

PPGI PROGRAMA
DE PÓS-GRADUAÇÃO
EM INFORMÁTICA

Universidade Federal do Rio de Janeiro

DISSERTAÇÃO DE MESTRADO

SÍRIUS THADEU FERREIRA DA SILVA

**ANÁLISE CONTEXTUAL DA REDE SOCIAL DE VÍTIMAS PARA
RECOMENDAÇÃO DE PESSOAS NO CENÁRIO DE EMERGÊNCIA**

Rio de Janeiro

2012



Instituto de Matemática



Instituto Tércio Pacitti de Aplicações
e Pesquisas Computacionais

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
INSTITUTO TÉRCIO PACCITI
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

SÍRIUS THADEU FERREIRA DA SILVA

ANÁLISE CONTEXTUAL DA REDE SOCIAL DE
VÍTIMAS PARA RECOMENDAÇÃO DE PESSOAS NO
CENÁRIO DE EMERGÊNCIA

Dissertação de Mestrado apresentada ao
Programa de Pós-Graduação em Informática,
Instituto de Matemática e Instituto Tércio Pacciti,
Universidade Federal do Rio de Janeiro, como
requisito parcial à obtenção do título de Mestre
em Informática

Orientadora: Jonice de Oliveira Sampaio

Co-Orientador: Marcos Roberto da Silva Borges

Rio de Janeiro
2012

S586 Silva, Sírius Thadeu Ferreira da

Análise contextual da rede social de vítimas para recomendação de pessoas no cenário de emergência / Sírius Thadeu Ferreira da Silva. – 2012

216 f.: il.

Dissertação (Mestrado em Informática) - Universidade Federal do Rio de Janeiro, Instituto de Matemática, Instituto Tércio Pacciti, 2012

Orientadora: Jonice de Oliveira Sampaio

Co-Orientador: Marcos Roberto da Silva Borges

1. Redes Sociais 2. Gestão de Emergências. 3. Recomendação
4. Dispositivos Móveis – Teses. I. Sampaio, Jonice de Oliveira Sampaio (Orient.). II. Borges, Marcos Roberto da Silva (Co-orient). III. Universidade Federal do Rio de Janeiro. Instituto de Matemática Programa de Pós-Graduação em Informática. IV. Título.

CDD.

SÍRIUS THADEU FERREIRA DA SILVA

**ANÁLISE CONTEXTUAL DA REDE SOCIAL DE
VÍTIMAS PARA RECOMENDAÇÃO DE PESSOAS NO
CENÁRIO DE EMERGÊNCIA**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática, Instituto de Matemática e Instituto Tércio Pacciti, Universidade Federal do Rio de Janeiro, como requisito parcial à obtenção do título de Mestre em Informática.

Aprovada em 30 de Novembro de 2012.

Prof^a. Jonice de Oliveira Sampaio, D.Sc., UFRJ

Prof. Marcos Roberto da Silva Borges, Ph.D., UFRJ

Prof^a. Adriana Santarosa Vivacqua, D.Sc., UFRJ

Prof. Paulo Victor Rodrigues de Carvalho, D.Sc., UFRJ

Prof^a. Renata Mendes de Araujo, D.Sc., UNIRIO

A minha amada esposa,
por todo apoio e dedicação durante este trabalho.

Agradecimentos

Primeiramente, gostaria de agradecer a meus pais (*in memoriam*) pelo zelo, dedicação e carinho durante todo o tempo que pudemos passar juntos nessa vida. Reconheço todos os dias que se não fosse pela criação e pelo exemplo de vida de ambos, eu não teria me tornado a pessoa que sou hoje, muito menos estaria onde me encontro. Gostaria também de agradecer a minha esposa pelo apoio e incentivo que me ajudaram a concluir esta importante empreitada. Essas três pessoas são uma parte muito especial e importante de minha vida, e sem elas eu não estaria completo.

Agradeço também a todos os professores que eu tive até então, por estimularem a minha formação acadêmica. Reconheço que todos eles, de uma forma ou de outra, me inspiraram a seguir este caminho acadêmico, que hoje concluo apenas mais uma etapa. Aos meus orientadores, Jonice Oliveira e Marcos Borges, expressei meu profundo agradecimento por me mostrarem a sempre buscar diferentes perspectivas sobre uma possível visão de mundo e fomentarem ainda mais meu interesse pela pesquisa e pela inovação. Meus agradecimentos especiais aos professores do grupo Greco (PPGI) pelo empenho e colaboração nos ensinamentos sobre a pesquisa científica.

Agradeço a todos os meus amigos pessoais e colegas de escola, faculdade, mestrado e trabalho que, de uma maneira ou de outra, acabaram contribuindo para meu crescimento como ser humano, seja na vida pessoal, na vida acadêmica ou na vida profissional. Nominalmente, gostaria de agradecer a Luciane Jasmin e Alayne Duarte pelo companheirismo e descontração nos momentos difíceis que passamos juntos no mestrado. Nesse sentido, em especial, gostaria de agradecer ao meu “fiel escudeiro” Samuel Apolonio pelos conselhos providenciais nos momentos mais críticos do mestrado e por sua cumplicidade durante todo o tempo que estivemos juntos nessa árdua etapa de nossas vidas.

Por fim, gostaria de agradecer à consciência cósmica por me proporcionar todas essas experiências que vivenciei até hoje. A cada dia que passa, aprendo novas lições de vida. E tenho a mais absoluta certeza de que todos esses momentos que se passaram, tenham eles sido bons ou ruins, não foram em vão. Pois no final das contas, nessa vida nada é por acaso...

“Ninguém é tão sábio que não tenha algo para aprender e nem tão tolo que não tenha algo para ensinar.”
Blaise Pascal

Resumo

SILVA, Sírius Thadeu Ferreira da. **Análise contextual da rede social de vítimas para recomendação de pessoas no cenário de emergência.** 2012. 216 f. Dissertação (Mestrado em Informática) – Instituto de Matemática, Instituto Tércio Pacitti, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2012.

O crescente uso de dispositivos móveis por parte da população e a alta popularidade das mídias sociais, como o Facebook e o Twitter, na sociedade atual produz cada vez mais informações, muitas delas com dados contextuais. O uso generalizado de dispositivos móveis e das mídias sociais sugere a utilização desses recursos para resolver problemas que podem evoluir ao longo do tempo, inclusive colocando vidas ou propriedades em perigo, e exigem uma resposta imediata por meio de procedimentos e recursos coordenados.

O trabalho aqui apresentado tem como proposta coletar informações contextuais das mídias sociais e dos dispositivos móveis, analisando-as a fim de ajudar em situações de emergência. Este trabalho foca na identificação da rede social de vítimas de emergência que estão desaparecidas ou inconscientes e na recomendação de pessoas que possam fornecer informações relevantes sobre essas vítimas.

A questão crítica que estamos abordando é a falta de informações confiáveis sobre essas vítimas de emergência e, portanto, a ideia principal deste trabalho é inferir as pessoas mais próximas da rede social da vítima e recomendá-las, ajudando a equipe de emergência na coleta de informações importantes e confiáveis sobre as vítimas.

Acreditamos que através da análise das interações em redes sociais e dispositivos móveis podemos identificar os amigos mais próximos de uma vítima. Nós também utilizamos a informação contextual para identificar as pessoas mais próximas ao local do desastre, melhorando assim o processo de recomendação.

Palavras-chave: Redes Sociais. Gestão de Emergências. Recomendação de Pessoas. Dispositivos Móveis.

