



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Frame Social: Um framework para acesso a dados em dispositivos móveis para o desenvolvimento de Redes Sociais Ad hoc

Djalma José da Silva Júnior
Mateus Borges Damasceno

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientadora
Prof.^a Dr.^a Maristela Terto de Holanda

Brasília
2014

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Coordenadora: Prof.^a Dr.^a Maria Emília Machado Telles Walter

Banca examinadora composta por:

Prof.^a Dr.^a Maristela Terto de Holanda (Orientadora) — CIC/UnB
Prof.^a Dr.^a Aletéia Patrícia Favacho de Araújo — CIC/UnB
Prof.^a Dr.^a Edna Dias Canedo — FGA/UnB

CIP — Catalogação Internacional na Publicação

da Silva Júnior, Djalma José.

Frame Social: Um framework para acesso a dados em dispositivos móveis para o desenvolvimento de Redes Sociais Ad hoc / Djalma José da Silva Júnior, Mateus Borges Damasceno. Brasília : UnB, 2014.

123 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2014.

1. bancos de dados móveis, 2. redes sociais mediadas por computador,
3. *Wi-Fi Direct*, 4. *Wi-Fi P2P*, 5. *SQLite*

CDU 004.4

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília-DF — Brasil

Dedicatória

Dedico aos meus pais, pela fé e confiança demonstrada, pelo apoio nas horas difíceis e por estarem sempre ao meu lado, ajudando-me a finalizar o curso.

(Mateus)

Dedico este trabalho à minha mãe, Cida (*in memoriam*),

ao meu pai, senhor Djalma, herói guerreiro,

à minha esposa, Paula, com quem aprendo o gosto de construir uma vida deslumbrante, e à minha filha, Amora, que me ilumina todo dia com seu sorriso ensolarado.

(Djalma)

Agradecimentos

Agradeço a Deus, por ter me dado saúde e força para superar as dificuldades.

À minha orientadora, pela paciência na orientação e por estar sempre disposta a ajudar no desenvolvimento deste projeto.

À minha família, por sua capacidade de acreditar em mim.

Ao Max Canedo, por se dispor a fazer vistas ao texto da monografia.

Ao Luiz Marcus, por nos ajudar a resolver problemas no desenvolvimento deste projeto.
(Mateus)

Agradeço ao Mateus por compartilhar essa jornada colaborativa de ensino.

À Professora Maristela, pela orientação deste estudo, pelo apoio no momento que perdi minha mãe, por me educar na especialidade que depois virou minha profissão.

Aos professores que já tive, sempre admirados.

Aos meus amigos, que são a família que a vida me permite escolher.

À minha família, o meu ninho.

(Djalma)

Resumo

A utilização de dispositivos móveis é crescente, assim como das redes sociais. Em relação aos meios de comunicação para ambientes de computação móvel, este trabalho pesquisa o protocolo *Wi-Fi Direct* na implementação da troca de dados em um ambiente móvel *ad hoc*, além de investigar o sistema gerenciador de banco de dados *SQLite* na implementação da persistência de dados nos dispositivos conectados. Portanto, o trabalho une os conceitos de computação móvel *Ad hoc*, persistência de dados e redes sociais.

Palavras-chave: bancos de dados móveis, redes sociais mediadas por computador, *Wi-Fi Direct*, *Wi-Fi P2P*, *SQLite*

Abstract

The mobile devices usage is increasing, as well as social networks usage. Regarding media to mobile computing environments, this study researches protocol Wi-Fi Direct to implement data exchange in a mobile environment *ad hoc*, besides researches database management system SQLite to implement data persistence on connected devices. Therefore, this study unites the concepts of mobile computing *ad hoc*, data persistence and social networks.

Keywords: mobile databases, computer-mediated social networks, Wi-Fi Direct, Wi-Fi P2P, SQLite

Sumário

1	Introdução	1
1.1	Objetivo Geral	2
1.1.1	Objetivos Específicos	2
1.2	Metodologia	3
1.3	Estrutura do Trabalho	3
2	Referencial Teórico	4
2.1	Redes Sociais Mediadas por Computador	4
2.1.1	Aplicativos de Redes Sociais Móveis	7
2.2	Redes de Computadores Móveis	8
2.2.1	Redes de Computadores Móveis	10
2.2.2	Redes de Computadores Móveis <i>Ad-hoc</i>	11
2.2.3	A tecnologia de Rede <i>Wi-Fi Direct</i>	12
2.3	Bancos de Dados em Dispositivos Móveis	15
2.3.1	Bancos de Dados	15
2.3.2	Bancos de Dados Móveis <i>SQLite</i>	18
2.3.2.1	Diferenciais do <i>SQLite</i>	19
2.4	Sistema operacional <i>Android</i>	20
2.4.1	Arquitetura do <i>Android OS</i>	23
2.4.1.1	Visão geral da API <i>Wi-Fi Direct</i> no <i>Android OS</i>	24
2.5	Trabalhos Relacionados	25
2.5.1	<i>Eiko</i>	25
2.5.2	<i>MobiSN</i>	25
2.5.3	<i>My-Direct</i>	25
2.5.4	<i>MobiSoft</i>	26
2.5.5	<i>MyNet</i>	26
3	<i>Frame Social: um framework para redes sociais móveis Ad hoc</i>	27
3.1	Arquitetura do <i>Frame Social</i>	27
3.2	Fluxo do Processo de Comunicação e Persistência entre Dispositivos	28
3.3	Implementação do Projeto no <i>Android</i>	30
3.3.1	Customização da Rede Social pelo Programador	30
3.3.2	Módulo de Comunicação	31
3.3.3	Opção de <i>Chat</i>	32
3.3.4	Opção de Envio de Perfil	32
3.3.5	Módulo de Persistência	32

3.3.6	Consulta dos Contatos	33
3.3.7	Operações Disponíveis sobre o Banco de Dados	34
3.3.8	ETL entre os Bancos de Dados <i>SQLite</i>	34
4	Aplicação da proposta	36
4.1	Resultados do <i>Frame Social</i>	36
4.1.1	Utilização da Rede Social pelo Usuário	37
5	Contribuições do Estudo e Conclusão	48
5.1	Análise dos Resultados	48
5.2	Trabalhos Futuros	50
	Referências	52

Lista de Figuras

2.1	Rede social representada em grafo	5
2.2	Estatística do uso do Facebook, por Nielsen (2012a)	6
2.3	Uma rede ponto-a-ponto.	9
2.4	Uma arquitetura genérica de uma plataforma móvel, (Elmasri and Navathe, 2004).	10
2.5	A arquitetura de uma rede <i>ad-hoc</i> móvel (MANET), extraído de Elmasri and Navathe (2004).	11
2.6	Rede móvel <i>Ad-hoc</i> (de Rezende, 2004).	12
2.7	O procedimento de convite permite que um dispositivo <i>Wi-Fi Direct</i> torne-se cliente de um grupo existente (Alliance, 2010).	14
2.8	Mapa conceitual para bancos de dados	16
2.9	Arquitetura de um sistema utilizando SGBD, por Elmasri and Navathe (2004).	17
2.10	Projeto de dados no <i>SQLite</i>	19
2.11	A tela inicial do <i>Frame Social</i> executado em emulador <i>Android</i>	22
2.12	Edição de perfil do usuário do <i>Frame Social</i> executada em emulador <i>Android</i>	22
2.13	Diagrama da arquitetura do <i>Android OS</i> (Android, 2013b).	24
3.1	Arquitetura abstrata do <i>framework Frame Social</i>	28
3.2	Fluxograma de uso do <i>framework Frame Social</i> em dois dispositivos.	29
3.3	Fluxograma da comunicação entre dispositivos pelo <i>framework Frame Social</i>	30
3.4	Modelo de dados do <i>framework Frame Social</i>	33
4.1	Código da classe <i>DefineRedeSocial.java</i>	37
4.2	Menu principal.	38
4.3	Cadastrando um perfil.	39
4.4	Busca de dispositivos.	40
4.5	Dispositivos encontrados.	40
4.6	Opções para conexão.	41
4.7	Convite para conexão.	41
4.8	Opções para comunicação.	42
4.9	Mensagem de perfil recebido.	43
4.10	Tela de conversa entre usuários.	43
4.11	Dispositivos exibindo uma conversa entre usuários.	44
4.12	Menu de contatos cadastrados.	45
4.13	Lista de amigos e perfil do amigo selecionado.	46
4.14	Lista de amigos de amigos e perfil do amigo selecionado.	46

Capítulo 1

Introdução

O avanço tecnológico na computação tem aumentado a utilização de dispositivos móveis e, conseqüentemente, a facilidade de comunicação entre pessoas. De acordo com um levantamento da [Nielsen \(2012b\)](#), provedora global de informações e *insights* "As vendas de *smartphones* encerraram o ano de 2011 com um crescimento de 179%, comparado ao ano de 2010". Os dispositivos móveis possibilitam o acesso à informação a qualquer momento e em todo lugar. Existe uma grande variedade de aplicativos móveis atualmente, desde jogos a mídias sociais, passando por educação, dentre outros. Embora haja um grande interesse em jogos, aplicativos de mídia social também têm grande preferência entre os usuários. Isso é devido à necessidade da pessoa de mostrar sua identidade no seu ambiente. Segundo [Tomaél et al. \(2005\)](#), as razões que se destacam para o uso de redes sociais são "laços ou conexões", "comunidade" e "compartilhamento". Por exemplo, pessoas compartilham fotos com amigos, compartilham vídeos como prova de identidade, compartilham músicas que gostam para passar uma impressão sobre elas às demais pessoas ([Tomaél et al., 2005](#)).

Unido a essa facilidade de acesso à informação, as redes sociais mediadas por computador se destacam como um espaço importante de compartilhamento. [Tomaél et al. \(2005\)](#), [Lemos \(2004\)](#), [Marteleto \(2001\)](#) e também [Takahashi \(2000\)](#) apontam as redes sociais como canais para atuação de aprendizagem colaborativa, que leva à inovação. Compartilhamento e comunicação são essenciais às pessoas e isso deve ser representado no campo tecnológico.

Nesse contexto, aplicativos de mensagem instantânea e compartilhamento de arquivos são os fatores chave. Com a seleção de alguns aplicativos relacionados, pode-se fazer e receber chamadas de voz e trocar mensagens sem pagar pelo modo de tarifação telefônica. Aplicativos grátis de mensagem estão pela primeira vez pressionando o modelo clássico, portanto, pago, de SMS (*Short Message Service*) da telefonia e permitindo compartilhamento de conteúdos de várias novas maneiras.

Para manter os sistemas mencionados acima, há a necessidade de conexão de dados no *smartphone*. Portanto, é necessária uma rede de telefonia móvel 3G ou um ponto de acesso à internet. Matérias publicadas como ([Nielsen, 2013b](#)), ([Boudreau and Arathoon, 2008](#)) e ([Nielsen, 2012c](#)) apontam que é inegável o forte interesse comercial em aplicações de redes sociais, que tenham grande número de usuários, por parte das empresas envolvidas nas diversas ramificações dessa questão. Sejam empresas de mídia, ou publicitárias, ou de telefonia, ou infraestrutura de conexão, ou indústria de sistemas e de eletrônicos,

até mesmo eleitorais. Como exemplo, existe interesse público em alguma aplicação que melhore a atuação de agentes em uma busca policial, possibilitando o compartilhamento em tempo real de imagens ou posicionamento de evidências para a localização de um desaparecido (talvez um foragido), em ambiente sem cobertura de rede celular (talvez após uma catástrofe climática).

A tecnologia 3G é usada em telefonia sem fio, acesso móvel à internet, chamadas de vídeo e TV móvel. No entanto, estabelecer conexão de rede entre os *smartphones* pode às vezes ser caro e menos efetivo. Por esta razão, pode ser necessária uma nova solução. Este projeto entra em cena para tratar esse problema. Entretanto, a tecnologia 3G não está sempre disponível. E às vezes é caro usá-la quando é preciso compartilhar informações ou mensagens entre as pessoas. Conectar-se a um ponto de acesso à rede ou 3G nem sempre pode ser possível, por exemplo, em um grande show, ou dentro de prédios de fábrica, túneis, museus etc, onde o local construído pode bloquear o sinal 3G, ou não existir pontos de acesso à rede. Portanto, conectar-se a um grupo de pessoas para trocar dados é um problema nessas situações.

Os problemas mencionados acima podem ser resolvidos por meio do estabelecimento de uma rede ponto-a-ponto entre dois ou mais dispositivos, usando suas antenas *Wi-Fi*. O protocolo *Wi-Fi Direct* surgiu com a proposta de possibilitar esse método de conexão de dispositivos, conforme apresentado por [Alliance \(2010\)](#), e desde 2010 é aperfeiçoado e embarcado cada vez mais em dispositivos. Justifica-se assim, a necessidade de pesquisas e testes sobre esse protocolo.

Este trabalho é um projeto na área da Ciência da Computação, no contexto de tecnologias de bancos de dados móveis, para auxiliar o desenvolvimento de redes sociais em ambientes *Ad hoc*. Este trabalho dá sequência aos estudos sobre comunicação de dispositivos móveis e bancos de dados móveis, publicados por [Castro and Holanda \(2011\)](#) e [Queiroz et al. \(2012\)](#).

1.1 Objetivo Geral

O objetivo deste trabalho é desenvolver um *framework* para a manipulação de dados que deve atuar como solução na implementação de uma rede social com dispositivos móveis conectados ponto-a-ponto, sem necessidade de um ponto de acesso central, isto é, redes móveis *Ad hoc*. O *framework* proposto deve fornecer ao usuário o suporte necessário para a comunicação entre dispositivos móveis e para persistência de dados.

1.1.1 Objetivos Específicos

O objetivo geral é resultado do alcance dos seguintes objetivos específicos:

- Especificar e implementar o **módulo de comunicação**, ou seja, troca de dados entre os dispositivos móveis em uma rede *Ad hoc*;
- Especificar e implementar o **módulo de persistência de dados** para redes sociais;
- Realizar uma aplicação prática do *framework*.

1.2 Metodologia

A metodologia utilizada é um estudo de caso de natureza exploratória e aplicada nos conceitos sobre redes sociais, redes de computadores e bancos de dados. Assim sendo, foram realizados:

- Estudo bibliográfico;
- Desenvolvimento de código em linguagem de programação *JAVA* na plataforma *Eclipse* em sistemas *Windows* e *Linux*;
- Testes em dispositivos *Android*;
- Análise dos resultados obtidos.

1.3 Estrutura do Trabalho

Este trabalho está estruturado, além deste capítulo, nos seguintes capítulos:

- O Capítulo 2 apresenta um referencial teórico sobre os conceitos necessários para o desenvolvimento da pesquisa. Na Seção 2.1 estão os conceitos de redes sociais e um panorama do estado atual das redes sociais mediadas por computador. A Seção 2.2 traz um resumo teórico de redes de computadores móveis descrevendo os pontos pertinentes da tecnologia *Wi-Fi Direct*. São apresentados na Seção 2.3 os conceitos de bancos de dados, alcançando o referencial necessário à aplicação dessas tecnologias em conjunto com redes de dados para a implementação de um *framework* útil para a implantação de uma rede social móvel. Nas Seções 2.3.2 e 2.4, as funcionalidades pertinentes da tecnologia *SQLite* embarcada no sistema operacional *Android* são explicadas. Finalizando o capítulo, são apresentados alguns trabalhos publicados com projetos relacionados a este, para comparação na Seção 2.5.
- No Capítulo 3 está descrita a implementação do *framework Frame Social* em código *Java* para o sistema *Android*. A Seção 3.1 consiste na visão geral da arquitetura de software. Além disso, a ideia da solução e os pacotes e classes de código são abordados neste capítulo, na Seção 3.3. As ferramentas e a estrutura da solução de comunicação, na Seção 3.3.2, e de persistência de dados, Seção 3.3.5, seguem esse capítulo.
- O Capítulo 4 relata os resultados da aplicação prática do *framework* proposto.
- Finalmente, no Capítulo 5 o leitor encontra as análises das respostas do problema do projeto, com a consideração de soluções sugeridas como parte da discussão. Os trabalhos futuros e mais investigações também podem ser encontrados no Capítulo 5.

Capítulo 2

Referencial Teórico

Neste capítulo, os temas redes sociais, redes sociais móveis, redes de computadores, banco de dados móveis e sistema operacional *Android* são abordados. Esses conceitos são apresentados tendo em vista o contexto da tecnologia aplicada para o desenvolvimento do trabalho proposto.

2.1 Redes Sociais Mediadas por Computador

As redes sociais mediadas por computador são espaços valorizados e importantes para o compartilhamento de informação e conhecimento que, sendo utilizados de maneira inteligente, tornam-se valiosos e importantes. Conforme [Boyd and Ellison \(2007\)](#), um sistema de computação é considerado uma rede social quando possibilita que o usuário: (1) crie um perfil público ou semi-público registrado nesse sistema; (2) organize uma lista de usuários com os quais ele compartilha conexão; e (3) veja a lista de conexões feitas entre os demais usuários desse sistema.

Um fator que contribui na construção das redes sociais é a disponibilidade de acesso à Internet, despertando o interesse das pessoas para a criação de softwares com esta finalidade. Estes softwares permitem uma interação entre amigos, conhecidos e pessoas que convivem por algum motivo. Sendo assim, a rede aumenta rapidamente, e pessoas com ideias e objetivos semelhantes começam a interagir, mantendo relações sociais. Neste cenário, as pessoas passam a querer expressar suas opiniões e pensamentos de forma bem clara, deixando de ser apenas receptoras de informações como afirmam [Tomaél et al. \(2005\)](#), [Lemos \(2004\)](#) e também [Takahashi \(2000\)](#).

[Milgram \(1967\)](#) defende em sua pesquisa uma tese chamada de Mundo Pequeno, em que seis pessoas nos separam de qualquer outra pessoa no mundo. Para chegar à essa conclusão, ele enviou cartas para 160 moradores de Boston, nos Estados Unidos, escolhidos ao acaso, pedindo que estas reenviassem a carta a um destinatário, por meio de conhecidos que pudessem ajudar na entrega. Por quem a carta passava, a assinatura era feita no envelope, e o número de intermediários médio entre quem enviou e o destinatário foi seis. Com essa pesquisa, pode-se observar que as redes sociais são ferramentas poderosas na conquista de um objetivo e um método eficiente na busca por uma pessoa-alvo.

O grafo mostrado na Figura 2.1 representa as relações entre entidades de uma rede social. Pode ser observado, por exemplo, que Maria trabalha para Casas Pernambucanas e é filha de Filomena.

