Twitter Client

Relatório Final

31/05/2015

Sistemas Distribuídos

3º ano do Mestrado Integrado em Engenharia Informática e Computação

Elementos do Grupo:

José Melo - ei12050 - ei12050@fe.up.pt

Rui Gomes - ei12038 - ei12038@fe.up.pt

Índice

Introdução

Como projeto final, o grupo decidiu implementar um cliente, para Android, do Twitter que permita organizar, por categorias, os utilizadores a seguir.
Assim, este cliente, para além das inúmeras funcionalidades que interagem com o Twitter, também permite criar categorias e associar utilizadores a estas, de forma a melhor organizar todos os utilizadores.

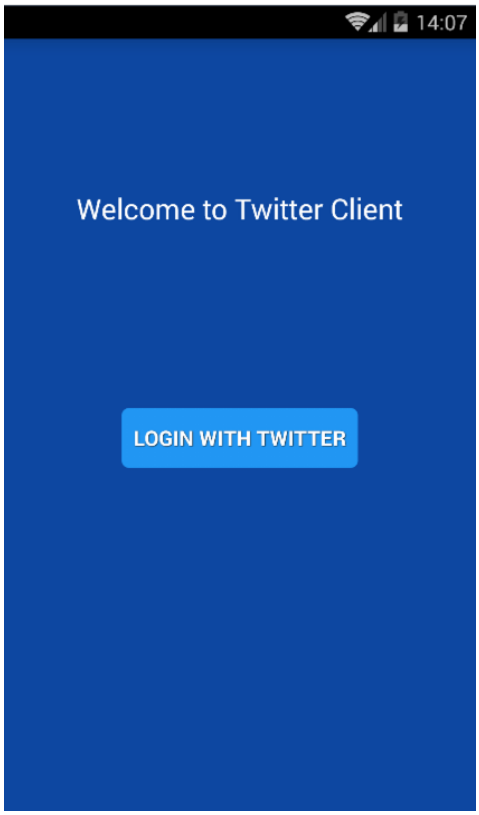
Funcionalidades

- Autenticar utilizador com uma conta do Twitter
- Criar e remover categorias
- Adicionar utilizadores às categorias
- Ler *tweets*
- Ver *tweets* não lidos
- Criar novo *tweet*
- Responder a tweets
- *Retweet*
- Pesquisar utilizadores
- Seguir novos utilizadores

Arquitetura

Login

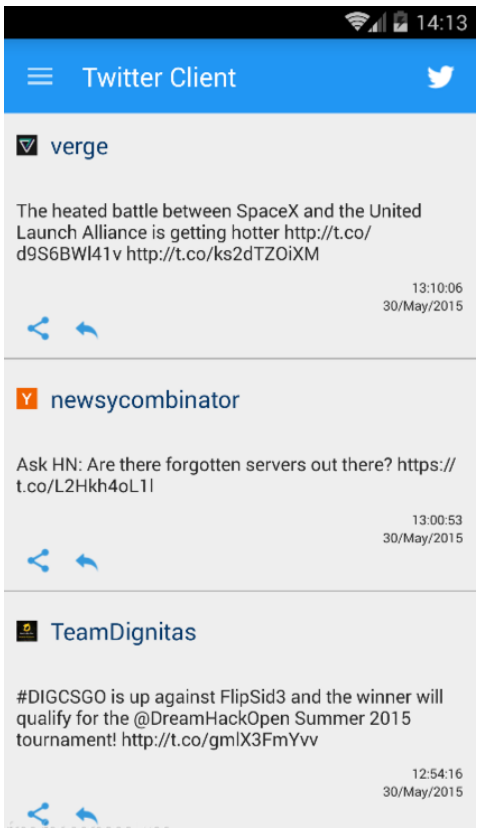
O login é feito na LoginActivity, e passa por verificar se existe algum token válido na memória do dispositivo, indicativo de um login já efetuado.
Caso este não exista, a aplicação irá fazer uma série de pedidos à API do Twitter de forma a obter um token de longa duração que poderá ser usado para obter dados da API.



Timeline

Após efetuado o login, a ClientActivity é lançada, instanciando um objeto User, que irá descarregar toda a informação relativa ao utilizar com login efetuado e a sua timeline, isto é, tweets de pessoas que escolheu seguir.