Abstract

SILVA, Sírius Thadeu Ferreira da. **Análise contextual da rede social de vítimas para recomendação de pessoas no cenário de emergência.** 2012. 216 f. Dissertação (Mestrado em Informática) – Instituto de Matemática, Instituto Tércio Pacitti, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2012.

The increasing use of mobile devices by the population and the high popularity of social media like Facebook and Twitter in our society produces more and more information, plenty of them with contextual data. The widespread use of mobile devices and social media suggests the use of these resources to solve problems that can evolve over time, putting lives or property in danger, and requiring an immediate response through coordinated procedures and resources.

The work presented here has the purpose of gathering contextual information from social media and mobile devices, and analyze them to help in emergency situations. This work focuses on identifying the social network of emergency victims who are missing or unconscious and on the recommendation of people who can provide relevant information about these victims.

The critical question we are addressing is the lack of reliable information on these emergency victims and, therefore, the main idea of this work is to infer the closest people in the social network of the victim and recommend them, thus helping the emergency response team to collect important and reliable information about the victims.

We believe that through the analysis of interactions in social networks and mobile devices we can identify the closest friends of a victim. We also use contextual information to identify the closest people to the disaster site, thus improving the recommendation process.

Keywords: Social Networks. Emergency Management. Person Recommendation. Mobile Devices.

Lista de Figuras

Figura 1.	Vista geral do deslizamento no morro do Bumba, em Niterói - RJ (UOL, 2010) .	23
Figura 2.	Vista aérea de uma cratera formada no morro do Bumba, em Niterói - RJ (R7, 2011)	23
Figura 3.	Esquema do Phased Emergency Framework Model (EIIF, 2009).....	34
Figura 4.	Capela Santo Antônio em Nova Friburgo - RJ (TRAGÉDIA, 2011)	41
Figura 5.	Acesso à rua Adelino Pereira Valente em Nova Friburgo - RJ (TRAGÉDIA, 2011)	41
Figura 6.	Rua Adelino Pereira Valente em Nova Friburgo - RJ (TRAGÉDIA, 2011).....	42
Figura 7.	Visão geral do processo da proposta de solução	77
Figura 8.	Fluxograma de atuação da abordagem no cenário de emergência	81
Figura 9.	Arquitetura da solução proposta.....	83
Figura 10.	Tratamento de dados e processamento da relevância social.....	84
Figura 11.	Componentes e módulos da ferramenta de solução proposta.....	93
Figura 12.	Processos e atividades do Servidor Web	95
Figura 13.	Arquitetura do Servidor Web	97
Figura 14.	Protótipo da tela de resposta do servidor comando e controle	98
Figura 15.	Resposta XML da API do Google Weather	102
Figura 16.	Resposta XML de uma requisição ao serviço do Climatempo.....	103
Figura 17.	Atividades e serviços do Servidor Comando e Controle.....	104
Figura 18.	Arquitetura do Servidor Comando e Controle.....	105
Figura 19.	Atividades e processos do Cliente Móvel	106
Figura 20.	Arquitetura do Cliente Móvel.....	108
Figura 21.	Distribuição de uso das diferentes versões, 01 de junho de 2012 (DEVELOPERS, 2012)	111
Figura 22.	O funcionamento da pilha de retorno – <i>Back Stack</i> (ANDROID, 2012a).....	120
Figura 23.	Trecho do arquivo de manifesto onde é declarada a atividade principal de uma aplicação (ANDROID, 2012a)	121
Figura 24.	Métodos de retorno fundamentais do ciclo de vida de uma atividade (ANDROID, 2012a)	123
Figura 25.	Ciclo de vida de uma atividade (ANDROID, 2012a).....	125
Figura 26.	Declaração de um serviço no arquivo de manifesto do aplicativo (ANDROID, 2012a)	128
Figura 27.	Métodos de retorno do ciclo de vida de um serviço (ANDROID, 2012a)	130
Figura 28.	Ciclo de vida de um serviço (ANDROID, 2012a).....	131
Figura 29.	Trecho do arquivo de manifesto de uma aplicação solicitando acesso de leitura ao provedor de conteúdo do dicionário do usuário (ANDROID, 2012a).....	134
Figura 30.	Estrutura de tabelas do provedor de conteúdo dos contatos (ANDROID, 2012a)	135
Figura 31.	Relações entre as tabelas de contatos, “contatos puros” e detalhes (ANDROID, 2012a)	138
Figura 32.	Fluxo de dados do provedor de conteúdo dos contatos (ANDROID, 2012a) ..	139
Figura 33.	Interação entre os três elementos principais do Social Design.....	143
Figura 34.	Conceito do Open Graph (FACEBOOK DEVELOPERS, 2012d)	144
Figura 35.	Multi-query em FQL (FACEBOOK DEVELOPERS, 2012c).....	147
Figura 36.	Módulos do protótipo e as interações entre eles.....	150
Figura 37.	O módulo Social Emergency Analytics e suas atividades.....	151
Figura 38.	O módulo Mobile Device Data Colector e suas atividades	153

Figura 39.	O módulo Facebook Data Colector e suas atividades	155
Figura 40.	Paginação e verificação de visibilidade dos resultados no Facebook API (FACEBOOK DEVELOPERS, 2011)	158
Figura 41.	Esquema de utilização do Google Person Finder (GOOGLE, 2010b).....	162
Figura 42.	Fluxo de trabalho em um cenário de emergência atual	170
Figura 43.	Fluxo de trabalho de nossa proposta de solução antes de um cenário de emergência	173
Figura 44.	Fluxo de trabalho de nossa proposta de solução durante um cenário de emergência	175
Figura 45.	Fluxo de trabalho alternativo de nossa proposta de solução durante um cenário de emergência.....	176
Figura 46.	Tela de pesquisa do Servidor Comando e Controle	177
Figura 47.	Exemplo de resposta de um Cliente Móvel.....	182

Lista de Quadros

Quadro 1.	Variáveis relativas às características da rede social	27
Quadro 2.	Variáveis relacionadas à distância social.....	85
Quadro 3.	Variáveis relacionadas à distância física	88
Quadro 4.	Diferentes versões do Android e distribuição de uso (DEVELOPERS, 2012)	110
Quadro 5.	Amostra da tabela do dicionário do usuário (ANDROID, 2012a)	133
Quadro 6.	Interação social do usuário com os contatos	179
Quadro 7.	Variáveis relativas à heurística de recomendação de pessoas	186
Quadro 8.	Variáveis de aderência da solução proposta ao trabalho real	188

Lista de Siglas

3G	<i>Terceira geração de padrões e tecnologias de telefonia móvel</i>
AIDL	<i>Android Interface Definition Language</i>
API	<i>Application Programming Interface</i>
ARM	<i>Advanced RISC Machine, originalmente Acorn RISC Machine</i>
CDMA	<i>Code Division Multiple Access</i>
CPU	<i>Central Processing Unit</i>
EDGE	<i>Enhanced Data rates for GSM Evolution ou Enhanced Data rates for Global Evolution</i>
EVDO	<i>Evolution-Data Optimized ou Evolution-Data Only</i>
FEMA	<i>Federal Emergency Management Agency</i>
FQL	<i>Facebook Query Language</i>
GPS	<i>Global Positioning System</i>
GSM	<i>Global System for Mobile Communications, originalmente Groupe Spécial Mobile</i>
HDD	<i>Hard Disk Drive</i>
HTTP	<i>HyperText Transfer Protocol</i>
HTTPS	<i>HyperText Transfer Protocol Secure</i>
IDEN	<i>Integrated Digital Enhanced Network</i>
IFRC	<i>International Federation of Red Cross and Red Crescent Societies</i>
IPC	<i>Inter-Process Communication</i>
J2ME	<i>Java 2 Platform, Micro Edition</i>
JSON	<i>JavaScript Object Notation</i>
LTE	<i>Long Term Evolution</i>
MMS	<i>Multimedia Messaging Service</i>
NFC	<i>Near Field Communication</i>
ONU	<i>Organização das Nações Unidas</i>
OpenGL ES	<i>OpenGL for Embedded Systems</i>
PC	<i>Personal Computer</i>
PDA	<i>Personal Digital Assistant</i>
PFIF	<i>People Finder Interchange Format</i>
RAM	<i>Random Access Memory</i>