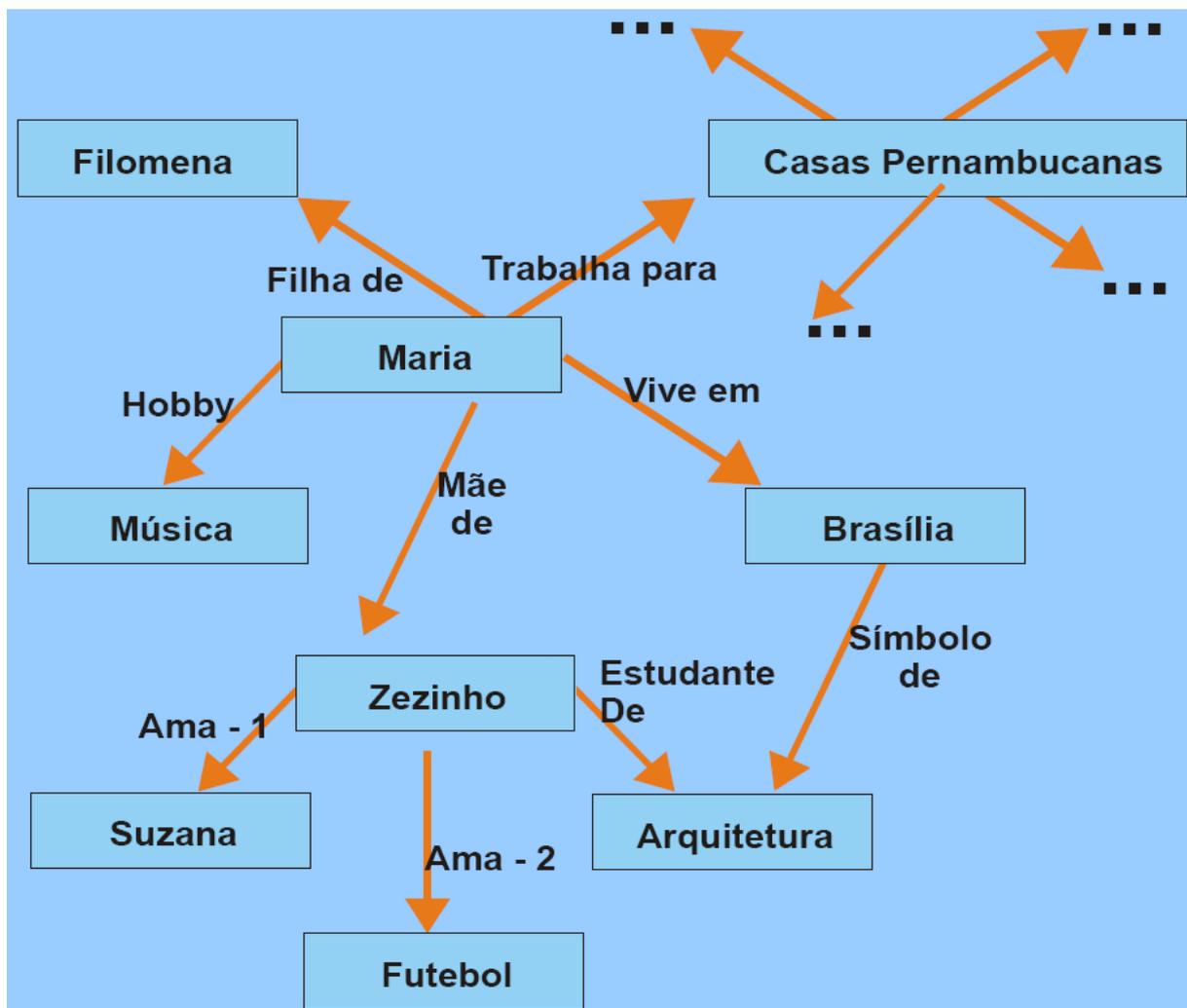


Figura 2.1: Rede social representada em grafo

Uma ferramenta de rede social bastante utilizada, atualmente, no Brasil e no Mundo é o *Facebook*. Ele foi fundado em 2004, com os estudantes da Universidade de Havard como público-alvo. Com o passar do tempo, foi se expandindo e permitindo o acesso de mais pessoas, até tornar-se público. Nele, as pessoas podem se conectar, expor ideias, compartilhar e trocar conhecimento e informação. Na Figura 2.2 pode-se ver uma estatística do uso desta rede social no Brasil, em pesquisa feita por Nielsen (2012a).

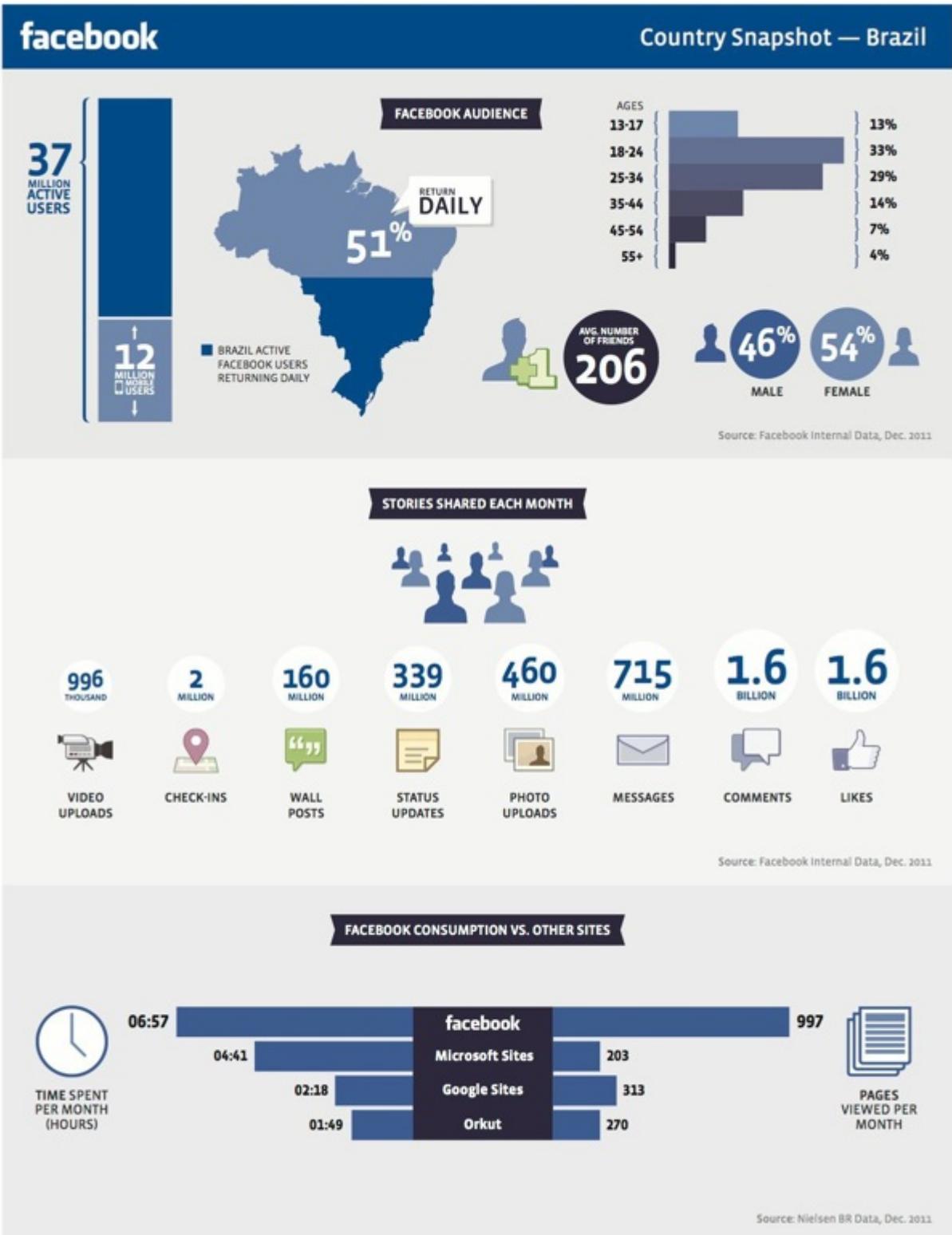


Figura 2.2: Estatística do uso do Facebook, por Nielsen (2012a)

Ainda segundo [Tomaél et al. \(2005\)](#), redes sociais são estruturas sociais compostas por pessoas ou organizações, conectadas por um ou vários tipos de relações, que partilham valores e objetivos comuns. Uma das características fundamentais das redes sociais é a sua abertura, que possibilita relacionamentos horizontais e não hierárquicos entre os participantes. Ou seja, uma rede social "é uma forma de organização caracterizada fundamentalmente pela sua horizontalidade, isto é, pelo modo de inter-relacionar os elementos sem hierarquia" ([Tomaél et al., 2005](#)). Um dos princípios das redes sociais, por ser uma ligação social, é a conexão fundamental entre as pessoas, que se dá por meio da identidade. As redes sociais mediadas por computador se destacam como um espaço importante de troca de conhecimento. [Tomaél et al. \(2005\)](#), [Lemos \(2004\)](#), [Marteleto \(2001\)](#) e também [Takahashi \(2000\)](#) apontam as redes sociais como canais para aprendizagem colaborativa, que leva à inovação.

Independentemente do tipo de pessoa que forma uma rede social, ou do tema que as une, um ponto em comum entre elas é o compartilhamento de informações, conhecimentos, interesses e esforços em busca de objetivos comuns. A intensificação da formação das redes sociais, nesse sentido, reflete um processo de fortalecimento da sociedade civil, em um contexto de maior participação democrática e mobilização social ([Takahashi, 2000](#)).

2.1.1 Aplicativos de Redes Sociais Móveis

Desde meados da década de 1990, redes sociais, baseadas na web, se ampliam e ganham mais espaço na vida das pessoas, conforme pesquisa de [Nielsen \(2012a\)](#). Com o crescente número de usuários em redes sociais, há o aumento da demanda por disponibilidade de conexão, mesmo em movimento. Para facilitar isso, tornou-se essencial a utilização de dispositivos móveis. Eles permitem e facilitam a comunicação entre pessoas, possibilitam a conexão com a internet e com outros dispositivos móveis.

As redes sociais móveis facilitam a interação de pessoas com interesses em comum. Permitem que elas troquem informações, como vídeos, fotos, currículos, músicas, dentre outros, como pode ser visto em ([Lemos, 2004](#)). Com isso, o relacionamento interpessoal pode ser aumentado, podendo ver outras pessoas expressarem suas idéias e opinar sobre isso. Junto a essa facilidade de acesso à informação, as redes sociais móveis representam um espaço importante de compartilhamento de informações para a troca de conhecimento, como descrito por [Marteleto \(2001\)](#).

Existe interesse comercial em aplicações de redes sociais com grande número de usuários, por parte das empresas envolvidas nas diversas ramificações dessa questão, sejam empresas de mídia, ou publicitárias, ou de telefonia, ou infraestrutura de conexão, ou indústria de sistemas e de eletrônicos, conforme [Nielsen \(2013b\)](#) e [Boudreau and Arathoon \(2008\)](#).

[Nielsen \(2013a\)](#) publicou uma seleção de estatísticas de diversos países, apresentando uma "visão profunda dos consumidores móveis e de como usam seus aparelhos ao redor do globo". Com o crescimento acelerado da quantidade de dispositivos móveis, várias empresas estão atuando no desenvolvimento de aplicativos. As empresas fabricantes de dispositivos móveis visam, cada vez mais, satisfazer o cliente e garantir sua fidelidade, criando aplicativos que facilitem sua vida e, de certa forma, provoquem alguma dependência deles. Eles têm como finalidade ajudar o usuário a realizar uma certa tarefa, que pode abranger várias situações cotidianas.

Google, *Microsoft* e *Apple* são exemplos de empresas fabricantes de aplicativos. Os aplicativos criados são disponibilizados em lojas virtuais e podem ser vendidos ou disponibilizados gratuitamente para usuários. Eles podem ser de diversas categorias, como: educação, redes sociais, jogos, paquera, escritório, localização, dentre outros.

O *Waze* é um exemplo de aplicativo de localização. Nele, com a utilização do GPS (*Global Positioning System*, sistema de posicionamento global), o usuário consegue descrever como está a situação do trânsito em um determinado local e hora do dia. O *Facebook*, o *Twitter* e o *Instagram* são exemplos de aplicativos que permitem a interação em redes sociais, com o objetivo de troca de informações entre as pessoas.

2.2 Redes de Computadores Móveis

Esta Seção traz um resumo teórico de redes de computadores, especificamente de redes móveis, para descrever os pontos pertinentes da tecnologia *Wi-Fi Direct*. Alcançando, assim, o referencial necessário à aplicação dessas tecnologias na implementação de um *framework* útil para a implantação de uma rede social móvel.

Uma LAN (*local area network*) é uma rede que interliga computadores em uma área limitada, como uma casa, uma escola, um laboratório de informática, ou um edifício de escritório usando mídia de rede (Tanenbaum and Weatherall, 2011). Uma LAN oferece alta taxa de transferência em área menor, onde não há necessidade de um serviço de internet. Embora ofereça muitas vantagens, há defeitos também. O mais óbvio é certamente sua dependência de conexão física, ou seja, cabo de rede. Para resolver esta questão, a tecnologia *wireless local area network* (WLAN) foi introduzida na década de 1990.

Uma WLAN interliga dois ou mais dispositivos por um método sem fio através de protocolos. Ela, geralmente, oferece conexão com a Internet e permite que os dispositivos se conectem a um ponto de acesso. Os tipos de WLAN podem ser categorizadas em três segmentos: *Bridge* (ponte), sistema de distribuição sem fio e *peer-to-peer* (ponto-a-ponto).

O termo *peer-to-peer* (ponto-a-ponto), também chamado P2P, refere-se ao conceito que em uma rede de pares usando sistemas de informação e comunicação apropriados, dois ou mais indivíduos sejam capazes de colaborar espontaneamente sem necessidade de uma coordenação central (Elmasri and Navathe, 2010). Em comparação às redes cliente-servidor, as redes P2P prometem escalabilidade melhorada, custo de posse reduzido, coordenação auto-organizada e descentralizada de recursos limitados ou previamente inutilizados, maior tolerância a falhas, e melhor suporte à montagem de redes *ad-hoc* (Schoder et al., 2005). Na Figura 2.3 visualiza-se uma rede ponto-a-ponto básica, onde cada computador está conectado a outro.

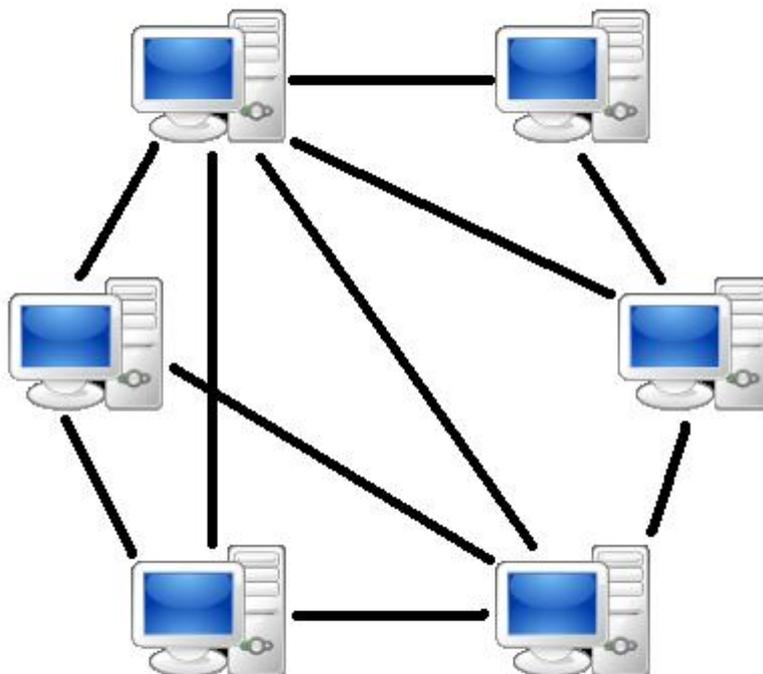


Figura 2.3: Uma rede ponto-a-ponto.

Tanenbaum and Weatherall (2011) mostra que, em relação ao *software*, as redes são compostas por hierarquia de protocolos, e estes em camadas, para diminuir a complexidade. Esse conjunto de camadas e protocolos é chamado de arquitetura de rede. Cada camada oferece serviços para as camadas superiores, sem detalhar o processo, e o que define estes serviços é a *interface*. Elas se comunicam com camadas iguais de uma outra máquina, e são chamadas de pares. Cada camada envia os dados e informações de controle para a camada inferior, até a mais baixa, e então, ocorre a transmissão pelo meio físico. No receptor isso ocorre de maneira inversa. Cada uma pode ser implementada de um jeito, mas os serviços devem ser iguais.

As próximas seções concentram-se em descrever as abordagens existentes e explorar as tecnologias de nova geração, especialmente em *smartphones* e demais dispositivos móveis. Também deve ser notado que o restante desta seção se baseia, principalmente, no sistema de *peer-to-peer Ad hoc*.

2.2.1 Redes de Computadores Móveis

As redes de computação móveis permitem que o usuário se comunique à distância, e utilize a rede em qualquer lugar que ele esteja, sem a necessidade de uma infraestrutura cabeada estabelecida em seu dispositivo.

A arquitetura genérica de uma plataforma móvel está ilustrada na Figura 2.4. É uma arquitetura distribuída onde um conjunto de computadores, os *hosts* fixos e estações base, é interconectado através de uma rede de alta velocidade, por exemplo. As estações base são configuradas para gerenciar unidades móveis, funcionando como *gateways* entre a rede fixa e as unidades móveis. Elas são equipadas com interfaces sem-fio e oferecem acesso à rede para as unidades móveis clientes.

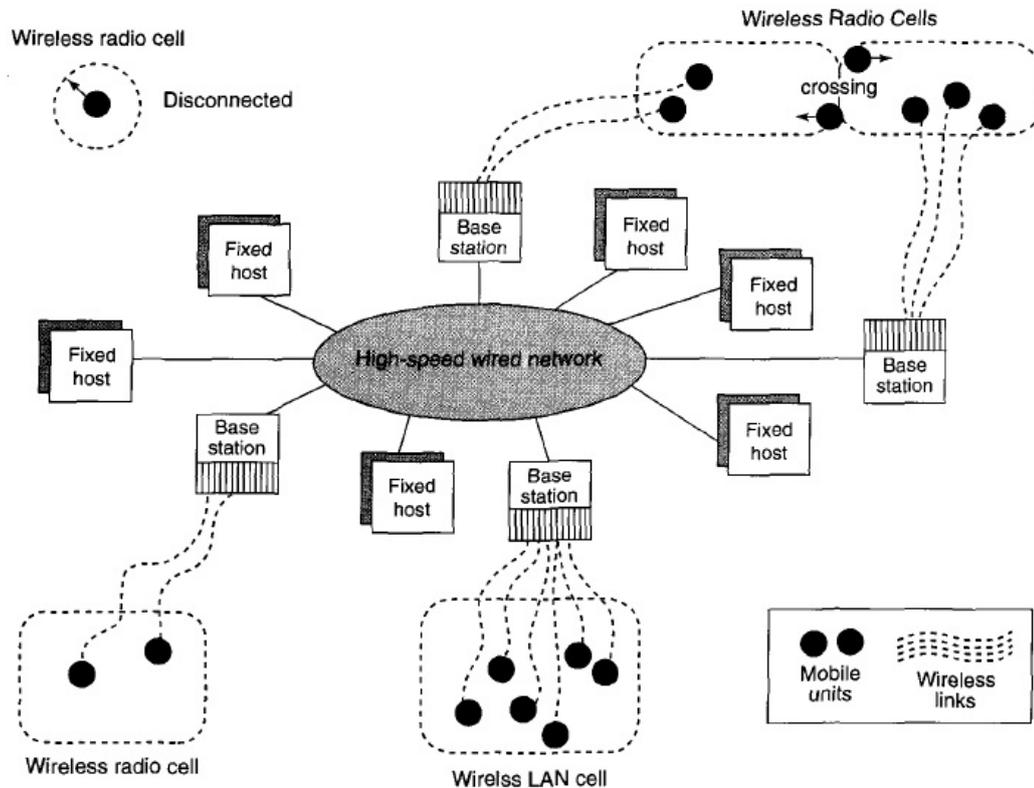


Figura 2.4: Uma arquitetura genérica de uma plataforma móvel, (Elmasri and Navathe, 2004).

Ainda na Figura 2.4, estão representados os conceitos:

- Comunicação sem-fio: O canal sem-fio pelo qual as unidades móveis se comunicam com as estações base têm largura de banda significativamente menor que a de uma rede cabeada. Apesar das taxas de dados, outras características também diferenciam as opções de conectividade sem-fio. Algumas dessas características incluem alcance, interferência, localização de acesso e suporte à troca de pacotes. As redes sem-fio modernas podem transferir dados em unidades chamadas pacotes, que são frequentemente usadas em redes cabeadas. Aplicações sem-fio precisam considerar essas características quando da escolha da opção de comunicação.