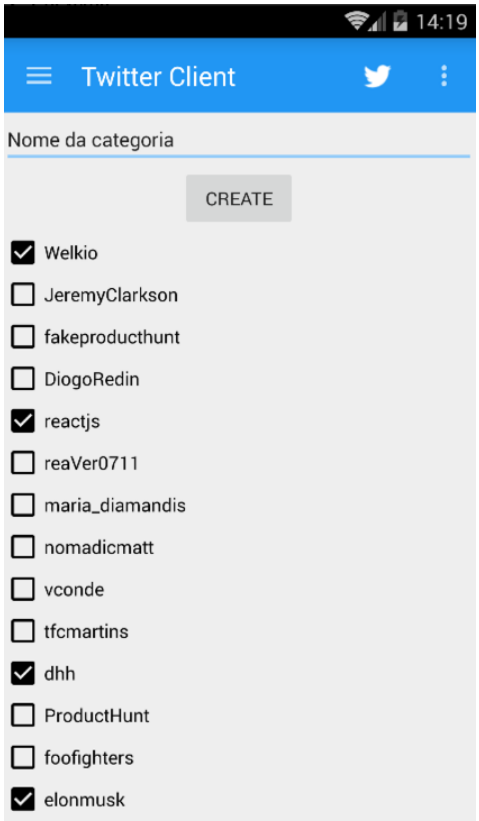
A interface mostra 2 botões por cada tweet, o primeiro para fazer retweet e o segundo para responder ao mesmo tweet.



Categorias

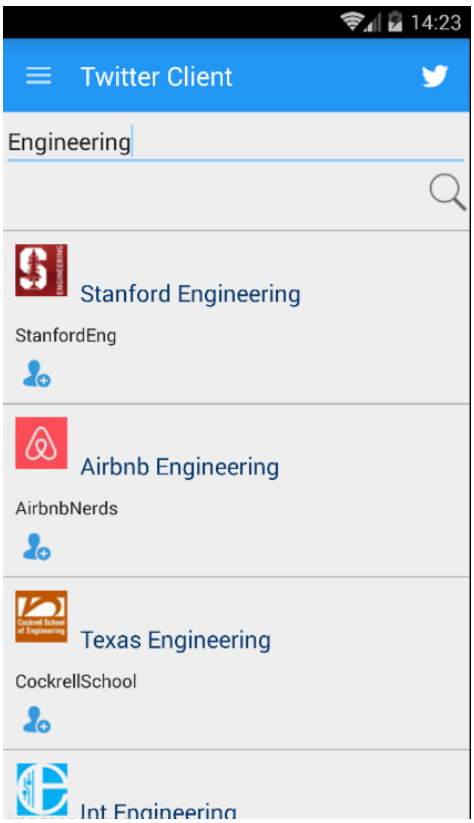
Através do menu lateral, o utilizador pode aceder à página de criar uma nova categoria, onde deve escolher os utilizadores a atribuir a essa categoria e um nome para a mesma.

Após criada a categoria, também através do menu lateral é possível abrir a mesma, onde serão mostrados apenas os tweets dos utilizadores selecionados anteriormente.



Pesquisar e seguir utilizadores

Também foi implementada a funcionalidade de pesquisar e seguir utilizadores. Através do menu lateral é possível aceder a página de pesquisa, onde facilmente podemos encontrar novos utilizadores e utilizar o botão apropriada para os seguir.



Implementação

O projeto consiste numa aplicação para Android. Como tal, a linguagem utilizada em todo o desenvolvimento foi Java.

Para atualizar o estado da aplicação, de forma a receber novo conteúdo ao sistema, são feitos pedidos à API do Twitter. Estes pedidos são efetuados recorrendo à TwitterApiRequest, objeto do tipo AsyncTask, que tem todos os métodos necessários para para obter dados do servidor, seja por GET ou POST.
Quando o objeto é criado, é iniciada uma tarefa assíncrona, permitindo fazer os pedidos sem interromper o funcionamento da aplicação.

OAuth 1.0

Todos os pedidos feitos à API do Twitter devem ser devidamente assinados utilizando o standard OAuth 1.0. Como tal, o grupo decidiu criar a sua própria implementação dos métodos necessários para a utilização deste standard.