RDF	<i>Resource Description Framework</i>
RPC	<i>Remote Procedure Call</i>
RSS	<i>Rich Site Summary, RDF Site Summary ou Really Simple Syndication</i>
SDK	<i>Software Development Kit</i>
SEDEC	<i>Secretaria de Defesa Civil</i>
SGML	<i>Standard Generic Markup Language</i>
SMS	<i>Short Message Service</i>
SQL	<i>Structured Query Language</i>
UML	<i>Unified Modeling Language</i>
UMTS	<i>Universal Mobile Telecommunications System</i>
URI	<i>Uniform Resource Identifier</i>
URL	<i>Uniform Resource Locator</i>
VGA	<i>Video Graphics Array</i>
VoIP	<i>Voice over Internet Protocol</i>
XML	<i>Extensible Markup Language</i>
WiMAX	<i>Worldwide Interoperability for Microwave Access</i>
WLAN	<i>Wireless Local Area Network</i>

SUMÁRIO

Capítulo 1 – Introdução	17
1.1 Motivação	17
1.2 Contextualização.....	22
1.3 Caracterização do Problema	24
1.4 Hipótese	25
1.5 Objetivos da Pesquisa e Enfoque da Solução	26
1.6 Organização do Documento.....	28
Capítulo 2 - Gestão de Emergências	29
2.1 Conceitos e Classificações de Emergência.....	30
2.1.1 Classificação dos desastres quanto à origem	31
2.1.2 Classificação dos desastres quanto à evolução	32
2.1.3 Classificação dos desastres quanto à intensidade do dano	33
2.2 Processos da Gestão de Emergências	33
2.2.1 Mitigação	34
2.2.2 Preparação	35
2.2.3 Resposta	35
2.2.4 Recuperação	38
2.3 Princípios da Gestão de Emergências.....	38
2.4 Abordagens de apoio à Gestão de Emergências	40
2.4.1 Abordagens de apoio à equipe de operação	42
2.4.2 Abordagens de apoio ao comando	43
2.4.3 Abordagens de apoio ao comando e às equipes de operação	45
Capítulo 3 - Sistemas de Recomendação	47
3.1 Tipos de Recomendação	49
3.1.1 Filtros colaborativos — <i>Collaborative Filtering</i>	50
3.1.2 Filtros baseados em conteúdo — <i>Content-based Filtering</i>	52
3.1.3 Filtros híbridos — <i>Hybrid Recommenders</i>	53
3.2 Desafios e Limitações	55
3.2.1 <i>Cold-Start</i>	55
3.2.2 Escalabilidade	56
3.2.3 Esparsidade	56
3.2.4 Fraude	57
3.2.5 Superespecialização	57
3.3 Sistemas de Recomendação Social	58
Capítulo 4 - Sistemas Baseados em Contexto nos Dispositivos Móveis, as Redes Sociais e o Cenário de Emergência	62
4.1 Sistemas Baseados em Contexto nos Dispositivos Móveis	67
4.2 Sistemas Baseados em Contexto nas Mídias Sociais.....	71
Capítulo 5 - Proposta.....	77
5.1 Organizações e o Cenário de Emergência	78
5.2 Descrição	82
5.2.1 Distância Social	84
5.2.2 Distância Física	87
5.2.3 Informação Contextual	88

5.3	Privacidade e Segurança	89
5.4	Arquitetura	92
5.4.1	Servidor Web	93
5.4.2	Servidor Comando e Controle	97
5.4.3	Cliente Móvel	105
5.5	Implementação	108
5.5.1	Android SDK	111
5.5.1.1	Atividades.....	119
5.5.1.2	Serviços	125
5.5.1.3	Provedores de Conteúdo.....	132
5.5.2	Facebook SDK	141
5.5.3	Protótipo	148
5.5.3.1	Social Emergency Analytics	150
5.5.3.2	Mobile Device Data Colector.....	152
5.5.3.3	Facebook Data Colector	155
5.6	Trabalhos Correlatos	159
5.6.1	Área de Busca por Informações sobre Vítimas de Emergência	160
5.6.2	Área de Dispositivos Móveis, Redes Sociais e Colaboração	164
Capítulo 6 – Ilustração de Uso		168
6.1	Cenário de Emergência Típico Atual.....	169
6.2	Utilização da Solução Proposta no Cenário de Emergência.....	170
6.3	Exemplo Prático Baseado no Protótipo Construído	176
6.4	Avaliação das Recomendações	183
6.5	Experimentações Futuras	184
6.6	Plano de Testes	185
6.6.1	Cenários de Teste para o Grupo dos Usuários Comuns	189
6.6.2	Cenário de Teste para o Grupo dos Usuários Avançados	191
Capítulo 7 - Conclusões e Trabalhos Futuros.....		193
7.1	Contribuições	196
7.2	Problemas Encontrados e Limitações	197
7.3	Trabalhos Futuros	200
Referências		203

Capítulo 1 – Introdução

Atualmente, os dispositivos móveis são de extrema utilidade em nossa sociedade. É muito raro encontrarmos hoje em dia alguma pessoa que não possua pelo menos um simples celular. Os dispositivos móveis se encontram tão amplamente difundidos em nossa sociedade que, segundo Ito (2004), “não ter um *keitai* (celular) é estar andando às cegas, desligado das informações *just-in-time* sobre onde e quando você está nas redes sociais de tempo e lugar”.

A sensibilidade ao contexto é uma das tendências crescentes da tecnologia móvel futura e devido aos avanços no desenvolvimento de tecnologia, novos conceitos de aplicação e serviços estão sendo desenvolvidos e demonstrados de forma sempre crescente. Muitos dos componentes necessários para se projetar e construir aplicações e serviços sensíveis ao contexto estão agora na fase em que eles podem ser utilizados na prática em qualquer domínio de tecnologia móvel — por exemplo, atualmente os celulares possuem GPS integrado, acelerômetros e outras funcionalidades sensoriais. A sensibilidade ao contexto pode permitir usos mais eficientes de aplicações e serviços móveis (por exemplo, oferecendo atalhos para informações situacionalmente relevantes) e também possui um grande potencial em facilitar a interação social e a colaboração (HÄKKILÄ et al., 2009).

Portanto, a fim de conseguirmos construir uma solução em dispositivo móvel que possa atuar no cenário de emergências, inicialmente devemos pesquisar trabalhos que foquem no estudo das informações contextuais, como capturá-las, processá-las e externalizá-las para que possam ser utilizadas na resolução de problemas intrínsecos a determinado cenário. Então estudamos formas de recomendação social, formação de times e técnicas para identificar potenciais parceiros para oferecer uma maior facilitação de “*collaborative matchmaking*” que utilizem essas informações contextuais na hora de recomendar pessoas para prestar informações confiáveis e íntegras sobre uma vítima de um cenário de emergência.