- Relacionamento cliente/rede: As unidades móveis podem se movimentar em um domínio de mobilidade geográfica, uma área que é circunscrita pela cobertura da rede sem-fio. Para gerenciar a mobilidade das unidades, o domínio de mobilidade geográfica é dividido em um ou mais domínios menores, chamados células (*cells*), cada qual é suportado por, pelo menos, uma estação base. A movimentação das unidades móveis deve ser irrestrito através das células de um domínio de mobilidade geográfica, ainda mantendo acessibilidade contígua à informação. Na Figura 2.4, essa movimentação de unidades está ilustrada.

Os tipos de redes de computadores móveis (ou sem fio) são:

- **Ponto de acesso**
- *Ad-hoc*

2.2.2 Redes de Computadores Móveis *Ad-hoc*

Além do conceito de rede de computação móvel, há campo para outro conceito que abrange a possibilidade de todos os dispositivos móveis de uma rede se comunicarem diretamente entre si. Essa é a chamada *Wireless ad-hoc network* ou também *Mobile ad-hoc network* (MANET). [Elmasri and Navathe \(2004\)](#) apresenta uma MANET como um tipo descentralizado de rede sem fio (sem envolvimento de ponto de acesso central).

Uma MANET é outra arquitetura possível de comunicação sem fio, que pode ser observada na Figura 2.5. Em uma MANET, unidades móveis co-localizadas não precisam se comunicar através de uma rede fixa, mas ao invés disso, através de suas próprias tecnologias, como a *Wi-Fi Direct* ou a *Bluetooth*. Em uma MANET, as unidades móveis são responsáveis por rotear os próprios dados, atuando tanto como estações base quanto como clientes. Devido a isso, precisam ser robustas o suficiente para manipular mudanças na topologia da rede, como a chegada ou partida de outra unidade móvel ([Elmasri and Navathe, 2004](#)).

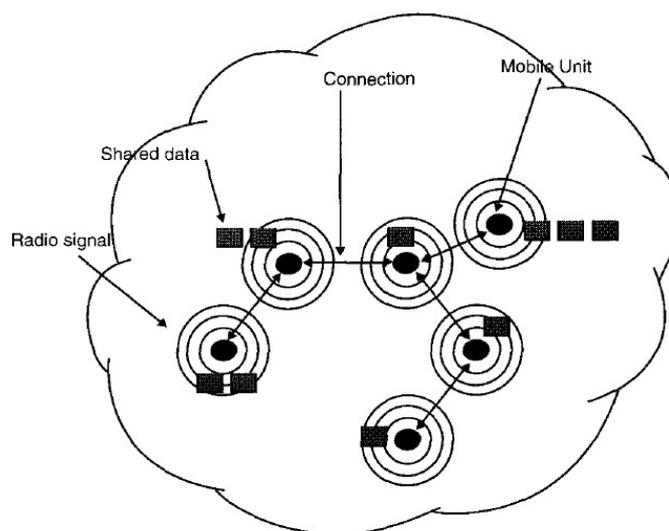


Figura 2.5: A arquitetura de uma rede *ad-hoc* móvel (MANET), extraído de [Elmasri and Navathe \(2004\)](#).

A Figura 2.6 mostra um exemplo de uma rede móvel *Ad-hoc* sem fio. Pode ser observada a rede de dispositivos com *Wi-Fi* habilitado interligados, mesmo na ausência de uma estação base, para o envio e recebimento de dados, compartilhando serviços.



Figura 2.6: Rede móvel *Ad-hoc* (de Rezende, 2004).

Três características importantes vistas nesse tipo de rede são (de Rezende, 2004):

- Topologia dinâmica: Permite que os nós formados se locomovam livremente, com mudanças frequentes, sem um controle centralizado e sem estrutura. Comunicação direta entre dispositivos.
- Largura limitada de banda: A capacidade dessa rede é menor que uma rede fixa cabeada.
- Segurança física limitada: Redes sem fio têm uma maior vulnerabilidade em relação a redes fixas, podendo ficar mais expostas a invasão.

Aplicações MANET podem ser consideradas ponto-a-ponto, significando que uma unidade móvel é, simultaneamente, um cliente e um servidor. Na Seção 2.2.3, será apresentado um estudo da tecnologia *Wi-Fi Direct*, que tem o objetivo de permitir que dispositivos com *Wi-Fi* habilitado se conectem, mesmo na ausência de uma estação base, para o envio e recebimento de dados.

2.2.3 A tecnologia de Rede *Wi-Fi Direct*

No decorrer desta seção estão os argumentos que fomentaram a decisão de desenvolver o projeto com a tecnologia *Wi-Fi Direct* como solução para a conectividade entre os nós da rede móvel *Ad hoc*. Essa tecnologia foi desenvolvida pela Alliance (2010) com o objetivo de permitir que dispositivos façam conexões diretas entre si, de forma rápida e conveniente, objetivando realizar ações como imprimir, sincronizar e compartilhar conteúdo, mesmo quando um ponto de acesso ou um roteador não estiver disponível (Alliance, 2010). Com *Wi-Fi Direct*, as estações de rede podem se comunicar ponto-a-ponto (P2P). Não ter

a necessidade de um ponto de acesso fixo significa simplicidade e funcionalidade com independência, inclusive, de provedor de conteúdo ou de telefonia. Além disso, por ser um produto licenciado, está disponível nos aparelhos sem exigir do usuário comum, e que não tenha habilidades de programação, que ele realize alterações como *rooting* ou *jailbreaking*, que são métodos que possibilitam que o usuário se torne superusuário do dispositivo, com privilégios extras para mudar ou manipular configurações no núcleo do sistema operacional.

Segundo [Alliance \(2010\)](#), dispositivos em grupos *Wi-Fi* P2P podem atuar tanto como um ponto de acesso, quanto como um cliente. O dispositivo que estiver atuando como um ponto de acesso é determinado como proprietário do grupo (*group owner*). Há duas abordagens a fim de determinar o proprietário do grupo. A primeira abordagem é uma escolha manual pelo usuário. A segunda abordagem é mais autêntica, na qual é preciso haver uma negociação entre os dispositivos. A negociação é tratada por um valor simples de intenção (*intent*). Esse valor de intenção depende de várias condições, como a situação de energia (bateria), força de sinal recebido, ou estado do dispositivo e se ele já é proprietário do grupo ou não. O dispositivo que estiver com maior valor de intenção é aceito como proprietário do grupo, enquanto os demais se tornam clientes.

Dispositivos *Wi-Fi Direct* suportam os mesmos perfis de performance de dispositivos *Wi-Fi* regulares. [Alliance \(2010\)](#) afirma que eles operam taxas de dados de cerca de 25 Mbps. Para dispositivos baseados no padrão *802.11 a* ou *802.11 g* (WLAN), as taxas de dados podem ser de 54 Mbps, com alcance de cobertura de cerca de 100 metros. No caso de dispositivos padrão *802.11 n* (WLAN), a velocidade pode alcançar 250 Mbps. A especificação *Wi-Fi Direct* foi desenvolvida pela *Wi-Fi Alliance*, para atuar em dispositivos padrão 802.11, tanto na faixa de 2,4 GHz, quanto em 5 GHz. Essa especificação possui alcance de transmissão de, aproximadamente, 200 metros (sob condições ótimas) e visa introduzir a capacidade de conexão ponto-a-ponto a dispositivos que já possuem *Wi-Fi* implantado, de acordo com [Alliance \(2010\)](#).

Em relação à velocidade e ao alcance descritos no artigo de [Alliance \(2010\)](#), tem-se que a maioria das aplicações e dos usuários atingem valores reais que são uma fração desses valores máximos. Extendendo a versatilidade, na qual o método de rede pode ser um-para-um ou um-para-muitos, o protocolo também permite que o usuário conecte-se a outra rede *Wi-Fi* comum, enquanto estiver conectado a uma rede *Wi-Fi Direct*. Finalmente, o *Wi-Fi Direct* fornece ao dispositivo a capacidade de descobrir outros dispositivos e informações limitadas sobre serviços desses dispositivos antes da associação (e antes de obter o endereço IP). Essa descoberta, ou pré-associação, aperfeiçoa a experiência do usuário que pode saber se um serviço desejado (por exemplo, impressão) está disponível em uma rede *Wi-Fi Direct*, antes de decidir conectar-se ([Alliance, 2010](#)).

Finalmente, em comparação à tecnologia *Bluetooth*, dispositivos *Wi-Fi Direct* podem se conectar com maior alcance e maior capacidade de transmissão de dados ([Alliance, 2010](#)):

- *Bluetooth*: alcance aproximado de 10 metros; taxas de transmissão aproximadas de 1 Mbps (versão 1.0), 3 Mbps (versão 2.0), e de 11 Mbps a 54 Mbps (versões 3.0 e 4.0). Em aplicações das versões 3.0 e 4.0 pratica-se a taxa média de 24 Mbps, segundo [Rinco \(2008\)](#).

- *Wi-Fi Direct*: alcance aproximado entre 50 e 200 metros; taxa de transmissão média aproximada entre 25 Mbps e 54 Mbps, com máxima de 250 Mbps.

De acordo com [Alliance \(2010\)](#), alguns dos benefícios dos dispositivos *Wi-Fi Direct* são:

- Mobilidade e portabilidade;
- Facilidade de uso;
- Conexões seguras por encriptação WPA2 (*Wi-Fi Protected Access*).

Os dispositivos *Wi-Fi Direct* podem se conectar formando grupos em uma topologia de um-para-um ou um-para-muitos. [Alliance \(2010\)](#) apresenta que, para interconectar os dispositivos e para a formação de grupos, estes são os mecanismos chave:

- Descoberta do dispositivo (*discovery*): Identifica outros dispositivos ou grupos *Wi-Fi Direct* e possibilita a seleção do usuário;
- Descoberta do serviço (*service discovery*): Permite a publicação de serviços suportados por aplicações;
- Formação do grupo: Um grupo pode ser criado por um único dispositivo *Wi-Fi Direct*. Quando uma conexão entre dois dispositivos é formada, um grupo pode ser formado automaticamente, neste caso, os dispositivos vão negociar para determinar qual dispositivo gerenciará o grupo. Depois que um grupo é formado, um dispositivo pode convidar outro dispositivo para se juntar ao grupo (descrição da Figura 2.7). Os dispositivos convidados têm a decisão de aceitar ou não o convite;
- Gerenciamento de energia: gerenciamento de energia em relação às interações entre os dispositivos.

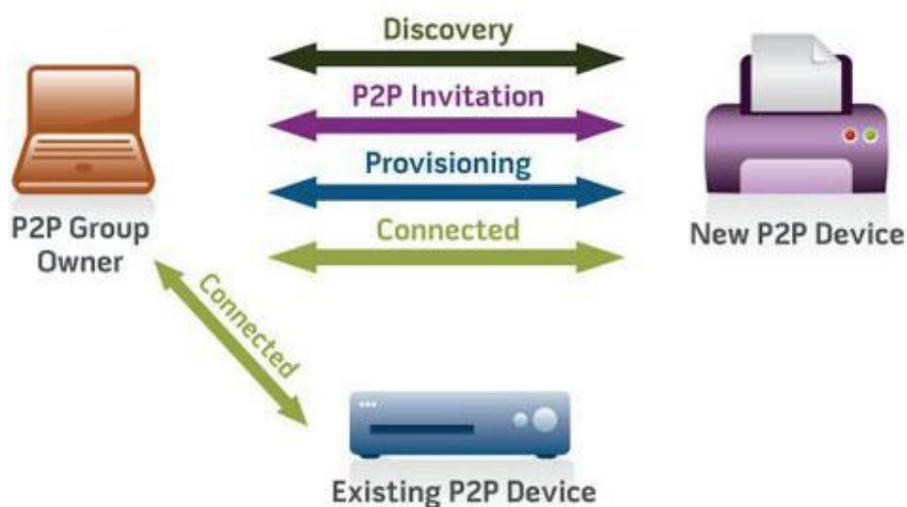


Figura 2.7: O procedimento de convite permite que um dispositivo *Wi-Fi Direct* torne-se cliente de um grupo existente ([Alliance, 2010](#)).

Em relação ao desenvolvimento de *software* na tecnologia *Wi-Fi Direct*, a Seção 2.4.1.1 apresenta uma visão geral da API *Wi-Fi Direct* na plataforma *Android*, que fornece habilidades efetivas para uso dos desenvolvedores de aplicativos.

O protocolo de comunicação *Bluetooth* IEEE 802.15.1 é utilizado para transmissão de dados a uma curta distância, aproximadamente dez metros, fazendo o uso de ondas de rádio. É uma tecnologia que tem um custo baixo e bem recomendada para dispositivos móveis. Segundo Rinco (2008), a ideia do *Bluetooth* começou em 1994 pela empresa Ericson, que estudava um sistema de baixo consumo de energia para substituir cabos a pequena distância. Dispositivos que têm a tecnologia *Bluetooth*, suportam conexão ponto-a-ponto, onde cada um funciona tanto como cliente como servidor (Rinco, 2008), (de Almeida Santos, 2014).

2.3 Bancos de Dados em Dispositivos Móveis

Nesta Seção são apresentados os conceitos de bancos de dados em dispositivos móveis.

2.3.1 Bancos de Dados

Um banco de dados é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico. O dado é um fato que pode ser registrado e tem seu significado implícito, conforme Elmasri and Navathe (2010). Em um sistema computacional, o dado é o que realmente está armazenado, e a informação é um significado contextualizado desse dado. Usa-se uma camada de software entre as aplicações e o banco de dados armazenado, chamado de Sistema Gerenciador de Banco de Dados (SGBD). O SGBD retira da aplicação cliente a responsabilidade de gerenciar o acesso, a manipulação e a organização dos dados, isolando o usuário de detalhes de hardware, segundo Date (2004). De forma resumida, um SGBD é uma coleção de programas que habilita usuários a criar e manter um banco de dados, facilitando o processo de definição, construção e manipulação desses bancos. A definição de um banco de dados envolve o processo de especificação dos tipos de dados a serem gravados no banco de dados, com uma descrição detalhada de cada um desses.

A descrição a seguir fornece uma leitura do mapa conceitual exibido na Figura 2.8. Um banco de dados é um conjunto de dados estruturados em registros. Um dado é um símbolo registrado. Uma informação é o significado contextualizado de um dado. A partir dos conceitos de dado e de informação, tem-se, por transitividade, que um registro representa uma informação. Um banco de dados pode ser controlado com a linguagem de programação SQL (*Structured Query Language*). As instruções básicas de SQL são inserir, pesquisar, atualizar e excluir. Um banco de dados é armazenado em arquivos de computador. Esses arquivos, ou seja, o banco de dados, são controlados através de um *software* SGBD (sistema gerenciador de banco de dados).

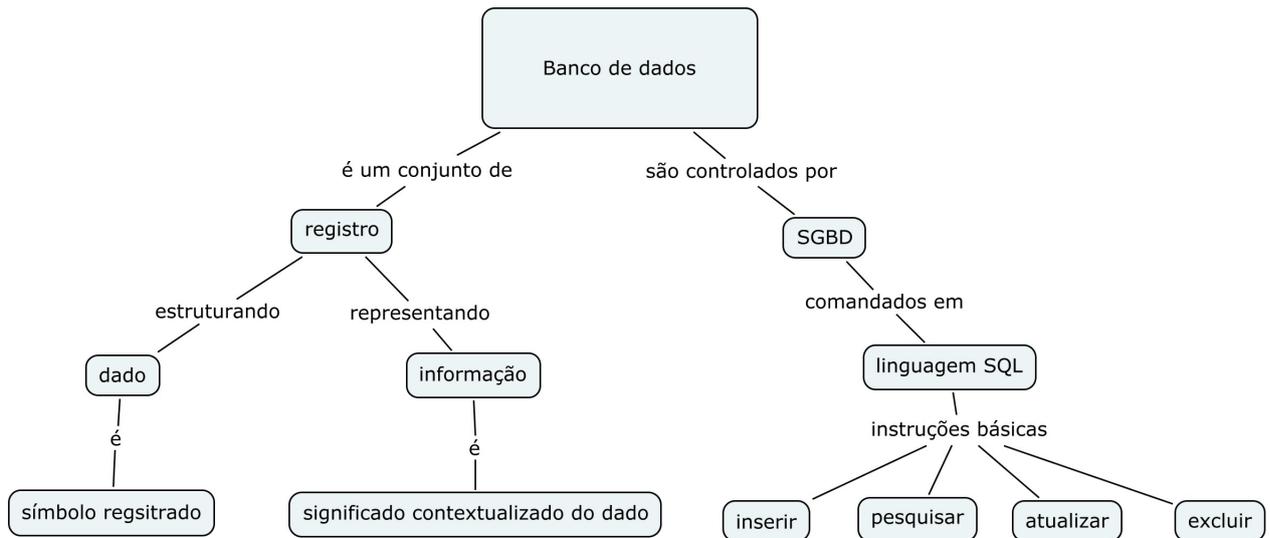


Figura 2.8: Mapa conceitual para bancos de dados

A maior vantagem de um gerenciador de banco de dados é a sua agilidade em localizar e reunir rapidamente informações armazenadas em relações de registros distintas. De acordo com [Molina et al. \(2001\)](#), os recursos oferecidos por um SGBD são:

- Armazenamento persistente: Um SGBD permite o armazenamento de dados para futuras consultas;
- Interface de programação: Um SGBD permite ao usuário acessar e modificar dados por meio de uma linguagem de consulta;
- Gerenciamento de transação: Um SGBD admite o acesso simultâneo a dados por muitos processos distintos e ao mesmo tempo.

A Figura 2.9 ilustra os conceitos da arquitetura de um sistema que utiliza SGBD, conforme [Elmasri and Navathe \(2010\)](#). Os usuários e programadores do sistema efetuam consultas (*queries*) ou executam programas (*procedures*) no SGBD (ou DBMS - *database management system*) com propósito de definir, construir, manipular ou compartilhar bancos de dados entre usuários ou aplicações. Definir um banco de dados envolve especificar os tipos de dados, estruturas e restrições dos dados a ser armazenados. A definição dos bancos de dados ou informação descritiva também é armazenada pelo SGBD, na forma de catálogos ou dicionários, chamados meta-dados (*meta-data*).

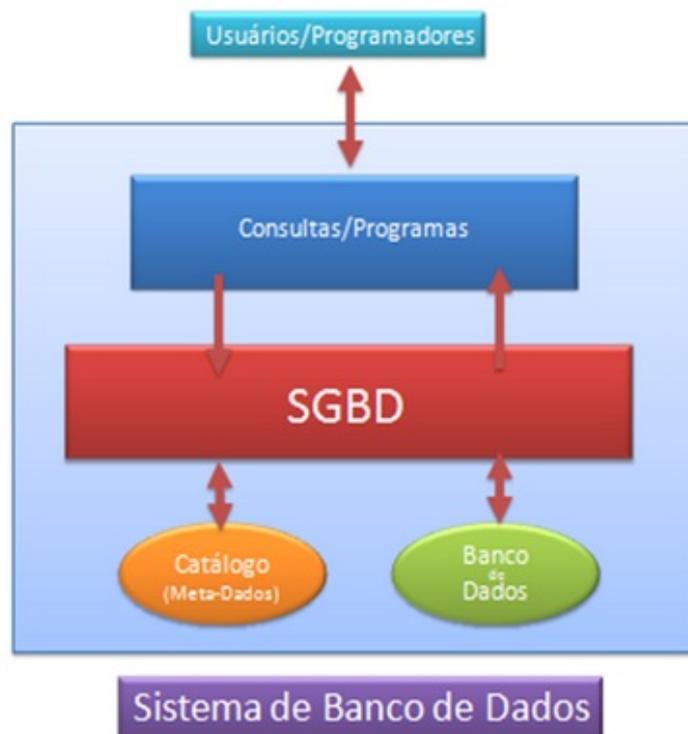


Figura 2.9: Arquitetura de um sistema utilizando SGBD, por [Elmasri and Navathe \(2004\)](#).