O método generateSignature(), na classe TwitterApiRequest recebe como argumentos os dados do pedido:

- Método HTTP
- URL
- Parâmetros
- Secret Token

De notar que os parâmetros terão que incluir no mínimo os seguintes dados:

- **oauth_consumer_key** - Consumer Key da aplicação Twitter
- **oauth_nonce** - string única e aleatória
- **oauth_timestamp** - UNIX timestamp do pedido
- **oauth_signature_method** - Método de encriptação da oauth_signature, neste caso HMAC-SHA1
- **oauth_token** - o token do utilizador atual (pode ser nulo caso se trate do pedido de login inicial)
- **oauth_version** - versão do standard OAuth a ser utilizado, neste caso o 1.0.

Com estes dados, o método ira criar uma string representativa do pedido, num formato especial, com todos os dados ordenados alfabeticamente, que será posteriormente encriptada com HMAC-SHA1, usando o Consumer Secret da aplicação do Twitter em conjunto com o Secret Token passado à função, e codificada em Base64.

Todos os dados relativos ao OAuth passados inicialmente são incluídos, em conjunto com a oauth_signature, no Authorization header dos requests feitos pela classe TwitterApiRequest.

POST e GET

Na aplicação são efetuados requests de dois tipos: POST e GET. Estes requests são implementados nos métodos GET e POST, na classe TwitterApiRequest, que recebem o url do request e os parâmetros a enviar. A resposta dos requests feitos é recebida no formato JSON.

Ambos os métodos criam um DefaultHttpClient, que é usado para executar o request, e um HttpPost/HttpGet, que contém o request a fazer, assim como os *headers* e parâmetros necessários.
Tal como foi mencionado anteriormente, os requests à API utilizam o standard O_Auth 1.0. Assim, antes de serem executados, é adicionado o header Authentication com todos os parâmetros necessários para validar o request através do O_Auth 1.0, explicados anteriormente.

Após recebidas as respostas do request efetuado, armazenadas num objeto do tipo HttpResponse, é feito parse para os objetos correspondentes.

A implementação dos métodos POST e GET podem ser encontrada no ficheiro TwitterApiRequest.java, na linha 141 e 220, respetivamente.

Armazenamento dos Dados

Para armazenar todo o conteúdo recebido pela API do Twitter e todo o conteúdo criado pelo utilizador, é utilizada a base de dados SQLite do Android, que é gerida recorrendo à DatabaseHandler, classe criada que contém todas as informações e métodos para criar, aceder e editar a base de dados. Assim, toda a implementação do código relativo ao gerenciamento da base de dados encontra-se no ficheiro DatabaseHandler.java.

Interface do utilizador

Todos os componentes da aplicação, que irão ser explicados de seguida, contêm uma barra de navegação lateral. Esta barra contém vários botões que permitem alternar entre as diversas componentes do sistema.
Estes botões, quando clicados, mudam a atividade a ser apresentada ao utilizador, apresentando a associada ao botão.

Login

Caso o utilizador não tenha um login já efetuado, a aplicação irá fazer um pedido ao seguinte endpoint:

https://api.twitter.com/oauth/request_token

Este endpoint irá devolver um token temporário, que servirá para redirecionar o utilizador para uma página onde este poderá autorizar a nossa aplicação a aceder à sua conta.

O endereço desta página é dado por: https://api.twitter.com/oauth/authenticate e tem como parâmetro o token temporário recibo anteriormente.

Após um login com sucesso, a página irá chamar um callback - oauth://sdis-twitter-client - que foi registado pela nossa aplicação, e como tal o sistema Android irá automaticamente redirecionar o pedido para a nossa aplicação.

Com este pedido, obtemos um token_verifier que servirá para obter um access_token de longa duração. Para tal, é necessário aceder ao endpoint: https://api.twitter.com/oauth/access_token e passar este mesmo verifier.

Obtém-se finalmente um token de longa duração que pode ser utilizado por tempo ilimitado para aceder a todas as funcionalidades da nossa aplicação.

