onClickListener** que iniciará uma nova actividade (**SingleNewActivity**) passando um **Extra** ao intent a iniciar com o id da notícia seleccionada.

- **Figura 6:** encontram-se 4 notícias de *PDM / 1415i / LI51D* e que por indicação do texto a estar como no estilo **bold**, é possível verificar que temos 2 notícias por ler e as outras 2 notícias já foram lidas, estando as lidas depois das não lidas
- **Figura 7:** menus **Settings** (*PreferencesActivity*) e **Refresh All** (novo pedido ao mesmo URL)

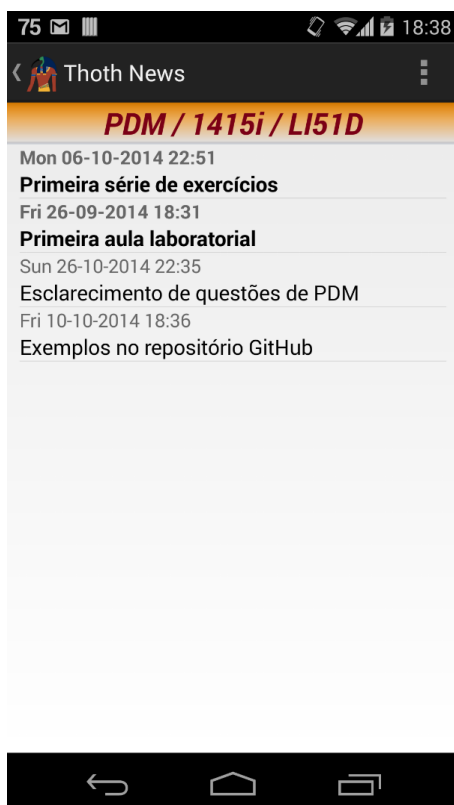


Figura 6

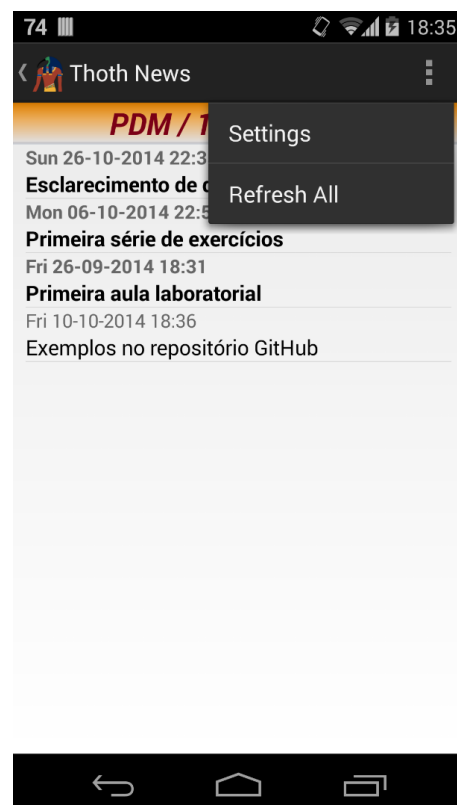


Figura 7

SingleNewActivity

Tem a responsabilidade de mostrar o conteúdo da notícia seleccionada em *NewsActivity*, fazendo então um pedido *HTTP* á *API Thoth* com o id da notícia clicada e que se encontra implementado em ***ExtractorSingleNew*** para obter o resultado esperado.

Se o conteúdo da notícia contiver links, esses links aos serem pressionados podem ser vistos no browser por definição, ou se tiver um e-mail, será aberto a aplicação responsável a esse tipo de pedido. Isto é possível através de uma funcionalidade que os criadores da plataforma Android nos oferecem e que consiste em adicionar ao *TextView* que irá conter o conteúdo da notícia o atributo ***android:autoLink="web"***.

- **Figura 8:** encontra-se um exemplo de uma notícia com a indicação no Link que de pode ser clicado para abrir a aplicação responsável ao tipo de pedido pretendido
- **Figura 9:** menu **Settings** (*PreferencesActivity*) acedido através desta actividade

