

Relatório do Projecto

NearTweet

Computação Móvel - 2012/13



Curso: MERC

Grupo: 01

Nome: Luís Carvalho Número: 55982 E-mail: luis.c.carvalho@ist.utl.pt

Nome: Bernardo Santos Número: 57437 E-mail: bernardompsantos@ist.utl.pt

Nome: Dinamene Barreira Número: 57447 E-mail: dinamene.barreira@ist.utl.pt

1. Trabalho Realizado

Este projecto tem como objectivo a realização de uma aplicação para dispositivos móveis, que visa trazer uma nova forma de utilização da aplicação existente *Twitter* (serviço de rede social e *microblogging*, que permite aos seus utilizadores partilharem conteúdo em mensagens pequenas até 140 caracteres) permitindo o uso deste tipo de dispositivos para a criação de “redes instantâneas” de proximidade criadas através de agrupamentos de pessoas, recorrendo ao uso de redes *wireless Ad-Hoc*, redes sem-fios descentralizadas que não necessita de uma infra-estrutura pré-existente e em que cada nó participa no encaminhamento dos dados.

Com esta aplicação é possível realizar partilha de mensagens para a rede, responder a mensagens e partilhar para outra aplicação (neste caso *Facebook*), partilhar conteúdo, como fotos, *links* para *sites*, localização do utilizador, realização de votações. Caso os utilizadores sintam que outros se encontram a cometer infracções, como por exemplo o por conteúdos impróprios, ainda oferecemos a possibilidade de notificar o servidor para lhes bloquear o acesso, marcando as mensagens como spam.

Devido a alguns problemas encontrados, optou-se por fazer a implementação das funcionalidades requisitadas para o projecto recorrendo ao uso da arquitectura cliente-servidor, para tal a actualização dos dados é efectuada sempre que os utilizadores se encontrem num *HotSpot*, cobertura de uma rede Wi-Fi que permite a ligação ao servidor.

2. Especificação

Esta aplicação *NearTweet*, pretende disponibilizar uma forma de comunicação através de dispositivos móveis para ajuntamentos de pessoas de acordo com o local onde se encontram, sem ter que recorrer ao uso de redes de dados, que muitas vezes se tornam dispendiosas. Sabendo que quando este tipo de agrupamentos ocorrem (concertos, maratonas, protestos, refeitórios), maior parte dos indivíduos não se conhecem, desta forma permite-se aos utilizadores a opção de partilhar novidades sobre o local onde estão, ou simplesmente informarem-se do que se passa à sua volta.

Nesta aplicação, como referido anteriormente, a actualização da informação para a aplicação ocorre através de *HotSpots* onde se encontram ligações com o servidor, por proximidade, assim sempre que os utilizadores se encontram perto de um *HotSpot*, automaticamente são notificados das actualizações, caso queiram partilhar algo e se encontrem numa zona escura, ou seja não coberta, será posto em secundário e actualizado automaticamente para o servidor assim que voltem a uma zona coberta.

3. Design

Esta aplicação foi desenhada para sistemas *Android*. Para tal é utilizada a arquitectura do mesmo, podendo ser observada na Figura 1. De modo a conseguirmos que a aplicação seja bem implementada recorreremos ao uso de elementos da arquitectura, como as *Activities*, *Services*, *AsyncTask*, *Handlers*, entre outros.