As seções a seguir estão encarregadas de clarificar mais a motivação deste trabalho, o problema que visamos resolver, as hipóteses levantadas durante o estudo desse problema, os principais objetivos dessa pesquisa e qual o enfoque da solução visionada.

1.1 Motivação

A ampla utilização de dispositivos móveis que ocorre atualmente na sociedade sugere o aproveitamento desses recursos na resolução de problemas e situações repentinas ou inesperadas que evoluem com o passar do tempo, podendo inclusive colocar vidas ou

propriedades em perigo e requerendo uma resposta imediata através do uso de procedimentos e recursos rotineiros coordenados por uma comunidade específica.

Para se resolver esses problemas dinâmicos¹, o tipo mais pertinente de informação a ser avaliada é a contextual, ou seja, aquela que está diretamente ligada ao contexto do problema. Por exemplo, informações que possam indicar como o problema começou, a ordem cronológica dos eventos resultantes, onde as pessoas envolvidas no cenário se encontravam nos momentos que se sucederam, etc. Nesse sentido, os dispositivos móveis, inclusive os celulares, possuem informações que carregamos conosco o tempo inteiro e que podem auxiliar na representação do contexto onde estamos inseridos: localização posicional (coordenadas GPS), fotos, vídeos, áudios, mensagens (SMS ou MMS), últimas ligações (realizadas ou recebidas), arquivos em geral, etc.

Uma pesquisa sobre *internet* móvel (MOBILE, 2009) atualizada em abril de 2011 nos mostra incríveis números sobre o crescimento e a utilização de dispositivos móveis ao redor do mundo. Ela nos mostra que a *internet* móvel está crescendo mais rapidamente e será maior do que a *internet* convencional, devido a cinco tecnologias convergentes e tendências de adoção social: 3G, redes sociais, vídeo, VoIP e impressionantes dispositivos móveis. Também verificamos que em 2010 os dispositivos móveis excederam a quantidade de 10 bilhões de unidades, o que em números da época chegavam a quase dois dispositivos móveis por habitante do planeta Terra. A penetração de assinantes 3G foi superior a 20% em 2010 e crescerá a mais de 40% até 2014, concentrados em nações desenvolvidas. Embora esse crescimento do 3G seja substancial, outras tecnologias sem fio (como o GPS, Wi-Fi e Bluetooth) estão crescendo na mesma velocidade ou ainda mais rápido do que 3G. Esse crescimento geral no uso de dispositivos móveis é impulsionado pelo fato de que cada inovação da tecnologia sem fio abaixa o custo por bit de dados móvel.

Um compêndio de estudos (MOBI, 2012) mostra o estado mais atual dos dispositivos móveis em todo o mundo, indicando que ao final de 2011 havia 6 bilhões de assinantes móveis (87% da população mundial), o que representa um enorme aumento de assinantes móveis em relação a 2010 (5,4 bilhões) e 2009 (4,7 bilhões). O crescimento nesse segmento de mercado é impulsionado pela demanda dos países em desenvolvimento, liderado pela rápida adoção móvel na China e na Índia, os países mais populosos do mundo. A previsão é de que os assinantes de telefonia móvel em todo o mundo chegarão a 6,5 bilhões até o final de 2012, 6,9 bilhões até o final de 2013 e 8 bilhões até o final de 2016. A impressionante marca

¹ Problemas que evoluem rapidamente com o passar do tempo, podendo atingir grandes proporções e com graves consequências à sociedade de um modo geral.

de 9 bilhões de assinaturas móveis deverá ser atingida em 2017, dos quais 5 bilhões serão conexões de banda larga móvel.

Além disso, a venda de dispositivos móveis cresceu em 2011 (11,1% em relação a 2010), com os *smartphones* mostrando um crescimento mais forte. A Nokia continua a ser a principal fabricante de dispositivos móveis (*smartphones* e celulares), com um *market share* variando entre 23,8% e 27,0% (dependendo da fonte de pesquisa). Em seguida vem a Samsung (*market share* entre 17,7% e 21,3%) e, em terceiro lugar, a Apple (*market share* entre 5,0% e 6,0%). Analisando somente a venda global de *smartphones* em 2011, as principais fabricantes desse tipo de dispositivo móvel são: a Samsung, com um *market share* variando entre 19,1% e 19,9% (dependendo da fonte de pesquisa), seguida pela Apple (*market share* de 19,0%) e, em terceiro lugar, a Nokia (*market share* entre 15,7% e 15,8%). A Nokia, até o início de 2011, também era a líder mundial em vendas de *smartphones*. Porém, sua surpreendente decisão de abandonar o sistema operacional Symbian em 2011 causou uma queda vertiginosa na venda de seus *smartphones* e no uso desse sistema operacional.

Atualmente, o mercado de sistemas operacionais para dispositivos móveis é liderado pelo Android (*market share* 2011 igual a 48,8%), seguido pelo iOS (19,1%) e, em terceiro lugar, o Symbian (16,4%), seguido pelo BlackBerry (10,5%), que possivelmente assumirá o posto de terceiro maior sistema operacional móvel utilizado mundialmente dentro de alguns meses. Em relação ao uso de *internet* móvel, esse estudo revela que 1,2 bilhão de pessoas em todo o mundo (17% da população mundial) acessou a *internet* móvel em 2011. As assinaturas de banda larga móvel cresceram 45% ao ano nos últimos quatro anos. Isso significa que, atualmente, para cada assinatura de banda larga fixa existem duas assinaturas de banda larga móvel. Nos países em desenvolvimento, a banda larga móvel é frequentemente o único método de acesso à *internet* disponível para muitas pessoas. A previsão é de que, mantido esse ritmo de crescimento, em 2017 haverá 5 bilhões de assinaturas de banda larga móvel.

Ainda, vemos que muitos usuários de *internet* móvel são *mobile-only*, ou seja, não usam, ou muito raramente usam um *desktop*, *laptop* ou *tablet* para acessar a *Web*. Mesmo nos países desenvolvidos existe uma considerável minoria dos usuários da *Web* móvel que são *mobile-only*, como nos EUA, por exemplo, onde 25% dos usuários da *Web* móvel são *mobile-only*. Já nos países em desenvolvimento, a maioria dos usuários da *Web* móvel são *mobile-only*, como no Egito (70% dos usuários da *Web* móvel são *mobile-only*) ou na Índia (59% dos usuários da *Web* móvel são *mobile-only*), por exemplo. Por fim, o estudo revela que em 2011 mais de 85% dos novos aparelhos são capazes de acessar a *Web* móvel, independentemente de serem *smartphones* ou celulares convencionais.

Em relação às trocas de mensagens em dispositivos móveis, esse estudo revela que 669,5 milhões de pessoas usaram *e-mail* móvel em 2011, e esse número deverá crescer para 2,4 bilhões até 2016. Mensageiros instantâneos móveis são usados por 7,9% dos assinantes de telefonia móvel, e esse uso deve subir para 30,9% em 2016. O tráfego global de mensagens instantâneas móveis vai aumentar de 1,6 trilhões de mensagens em 2011 para 7,7 trilhões de mensagens em 2016, duplicando a sua quota de tráfego de mensagens globais de 17,1% em 2011 para 34,6% em 2016. Vemos também que 207 bilhões de MMS's foram enviados em 2011. Em 2016, 387,5 bilhões de MMS's serão enviados, o que representará 1,7% do tráfego de mensagens globais.