Uma transação é o fluxo de operações necessário para realizar escritas e leituras no banco de dados. As transações devem apresentar as propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade). Implementações dessas propriedades são descritas em [Silberschatz et al. \(2006\)](#), [Elmasri and Navathe \(2010\)](#):

- Atomicidade: Uma transação deve ser executada totalmente ou não terá efeito;
- Consistência: Uma transação isolada leva o banco de dados de um estado consistente para outro estado consistente;
- Isolamento: O sistema deve garantir que não deve haver interferência entre as transações. Diversas transações podem ser executadas de forma concorrente. Cada transação deve ocorrer de forma isolada;
- Durabilidade: Quando uma transação é efetuada com sucesso, seus efeitos devem ser persistentes, até mesmo se houver falhas no sistema.

Uma linguagem padrão para programação e administração de bancos de dados é a SQL (*structured query language*). É uma linguagem de pesquisa declarativa, base para utilização em bancos de dados relacionais que foi desenvolvida nos anos 70, pela IBM, e é simples e fácil de ser utilizada. Desde 1986, duas entidades vêm publicando padrões de especificação da linguagem: ANSI (*American National Standards Institute*) e ISO (*International Standards Organization*), conforme [Staff \(2012\)](#).

Apesar de existirem normas para essa linguagem, ela pode possuir características diferentes ao se comparar os fabricantes dos SGBD. Isso porque a norma SQL não é simples

de ser implementada em todo sistema. Além de que cada empresa procura diferenciar seu produto de acordo com suas estratégias de mercado ou a aplicabilidade de seus sistemas, por exemplo. Entre alguns SGBD conhecidos, que utilizam a linguagem SQL, estão *SQLite*, *Oracle*, *SQL Server*, *DB2*, *Access*, *OpenOffice BASE*, *MySQL*, *Sybase ASE*, *Sybase IQ* e *IDMS*. Dependendo do SGBD, dados como vídeo, áudio e imagem podem ser estruturados em bancos de dados. Isso é feito utilizando objetos tipo BLOB (*binary large object*, objeto binário grande).

2.3.2 Bancos de Dados Móveis *SQLite*

Os dispositivos móveis tratam o gerenciamento de dados e o controle de transações com bancos de dados que levam em consideração as questões de conectividade limitada e intermitente, de limite de energia, e de frequência de movimentação física dos elementos de rede (Elmasri and Navathe, 2004).

A maioria dos aplicativos para dispositivos móveis utilizam persistência de dados. Isso é necessário para que os dados continuem armazenados mesmo depois que o dispositivo é desligado. O SGBD usado para implementar o banco de dados no sistema operacional *Android* é o *SQLite*. Formalmente, o *SQLite* é um módulo do *Android*. Como apresentado por Vogel (2013), o *SQLite* gerencia o banco de dados usando apenas 250 *kilobytes* da memória do dispositivo. Os tipos de dados implementados podem ser *NULL*, *TEXT*, *INTEGER* ou *REAL*. Quando necessitar de outro tipo de dados, como *datetime*, por exemplo, o desenvolvedor deverá implementar a conversão de dados na modelagem do banco de dados.

O *SQLite* é um sistema de banco de dados auto-contido, compacto, e sua versão 3 tem suporte nativo no *Android* e sem necessidade de configuração ou instalação. Isto torna-o a escolha natural para um ambiente em que deve-se prezar por desempenho, disponibilidade de memória e praticidade de uso. Citando Vogel (2013):

O *SQLite* é uma ferramenta poderosa e eficiente. É uma biblioteca compacta de software que é implementada sem um servidor, de forma auto suficiente. Seu código-fonte pode ser visualizado por quem quiser e pode ser utilizado para fins comerciais e pessoais. Ele geralmente tem um desempenho muito bom e pode ser utilizado em ambientes com pouca memória. Pode ser considerado confiável por ser muito testado antes do lançamento de cada versão e por ter estrutura simples, tem menos complicação, e apresenta menos chance de erros.

O *SQLite* é muito usado em aplicações populares, por exemplo, o *smartphone iPhone* e o navegador *Mozilla Firefox*. Algumas características importantes do *SQLite* serão descritas a seguir. Ele é simples de administrar, operar, manter e personalizar. Uma equipe internacional de desenvolvedores trabalham para expandir a capacidade do *SQLite* e aumentar seu desempenho e confiabilidade (SQLite, 2013). Conteúdos diversos que seriam armazenados como pilhas, podem ser encapsulados em um único arquivo de disco. O aplicativo apenas precisa carregar dados que serão utilizados ao invés do arquivo completo, tornando o acesso mais rápido. Ele é simples de configurar e utilizar, bastando copiar o executável para o computador destino e executar.

O *SQLite* não é recomendado para algumas situações. Quando existem várias pessoas acessando um banco comum em uma rede, simultaneamente, o desempenho não será

muito bom, e poderá ocorrer uma série de *bugs*. O banco é limitado a um tamanho de 140 terabytes, e armazena em um disco único. Se for necessário usar um banco deste tamanho, deve ser considerado outro SGBD, que espalhe o conteúdo armazenado em mais de um disco. O *SQLite* suporta vários leitores simultaneamente, mas permite que somente um usuário, por vez, escreva. Mesmo que para escrever seja muito rápido, isso não é garantido em todas as ocorrências.

A Figura 2.10 mostra um exemplo de um projeto em desenvolvimento que utiliza o *SQLite* para persistência de dados.

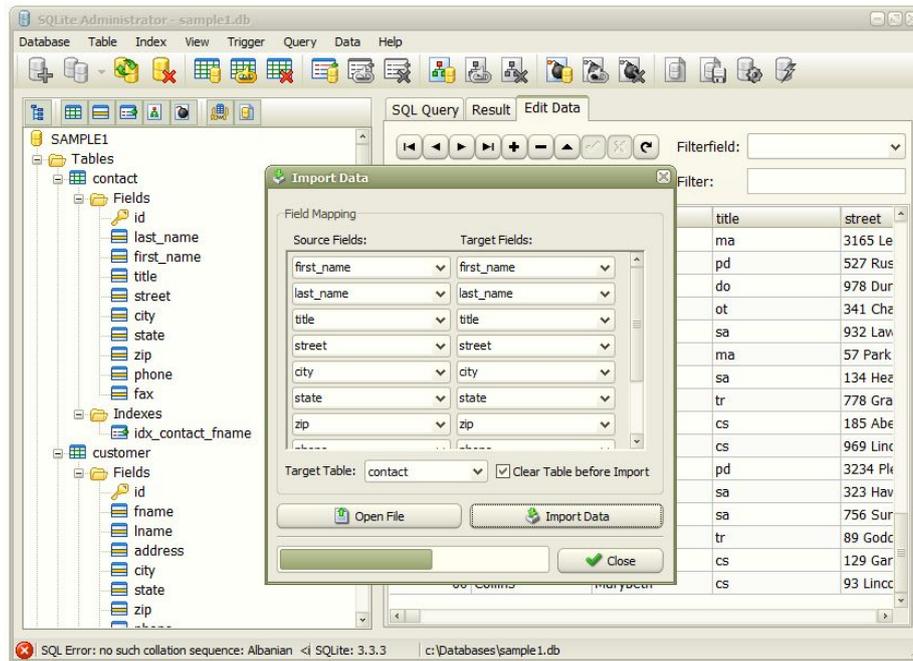


Figura 2.10: Projeto de dados no *SQLite*.

O *SQLite* é uma boa indicação para celulares, PDAs, dentre outros dispositivos, por requerer pouca administração para dispositivos que devem trabalhar de forma autônoma e sem apoio humano.

2.3.2.1 Diferenciais do *SQLite*

Algumas características diferem o *SQLite* dos demais sistemas gerenciadores de bancos de dados utilizados em aplicações. Um dos seus diferenciais é a não necessidade de ser instalado antes de ser utilizado. Sendo assim, não existe um *setup* ou um processo que necessite ser configurado. O administrador não precisa criar uma nova instância de banco de dados ou conceder permissões aos usuários. Com o *SQLite*, o processo que deseja acessar o banco de dados lê e escreve direto em arquivos de banco de dados no disco. A vantagem disso é que não há necessidade de instalar ou configurar algum tipo de servidor, e um programa que o utiliza não precisa de apoio administrativo, basta acessar o disco.

Em outros mecanismos de banco de dados, o armazenamento é feito como uma grande coleção de arquivos. Estes arquivos podem estar em um local padrão que somente o SGBD pode acessar, deixando os dados mais seguros, porém mais difíceis de acessar. No

SQLite, isso funciona de forma diferente, o banco de dados é um arquivo único comum que pode ser localizado em qualquer lugar do diretório. Esse arquivo pode ser criado em um computador e migrado para outro. O aplicativo apenas precisa carregar dados que serão utilizados ao invés do arquivo completo, tornando o acesso mais rápido. Essa abordagem facilita o uso, bastando copiar o executável para o computador destino e executar.

A biblioteca que o *SQLite* tem o tamanho inferior a 400 KB, e caso sejam desabilitados alguns recursos que estejam desnecessários, pode reduzir a 190 KB. No *SQLite*, o tipo de dados é uma propriedade do valor registrado, ou seja, é permitindo armazenar valores de qualquer tipo de dados em uma coluna, independente do tipo declarado nessa coluna. Isso apenas não funciona para a coluna que tem chave primária do tipo *integer*. Esse modelo é um diferencial dos demais SGBD, onde cada coluna tem um tipo de dados definido. Essa preferência na digitação de campos é uma importante funcionalidade que demonstra a versatilidade do *SQLite*. Por exemplo, a inserção de um registro, com valor textual, em um campo que foi previamente declarado do tipo *real* é efetivada no *SQLite* e não significa erro.

Outra característica interessante é que o *SQLite* não precisa alocar espaço em disco para determinar o tamanho de cada linha. Ele só utiliza o espaço necessário para armazenar o dado inserido, e o primeiro *byte* grava o tipo do dado e seu comprimento. Essa estratégia permite que os dados utilizem um espaço menor para o armazenamento e, assim, faz com que a execução seja mais rápida, pois há menos informação para buscar. Este SGBD utiliza uma máquina virtual que permite aos programadores ver o código e o que ele está tentando fazer, ajudando a depuração.

O código fonte do *SQLite* é de domínio público, não contém nenhum tipo de licença para ver ou copiar, e qualquer pessoa pode fazer legalmente o que quiser com o código fonte.

2.4 Sistema operacional *Android*

Uma necessidade específica deste estudo é criar um sistema simples e funcional sem necessidade do usuário fazer nenhuma manipulação avançada no dispositivo móvel. Desse ponto de vista, o *Android OS* foi escolhido, pois oferece APIs¹ adequadas, além de considerada popularidade e do fato de ser usado na maioria dos *smartphones* no mercado, conforme Nielsen (2013a). De acordo com um levantamento de Nielsen (2012b), "O sistema operacional *Android* atingiu a marca dos 61% de participação de mercado no quarto trimestre de 2011, o que representa três em cada cinco novos *smartphones*".

O sistema operacional *Android* foi originalmente projetado para câmeras, onde seria uma plataforma para armazenamento *on line* de fotos. Essa proposta mudou a partir do momento em que o mercado de *smartphones* começou a expandir. É um sistema operacional desenvolvido pela *Open Handset Alliance*, uma aliança entre várias empresas liderada pela *Google*. Ele gerencia os processos dos aplicativos e do hardware do dispositivo e é utilizado em *smartphones*, TVs e *tablets*. Desenvolvido sobre o *kernel Linux* versão 2.6, ele é um projeto de código aberto e possui um vasto número de desenvolvedores escrevendo aplicativos, que podem ser acessados, principalmente, na loja *online* oficial chamada *Google Play*. Atualmente, há mais de 600 mil aplicativos disponíveis na loja, que

¹ *Application Programming Interface* ou Interface de Programação de Aplicativos

já contabiliza mais de 20 bilhões de *downloads* (Android, 2013a). Apesar das aplicações *Android* serem escritas em linguagem *Java*, não se usa a *Java Virtual Machine* (JVM), mas sim a *Dalvik Virtual Machine*, que é aberta para desenvolvimento, de acordo com o Android (2013a). Esta máquina virtual é desenvolvida para uso em dispositivos móveis e, permite que programas sejam distribuídos em formato binário. Dessa forma, podem ser executados em qualquer dispositivo *Android*, independente do processador.

Uma das principais vantagens do sistema *Android* é a variedade de aplicativos encontrados em sua loja virtual. Ele recebe também uma coleção de aplicativos criados pela *Google*, como *Gmail*, *Google Maps*, *YouTube*, *Google Drive*, dentre outros. Ele permite a integração destes serviços da *Google*, a partir de uma conta que o usuário já possua (Android, 2013a). Algumas marcas fabricantes de dispositivos também fornecem a seus clientes usuários aplicativos personalizados, como por exemplo o *Optimus UI* da LG, *TouchWiz* da *Samsung*.

O *Android* é compatível com dispositivos VGA maiores, gráficos 2D, bibliotecas gráficas 3D, baseadas em *OpenGL ES*. Ele utiliza sistema gerenciador de banco de dados *SQLite*, que será apresentado na Seção 2.3.2. O ambiente de desenvolvimento inclui um emulador, memória e análise de performance. A interface de desenvolvimento (IDE) *Eclipse* pode ser utilizada, desde que o *plugin Android Development Tools* (ADT) seja adicionado (Android, 2013b).

O primeiro dispositivo a utilizar o *Android* como sistema operacional foi o *T-Mobile (HTC Dream)*, lançado em 22 de outubro de 2008. Desde então, vários celulares contam com o sistema, por exemplo o *Motorola Milestone*, o *Samsung Galaxy S* e o *LG Optimus One*, todos disponíveis para os usuários do Brasil. Aqui, o primeiro aparelho a executar o sistema da *Google* foi o *HTC Magic*, em setembro de 2009 (Android, 2013a).

Nas Figuras 2.11 e 2.12, pode-se ver um emulador de *Android* da plataforma *Eclipse*. As imagens foram retiradas da parte prática deste trabalho, e mostra a simulação de uma aplicação básica.

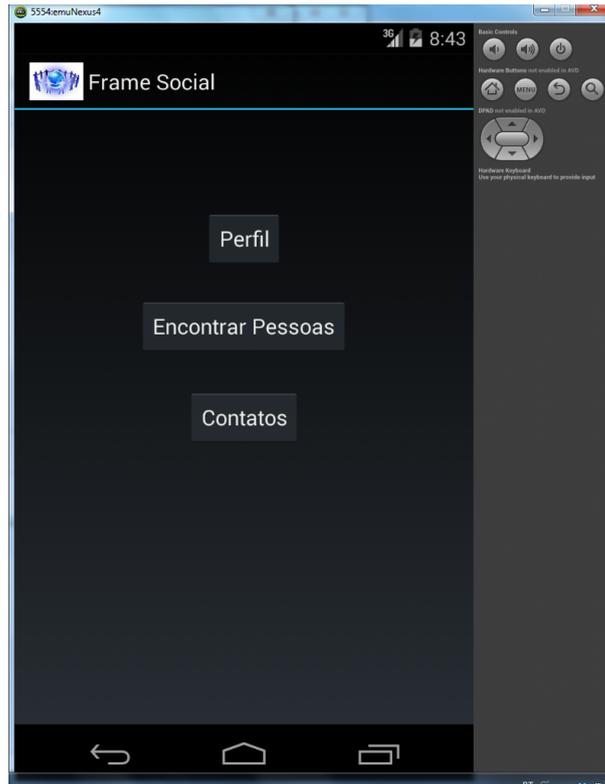


Figura 2.11: A tela inicial do *Frame Social* executado em emulador *Android*.

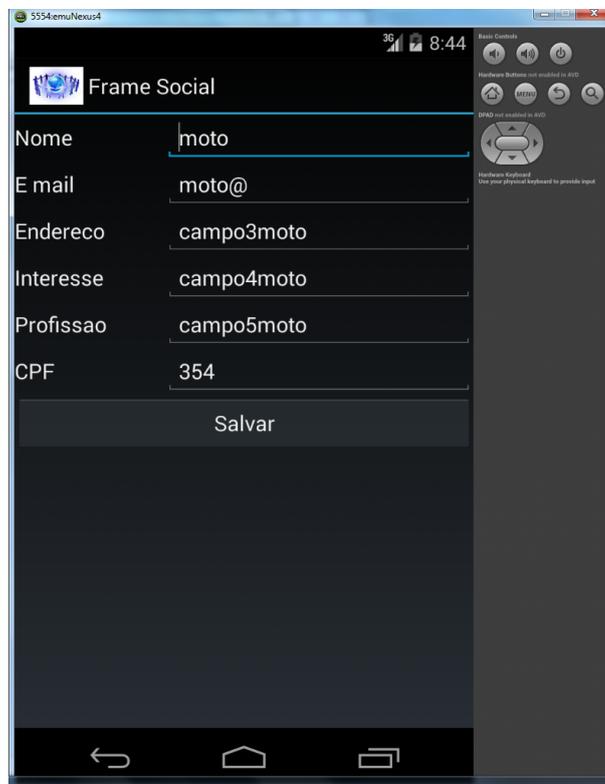


Figura 2.12: Edição de perfil do usuário do *Frame Social* executada em emulador *Android*.

2.4.1 Arquitetura do *Android OS*

O *Android* executa em *Linux* com serviços e bibliotecas escritas na linguagem *C*. Ele usa a *Dalvik Virtual Machine* para executar o *Dalvik Executable Code* traduzido do *bytecode Java*. Todas as APIs padrões são definidas em classes, interfaces, métodos e objetos. Em termos de plataforma de hardware, a arquitetura ARM é a principal. Contudo, também há suporte para a arquitetura x86 (Android, 2013b). *Advanced RISC Machine* (ou máquina RISC avançada) é uma arquitetura de processadores com sistema embutido que possui como forte característica o baixo consumo de energia. Os processadores de arquitetura x86 usam o modelo CISC. Essa arquitetura é capaz de executar instruções complexas em vez de várias instruções simples como é utilizado na arquitetura RISC e é implementada diretamente no processador em linguagem de máquina. A diferença entre essas arquiteturas é basicamente uma relação entre processamento e consumo de energia. Um processador ARM utilizando o modelo RISC é bastante simples se comparado ao x86, o que por um lado faz com que o seu desempenho seja menor, mas faz também com que ele precise de uma quantidade ínfima de energia para operar, o que o torna ideal para *smartphones* e *tablets*, dispositivos que trazem uma bateria bastante reduzida se comparado a um *laptop* moderno.

Antes de ser interpretado pela Máquina Virtual Java (JVM), o código de um programa escrito na linguagem *Java* é compilado para uma forma intermediária de código chamada *bytecode* (Haggar, 2000). Essa é a característica que faz com que os programas *Java* tenham alto grau de independência de plataforma, funcionando em qualquer sistema que possua uma JVM.

A *Dalvik* tem um compilador em tempo de execução (JIT, ou *Just-In-Time*) onde o *bytecode* armazenado na memória é compilado para linguagem de máquina. O *bytecode* pode ser considerado como o "nível intermediário". O compilador JIT lê o *bytecode* em muitas seções e compila dinamicamente, a fim de executar o programa mais rápido. O *Java* desempenha checagens em porções diferentes do código, e assim o trecho de código é compilado justamente antes de ser executado. Uma vez que ele é compilado, é posto em *cache* e estabelecido para estar pronto para uso posterior.