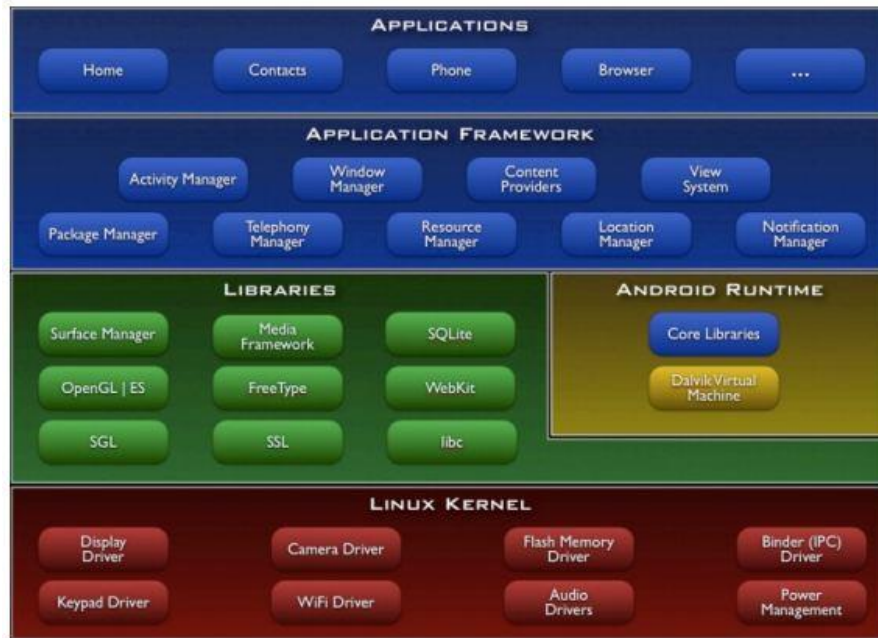


Figura 1 – Arquitectura do sistema Android

A rede definida para a implementação desta aplicação é uma rede cliente-servidor, apesar desta rede não ser tão desejada como o *WiFi Direct*, permite na mesma aos utilizadores uma sensação de obiquidade, pois estes só precisam de se encontrar numa zona coberta por um *HotSpot*. Pode ser observado na Figura 2, a imagem de uma comunicação de um dispositivo móvel com o servidor.

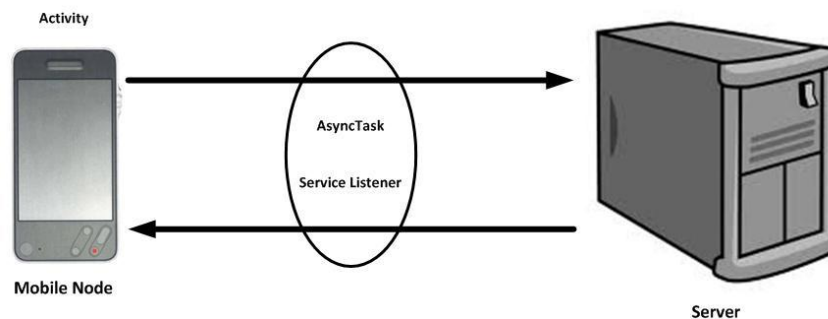


Figura 2- Arquitectura da Rede

A comunicação é realizada recorrendo ao uso de sockets, que fazem a escuta da comunicação entre os clientes e o servidor, apesar de na figura só se encontrar um dispositivo móvel na realidade são vários os dispositivos que se ligam em simultâneo ao servidor.

Esta aplicação funciona da seguinte forma, o utilizador quando escolhe uma opção gera uma *Activity*, esta por sua vez manda a informação para o servidor recorrendo a uma *AsyncTask* para realizar a comunicação. A aplicação tem um serviço que corre em segundo plano (*ServiceListener*) estando sempre à escuta para ver se ocorreram novas mensagens, para poder assim notificar o utilizador das mesmas. Este *Listener* fornece também ao utilizador a lista actualizada de mensagens sempre que ocorreu uma acção (*Tweet*, *Reply*, etc).

4. Escolhas de Implementação

Para o desenvolvimento desta aplicação optou-se pela utilização do sistema operativo *Android* que é baseado no Linux para dispositivos móveis. Este sistema permite a criação de novas aplicações com a integração de serviços e aplicações já existentes. Durante o desenvolvimento desta aplicação escolheu-se o uso da linguagem JAVA para a sua implementação, devido à familiaridade existente com esta e à quantidade de informação disponível para aplicações para este tipo de sistema.

O sistema *Android* escolhido para a implementação desta aplicação é o *Jelly Bean* (4.2.2). Para a realização de testes e de utilização da aplicação recorreremos à emulação de dispositivos *Android*, utilizando as ferramentas disponibilizadas pelo Eclipse, *software* de programação que suporta o uso de diferentes linguagens de programação.

Do ponto de vista do sistema *Android*, fazemos uso das seguintes estruturas de elementos, *layouts*, permitenos criar a interface gráfica das actividades, *values*, como strings e *ids* permitindo estruturar o tipo de informação obtida pelo utilizador, *activities*, que nos permite criar as funcionalidades desejadas da aplicação, *services*, podendo assim ter operações a correr em simultâneo em *background* com a aplicação, sendo muito útil para a actualização da informação desta, *handlers*, responsável pelas mensagens enviadas pelos serviços para a actividade que o está a invocar.