Como uma menção digna de nota desse estudo, vale ressaltar que, apesar de toda a tecnologia disponível nos mais recentes dispositivos móveis, o SMS continua sendo o rei das mensagens móveis e permanecerá assim por um bom tempo. Em 2011, 7,8 trilhões de mensagens SMS foram enviadas. O tráfego de SMS deve chegar a 9,6 trilhões em 2012. Uma consulta realizada pela Anatel (Agência Nacional de Telecomunicações) revelou um relatório indicando que o Brasil atingiu a marca impressionante de 256 milhões de celulares habilitados em julho de 2012. Desse total, 81,49% são celulares pré-pagos (TELECO, 2012).

Por outro lado, também temos o crescente uso das mídias sociais, que permite a produção de conteúdos de forma descentralizada, ou seja, as informações são procedentes de diversas fontes espalhadas pelo mundo, e sem o controle editorial de grandes grupos. Dessa forma, diariamente temos a produção de conteúdos diversificados da parte dos próprios usuários para todas as pessoas que acessarem determinada mídia social. As mídias sociais são um grupo de aplicações para *internet* construídas com base nos fundamentos ideológicos e tecnológicos da *Web 2.0*, e que permitem a criação e troca de conteúdo gerado pelo utilizador (KAPLAN; HAENLEIN, 2010). Elas podem vir em diferentes formatos como *blogs* (*weblogs* ou *microblogging*), *videologs*, redes sociais, *scrapbooks*, compartilhamento (de fotos, vídeos, músicas ou conhecimento), *crowdsourcing*, realidade virtual, grupos de discussão, entre outros. Assim sendo, através do simples uso de indexadores de conteúdo da *internet*, como o Google Search (GOOGLE, 1997), por exemplo, temos à nossa disposição uma rica gama de informações sobre determinada pessoa, com facilidade e rapidez.

Pesquisas recentes na área das mídias sociais revelaram estatísticas interessantes sobre a utilização delas no Brasil e no mundo. Com base em uma dessas pesquisas (ARNO, 2012), destacamos que, ainda em 2012, haverá 1,43 bilhões de usuários em redes sociais (um quinto da população total do mundo), o que representa um aumento de 19,2% sobre os números de 2011. Há sinais de que alguns países desenvolvidos podem estar se aproximando de seus

pontos de saturação, mas com as taxas de penetração da *internet* ainda em ascensão em grandes mercados emergentes como a Índia, a China e a América do Sul (principalmente o Brasil), ainda há muito espaço para crescimento. Nesses países, muitas pessoas acessam as mídias sociais através de dispositivos móveis. Mas essa tendência não se limita exclusivamente às nações em desenvolvimento. Um relatório da ComScore revelou que 72 milhões de norte americanos usaram as mídias sociais através de dispositivos móveis em 2011, o que representa um aumento de 37% em relação ao ano anterior.

Embora haja um número crescente de plataformas de mídia social disponíveis, essa pesquisa revela que os usuários ativos de *internet* não estão mais criando novos perfis. Isso sugere que os usuários tendem a ficar com as plataformas a que estão familiarizados e que eles estão gastando mais tempo do que nunca em seus perfis existentes. Na verdade, a pesquisa sugere que as pessoas entre 16 e 24 anos de idade estão passando tanto tempo em sites de mídia social quanto assistindo à televisão, e muito mais tempo do que eles costumam ouvir rádio ou ler revistas e jornais. O número de contatos sociais mantidos também continua a crescer. Isso se refere às pessoas com quem os usuários regularmente mantêm contato, em vez de apenas nomes em uma lista de amigos.

Um estudo sobre o crescimento das mídias sociais (MBA, 2012) aponta os seguintes números sobre os usuários de *internet* dos Estados Unidos da América: 66% dos usuários adultos *online* participam de uma ou mais plataformas de mídia social. O uso dessas plataformas e ferramentas tem aumentado incrivelmente nos últimos 10 anos, tanto por motivos pessoais ou de negócios. A mesma pesquisa mostra ainda que o Facebook (FACEBOOK, 2004), líder mundial em usuários com um bilhão de usuários ativos em todo o mundo (FACEBOOK NEWSROOM, 2012), possui em média 200 milhões de visitantes móveis diariamente. Os usuários do Facebook acessam o *site*, em média, 40 vezes por mês e, em cada uma dessas visitas, gastam em média aproximadamente 23 minutos.

Esse mesmo estudo ainda revela que o Twitter (TWITTER, 2006) possui 127 milhões de usuários ativos, dos quais 54% acessam o *site* via dispositivos móveis. Os usuários do Twitter gastam, em média, aproximadamente 12 minutos no *site* a cada visita e 36% dos usuários fazem pelo menos um “*tweet*” (forma como as pessoas disponibilizam informações) por dia. Por fim, esse estudo nos mostra que a grande maioria dos norte americanos utilizam as mídias sociais com a finalidade de se manterem conectados com os amigos (67%) ou com a família (64%), ou ainda se reconectarem com velhos amigos (50%).

Outra coletânea de estatísticas atuais sobre as mídias sociais (BULLAS, 2012) indica que no Facebook existem 100 bilhões de conexões e, a cada dia, 250 milhões de fotos são

inseridas e 2,7 bilhões de “likes” (interações entre os usuários) são dados. Ainda, 20% de todas as visualizações de páginas na *web* são no Facebook. Em relação aos seus usuários ao redor do mundo, citando apenas as regiões onde o Facebook é mais acessado, vemos que 50,3% dos norte americanos participam do Facebook, seguidos de 37,7% de toda a população da Oceania, 27,5% dos europeus e 25,5% da população da América Latina. Em relação ao Twitter vemos que, diariamente, 175 milhões de *tweets* são feitos e um milhão de contas são criadas. Os três países com mais usuários no Twitter são os EUA (107 milhões), o Brasil (33 milhões) e o Japão (quase 30 milhões).

1.2 Contextualização

Para auxiliar no entendimento do problema, pensemos no cenário de emergência real que aconteceu em Niterói há mais de dois anos, em abril de 2010. O Morro do Bumba, na região do Cubango, sofreu um grave deslizamento de terra, como consequência das fortes chuvas que assolaram a cidade no mesmo período (G1, 2010). Mais de 200 pessoas foram soterradas e a equipe de resposta a emergências tentava encontrar onde elas estavam para poder resgatá-las. Esse trabalho de obtenção das informações sobre as vítimas da emergência não foi nada fácil, pois no meio da situação caótica instaurada pelo deslizamento de terra, a equipe de resposta a emergências tinha que encontrar pessoas que tivessem presenciado o desastre, ou que soubessem o que havia ali antes do soterramento, e que estivessem dispostas a disponibilizar informações importantes sobre as vítimas para o processo de resgate.

A Figura 1 mostra uma visão frontal desse deslizamento de terra ocorrido. Através dela podemos ter uma noção da magnitude do desastre e da dificuldade de acesso à área onde ocorreu a emergência. Podemos notar que uma grande parte do morro foi afetada, levando consigo as casas que ali estavam durante o desabamento. Já a Figura 2 mostra uma visão aérea do local da emergência. Através dela podemos ter uma noção mais completa da extensão da área afetada e da situação do soterramento das casas e do desgaste do terreno em questão.

Todo esse trabalho manual que a equipe de emergência experimentou para obter informações sobre as vítimas poderia ser automatizado, visando melhorar a agilidade nesse processo de obtenção de informações sobre as vítimas de uma emergência. Então, partindo do princípio da Engenharia Social, onde todas as recomendações são realizadas ativamente, sem o auxílio de recursos computacionais avançados — as pessoas procuram pares entrando em contato de maneira convencional, utilizando-se de conversas ao telefone, *e-mail*, sala de *chats*, mensageiros instantâneos ou até mesmo presencialmente —, possuímos uma tendência a fornecer uma maior agilidade a esse processo por meio do apoio da computação móvel.