Essencialmente, o sistema *Android* tem quatro camadas, de acordo com o Android (2013b):

- A camada de **Aplicativos** é o topo, onde o usuário interage com o dispositivo. Ela é escrita em *Java* e executada em *Dalvik*;
- O próximo nível, **Framework de Aplicação**, consiste em serviços e bibliotecas. Aqui, códigos de aplicativos e *frameworks* também são executados em *Dalvik*;
- A camada de **Bibliotecas** inclui bibliotecas nativas, *daemons* e serviços, os quais são escritos em *C/C++*;
- Por último está o *Linux Kernel*, que inclui *drivers* para muitas funcionalidades, como rede, acesso ao sistema de arquivos, IPC (*Interprocess Communication Portocol* ou Protocolo de Comunicação Interprocessos).

A Figura 2.13 mostra uma visão geral da arquitetura do sistema *Android* nas diferentes camadas descritas acima.

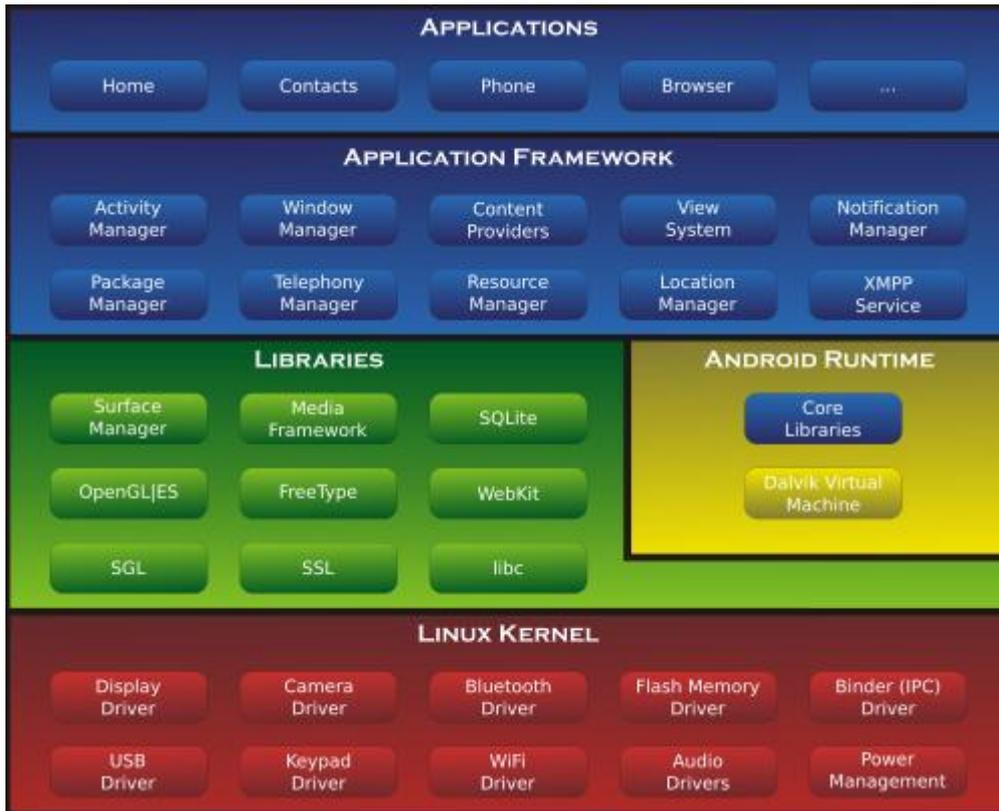


Figura 2.13: Diagrama da arquitetura do *Android OS* (Android, 2013b).

2.4.1.1 Visão geral da API *Wi-Fi Direct* no *Android OS*

Em adição às funcionalidades descritas na Seção 2.2.3, a API *Wi-Fi Direct* fornece habilidades de desenvolvimento efetivas para os programadores no *Android OS*. Começando na versão 4.0 (ICS - *ice cream sandwich*), o *Google* introduziu sua nova API (nível 14) que permite o desenvolvimento de aplicativos para *Wi-Fi Direct*. Segundo Android (2013b): "Usando essa API em conjunto com outras APIs *Android*, pode-se descobrir pares, conectar a outros dispositivos e se comunicar por uma conexão rápida, através de distâncias maiores que em uma conexão *Bluetooth*". Há vários pacotes para criação de aplicativos *Wi-Fi Direct*, tais como:

- *android.net.wifi.p2p*: Fornece classes para criar conexões ponto-a-ponto (P2P) com *Wi-Fi Direct*;
- *java.net*: Consiste em bibliotecas que sustentam conexões por *socket*;
- *java.io*: Contém pacotes para operações de entrada e saída.

As operações de núcleo são mais manipuladas por uma classe chamada *WifiP2pManager*, sob o pacote *android.net.wifi.p2p*. Essa classe provê conectividade P2P, bem como descoberta, configuração de conexão entre os dispositivos e consulta na lista de pares. As outras classes são: *WifiP2pDevice*; *WifiP2pDeviceList*; *WifiP2pGroup*; *WifiP2pInfo*; and *WifiP2pConfig*.

2.5 Trabalhos Relacionados

O projeto proposto nesse trabalho segue uma sequência de estudos previamente publicados sobre comunicação de dispositivos móveis e bancos de dados móveis. Nesta sessão serão apontados alguns trabalhos recentes relacionados ao projeto do *Frame Social*, com comparações de propostas que evidenciam a inovação presente no projeto.

2.5.1 *Eiko*

O aplicativo *Eiko* (Castro and Holanda, 2011) foi desenvolvido como protótipo de pesquisa para redes sociais móveis *ad hoc*. O *Eiko* é uma rede social móvel para compartilhamento de informações profissionais e currículos em conferências. Utilizaram a linguagem orientada a objetos *Java Micro Edition* (JME), a tecnologia *Bluetooth* para conexão entre dispositivos e o armazenamento de dados foi diretamente em registros, sem a utilização de um SGBD.

Diferentemente da proposta apresentada nesta monografia, o *Eiko* é uma rede social fixa, e não um *framework* para o desenvolvimento de outras redes sociais.

2.5.2 *MobiSN*

MobiSN é um *framework* desenvolvido na Universidade de Brasília por Queiroz et al. (2012). É utilizado para desenvolvimento de redes sociais móveis MANET, fornecendo suporte para o desenvolvedor na criação da interface gráfica com o usuário. Os dispositivos buscam por pares para comunicação, promovendo a interação social. Foi organizado em três camadas, a camada mais alta, a camada intermediária e a camada mais baixa. São elas, respectivamente:

- Interface gráfica de usuário: criação de módulos com interfaces;
- Comunicação: utiliza a tecnologia *Bluetooth* para estabelecer uma conexão entre dispositivos;
- Persistência de dados: armazena os dados em arquivo binário com extensão ".dat" (sem utilizar SGBD), realiza o CRUD utilizando arquivos auxiliares.

As principais vantagens desse *framework* são permitir a criação de redes sociais em diferentes contextos, onde o programador pode definir os campos de acordo com seus interesses, e permitir a comunicação sem um ponto de acesso. Isso é feito por um arquivo texto.

Diferentemente do trabalho apresentado nesta monografia, Queiroz et al. (2012) não utiliza um SGBD para o armazenamento dos dados e a comunicação é feita utilizando a tecnologia *Bluetooth*.

2.5.3 *My-Direct*

O sistema *My-Direct* (de Almeida Santos, 2014) é um *middleware* utilizado para desenvolvimento de redes sociais em P2P, que proporciona a comunicação flexível e privada. Permite a conexão entre dois dispositivos através da tecnologia *Wi-Fi Direct* e *Bluetooth*.

O gerenciamento dessas duas formas de comunicação é feito pelo próprio sistema, fazendo com que uma funcione quando a outra não estiver disponível.

Para cada grupo de parceiros, o usuário pode ter um conjunto de atividades sociais, podendo ser elas: um *chat*, compartilhamento de imagens ou um jogo em rede. Há um mecanismo de privacidade baseado em informações presentes no dispositivo móvel.

Diferentemente do trabalho apresentado nesta monografia, o *My-Direct* não é um *framework* para desenvolvimento de redes sociais.

2.5.4 *MobiSoft*

Steffen Kern (2006) desenvolveu o *MobiSoft*, um *middleware* que permite a criação de uma rede social, onde o usuário pode cadastrar seu perfil e uma lista de tarefas. Visa facilitar a interação humana por meio de dispositivos, que procuram por outros dispositivos nas proximidades e se conectarem através de uma MANET (*Mobile Ad hoc Network*) utilizando o *Bluetooth*. Após conectados, é possível trocar informações cadastradas em um espaço digital. É organizado em três camadas, sendo elas:

- Baixa: Nesta camada está o micro *kernel* e agentes do sistema;
- Intermediária: Nesta camada têm vários *plugins*, utilizados para atender as exigências de funcionalidades contidas no sistema;
- Alta: Contém assistentes sociais móveis, responsável pela formação de grupos com interesses comuns e troca de informações entre usuários conectados.

Suas vantagens são a descentralização, a facilidade de configuração e a não necessidade de se conectar à Internet para troca de informações. Uma desvantagem é não ter controle de privacidade.

Diferentemente da proposta apresentada nesta monografia, o *MobiSoft* se comunica utilizando a tecnologia *Bluetooth*.

2.5.5 *MyNet*

A *MyNet*, desenvolvida por Kalofonos (2006), é uma plataforma utilizada para serviços de redes sociais, permitindo que usuários organizem e compartilhem recursos com outros usuários que estão por perto, em tempo real. A interface de gerenciamento de grupos de dispositivos distribuídos e conectividade oblíqua da *MyNet* é chamada UIA (*Unmanaged Internet Architecture*). Essa plataforma permite que a troca de informação seja feita de forma segura, utilizando um endereço permanente em cada dispositivo, que permite conexões criptografadas e a necessidade de autenticação. O *Personal Device Cluster* (PDC) é a rede pessoal onde um usuário e todos os dispositivos podem autenticar-se para fazer parte do mesmo PDC. Cada dispositivo pode criar grupos e executar um ou mais serviços. A disponibilidade do conteúdo e de serviços da rede pode não ser garantida, pois os dispositivos podem estar desligados.

O *Frame Social* representa uma sequência direta dos estudos propostos do *Eiko* (Castro and Holanda, 2011) e do *MobiSN* (Queiroz et al., 2012), introduzindo a pesquisa de comunicação de dispositivos com a tecnologia *Wi-Fi Direct* e agregando a persistência de dados com o SGBD móvel *SQLite*.

Capítulo 3

Frame Social: um *framework* para redes sociais móveis *Ad hoc*

Este capítulo apresenta a estrutura global da solução proposta ao problema apresentado, partindo da apresentação da arquitetura do *framework Frame Social*, seguida da descrição da implementação do *framework*. Para finalizar, este capítulo compara alguns trabalhos publicados com projetos relacionados a este projeto.

O *Frame Social* é um *framework* para o desenvolvimento de redes sociais que fornece suporte para conexão entre dispositivos para formação de rede móvel *ad hoc* (MANET), além de oferecer persistência de dados com SGBD. O *Frame Social* representa uma sequência direta dos estudos propostos do aplicativo *Eiko* (Castro and Holanda, 2011) e do *framework MobiSN* (Queiroz et al., 2012), agregando a pesquisa de desenvolvimento de *software* com a tecnologia *Wi-Fi Direct* e a persistência de dados com o SGBD móvel *SQLite*.

O *Frame Social* é customizável por programação e está preparado para que o usuário da rede social crie e edite seu perfil, e possa buscar outros usuários e estabelecer conexão. Assim, com as informações de sua rede social armazenadas, o usuário consulta os perfis dos seus contatos. Uma vez conectados, dois usuários podem conversar via *chat* no *Frame Social*.

3.1 Arquitetura do *Frame Social*

A arquitetura do *Frame Social* é apresentada a partir do diagrama na Figura 3.1, e está descrita detalhadamente na Seção 3.3. Os destaques do *framework* são as camadas de **Persistência** e de **Comunicação**, pois implementam os objetos de estudo deste projeto com tecnologias apropriadas:

- A contribuição implementada na camada de **Persistência** está na pesquisa de armazenamento e migração de dados com um SGBD em ambiente móvel;
- A camada de **Comunicação** resolve o problema questionado de transmissão de dados em uma rede MANET (móvel e conectada ponto-a-ponto, sem ponto de acesso central), desenvolvida na tecnologia *Wi-Fi Direct*;

- A camada *Interface* executa a apresentação do aplicativo ao usuário, e mantém a interação entre usuário e sistema operacional. Não é o objeto de estudo deste projeto;
- A camada *DefineRedeSocial.java* é composta pela classe onde o programador define os parâmetros necessários para que o *framework* crie a rede social;
- A camada *Dalvik Virtual Machine* representa a máquina virtual *Java* que executa o sistema operacional *Android*. É a camada responsável por gerenciar a memória, *threads*, pilhas de execução, etc. Esta camada não é o objeto de estudo deste projeto.

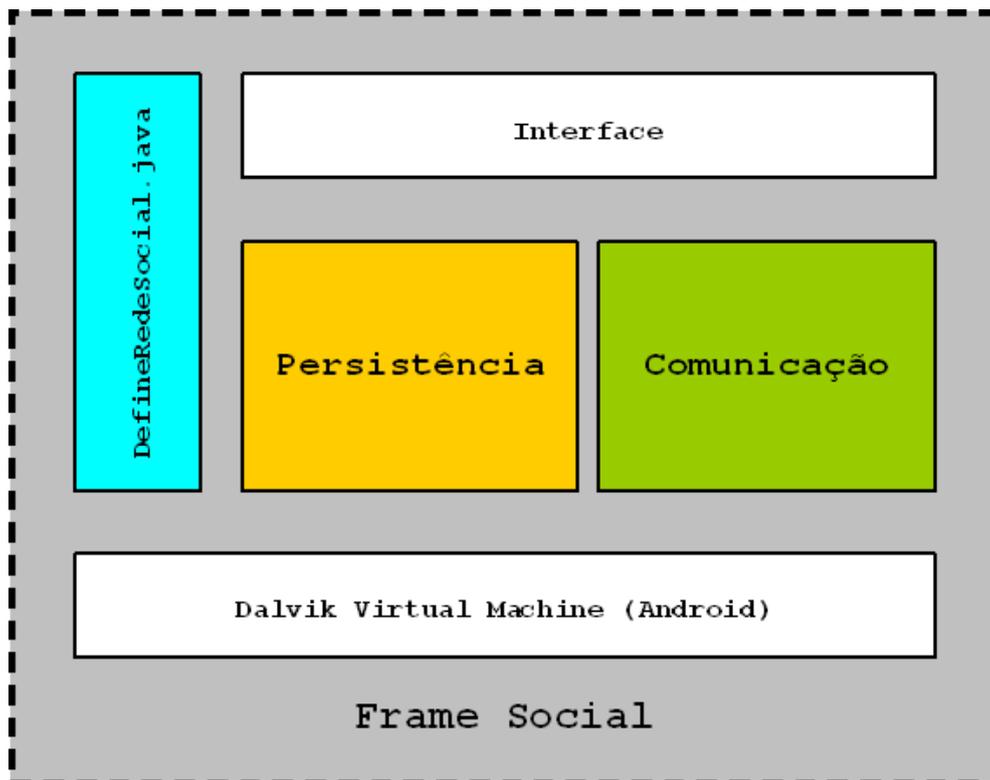


Figura 3.1: Arquitetura abstrata do *framework Frame Social*.

3.2 Fluxo do Processo de Comunicação e Persistência entre Dispositivos

O processo de funcionamento do *framework Frame Social*, conforme as descrições que estão nas próximas seções do capítulo, está ilustrado no fluxograma apresentado na Figura 3.2. A *activity* (atividade) "estabelecer conexão" é detalhada na Figura 3.3. *Activity* é uma tarefa que uma aplicação pode fazer, algo que torna a aplicação apresentável para o usuário. Uma *activity* é um componente do *Android OS*, que permite a classes herdarem suas características para haver interação com o usuário.

As atividades do processo **Conexão e persistência de dados**, Figura 3.2, são:

- Iniciar Aplicativo: O usuário inicia o aplicativo e é direcionado para a tela inicial;

- Cadastrar Perfil: Permite que o primeiro usuário cadastre seus dados pessoais/profissionais no aplicativo;
- Estabelecer Conexão: Subprocesso que permite que usuários se conectem;
- *Iniciar Chat*: O primeiro usuário opta por iniciar uma conversação com o segundo usuário;
- Enviar Perfil: O primeiro usuário opta por enviar seus dados para o segundo usuário;
- Receber Perfil: O segundo usuário recebe o perfil com dados do primeiro usuário;
- Consultar Contatos: Permite que o segundo usuário visualize em seus contatos os dados do primeiro usuário.

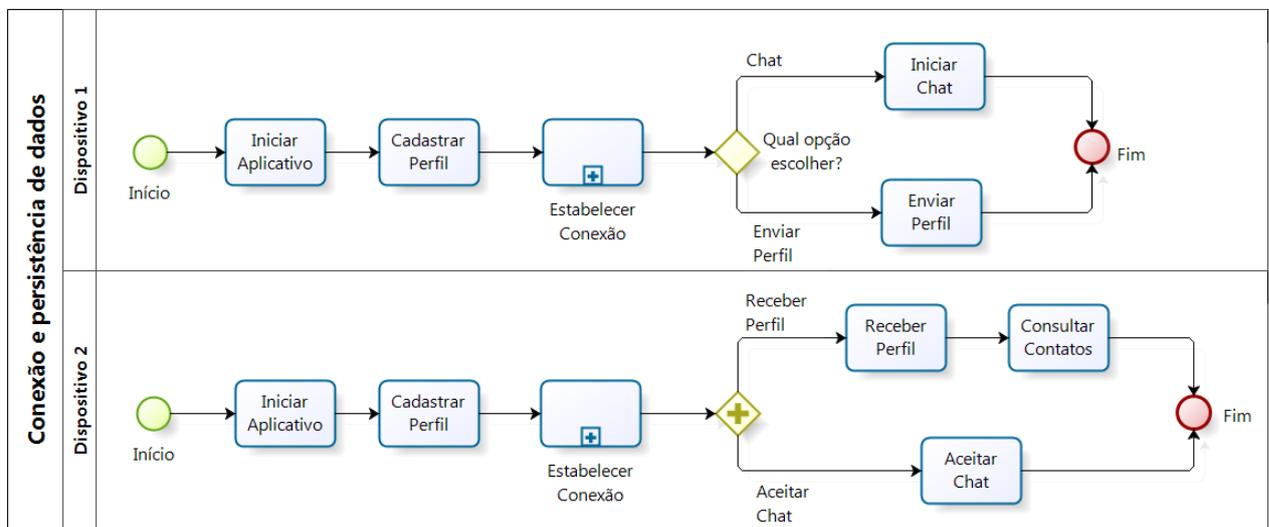


Figura 3.2: Fluxograma de uso do *framework Frame Social* em dois dispositivos.

As atividades do processo **Estabelecer Conexão**, Figura 3.3, são:

- Ligar *Wi-fi*: A antena *Wi-fi* do dispositivo deve ser ligada para que o *Wi-Fi Direct* funcione e permita que ele se conecte com outro dispositivo;
- Buscar Dispositivos: Os usuários fazem busca por dispositivos que estão no alcance;
- Solicitar Conexão: O primeiro dispositivo solicita permissão para se conectar com o segundo dispositivo, que pode aceitar ou não;
- Conectar: Caso o segundo dispositivo aceite a conexão, eles se conectam. Caso não aceite, o primeiro dispositivo pode pedir para se conectar novamente.