5. Avaliação Experimental

Este projecto pode ser testado recorrendo a um simulador de *Android* disponibilizado pela ferramenta de *eclipse*, para tal é necessário iniciar duas instâncias do simulador e definir as ligações localmente recorrendo à consola com a ferramenta *telnet*.

Começa-se por executar o servidor identificado por *NearTweet_Server*, em cada uma das instâncias por a aplicação a executar recorrendo ao projecto *NearTweet_Client*. Após a realização destes, tem que ser feito em cada aplicação o login para cada uma das instâncias.

a. Teste 1 : Tweet

O utilizador A clica no botão *Tweet*, escreve a mensagem desejada e depois clica novamente no botão *SendTweet* e poderá ver a sua mensagem a aparecer na lista de *Tweets* da aplicação. O utilizador B é notificado de que ocorreu uma alteração e a sua lista de *Tweets* é actualizada, com a mensagem enviada pelo utilizador A.

b. Teste 2 : Reply

O utilizador B escolhe um *Tweet* e pressiona longamente até lhe aparecer um *Pop Up Menu* o qual deverá escolher a opção *Reply*, após o que deve proceder à escrita da mensagem desejada. Quando acabou de escrever a mensagem deve carregar então no botão *SendTweet*. Como isto é um *Reply*, não irá aparecer na lista de *Tweets*, ficando associado ao *Tweet* específico. Para ver o a mensagem de resposta deve-se clicar no próprio *Tweet*. Tanto para o utilizador A como para o tilizador B.

c. Teste 3 : Reply All

Este teste permite que o utilizador faça uma resposta para todos, ou seja, uma resposta pública, neste caso deve fazer os mesmos procedimentos que no ponto anterior, com a diferença em que deve seleccionar o botão *ReplayAll* e esta mensagem aparece na lista de *Tweets* ficando visível para todos os utilizadores.

d. Teste 4 : ReTweet

O utilizador A escolhe um *Tweet* e pressiona longamente até lhe aparecer o *Pop Up Menu* o qual deverá escolher a opção *ReTweet*, para este teste funcionar é necessário que o simulador tenha uma conta e aplicação do *Facebook* activa, a aplicação de *Facebook* procede a uma confirmação de autorização, assumindo que o utilizador o aceita, a aplicação procede à publicação do *Tweet* seleccionado no mural do utilizador.

e. Teste 5 : Poll

O utilizador pode recorrer ao botão de menu do próprio telemóvel, neste caso o simulador, para criar uma eleição, com várias perguntas e respostas possíveis em que os outros utilizadores podem seleccionar essa *Poll* e responder simplesmente escolhendo qual a resposta que mais se adequa para si.

f. Teste 5 : Sharing

O utilizador A procede à realização de um *Tweet* normal, dentro deste existe um menu em que permite a escolha de opções de *Sharing*, Como a partilha de fotos, botão *Photo*, ou mesmo da localização do utilizador, com o botão *GPS*.

i. Photo :

Esta opção permite ao utilizador ir buscar imagens de três sítios diferentes, como o *Clipboard*, de um URL, ou mesmo da câmara do equipamento, após a escolha de uma destas opções escreve a mensagem desejada e depois e só fazer *SendTweet*, e fica partilhado na lista de *Tweets*.

ii. GPS :

Esta opção permite ao utilizador partilhar a sua localização, utiliza-se o mesmo procedimento que um *Tweet* simples e aparece público na lista de *Tweets*.

g. Teste 7 : Spam

Quando um utilizador acha que outro se encontra a enviar informação imprópria, pode notificar o servidor de que esse utilizador o está a fazer, para tal deve escolher o *Tweet* correspondente ao utilizador, pressionar durante algum tempo até aparecer o *Pop Up Menu* e escolher a opção *SPAM*. Para verificar que isto se encontra funcional é necessário marcar um *Tweet* como *SPAM* 2 vezes, o servidor após este acontecimento o servidor bloqueia o utilizador responsável pelos *Tweets*.

6. Conclusões

Este trabalho notou-se complicado do ponto de vista de nos conseguirmos familiarizar com a forma como a estrutura de *Android* se encontra, nomeadamente, a interacção entre componentes nativos, do sistema, face aos requisitos pedidos para este trabalho. No entanto, foi-nos possível realizar as funcionalidades mínimas pretendidas como referido anteriormente.

Para recorrer ao uso do simulador pretendido, neste caso o *WDSim*, a sua apresentação e disponibilidade deve ser feita o mais cedo possível face ao calendário da cadeira.