Figura 1. Vista geral do deslizamento no morro do Bumba, em Niterói - RJ (UOL, 2010)



Figura 2. Vista aérea de uma cratera formada no morro do Bumba, em Niterói - RJ (R7, 2011)

Assim, realizando a troca de informações e recomendações de pessoas de forma passiva (ou proativa) — sem a necessidade do disparo do evento inicial pelo ser humano —, o processo de colaboração na resolução de problemas impostos por determinados cenários críticos seria facilitado. Dessa forma, poder-se-ia focar na troca de informações presentes nos dispositivos móveis da população que estivesse no local naquele momento e na análise contextual da rede social da vítima, visando à recomendação de pessoas para auxiliar de diversas formas na emergência que estava ocorrendo, seja prestando informações sobre o local e pessoas desaparecidas, ou até mesmo ajudando no resgate dessas.

Portanto, identificando as pessoas mais qualificadas para auxiliar na resolução do problema, tais como familiares ou amigos de pessoas desaparecidas (para auxiliar no resgate dos envolvidos), médicos ou enfermeiros transeuntes (para cuidar dos feridos, prestando os primeiros socorros), moradores locais (para dar informações sobre a área afetada pelo problema) e afins, poderíamos recomendar essas pessoas à equipe de emergência, a fim de que elas colaborassem com informações confiáveis e íntegras sobre as vítimas do acidente.

1.3 Caracterização do Problema

Em qualquer processo diário de nossas vidas, a informação é identificada cada vez mais como um recurso essencial para a tomada de decisões. Dada à importância que a informação tem assumido atualmente em todos os setores da sociedade global, é natural e aceitável que se levantem preocupações relativas à sua qualidade. Deveras, uma informação incorreta ou desatualizada pode levar a uma decisão errônea, produzindo assim um impacto negativo durante, por exemplo, a fase de resposta a uma emergência. De acordo com Oliveira e Amaral (1999),

Todas as decisões que se tomam são baseadas num conjunto de informação que está disponível no processo de tomada de decisão, algo que faz depender a decisão final das características da informação que lhe serviu de *input*. Informação com insuficiente qualidade não produzirá uma decisão adequada que, quando aplicada, produza os resultados esperados.

Dessa forma, vemos que para obtermos os resultados esperados durante o processo de tomada de decisões, principalmente quando aplicado à fase de resposta a uma emergência, é fundamental termos amplo acesso a informações confiáveis e íntegras. A questão que nos falta responder agora é como conseguiremos esse acesso às informações com qualidade, para aplicarmos como entrada em tomadas de decisões críticas, como aquelas que são necessárias na resposta a emergências? Em tese, as informações de melhor qualidade deverão vir das pessoas que estavam no local onde e quando ocorreu o desastre, mais especificamente dos moradores ou frequentadores assíduos do local, caso esses tenham presenciado o desastre e estejam em condições de prestar maiores informações sobre o ocorrido.

Então, a questão crítica que visamos atacar é a ausência de informações confiáveis e íntegras sobre eventos ou objetos, que possam auxiliar no processo de tomada de decisão. No caso das vítimas em determinadas situações de emergência, tais informações poderiam vir de pessoas que possuem algum convívio com elas e, portanto, para sua obtenção necessitaremos abordar o problema de detectar e recomendar pessoas, baseados em seus perfis sociais, para colaborar na resolução de um determinado cenário-problema, utilizando informações contextuais obtidas pelos indivíduos ou especialistas envolvidos nesse cenário. O problema de

detecção e recomendação de pessoas nesse caso não depende somente de perfis sociais, que podem se enquadrar como objetos mais estáticos nessa análise, mas também do contexto onde essa rede social está inserida.

A questão contextual depende de objetos mais dinâmicos que estão inseridos nessa análise, tais como onde estão localizadas as pessoas participantes dessa rede social, quando ocorreu a última interação entre essas pessoas e qual a intensidade dessa interação, além de ingredientes contextuais específicos, assumidos durante determinados instantes, e que representam atividades usuais ou informações sobre o ambiente em torno do qual está inserida a questão – como, por exemplo, qual seria a melhor trajetória para pessoa que prestará as informações chegar até o local da emergência (e nesse caso deveremos analisar o trânsito e, por conseguinte, a condição climática dos locais de origem e destino, pois esta afeta diretamente o tráfego entre esses pontos).

Outra questão a ser tratada quando nos deparamos com problemas desse tipo é a sinergia entre os nós dessa rede social. Nesse caso, o contexto se refere aos valores de certas características específicas dessa rede social, pois a qualidade da informação (confiabilidade e integridade) dependerá diretamente da importância que determinada pessoa tem para a vítima. Ou seja, a fim de recomendarmos a pessoa correta para colaborar nesse cenário de emergência, deveremos analisar os aspectos dos tipos de relação da pessoa nessa rede social, o peso dessas relações, a quantidade de conexões dessa pessoa com a vítima e, principalmente, qual a distância da relação (grau de separação) entre a pessoa e a vítima.

1.4 Hipótese

Nosso problema, conforme citado anteriormente, é a obtenção de informações confiáveis e íntegras sobre eventos ou objetos, que possam auxiliar no processo de tomada de decisão. Analisemos o caso específico de uma vítima de um determinado cenário de emergência, que não esteja consciente ou que esteja desaparecida. Como poderíamos obter informações precisas sobre essa vítima? Quais são os maiores candidatos a conhecer a vítima tão bem a ponto de oferecer um *background* sobre ela mais próximo da realidade possível? Os vizinhos? Colegas de trabalho? Amigos de infância? Parentes ou familiares? É bem provável e possível que quanto mais próximas as pessoas forem da vítima, quanto mais tempo passarem com ela, quanto mais íntimas essas pessoas forem da vítima, mais confiáveis, corretas e íntegras as informações dessas pessoas sobre a vítima serão.

Portanto, baseado nessa inferência da rede social de uma vítima de emergência, construímos a seguinte hipótese para a confecção deste trabalho:

A identificação da distância social entre duas pessoas a partir de suas interações nas mídias sociais e dispositivos móveis, acrescida de informações contextuais sobre os locais e os eventos em que essas pessoas e suas redes sociais estão inseridas, pode melhorar a recomendação de pessoas a fim de disponibilizar informações confiáveis e íntegras para auxiliar em um processo de tomada de decisão.

1.5 Objetivos da Pesquisa e Enfoque da Solução

Nossa pesquisa tem por objetivo a melhoria no processo de recomendação de pessoas através da análise contextual de sua rede social, bem como as relações e interações sociais entre essas pessoas, além de inspecionar aspectos inerentes à localidade delas. Esse processo de recomendação verifica a rede social de um indivíduo em busca de pessoas próximas o suficiente para prestarem informações confiáveis e íntegras sobre ele. Em adição à análise contextual dessa rede social, são utilizadas as informações contidas nos dispositivos móveis das pessoas e informações disponibilizadas através das mídias sociais na *internet*.

O experimento do mundo pequeno (MILGRAM, 1967) procurou examinar o comprimento médio do caminho entre as pessoas contidas em uma determinada rede social nos Estados Unidos. Essa pesquisa foi baseada na sugestão de que a sociedade humana é uma rede de conexões típicas de um mundo pequeno, caracterizada por caminhos de comprimentos curtos na conexão entre as pessoas dessa rede. O resultado desse trabalho nos mostrou que o mundo está cada vez mais interconectado ao afirmar que somente cinco intermediários bastariam, em média, para conectar dois indivíduos escolhidos aleatoriamente, independentemente de onde eles vivam nos Estados Unidos.