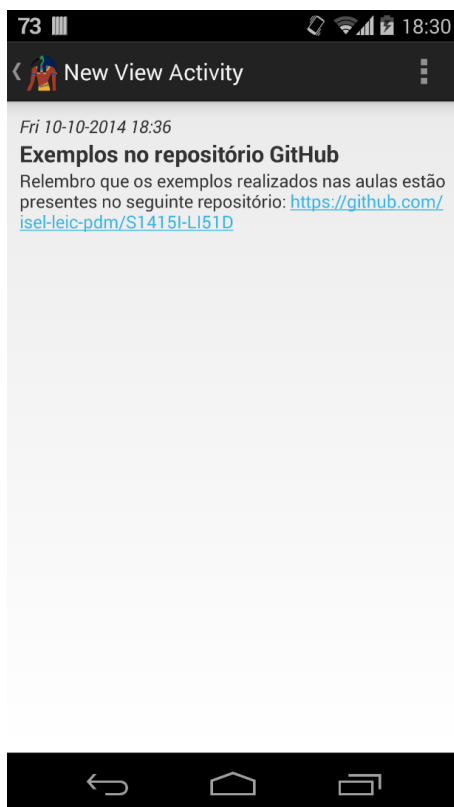


Figura 8

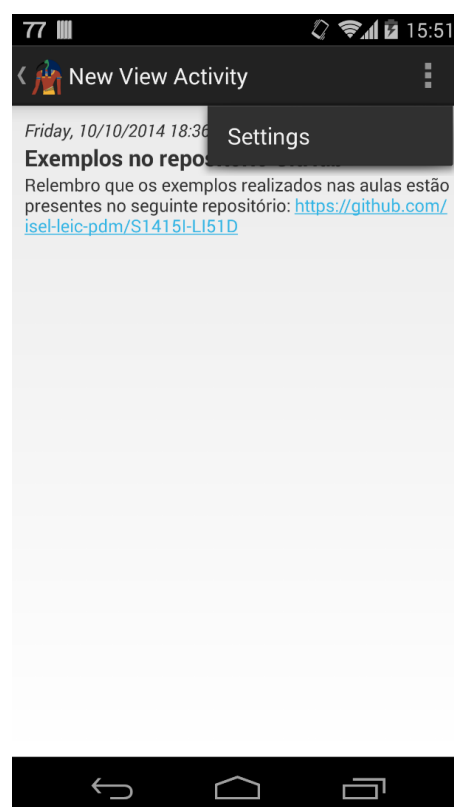


Figura 9

Extractors

Estas classes que estendem de *AsyncTasks* (fazem os pedidos e *bindings* nas worker threads e por isso trabalham assincronamente com o resto da aplicação) têm o objectivo de fazer o pedido *HTTP* com o *URL* pretendido, obtendo a informação necessária para preencher o modelo correspondente.

É necessário realizar o processamento noutra *thread* - via *AsyncTasks* - diferente da UI *thread* porque se tratam de operações de processamento demorado, e caso fossem realizadas na UI *thread* a mesma o utilizador ficaria com a sensação que a aplicação bloqueou.

O modelo referente ao pedido encontra-se nessa mesma classe, sendo assim mais fácil verificar o que necessita para fazer *bind* do que é recebido no pedido e na criação dos objectos a serem mostrados.

Exercício 2 – Anniversary Reminder

Objectivo:

A aplicação do exercício 2, que decidimos atribuir o nome Anniversary Reminder tem como objectivo informar o utilizador dos aniversários que vão acontecer brevemente. Tem a possibilidade de listar os contactos com aniversários a acontecer entre 1 a 4 semanas, e permite também ao utilizador introduzir uma nova data de aniversário associado a um contacto, ou actualizar essa data caso o contacto já tivesse associado uma data de aniversário anteriormente.

Para aceder à informação dos contactos, e actualizar as mesmas é necessário recorrer ao *Content Provider* de contactos, que nos permite aceder de forma estruturada.

Activities

MainActivity

Nesta actividade é possível consultar a lista de aniversários a acontecer em breve, aceder às preferências para definir o tempo máximo, entre 1 a 4 semanas a filtrar. Para além disso é possível escolher adicionar um aniversário, que nos remete para a *AddAnniversaryActivity*.

- **Figura 9:** encontra-se um exemplo de aniversários de 3 contactos. É possível verificar que visualmente existe uma diferença entre um aniversário que acontece no dia de hoje, amanhã ou noutro dia.

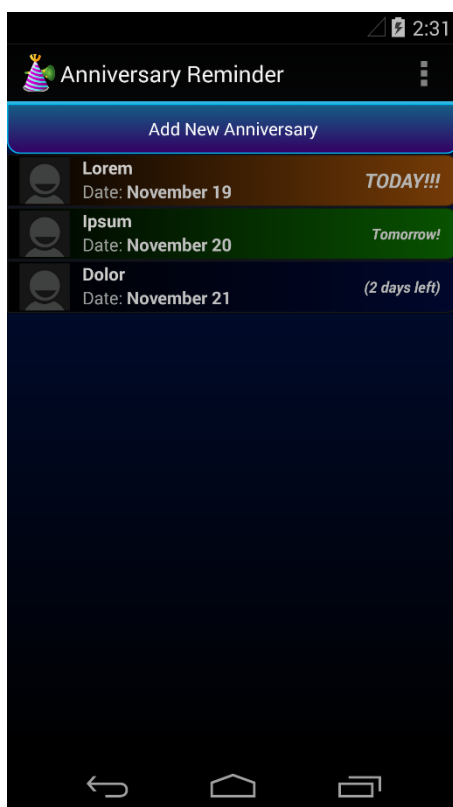


Figura 9

Queries

Para poder apresentar a lista de contactos que fazem aniversário brevemente é necessário obter o nome de contacto, fotografia e data de aniversário de cada um deles que vão ocorrer num futuro próximo.