Quando a aplicação é aberta pela primeira vez, é atualizada a *timeline*, os *tweets* inseridos e os utilizadores seguidos. Para tal, recorre-se à tarefa assíncrona referida anteriormente - *TwitterApiRequest* - para obter o conteúdo do sistema. Os resultados obtidos são imediatamente adicionados à base de dados local, de forma a evitar perdas de dados e a evitar fazer pedidos à API sempre que se abre a aplicação.

Timeline

Na timeline são apresentados todos os tweets inseridos pelos utilizadores a seguir, por ordem cronológica. O conteúdo apresentado nesta interface é obtido acedendo, primeiramente, à base de dados local para obter os dados já obtidos pelos pedidos ao servidor. Caso a base de dados ainda não contenha quaisquer dados, é feito um pedido ao servidor (https://dev.twitter.com/rest/reference/get/statuses/home_timeline) para atualizar a base de dados e, posteriormente, apresentar todos os resultados obtidos na interface.

Neste componente do sistema é permitido ao utilizador atualizar o estado da aplicação, recebendo os conteúdos mais atualizados do Twitter. Para tal, é utilizado um SwipeRefreshLayout, que quando é acionado faz novos pedidos à API - https://dev.twitter.com/rest/reference/get/statuses/home_timeline -, através da tarefa

assíncrona de acesso à API, e atualiza o conteúdo apresentado ao utilizador, assim como os dados presentes na base de dados.
Os novos tweets recebidos e ainda não lidos, são colocados acima da timeline. Sempre que a *timeline* é apresentada, a aplicação faz *scroll down* automaticamente de forma a colocar os tweets não lidos na parte superior da timeline, sendo necessário fazer *scroll up* para os visualizar. À medida que o utilizador faz *scroll up* e visualiza os tweets ainda não lidos, estes são marcados na base de dados como lidos.

Cada *tweet* apresentado na *timeline* tem dois botões:

1. *Retweet* - permite fazer *retweet* do *tweet* apresentado. Quando clicado, este botão envia um POST, com o id do *tweet*, para a API - https://api.twitter.com/1.1/statuses/retweet/:id.json. O servidor do Twitter é encarregue de fazer o *retweet* pedido pelo POST enviado.
2. *Reply* - permite responder ao *tweet* apresentado. Quando clicado, é apresentada uma caixa de texto para escrever a resposta ao *tweet* e dois botões, para cancelar e para responder. Quando o botão para responder é pressionado, é enviado um POST, com a resposta e com o id do *tweet* a que se esta a responder, para a API - https://api.twitter.com/1.1/statuses/update.json. O servidor do Twitter é encarregue de adicionar a resposta ao sistema.

Por fim, na barra superior tem um botão para criar um novo *tweet*. Quando pressionado, uma caixa de texto para escrever o conteúdo do *tweet* e dois botões, para cancelar e para criar. Quando o botão para criar é pressionado, é enviado um POST, com o conteúdo do *tweet*, para a API (https://api.twitter.com/1.1/statuses/update.json). O servidor do Twitter é encarregue de adicionar o tweet ao sistema.

Categorias

Esta componente da aplicação contém três interfaces distintas:

1. Interface de criação de uma nova categoria
2. Interface para visualização de todas as categorias
3. Interface para visualização dos *tweets* de uma categoria especifica

A primeira interface contém uma EditText para atribuir um nome à categoria, uma lista com todos os utilizadores a seguir, para se selecionar quais adicionar à categoria, e um botão para criar a categoria.
Quando o botão é pressionado, e de forma a adicionar a categoria, são efetuados os seguintes passos:

1. Verifica-se se foi atribuído um nome à categoria
2. Adiciona-se a categoria à tabela das categorias, na base de dados
3. Percorre-se a ArrayList que contém todos os utilizadores adicionados à categoria e adiciona-se, um a um, os utilizadores à tabela dos utilizadores da categoria, na base de dados.

A segunda interface contém uma lista com todas as categorias do utilizador da aplicação. Cada categoria presente na lista contém dois botões:

- Para visualizar a categoria
    - Abre a terceira interface. Para tal, cria uma atividade do tipo SpecificCategoryActivity, passando-lhe como parâmetro a categoria em questão.
- Para remover a categoria
    - Remove a categoria da base de dados, recorrendo ao método removeCategory(Category category), implementado num objeto do tipo DatabaseHandler, que contém a base de dados da aplicação.
    - Atualiza a interface, removendo da lista a categoria selecionada.