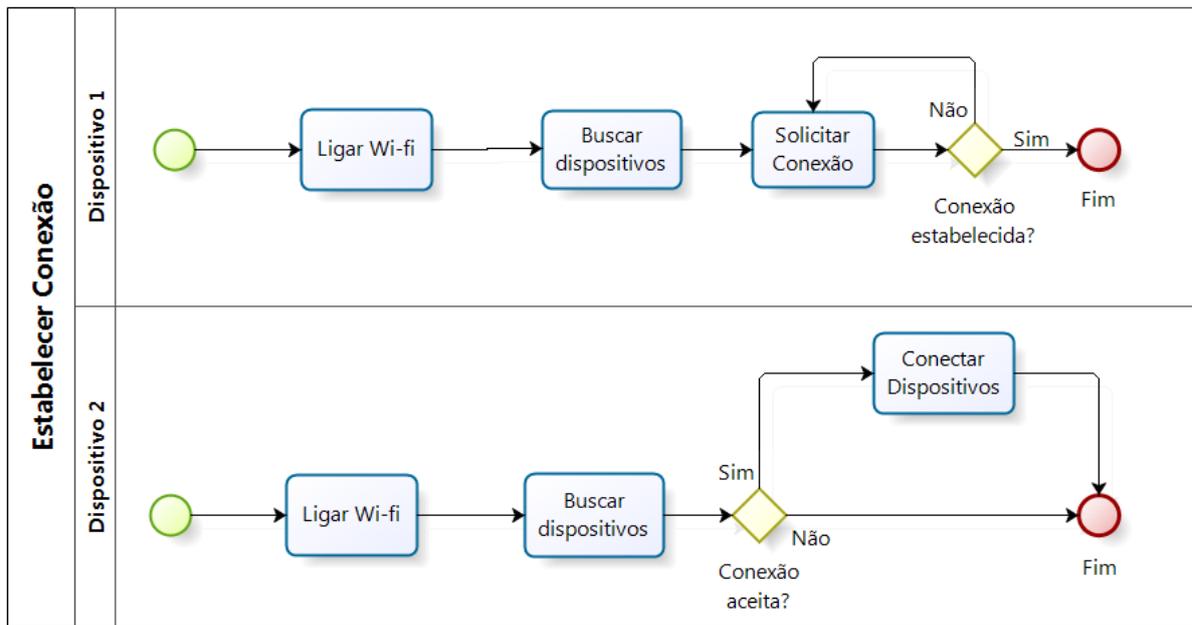


Figura 3.3: Fluxograma da comunicação entre dispositivos pelo *framework Frame Social*.

3.3 Implementação do Projeto no *Android*

Nesta seção estão apresentadas as classes e pacotes do projeto *Android*, desenvolvidas em plataforma *Java*, contendo os códigos do *framework FrameSocial*. A interface de desenvolvimento (IDE) *Eclipse*, é preferida para o desenvolvimento do *framework*, graças a sua contribuição ao desenvolvimento do *Android OS* e, principalmente, por possuir grande suporte do *Android* por ser sua ferramenta de desenvolvimento oficial. O *plugin Android Development Tools* (ADT) deve ser adicionado. Uma tela da IDE *Eclipse* é mostrada na Figura 4.1.

3.3.1 Customização da Rede Social pelo Programador

O *Frame Social* permite que um programador elabore uma rede social *ad hoc*, definindo os campos de dados da tabela que mantém os perfis dos usuários da rede social. Como restrição, há dois campos obrigatórios pelo *framework*:

- campo *nome* (tipo *text*) - Embora o nome atribuído a cada perfil não tenha restrição de unicidade, o campo é obrigatório;
- campo *email* (tipo *text*) - Por ser a chave de identificação dos usuários na rede social, segue as restrições de obrigatoriedade e unicidade.

Os demais campos são definidos na classe *DefineRedeSocial.java* (no pacote *com.defineRedeSocial*), de acordo com o interesse do programador. Nessa classe, o programador deve inserir quatro nomes para as colunas, além dos tipos de dados para cada coluna da tabela. O *FrameSocial* suporta os seguintes tipos de dados relativos ao SGBD *SQLite 3* em [SQLite \(2013\)](#):

- *null* - Registro com valor nulo;
- *integer* - O registro é um inteiro com sinal, podendo ser armazenado em 1, 2, 3, 4, 6 ou 8 *bytes*, conforme a magnitude do valor;
- *real* - O registro é um valor com ponto flutuante, armazenado como um número com ponto flutuante *IEEE* de 8 *bytes*;
- *text* - O registro é uma cadeia de texto.

Quando o aplicativo *Frame Social* é executado, a classe *MainActivity.java*, no pacote *com.activities*, executa a tela da Figura 4.2, onde o usuário pode cadastrar seu perfil pessoal na rede social. Na tela de cadastro, Figura 4.3, são apresentados os campos obrigatórios e os campos que foram definidos pelo programador através da classe *DefineRedeSocial.java*.

3.3.2 Módulo de Comunicação

As classes que mantêm as instruções referentes à comunicação entre os dispositivos dos usuários do *Frame Social* estão implementadas no pacote *com.wifidirect*.

O fluxograma de uso do *framework* está apresentado na Seção 3.2. Após concluir o preenchimento do seu perfil no *Frame Social*, o usuário pode solicitar a atividade de conexão, efetuando a busca de usuários com dispositivos que estejam localizados na área de alcance da antena *Wi-Fi Direct* do seu dispositivo, Figura 4.4. Nesse momento, o *framework* gera um arquivo CSV, com os dados do seu perfil, além dos dados da sua lista de contatos, já pareados. Esse arquivo CSV é gravado no armazenamento externo do dispositivo. No *framework*, a busca de dispositivos está implementada na classe *DeviceListFragment.java*. Essa classe traz um *ListFragment*¹ que lista os pares disponíveis após a busca, e aguarda as eventuais interações da *Activity*, permitindo a escolha de um par para solicitar conexão, Figura 4.5. O segundo usuário recebe em sua tela a solicitação, podendo aceitar ou recusar. A classe trata o *status* do dispositivo, que pode ser:

- disponível;
- convidado;
- sem sucesso;
- indisponível;
- desconhecido.

Uma vez conectados, os usuários têm a opção de se comunicar por um *chat*, além de trocarem seus perfis, caracterizando um relacionamento social.

¹*Fragments* representam uma parte da interface da tela. Eles são executados dentro de uma *activity* principal e possuem ciclo de vida bem definido: *create/pause/resume/destroy*. A idéia destes é modularizar componentes para que possam ser reutilizados.

3.3.3 Opção de Chat

Quando os usuários iniciam um *chat*, o pacote *com.chat.msg* é chamado. O pacote registra um serviço local no *Android OS* e hospeda um par de fragmentos para gerenciar as operações do *chat*. O pacote torna público o serviço de *chat* e tenta descobrir os serviços publicados pelo segundo dispositivo conectado. O pacote verifica o *status* de conexão. Se ele confirma que a conexão continua estabelecida, e que o dispositivo pareado publicou o mesmo serviço de *chat*, o pacote abre *sockets* para iniciar uma conversa. Em seguida, o código da classe *WiFiChatFragment.java* é adicionado à atividade principal do pacote, *WiFiServiceDiscoveryActivity*, para gerenciar a interface e transferências necessárias a uma sessão de conversa.

A classe *WiFiChatFragment.java* implementa um *fragment* que mantém a interface referente ao *chat*, que inclui uma exibição em lista das mensagens trocadas e um campo de entrada de texto com um botão "enviar".

A classe *ChatManager.java* implementa a leitura e a escrita das mensagens com *socket buffers*. Ela usa um *handler*² para publicar as mensagens à trilha (*thread*), para atualizar a exibição ao usuário.

3.3.4 Opção de Envio de Perfil

Quando o usuário escolhe a opção de enviar perfil, o arquivo CSV é transmitido para o dispositivo conectado. Ao receber o arquivo, o segundo dispositivo faz a importação para o banco de dados *SQLite* local da aplicação *Frame Social*. O código responsável pela ETL (extração, transformação e carga) do arquivo CSV está na classe *DeviceDetailFragment.java*. Além do perfil do primeiro usuário, o arquivo CSV contém os amigos desse usuário. A descrição desse procedimento será abordada logo a seguir, na Seção 3.3.5.

3.3.5 Módulo de Persistência

No modelo de dados do *Frame Social*, conforme Figura 3.4, descreve-se as tabelas do banco de dados *SQLite* local da aplicação:

- tabela *perfil* - Armazena o perfil do usuário do dispositivo;
- tabela *amigo* - Armazena os perfis que já foram recebidos de amigos conectados;
- tabela *amigoDeAmigo* - Armazena a lista de amigos dos amigos.

²*Handler* é uma classe do pacote *android.os*, utilizado como manipulador de mensagens e ações. É utilizado para agendar mensagens que serão executadas e também para enfileirar uma ação que será executada em um segmento diferente do qual foi enviada.

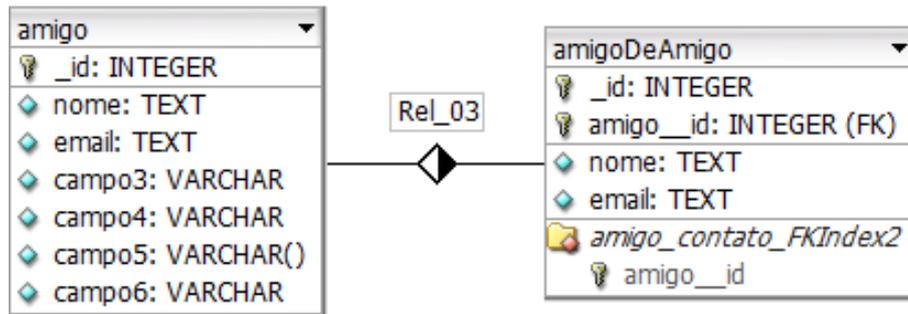


Figura 3.4: Modelo de dados do *framework Frame Social*.

O banco de dados *projeto.db* é criado pela classe *DbHelper.java* no armazenamento interno de cada dispositivo. A classe *DbHelper.java* possui o procedimento *onCreate*, que busca as classes *TabelaPerfil.java*, *TabelaAmigo.java* e *TabelaAmigoDeAmigo.java* para a criação das tabelas do *framework*, buscando também a classe *DefineRedeSocial.java*, onde o programador da rede social definiu as colunas opcionais. Essas classes estão no pacote *com.banco*. Cada registro das tabelas é identificado no *SQLite* pela coluna *_id*, única, do tipo *integer*.

O arquivo CSV é composto pelas colunas "nome", "email", "campo3", "campo4", "campo5" e "campo6" da tabela "perfil", além das colunas obrigatórias "nome" e "email", da tabela "amigo". Os nomes dos campos de 3 a 6, naturalmente, são denominados conforme as definições apontadas pelo programador. O dispositivo receptor do arquivo armazena as colunas da tabela "perfil" do primeiro dispositivo em sua tabela "amigo", e armazena as duas colunas obrigatórias da tabela "amigo", do primeiro dispositivo em sua tabela "amigoDeAmigo". No dispositivo destino, a tabela "amigoDeAmigo" relaciona-se com a tabela "amigo", onde o campo "amigo._id" é igual ao campo "amigoDeAmigo._id_amigo". Dessa forma, a rede de relacionamentos do dispositivo pode ser listada.

3.3.6 Consulta dos Contatos

Após o usuário entrar na opção "Contatos", caso ele escolha "Amigos", é exibida uma lista dos amigos com os quais ele já tenha recebido o perfil. Essa lista é implementada pela classe *ContentManipulaDadosAmigo.java*, com o uso de consulta que está implementada na classe *MyContentProviderAmigo.java*. Ao escolher um amigo da lista, a classe *ContentDetalheDadosAmigo.java* busca o perfil desse amigo na tabela "amigo".

Ainda na opção "Contatos", também está a opção "Amigos de amigos", por onde as classes *ContentManipulaDadosAmigoDeAmigo.java* e *MyContentProviderAmigoDeAmigo.java* implementam a lista de amigos com os quais seus amigos receberam perfis. Ao escolher um amigo de amigo, a classe *ContentDetalheDadosAmigoDeAmigo.java* lista o nome e o email desse terceiro contato, além disso, a classe consulta a qual dos seus amigos este terceiro foi conectado. Afinal, não faz sentido que o usuário receba o perfil completo de um terceiro com o qual ele possa nunca ter se conectado. E assim fica apresentada a abstração da rede social e dos relacionamentos de seus membros até o segundo grau.

3.3.7 Operações Disponíveis sobre o Banco de Dados

A classe *MyContentProviderPerfil.java* implementa as operações básicas de inserir, pesquisar, atualizar ou excluir dados (CRUD) sobre a tabela "perfil". Trata-se do *content provider* que controla o acesso aos dados dessa tabela, conforme as solicitações feitas por outras partes do *framework*. *Content providers* são responsáveis por fornecer os dados às aplicações, conteúdo essencial para seu funcionamento. A classe *CadastraPerfil.java*, por exemplo, chama a classe *MyContentProviderPerfil.java* para efetuar a inserção dos campos na tabela, quando o usuário cria seu perfil na rede social. A classe *ContentManipulaDadosPerfil.java* implementa a listagem do perfil do usuário, além de permitir a exclusão do perfil, atendendo assim as ações "pesquisar" e "apagar" no banco de dados. A classe *ContentDetalheDadosPerfil.java* oferece pesquisa dos registros da tabela, além de implementar inserção, atualização, e exclusão de campos solicitados. Todas as ações CRUD implementadas obedecem as restrições definidas para a tabela "perfil", no banco de dados do *framework*, conforme a Figura 3.4.

Também está implementado um *content provider* para a tabela "amigo" e outro para a tabela "amigoDeAmigo". São as classes *MyContentProviderAmigo.java* e *MyContentProviderAmigoDeAmigo.java*, respectivamente. As ações de CRUD nessas tabelas são solicitadas para essas classes, mas nem todas são previstas, pois a regra de negócio do *framework* determina que não é possível que o usuário altere no seu dispositivo os campos do perfil de um amigo. Entretanto, ele pode apagar o perfil de um amigo do seu dispositivo. O *content provider* da tabela "amigoDeAmigo" efetua a pesquisa dos amigos de amigos. Por essas classes o *framework* apresenta a abstração da rede social, quando informa a qual amigo está relacionado cada amigo de amigo.

As propriedades ACID (atomicidade, consistência isolamento e durabilidade) estão previstas nas transações efetuadas no banco de dados do *framework*. O *framework* trata a integridade referencial entre as tabelas, impedindo a corrupção dos registros. Por exemplo, ao excluir um perfil da tabela "amigo" os registros relacionados a esse amigo também são excluídos da tabela "amigoDeAmigo".

3.3.8 ETL entre os Bancos de Dados *SQLite*

Um ponto crítico pra o funcionamento da rede social, suportada pelo *framework*, é o tratamento da sincronia entre os bancos de dados *SQLite* locais de cada dispositivo participante da rede. O sucesso dessa sincronia é tratado com a transmissão de dados já descrita na Seção 3.3.4. Agora são descritos os procedimentos de extração, transformação e carga (ETL) de dados no contexto de *SQLite* e *Android* do *framework*.

A exportação de dados das tabelas é solicitada pela classe *MainActivity.java* quando o usuário entra no módulo de comunicação. Esse fluxo de processo foi descrito na Seção 3.2. Essa classe usa a classe *java.io.FileOutputStream.class*, nativa da linguagem, para escrita em um arquivo armazenado na memória auxiliar do dispositivo, conforme o sistema de arquivos do *Android*. O banco de dados é acessado via classe *DbHelper.java* do *framework*. O arquivo gravado atende aos requisitos definidos nas resoluções RFC 4180 e RFC 7111, por Shafranovich (2005) e Hausenblas et al. (2014), respectivamente. Trata-se de um arquivo CSV utilizando o caractere ";" como separador. Por regra, é preciso que o usuário já tenha gravado seu perfil no dispositivo, para que funcione a extração de dados. Caso o usuário não tenha gravado seu perfil, o *framework* solicita que isso seja feito.

A importação dos dados acontece na classe *DeviceDetailFragment.java*, usando a classe *java.io.BufferedReader.class* para ler o arquivo CSV de forma estruturada. O *content provider* de cada tabela formaliza a organização dos campos de acordo com a inserção solicitada, obedecendo as restrições do modelo de dados.

Capítulo 4

Aplicação da proposta

Neste capítulo são apresentados os resultados de uma aplicação prática realizada com o *framework Frame Social*. A descrição do processo de uso do *framework* é acompanhada das telas do aplicativo, ilustrando as ações referentes ao escopo do projeto: comunicação entre dispositivos e persistência de dados.

4.1 Resultados do *Frame Social*

Um exemplo de uso do *Frame Social* pode ser aferido como uma rede social de currículo acadêmico, podendo ser útil em congressos e conferências. Neste caso, o *Frame Social* permite que um usuário saiba as principais características pessoais e profissionais de outro usuário do aplicativo que esteja ao alcance de sua antena. Para concluir isto, foram feitos testes em três telefones celulares com *hardware* compatível com o padrão *WiFi Direct*, sendo eles:

- *Motorola Moto G: Android* versão 4.4.2;
- *LG Optimus L5*, modelo LG-E450: *Android* versão 4.1.2;
- *LG Nexus 4*, modelo LG-E960: *Android* nativo versão 4.4.2.

O programador da rede social definiu os campos opcionais do perfil de usuários na classe *DefineRedeSocial.java*:

- coluna 3: "Telefone", tipo *text*;
- coluna 4: "Profissao", tipo *text*;
- coluna 5: "LocalDeTrabalho", tipo *text*;
- coluna 6: "Interesses", tipo *text*.

A Figura 4.1 exibe o código da classe *DefineRedeSocial.java* alterado com as definições dos campos 3, 4, 5 e 6, além da tipagem de dados desses campos.

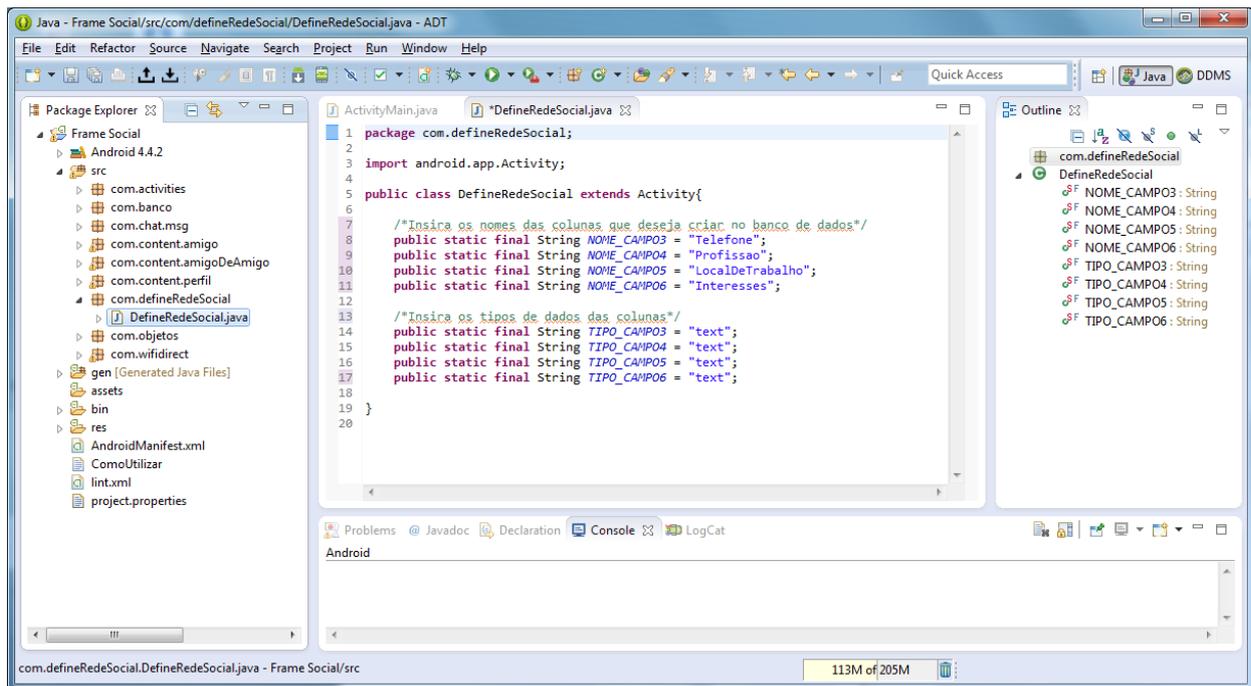


Figura 4.1: Código da classe *DefineRedeSocial.java*.