Esse trabalho ganhou grande notoriedade mundial, apesar de gerar certas controvérsias no mundo científico, e foi associado ao termo “seis graus de separação” cunhado em 1929 originalmente por Frigyes Karinthy² e popularizado através de uma peça homônima de 1990 escrita por John Guare³, para se referenciar à ideia de que todo mundo está, em média, aproximadamente seis passos, por vias de apresentação, de qualquer outra pessoa no planeta. Ou seja, uma cadeia do tipo “amigo de um amigo” poderia ser criada e, em média, seriam necessários seis passos para se conectar quaisquer duas pessoas no mundo.

² Frigyes Karinthy (25 de Junho de 1887 – 29 de Agosto de 1938), nascido na Hungria, foi escritor, dramaturgo, poeta, jornalista e tradutor. Ele foi o primeiro proponente do conceito de “seis graus de separação”, em sua curta história de 1929 chamada “Correntes” (*Láncszemek*, no título original).

³ John Guare (nascido em 05 de Fevereiro de 1938) é um dramaturgo americano. Ele é o autor da peça “Seis Graus de Separação” (1990), que explora a premissa existencial que todas as pessoas no mundo estão ligadas a todas as outras no mundo por uma cadeia de não mais que seis conhecidos.

Então, o conceito de distância social incorporado pelo presente trabalho, além de ser inspirado pela “Teoria do Mundo Pequeno”, também é encorpado pela avaliação das variáveis contextuais presentes na relação entre os indivíduos da rede social que são apresentadas no Quadro 1 abaixo.

Variável	Significado
Tipo da Relação	Amigo, colega, conhecido, parente, parceiro, etc.
Peso da Relação	Importância, grau de afinidade, proximidade, etc.
Quantidade de Conexões	Quantidade de tipos de relações distintas entre duas pessoas.

Quadro 1. Variáveis relativas às características da rede social

Inicialmente, essa abordagem móvel de recomendação de pessoas será vetorizada através da utilização de celulares, por ser atualmente o meio de comunicação e troca de informações mais utilizado mundialmente. O ferramental construído poderá auxiliar, por exemplo, em situações de emergência, na fase de resposta, visando amenizar as principais dificuldades encontradas (por exemplo, a ausência de dados estruturados) e atuando fortemente nas características dessa fase (por exemplo, fornecendo rapidez na solução).

Tomando a área de gestão de emergências como exemplo de domínio de aplicação, durante a fase de resposta, a distância física é baseada no trajeto da pessoa até o local da emergência, ou seja, é o caminho pelo qual cada uma das pessoas recomendadas deverá seguir até chegar ao local da emergência. Dessa forma, existem certas informações contextuais que deverão ser analisadas para que a recomendação possa sugerir pessoas que, além de estarem o mais próximo possível do local da emergência, possam chegar lá mais rapidamente. Então, apesar da distância física entre as pessoas recomendadas e o local onde ocorreu a emergência se dar basicamente através do cálculo de suas posições geográficas (coordenadas GPS), deveremos também analisar outras variáveis de contexto dinâmico como a condição climática ou o tráfego, que influenciam diretamente no trânsito das pessoas recomendadas até o local.

Portanto, o enfoque de solução desta proposta é trabalhar nos aspectos contextuais das redes sociais, utilizando-se de computação móvel para dar mais agilidade na recomendação de pessoas que consigam disponibilizar informações confiáveis e íntegras a fim de auxiliar em um processo de tomada de decisão. Quando aplicada à área de gestão de emergências, a solução proposta visaria amenizar as principais dificuldades encontradas (por exemplo, a ausência de dados estruturados) e atuaria fortemente nas características da fase de resposta a uma emergência (por exemplo, fornecendo rapidez na solução).

As informações contidas nos dispositivos móveis (tais como coordenadas, agendas, dados textuais, etc.) poderão ser aliadas a outras informações pessoais (como por exemplo, *tweets*, *torpedos*, dados pessoais, etc.) com o objetivo de ajudar na recomendação de pessoas para o fornecimento das informações necessárias para o processo de tomada de decisão. No caso de uso em um cenário de emergência, a utilização das informações contextuais existentes pelo nosso algoritmo da solução culminará com a seleção de pessoas próximas à vítima para colaborar com informações precisas sobre ela. O algoritmo utilizado vasculha a rede social da vítima, tanto através das mídias sociais quanto através das informações contidas em seu dispositivo móvel, a fim de realizar a recomendação da pessoa ideal para prestar informações sobre a vítima, ou seja, a pessoa com mais afinidade à vítima a ponto de poder prestar informações úteis e de qualidade sobre ela.

1.6 Organização do Documento

O presente trabalho está estruturado da seguinte maneira: no próximo Capítulo são discutidas as situações de emergência, suas fases e processos de trabalho, tudo isso com um foco maior na fase de resposta, que é onde nossa solução é aplicada. O Capítulo 3 faz uma revisão sobre sistemas de recomendação e os diversos tipos de recomendação existentes, além da área de formação de times que serviram como inspiração durante a elaboração da solução proposta nesta dissertação. A importância da percepção situacional nesse processo e os diversos sistemas baseados em contexto aplicados ao cenário de emergências estudados são discutidos logo no Capítulo seguinte. O Capítulo 5 descreve detalhadamente o modelo de solução proposto, apresentando a heurística do processo elaborado para a recomendação de pessoas, a arquitetura da solução e o desenvolvimento de uma ferramenta computacional que aplique essa solução proposta para apoiar a equipe de resposta à emergência na obtenção de informações de qualidade sobre as vítimas de um determinado cenário de emergência. No Capítulo seguinte é realizada a ilustração de uso da ferramenta construída, com a descrição de um caso de uso exploratório utilizando-se o protótipo construído durante o desenvolvimento deste trabalho. Encerrando a análise deste trabalho, no Capítulo 7 todo o retrospecto da pesquisa é criticado e seus produtos gerados são discutidos. As conclusões sobre as contribuições, limitações e oportunidades futuras são apresentadas nesse mesmo Capítulo.

Capítulo 2 - Gestão de Emergências

Uma emergência pode ser definida como uma “situação crítica” ou “caso de urgência” (BRASIL, 2008). Eventos desse tipo possuem graus de severidade muito variáveis e não padronizados que vão desde pequenos acidentes, cujos danos podem ser de certa forma suportáveis pela população, até catástrofes de proporções gigantescas, onde se faz necessário o apoio de organizações internacionais. Apesar de acontecimentos desse tipo serem comuns em toda parte do globo, nos países em desenvolvimento seus danos costumam ser maiores devido ao adensamento populacional, ocupação de áreas de risco e falta de infraestrutura e de investimentos na área de Gestão de Emergências (MECHLER, 2004; MIRZA, 2003).

A área de emergência, em termos gerais, é um pouco nebulosa devido à sua complexidade e falta de definição de padrões globais de termos e estruturação. Justamente por isso, há uma grande quantidade de tópicos e definições diferentes que podem ser encontrados em várias literaturas, mas, de forma geral, uma emergência pode ser definida como um evento súbito e inesperado que geralmente exige medidas imediatas para minimizar suas consequências adversas (DHA/ONU, 1992).