Fazemos uma query sobre a tabela localizada no URI definido pela constante

ContactsContract.Data.CONTENT_URI fazendo uma selecção das colunas:

- Contacts.DISPLAY_NAME – Nome do contacto
- Event.DATA – Data associada a um evento, neste caso de aniversário
- Contacts.PHOTO_THUMBNAIL_URI – URI da fotografia do contacto

Com uma cláusula where com a seguinte estrutura:

ContactsContract.Data.MIMETYPE = Event.CONTENT_ITEM_TYPE

AND Event.TYPE = Event.TYPE_ANNIVERSARY

AddAnniversaryActivity

Nesta actividade é possível escolher o contacto ao qual pretendemos definir a data de aniversário, ou caso já se encontre definida anteriormente, permitirá actualizar a mesma.

- **Figura 10:** exemplo de ecrã para definir a data de aniversário de um contacto.
- **Figura 11:** lista de contactos, para onde somos encaminhados quando pretendemos escolher um contacto ao qual vamos atribuir uma data de aniversário.
- **Figura 12:** escolha da data de aniversário a associar a um contacto.

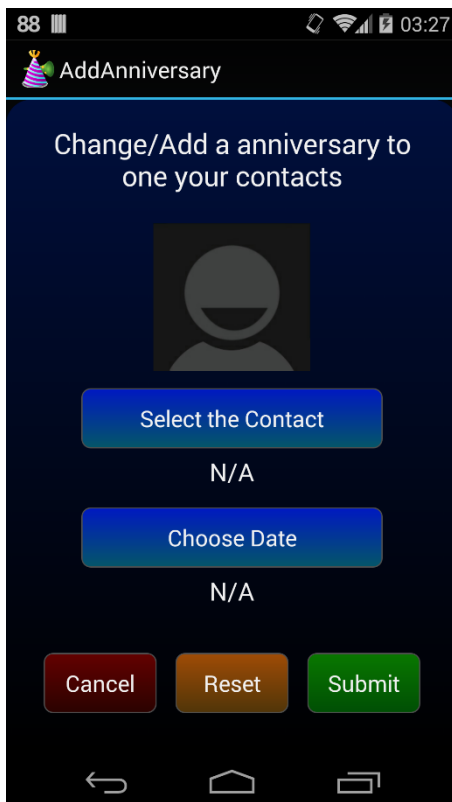


Figura 10

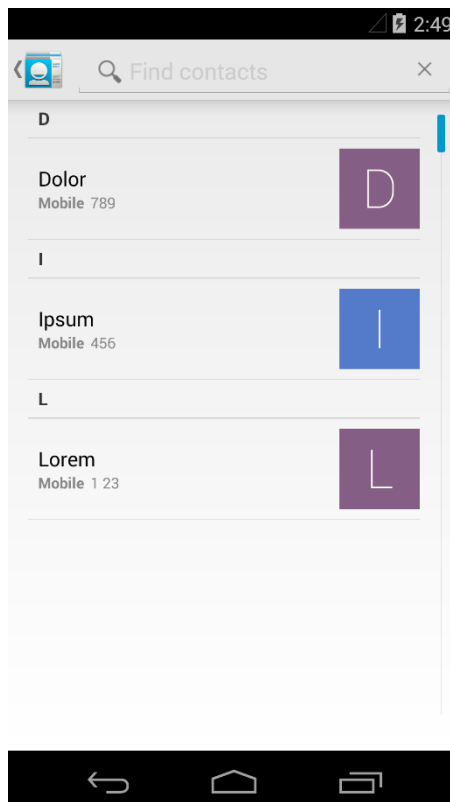


Figura 11

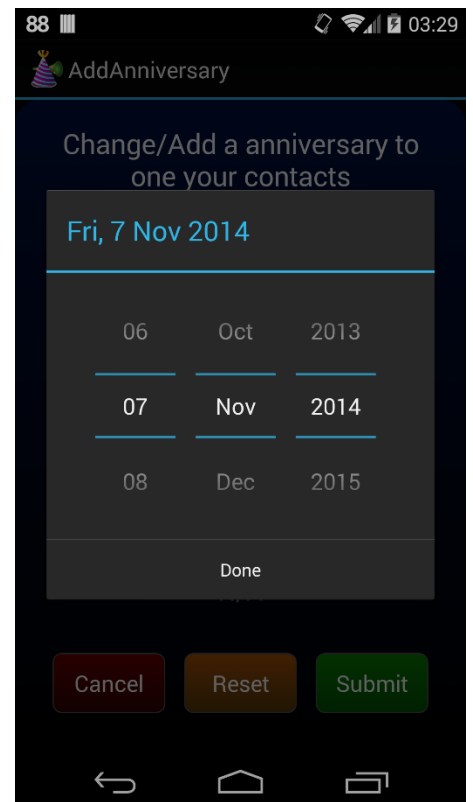


Figura 12

Queries

Para fazer manipulação de uma tabela é necessária definir a localização da tabela onde é armazenada essa informação. Adicionalmente é necessário definir um contentor com os pares chave, valor com os dados a introduzir.