4.1.1 Utilização da Rede Social pelo Usuário

Quando *Frame Social* é executado, apresenta uma tela de boas vindas ao usuário. Em seguida é mostrado o menu principal, como na Figura 4.2, onde o usuário escolhe o que deseja fazer. Na primeira opção, "Perfil", ele pode cadastrar o seu perfil, que é obrigatório para que ele consiga se conectar a outro usuário. A conexão acontece na segunda opção, "Encontrar Pessoas", onde o usuário pode conversar com outro através de *chat*, e também enviar seu perfil. Na terceira opção desta tela, "Contatos", o usuário pode visualizar o perfil de seus amigos e o nome e o e-mail dos amigos deles.



Figura 4.2: Menu principal.

A Figura 4.3 mostra um exemplo de um perfil fictício sendo cadastrado, em que um usuário digita seus dados pessoais. Esses dados formam o seu currículo, que posteriormente outros usuários poderão ver. Os valores podem ser editados e deletados pelo usuário. Não é permitido que um usuário cadastre seu perfil sem preencher nome e e-mail.



Figura 4.3: Cadastrando um perfil.

Ao escolher a segunda opção do menu inicial, "Encontrar Pessoas", e tocar na lupa (canto superior esquerdo), é feita uma busca por dispositivos que possuem o aplicativo instalado e que estejam com a antena *Wi-Fi* ligada, Figura 4.4. A Figura 4.5 mostra uma lista que é exibida para que o usuário escolha quem ele quer convidar para estabelecer conexão. Quando ele escolhe algum dispositivo da lista, são exibidas as opções mostradas na Figura 4.6. Nessa parte, mesmo aparecendo a opção de *Chat*, não é possível iniciar um *chat* antes que a conexão seja estabelecida. A Figura 4.7 exibe a atividade de convite em ambos os dispositivos envolvidos. Depois que o outro usuário aceita o pedido de conexão, ele pode enviar o seu perfil ou iniciar um *chat* com o outro usuário. A Figura 4.8 mostra a tela exibida com as opções.

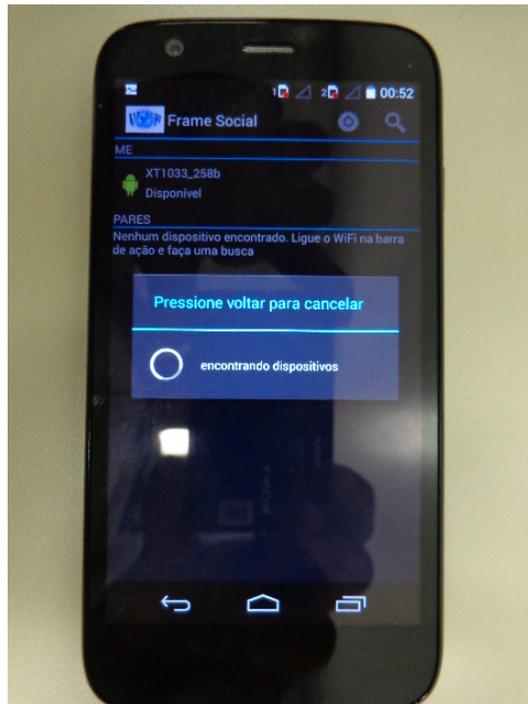


Figura 4.4: Busca de dispositivos.

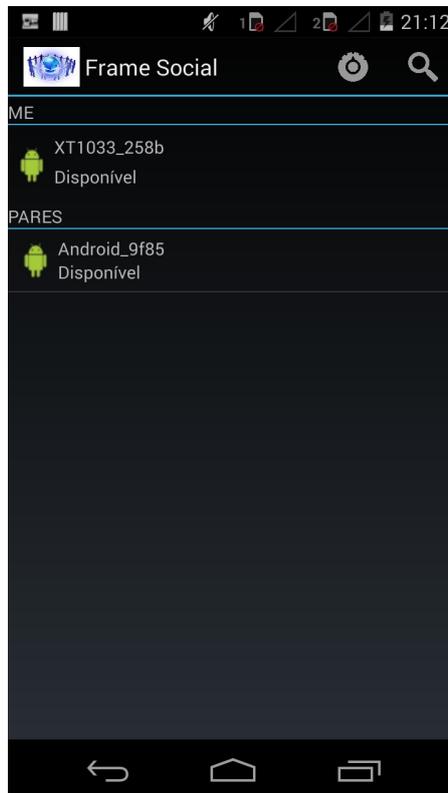


Figura 4.5: Dispositivos encontrados.

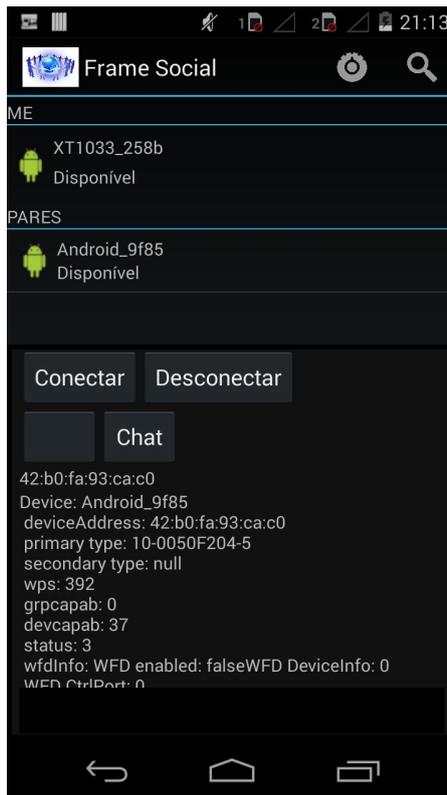


Figura 4.6: Opções para conexão.

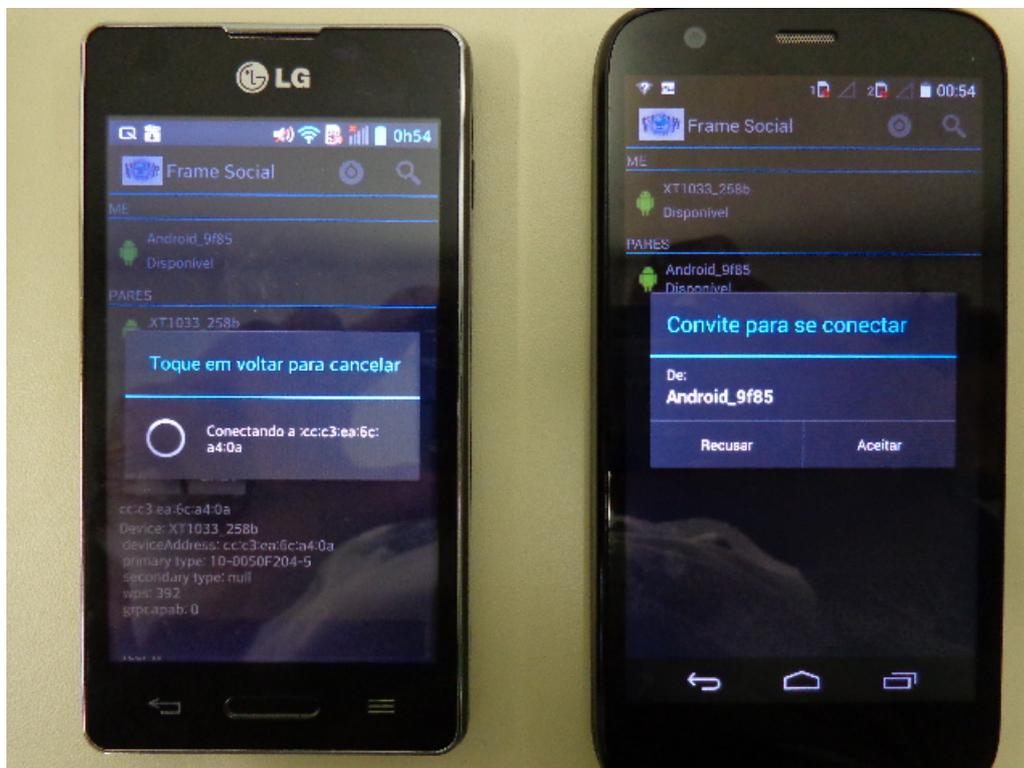


Figura 4.7: Convite para conexão.

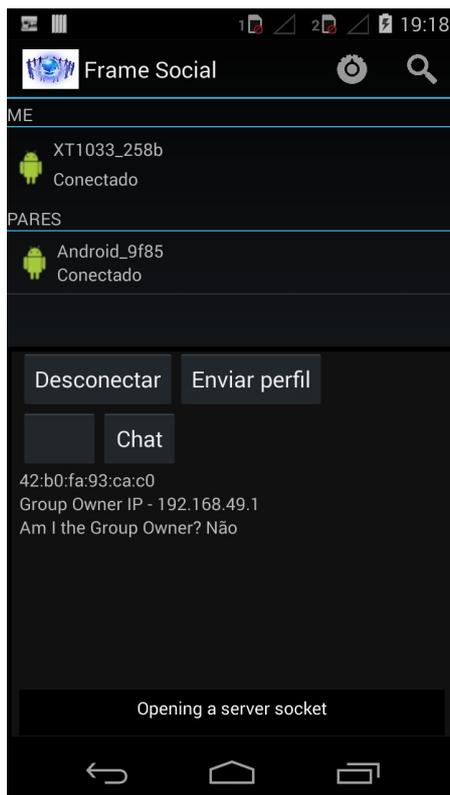


Figura 4.8: Opções para comunicação.

Quando a opção de "Enviar perf" é escolhida, no submenu, o outro usuário recebe o currículo, e o usuário que enviou passa a ser amigo dele. Além do perfil, a lista de seus amigos também é enviada, e depois pode ser visualizada pelo outro usuário. A Figura 4.9 mostra a mensagem que aparece no dispositivo quando um usuário recebe o currículo de outro usuário. Se a opção escolhida for *Chat*, uma conversa é iniciada, como mostram as Figuras 4.10 e 4.11. Caso a opção escolhida seja "Desconectar" o nome do dispositivo permanece na lista de dispositivos encontrados, porém não aparece nenhuma opção para estabelecer conexão.



Figura 4.9: Mensagem de perfil recebido.

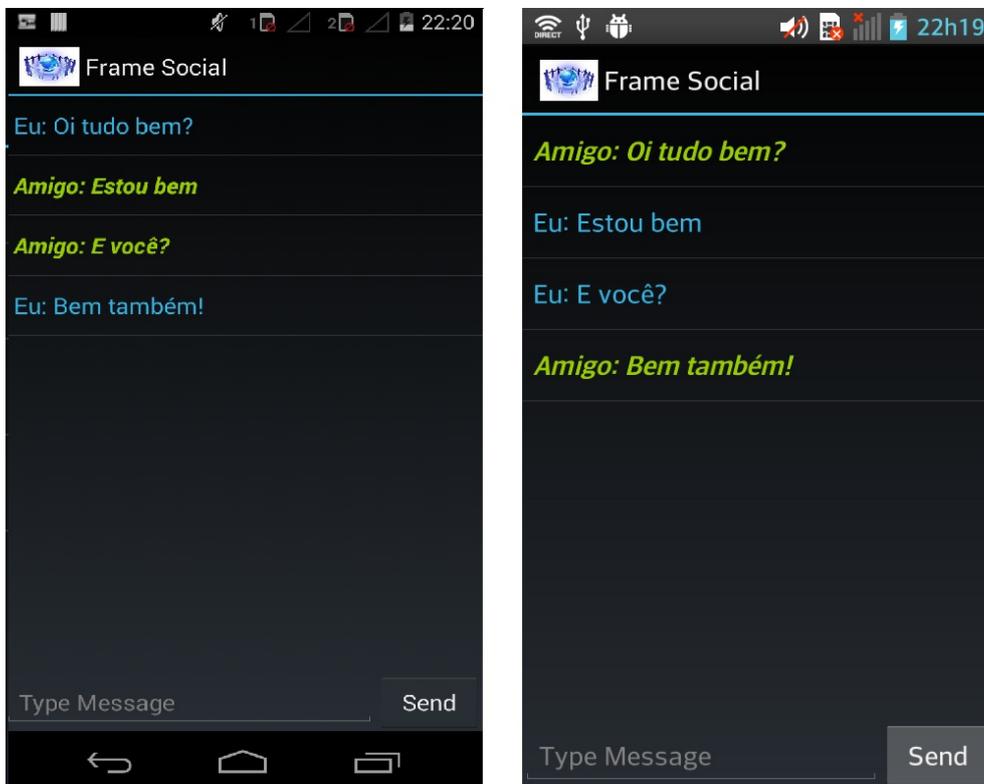


Figura 4.10: Tela de conversa entre usuários.

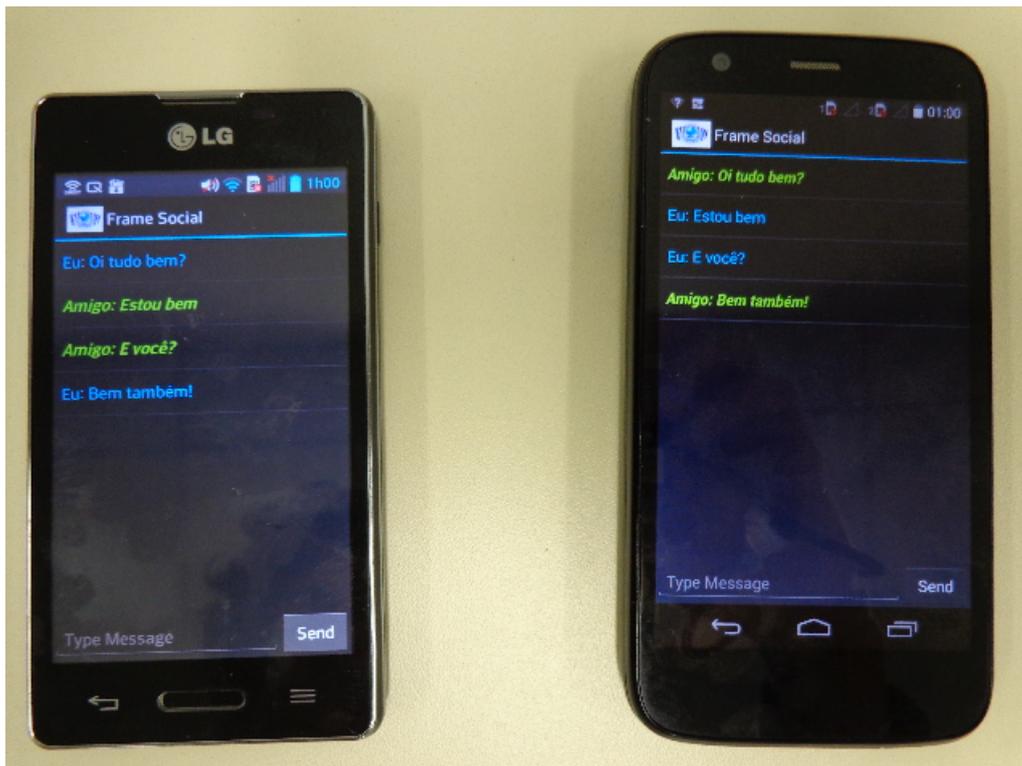


Figura 4.11: Dispositivos exibindo uma conversa entre usuários.

Caso a opção escolhida no menu principal seja "Contatos", o usuário entra em outro menu, como mostra a Figura 4.12. Se o usuário escolher a opção "Amigos", é possível visualizar uma lista de amigos que estão cadastrados em seu dispositivo, e selecionando um deles, o currículo é mostrado. A Figura 4.13 mostra esta lista exibida, e também o currículo do amigo selecionado para visualização. Quando a opção escolhida for "Amigos de amigos", uma lista é exibida contendo os amigos dos amigos que estabeleceram conexão com o usuário. Se algum amigo desta lista for escolhido é possível ver o nome, o email e o nome do amigo ao qual ele pertence. Isso é mostrado na Figura 4.14.

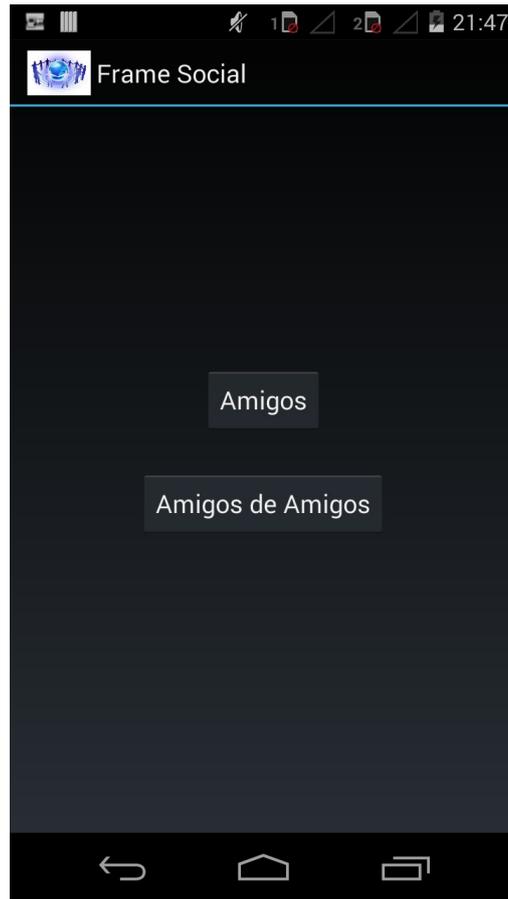


Figura 4.12: Menu de contatos cadastrados.

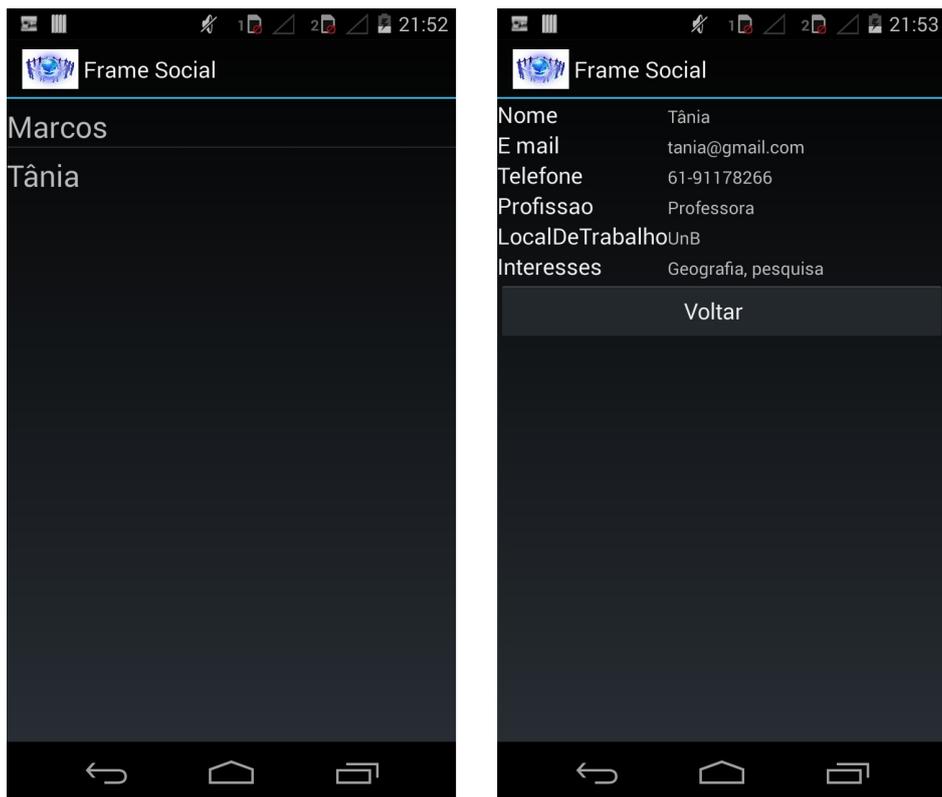


Figura 4.13: Lista de amigos e perfil do amigo selecionado.