A terceira interface funciona de forma semelhante à *Timeline.* No entanto, filtra os *tweets* apresentados, mostrando apenas os *tweets* dos utilizadores que foram adicionados à categoria.

Pesquisar e seguir utilizadores

Esta componente permite ao utilizador da aplicação pesquisar por outros utilizadores e segui-los, caso deseje.
Nesta componente é apresentada uma EditText onde é inserida a query de pesquisa e um botão para fazer a pesquisa. Quando o botão é pressionado, é feito um pedido à API - https://api.twitter.com/1.1/users/search.json, passando como parâmetro a *string* inserida na EditText. Neste pedido é recebida uma lista com os utilizadores que contêm, no seu *username* ou nome, a *string* inserida na pesquisa.
Após se fazer parse do resultado obtido, são apresentados todos os resultados numa lista.
Cada utilizador apresentado na lista, contém um botão que permite seguir o utilizador em questão. Quando este último botão é pressionado, é feito um pedido, do tipo POST, ao servidor para seguir o utilizador - https://api.twitter.com/1.1/friendships/create.json.

Problemas Relevantes

Segurança

A aplicação requer segurança quando está a autenticar o utilizador com a conta do Twitter e quando está a haver transferência de dados para o servidor.

Como foi anteriormente descrito, o grupo utilizou uma implementação própria do standard OAuth 1.0 e todos os pedidos HTTP são feitos de forma segura, utilizando SSL.

A utilização de OAuth permite garantir que apenas o utilizador que fez login na aplicação irá enviar dados para a API do Twitter, dados esses que são encriptados por SSL, usando o certificado SSL da API do Twitter.

Um exemplo da utilização de OAuth no código pode ser encontrado no método POST(String url, List<NameValuePair> params) presente no ficheiro TwitterApiRequest.java, na linha 141.

Tolerância a erros

A aplicação vai gravando os dados transferidos do Twitter na base de dados SQLite do Android. Assim, e tendo em conta que os dados são gravados imediatamente à medida que a aplicação é usada, não devem existir quaisquer problemas quanto a perdas de dados em caso de erro da aplicação.

Para gravar todos os dados criamos a classe DatabaseHandler, presente no ficheiro DatabaseHandler.java. Esta classe contém todos os métodos relacionados com a criação e edição da base de dados, que são chamadas à medida que novos dados são recebidos pelos pedidos feitos à API. Um exemplo desta utilização encontra-se no ficheiro TwitterAPIRequestThread.java, na linha 72.

Conclusão

O cliente para o Twitter implementado permite realizar todas as operações mais importantes do Twitter de uma forma simples e rápida. No entanto, a maior utilidade e o ponto mais forte da aplicação implementada é o facto de permitir organizar os utilizadores que se está a seguir por categorias.

Ao nível da segurança e das possíveis falhas, a aplicação é bastante segura, na medida em que utiliza protocolos testados e seguros e uma base de dados para manter sempre os últimos estados e dados da aplicação guardados, evitando perdas aquando a ocorrência de falhas.

Apesar de já estar a permitir realizar as operações mais importantes e de já ser bastante seguro, ainda existem imensos pontos a melhorar, nomeadamente ao nível da interface e da intuitividade, e funcionalidades a implementar para tornar a aplicação mais rica, como por exemplo apresentar o números de *tweets* ainda por ler em cada categoria, ou visualizar o perfil de outros utilizadores.

Divisão do trabalho

José Melo - Responsável pelo desenvolvimento dos acessos à base de dados SQLite, pedidos HTTP que obtêm informação sobre utilizadores, tweets e pesquisa. Participante no desenvolvimento da interface gráfica Android.
Autor de 50% do trabalho.

Rui Gomes - Responsável pelo desenvolvimento das chamadas HTTP relacionadas com o login e informação sobre utilizadores, geração das assinaturas e implementação do standard OAuth, e também responsável pelo desenvolvimento da parte gráfica Android.
Autor de 50% do trabalho.

Bibliografia

Para a implementação desta aplicação recorremos aos seguintes websites:

- https://dev.twitter.com/rest/public
- http://developer.android.com/
- http://stackoverflow.com
- http://quonos.nl/oauthTester/
- http://oauth.net/core/1.0/