- Para inserir informação de aniversário de um contacto definimos o *ContentValues* com os seguintes pares chaves valor:

Chave	Valor
Data.MIMETYPE,	Event.CONTENT_ITEM_TYPE
Data.RAW_CONTACT_ID	ID do contacto seleccionado
Event.TYPE	Event.TYPE_ANNIVERSARY
Event.START_DATE	Data de aniversário seleccionada

DATA.MIMETYPE – corresponde o tipo de informação que pretendemos associar a um contacto, sendo do tipo evento.

EVENT.TYPE – define o tipo exacto de evento.

EVENT.START_DATE – data de início do evento.

- Para actualizar a informação de aniversário - que acontece quando esta foi definida anteriormente colocamos exactamente a mesma informação, com a uma condição WHERE que seja igual aos pares chave, valor definidos anteriormente, em que a única coluna modificada é EVENT.START_DATE visto que foi apenas a data de aniversário que se alterou.

PreferencesActivity

Nesta actividade é possível definir as opções de filtragem do número de semanas para qual pretendemos ver os aniversários. É possível filtrar por 1, 2, 3 ou 4 semanas.

- **Figura 13:** ecrã de definições onde é possível definir as opções de filtragem.
- **Figura 14:** ecrã das diferentes opções de filtragem.

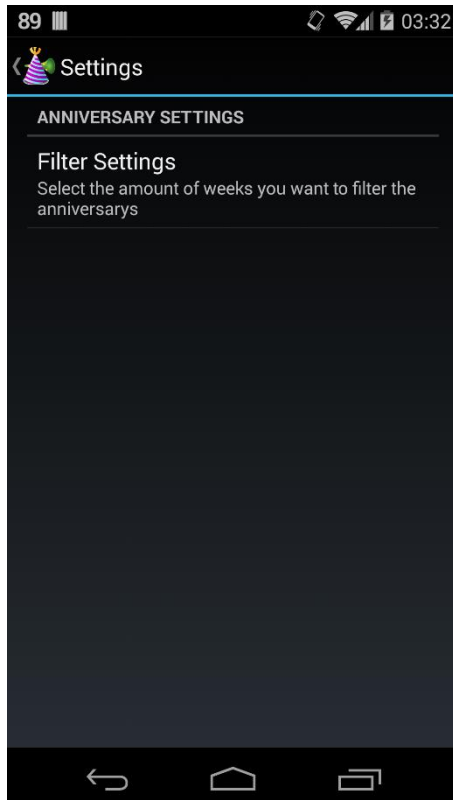


Figura 13

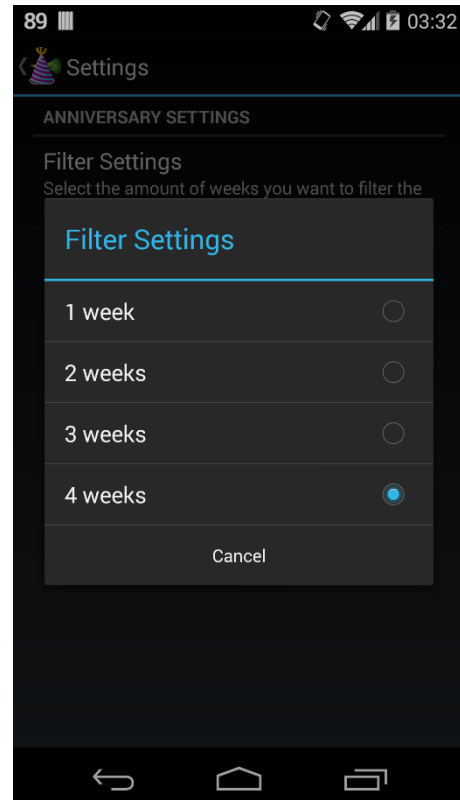


Figura 14

Bibliografia

Inc, G. (Novembro de 2014). *Android 4.4 Reference*. Obtido de <http://developer.android.com/reference/android/provider/ContactsContract.Data.html>

Inc., G. (Novembro de 2014). *Contacts Provider*. Obtido de <http://developer.android.com/guide/topics/providers/contacts-provider.html>