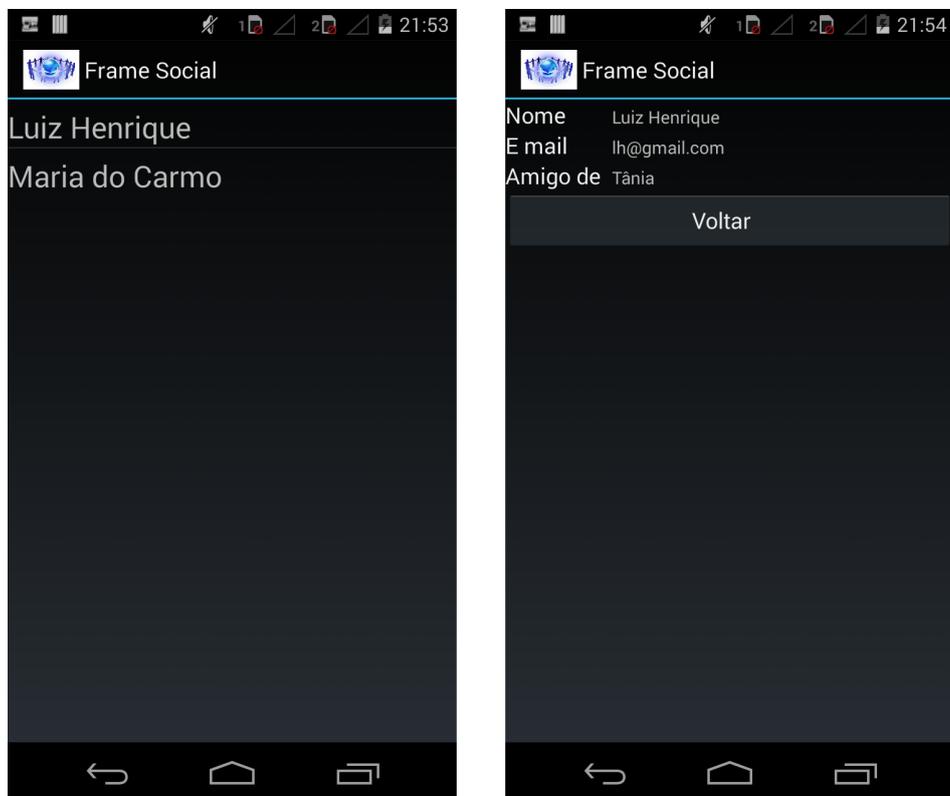


Figura 4.14: Lista de amigos de amigos e perfil do amigo selecionado.

Os testes mostrados neste capítulo foram realizados em três dispositivos celulares diferentes, das marcas *LG* e *Motorola*. As telas representam as ações durante o uso do aplicativo de uma Rede Social desenvolvido com o *Frame Social* em um caso de uso específico. Estas ações são referentes às atividades descritas nos fluxogramas da Seção 3.2. Pode-se ver nas telas e fotos que todas as funcionalidades do aplicativo funcionaram conforme a proposta do projeto.

Capítulo 5

Contribuições do Estudo e Conclusão

Este estudo lidou com o problema da comunicação e troca de dados por um grupo de pessoas onde a conexão a um ponto de acesso ou a uma rede 3G nem sempre é possível. Também tratou da persistência de dados nessa rede. A solução proposta foi a criação de um *framework* para dispositivos móveis usando *Wi-Fi Direct* e *SQLite*.

Este capítulo apresenta uma avaliação do projeto, a fim de mostrar uma síntese do trabalho. Essa síntese consegue, por sua vez, abranger os resultados que foram alcançadas ao longo do estudo. O capítulo também levanta questões que surgiram como problemas, assim como as possíveis soluções para eles. Por fim, são apresentadas possibilidades para o desenvolvimento futuro do que foi proposto na pesquisa.

5.1 Análise dos Resultados

O objetivo do trabalho foi o desenvolvimento de um *framework* para atuar como solução na implementação de uma rede social com dispositivos móveis conectados ponto-a-ponto por redes móveis *Ad hoc*. Também foi proposto que o *framework* suportasse a comunicação entre dispositivos móveis e a persistência de dados. O Capítulo 4 descreveu a aplicação prática no estudo de caso do *Frame Social*, que estabelece a conexão entre os dispositivos móveis e mantém a transferência de dados, sem o uso de qualquer serviço 3G ou ponto de acesso, com persistência de dados local em cada dispositivo, e aplicando isso à implementação de uma rede social.

Portanto, o objetivo é cumprido com o *Frame Social*, onde é possível implementar uma rede social por conexão ponto-a-ponto, com transferência e persistência de dados. Além disso, o trabalho foi desenvolvido para orientar futuros desenvolvedores sobre a visão geral desse sistema, e, finalmente, denotar os problemas encontrados. Com a ajuda deste documento, o desenvolvedor pode obter informações adequadas ao estudo sobre a implementação, bem como a teoria e o contexto.

A tecnologia *Wi-Fi Direct* foi pesquisada e bem compreendida, sua aplicação relevante foi desenvolvida ao propósito do estudo. A descoberta de pares é bem sucedida e a conexão ponto-a-ponto móvel é mantida nos dispositivos *Android*. Além disso, a conexão entre dois dispositivos é estabelecida com sucesso e, o mais importante, a transferência e persistência de dados entre eles é realizada. Posteriormente, os dados da rede social são gravados em bancos de dados *SQLite* de cada dispositivo, transferidos entre os pares e gerenciados pelo aplicativo do *framework*.

Uma ferramenta de envio de mensagem simples e básica foi adicionada ao aplicativo, onde os pares podem enviar e receber mensagens instantâneas entre si. Tecnicamente, isso é um processo muito mais fácil, em comparação com transferência de arquivo ou áudio. A passagem de *strings* (fluxos), através de *sockets*, pode ser feita facilmente, usando vários métodos específicos de classes de *I/O stream* (fluxo de entrada-saída).

Como resultado, o projeto garante uma solução prática para o problema proposto, independentemente dos problemas menores. Existem algumas questões que ainda permanecem como possíveis problemas e precisam ser resolvidos. Abaixo está a lista atual de questões, juntamente com informações adicionais sobre o problema em si.

- **Remoção de grupos:** Depois de estabelecer a conexão e a transferência de dados com sucesso, a operação de desconexão não responde a necessidades de usuários de um lado. Por exemplo, quando o botão de desconectar é pressionado, os grupos são realmente removidos, mas o menu ainda mostra que o par está conectado. Esta questão não é apenas de layout, mas também de descoberta que basicamente funciona através de *broadcast receivers* (receptores de radiodifusão). Pois, quando o grupo é removido, o sistema automaticamente procura por pares e descobre os pares instantaneamente. Portanto, a lista de pares não é atualizada e a reconexão só pode ser estabelecida quando o aplicativo for reiniciado.

Uma possível solução pode ser a adição de exceções ao *broadcast receiver*, na qual o status da conexão e dos pares é verificado.

- **Pares:** Outra questão é sobre uma característica da classe *WiFiP2pManager*. Depois de uma conexão e possível transferência de dados bem-sucedida, os pares permanecem conectados por um tempo limitado, que é cerca de cinco minutos. Durante este período, os pares podem se conectar, se desconectar e transferir dados várias vezes. No entanto, a operação de descoberta tem de ser chamada novamente para a busca de pares após cinco minutos. Isto leva a inconsistências no sistema e restringe a capacidade de conexão constante.

Não há atualmente nenhuma solução possível para este problema. No entanto, uma nova versão da API do *Wi-Fi Direct* pode resolvê-lo.

- **Transmissão Bidirecional** de dados: A questão é solucionada com a implementação em modo *multi thread* (múltiplos segmentos) com serviços do sistema operacional. Mas quando o serviço não pode ser interrompido de forma explícita, não acontece a próxima etapa, que é o *socket* de conexão. E, certamente, a transferência de dados não pode ser concluída sem *socket* de conexão, que é uma questão importante para este estudo.

Não há atualmente nenhuma solução possível encontrada em relação a este problema. No entanto, suspeita-se que o trabalho com *multi thread* (vários segmentos) provoca problemas de concorrência e possíveis melhorias no tratamento de *threads* podem resolver esse problema.

Além das questões acima, há problemas que foram tratados, embora ainda permaneçam como uma questão de funcionalidade. A API do *Wi-Fi Direct* já foi uma grande questão para todos na indústria e desenvolvimento com *Wi-Fi Direct*, essencialmente devido a alguns problemas de estabilidade de *drivers* na versão ICS (*Ice Cream Sandwich*)

do *Android OS*. O problema era sobre o processo de reconexão. Dispositivos estavam se desligando a cada quarto processo de conexão, ou uma nova conexão só podia ser estabelecida uma vez que o dispositivo fosse religado. Embora isso tivesse sido tratado na versão 4.1 JB (*Jelly Bean*) do *Android OS*, com aprimoramentos de recursos e atualizações de API, ainda restam ocorrências relacionadas à instabilidade, conforme relatos nos fóruns da comunidade de desenvolvedores. Nas versões seguintes do *Android*, o *Google* resolveu *bugs* e adicionou características *Wi-Fi Direct*. Mesmo na atual versão 4.4 *KitKat*, ainda há questões a solucionar sobre várias falhas que precisam ser revistas, tanto em termos de funcionalidade, quanto de performance. O *Wi-Fi Direct* ainda é uma tecnologia emergente. Disso também deriva a contribuição desse estudo.

5.2 Trabalhos Futuros

Novas funcionalidades podem ser agregadas ao *framework* com o tratamento das seguintes questões, que podem ser indicadas em trabalhos futuros:

- Transferência e a persistência de uma foto do perfil de usuário da rede social. A implementação de um campo do tipo *blob* no banco de dados *SQLite*, juntamente com sua transmissão em um objeto binário pela rede.
- Suporte a vários perfis que o usuário queira criar no dispositivo, possibilitando que ele alterne seus perfis entre redes sociais diferentes, de acordo com seu contexto de uso.
- Transferência de quaisquer arquivos pessoais dos usuários, sem restrição da extensão do arquivo, implementada sobre o armazenamento local dos dispositivos.
- Implementação de uma demonstração do grafo da rede social, abstraída a partir das informações dos amigos conectados e seus amigos de amigos (descrição na Seção 3.3.5).

Apesar das muitas utilidades do estudo, tais como transferência de arquivos, descoberta de pares e conexão, o *framework* não propôs a transferência de áudio. Um futuro trabalho pode desenvolver uma implementação, em que o áudio seja gravado e seus dados sejam armazenados em um arquivo. Desta forma, este arquivo pode ser enviado para os pares através de *sockets*. A conexão por *sockets* é obtida pelo mesmo método que a transferência de arquivos. Ainda, um projeto futuro pode fornecer uma solução em termos de transmissão ao vivo (*streaming*), lidando com os problemas e limitações já conhecidos e os erros de *buffer*. Um exemplo de utilização sugerida pode ser uma solução de "babá eletrônica", com os dispositivos móveis. Ou ainda, uma conversa de áudio, independente de rede telefônica (ou tarifação por operadoras).

Embora a segurança não é um problema deste estudo, é necessário mencionar as deficiências do *Wi-Fi Direct* sobre a segurança. O *Wi-Fi Direct* é protegido com a criptografia WPA2 (*Wi-Fi Protected Access 2*, nome comercial para o padrão 802.11i de Alliance (2010)) mas opera independente do sistema de segurança de um ponto de acesso *Wi-Fi* (por exemplo, um roteador). Isso significa que não é necessário ter credenciais na infraestrutura de rede do ponto de acesso para se conectar à rede *Wi-Fi Direct*. Existem

potenciais problemas em relação a como os dispositivos se interconectam e sobre as medidas para prevenir que pessoas ou dispositivos indesejados se conectem ao dispositivo *Wi-Fi Direct* do usuário. Afinal, o usuário provavelmente não quer que qualquer um no aeroporto seja capaz de transferir todas as fotos de suas férias de sua câmera digital ou *smartphone*. Um trabalho futuro poderá avançar essa investigação.

O estudo pode ser ampliado e as questões futuras precisarão ser tratadas, acompanhando novas melhorias na API *Wi-Fi Direct* do *Android*. Por exemplo, o processo de *streaming* ao vivo pode ser redesenvolvido em relação a soluções oferecidas ou usando outros métodos. Além disso, existem mais algumas particularidades que podem ser adicionadas ao sistema, a fim de inserir novos recursos e aumentar a facilidade de utilização. A análise de duplicação de código também pode ser considerada como um trabalho futuro, para manter o programa sólido.

Outra implementação pode ser a adição da funcionalidade de compartilhamento de Internet entre os dispositivos. Atualmente, a API *Wi-Fi Direct* não oferece esse recurso para o sistema *Android*. No entanto, a própria *Wi-Fi Direct* praticamente oferece compartilhamento de Internet em outras plataformas. Por esta razão, esta melhoria pode ser declarada como um trabalho futuro. Ao fazê-lo, assim como para uma investigação mais aprofundada, usuários serão capazes de compartilhar sua conexão com a Internet, em áreas particulares, e esta partilha pode avançar para outros pontos onde, eventualmente, todos no grupo podem usar uma conexão de Internet sem pagar taxas extras.

Referências

- Alliance, W.-F. (2010). Wi-fi certified wi-fi direct, personal, portable wi-fi technology. Online; <http://www.wi-fi.org/knowledge-center/white-papers/wi-fi-certified-wi-fi-direct-personal-portable-wi-fi-technology-2010>; accessed 04-agosto-2013. [vii](#), [2](#), [12](#), [13](#), [14](#), [50](#)
- Android (2013a). Android. Online; <http://www.andorid.com/>; accessed 15-junho-2013. [21](#)
- Android (2013b). Android. Online; <http://www.developer.andorid.com/>; accessed 15-junho-2013. [vii](#), [21](#), [23](#), [24](#)
- Boudreau, E. B. and Arathoon, L. (2008). O futuro das redes sociais móveis. Online; http://www.e-thesis.inf.br/index.php?option=com_content&task=view&id=2752&Itemid=52; accessed 1-abril-2013. [1](#), [7](#)
- Boyd, D. M. and Ellison, N. B. (2007). Social network sites: Denition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13:210–230. [4](#)
- Castro, J. M. and Holanda, M. (2011). Eiko: A social mobile network for manet. *Information Systems and Technologies (CISTI), 2011 6th Iberian Conference on*, pages 1–5. [2](#), [25](#), [26](#), [27](#)
- Date, C. J. (2004). *Introdução a sistemas de bancos de dados*. Elsevier, Brasil. [15](#)
- de Almeida Santos, L. M. M. (2014). My-direct: Um middleware para desenvolvimento de redes sociais móveis p2p. Programa de Pós-Graduação em Ciência da Computação (PROCC), Universidade Federal do Sergipe. [15](#), [25](#)
- de Rezende, N. S. (2004). Redes móveis sem fio ad-hoc. Pós-graduação Lato Sensu em Gerência de Redes de Computadores e Tecnologia Internet, Núcleo de Computação Eletrônica, Universidade do Rio de Janeiro. [vii](#), [12](#)
- Elmasri, R. and Navathe, S. B. (2004). *Fundamentals of Database Systems - 4ed*. Addison-Wesley, Vancouver. [vii](#), [10](#), [11](#), [17](#), [18](#)
- Elmasri, R. and Navathe, S. B. (2010). *Fundamentals of Database Systems - 6ed*. Addison-Wesley, Boston. [8](#), [15](#), [16](#), [17](#)
- Haggar, P. (2000). *Practical Java Programming Language Guide*. Addison-Wesley, EUA. [23](#)

- Hausenblas, M., Wilde, E., and Tennison, J. (2014). Rfc 7111 - uri fragment identifiers for the text/csv media type. Online; <http://tools.ietf.org/html/rfc7111>; accessed 29-maio-2014. 34
- Kalofonos, D.N., A. Z. . R. F. V.-K. M. S. J. W. P. (2006). Mynet: A platform for secure p2p personal and social networking services. In: Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications. 26
- Lemos, A. (2004). Cibercultura e mobilidade: a era da conexão. 1(41):1–21. 1, 4, 7
- Marteleteo, R. M. (2001). Análise de redes sociais - aplicação nos estudos de transferência da informação. *Ciência da Informação Brasília*, 30(1):71–81. 1, 7
- Milgram, S. (1967). The small world problem. *Psychology Today*, 2:60–67. 4
- Molina, H. G., Ullman, J. D., and Widom, J. (2001). *Implementação de Sistemas de Banco de Dados*. Campus, Rio de Janeiro. 16
- Nielsen (2012a). Pesquisa global sobre mídias sociais. Online; accessed 05-julho-2013. vii, 5, 6, 7
- Nielsen (2012b). Queda no preço de smartphones contribui para crescimento de 179% em 2011. Online; accessed 05-julho-2013. 1, 20
- Nielsen (2012c). State of the media - the social media report 2012. Online; accessed 05-novembro-2013. 1
- Nielsen (2013a). O consumidor móvel. Online; accessed 05-julho-2013. 7, 20
- Nielsen (2013b). Running digital audiences, walking advertising dollars. Online; accessed 05-novembro-2013. 1, 7
- Queiroz, A., Holanda, M., Araújo, A., and Castro, J. M. (2012). Mobisn: A framework for developing mobile social networks in manet. *Information Systems and Technologies (CISTI), 2012 7th Iberian Conference on*, pages 1–5. 2, 25, 26, 27
- Rinco, M. A. G. (2008). Protótipo de comunicação bluetooth iee 802.15.1 em tecnologia móvel parte 1. Online; <http://www.devmedia.com.br>; accessed 05-julho-2014. 13, 15
- Schoder, D., Fischbach, K., and Schmitt, C. (2005). Core concepts in peer-to-peer networking. In *Peer-to-Peer Computing: The Evolution of a Disruptive Technology*, chapter 1, pages 1–27. Idea Group, USA and UK. 8
- Shafranovich, Y. (2005). Rfc 4180 - common format and mime type for comma-separated values (csv) files. Online; <http://tools.ietf.org/html/rfc4180>; accessed 29-maio-2014. 34
- Silberschatz, A., Korth, H. F., and Sudarshan, S. (2006). *Sistema de Banco de Dados*. Makron Books, São Paulo. 17
- SQLite (2013). Sqlite. Online; <http://www.sqlite.org/>; accessed 15-junho-2013. 18, 31

- Staff, I. (2012). The sql reference for cross-platform development. Online; <http://www.ibm.com/developerworks/data/library/techarticle/0206sqlref/0206sqlref.html>; accessed 15-julho-2013. 17
- Steffen Kern, Peter Braun, W. R. (2006). Mobisoft: an agent-based middleware for social-mobile applications. Computer Science Department, Friedrich Schiller University Jena. 26
- Takahashi, T. (2000). *Sociedade da Informação no Brasil: Livro Verde*. Ministério da Ciência e Tecnologia, Brasília. 1, 4, 7
- Tanenbaum, A. S. and Weatherall, D. J. (2011). *Computer Networks*. Pearson Education publishing as Prentice Hall. Ed. 5, Boston. 8, 9
- Tomaél, M. I., Alcará, A. R., and Chiara, I. G. D. (2005). Das redes sociais à inovação. *Ciência da Informação Brasília*, 34(2):93–104. 1, 4, 7
- Vogel, L. (2013). Android sqlite database and content provider - tutorial. Online; <http://www.vogella.com/articles/AndroidSQLite/article.html>; accessed 1-abril-2013. 18