Várias organizações de resposta a emergências ao redor do mundo tentam evitar essa falta de definições formais no campo de emergência gerando seus próprios vocabulários padronizados, como, por exemplo, a *International Federation of Red Cross and Red Crescent Societies* (IFRC, 2012), a *Federal Emergency Management Agency* (FEMA, 2012a) e a Secretaria Nacional de Defesa Civil (BRASIL, 2012a). Essa estratégia visa ampliar e facilitar os esforços de cooperação local e internacional em lidar com grandes catástrofes. No entanto, infelizmente, não há um vocabulário único que seja o padrão global dessa área.

Como a gestão de emergências é um processo complexo que lida com informações de diferentes fontes num ambiente dinâmico e as decisões são tomadas sob pressão, torna-se necessário garantir um entendimento comum entre todos os atores envolvidos em diferentes fases para a obtenção do sucesso nesse processo. Portanto, nos últimos anos, diversos estudos (XU; ZLATANOVA, 2007; DI MAIO, 2007; LI et al., 2008; KRUCHTEN et al., 2008; TRUPTIL; BÉNABEN; PINGAUD, 2009) têm procurado demonstrar o uso de ontologias para viabilizar a interoperabilidade semântica dos sistemas envolvidos e uniformizar ou explicitar diferentes perspectivas sobre os conhecimentos dessa área.

2.1 Conceitos e Classificações de Emergência

Na gestão de emergências, especialmente na fase de resposta, diversas equipes com diferentes perspectivas e atribuições precisam colaborar tendo em vista um objetivo comum. Com o aumento na utilização de sistemas de informação distribuídos e a proliferação de interfaces públicas de base de dados, de tecnologias de mediação e de padrões de formato para troca de dados, este cenário tornou-se ainda mais complexo, ocasionando uma verdadeira sobrecarga de informações heterogêneas para os tomadores de decisão.

Ao considerarmos o enorme volume de informações disponíveis no cenário de emergências, sejam elas na forma de normas, regulamentos, bases de relatos de situações de emergência, boletins de ocorrência, dentre outras, teremos disponível um rico *corpus* de referência, com um representativo conhecimento acerca de diferentes domínios (SANTOS, 2007). Este *corpus* não serve apenas de base para o levantamento do vocabulário típico utilizado no contexto de cada domínio, mas também para a captura de importantes conceitos e associações relativas a ele (MAEDCHE; STAAB, 2000).

A Agência Federal de Gestão de Emergências dos Estados Unidos (FEMA, 2012b) define emergência como “qualquer incidente, natural ou provocado pelo homem, que requer uma ação de resposta para proteger a vida ou a propriedade”. Diversas agências envolvidas com a área de gestão de emergências ao redor do mundo adotam a clássica definição de emergência, que se refere a “um acontecimento inesperado que coloca a vida e/ou a propriedade em perigo e exige uma resposta imediata através dos recursos e procedimentos de rotina da comunidade. Exemplos: um acidente envolvendo vários automóveis com feridos ou mortos; um incêndio causado por um relâmpago que se espalha a outros edifícios” (DRABEK, 1996).

A visão da ONU para o conceito de emergência também segue a linha de definições supracitadas, uma vez que se refere a uma situação em que a normalidade já foi rompida e se restringe apenas a determinar a necessidade de se tomar “medidas imediatas para minimizar suas consequências adversas” (DHA/ONU, 1992), sem procurar delimitar a intensidade da ocorrência. Para a FEMA, porém, o termo emergência delimita os acontecimentos pela capacidade da comunidade afetada de superar os danos e prejuízos com ou sem o auxílio de recursos externos à localidade. Portanto, uma emergência é uma ocasião extraordinária que necessita de ajuda Federal, seja através de recursos financeiros ou operacionais, para suplementar os esforços do Estado e da localidade vitimada.

Analisando pela ótica empresarial/industrial, uma emergência é “qualquer evento não planejado que pode causar mortes ou ferimentos significativos para funcionários, clientes ou o público em geral, podendo fechar aquele negócio, interromper suas operações, causar danos físicos ou ambientais, e ainda ameaçam a capacidade financeira da instalação ou sua imagem pública” (FEMA, 1993).

No Brasil, segundo a Secretaria Nacional de Defesa Civil (BRASIL, 2007), os desastres são resultados de eventos adversos, naturais ou provocados pelo homem, sobre um ecossistema vulnerável, causando danos humanos, materiais e ambientais e consequentes prejuízos econômicos e sociais. A intensidade de um desastre depende da interação entre a magnitude do evento adverso e a vulnerabilidade do sistema e é quantificada em função de danos e prejuízos.

Essas ocorrências causadoras de danos e prejuízos (também chamadas de catástrofes, dependendo da intensidade do evento) alteram gravemente a ordem da sociedade, provocando um grande número de vítimas. Uma catástrofe envolve um número excepcional de vítimas e prejuízos. Já os desastres são classificados quanto à origem, evolução e intensidade de danos e prejuízos (BRASIL, 1999). A seguir mostraremos algumas classificações de emergências de acordo com essas literaturas encontradas na área.

2.1.1 Classificação dos desastres quanto à origem

Quanto a sua origem, os desastres são classificados em três tipos, a saber: naturais, humanos ou antropogênicos e mistos. Essa classificação é dada conforme o agente causador do fenômeno ou evento adverso.

Os desastres naturais são provocados por desequilíbrios na natureza e fatores de origem externa que atuam independentemente da ação direta humana. Podem ser relacionados a fenômenos siderais, meteorológicos, tectônicos ou outros fatores naturais. Abrangem, portanto, eventos como impacto de meteoritos, vendavais, chuvas de granizo, inundações, geadas, secas, ciclones, terremotos, tsunamis, erupções vulcânicas e pragas.

Os desastres provocados pelas ações ou omissões humanas dividem-se em tecnológicos, sociais e biológicos. Esses se relacionam com a atuação do próprio homem, enquanto agente e autor, podendo produzir situações capazes de gerar grandes danos à natureza, ao habitat humano e ao próprio homem, enquanto espécie. Os primeiros são consequência do adensamento populacional urbano sem infraestrutura adequada e comprometimento com a segurança, tais como problemas com produtos perigosos (químicos, radioativos, etc.), acidentes com transportes, explosões industriais etc. Mas, normalmente, os

desastres humanos são consequências de ações desajustadas geradoras de desequilíbrios no relacionamento socioeconômico e político entre os homens, o que provoca fome, banditismo, tráfico de drogas, desnutrição e violência. Já os desastres humanos de natureza biológica estão vinculados a profundas e prejudiciais alterações em seu ambiente ecológico, como interações biológicas que acabam por provocar epidemias e outras doenças contagiosas ou pragas (dengue, febre amarela, cólera, sarampo, AIDS e outras DSTs são alguns exemplos).

Os desastres de causa mista ocorrem quando as ações ou omissões humanas contribuem para intensificar, complicar ou agravar os desastres naturais. Além disso, também se caracterizam quando intercorrências de fenômenos adversos naturais, atuando sobre condições ambientais degradadas pelo homem, provocam desastres. As ocorrências desta categoria costumam afetar profundamente grandes extensões do meio ambiente e alterar o ecossistema. Desertificação, salinização, chuva ácida, efeito estufa e a redução da camada de ozônio são alguns exemplos de desastres de causa mista.

2.1.2 Classificação dos desastres quanto à evolução

De acordo com a Secretaria Nacional de Defesa Civil (2007), a evolução de desastres pode ocorrer de forma súbita (ou de evolução aguda), gradual (ou de evolução crônica) ou por soma de efeitos parciais. Os desastres súbitos ou de evolução aguda caracterizam-se pela subtaneidade, pela velocidade de evolução e, normalmente, pela violência dos eventos adversos. Podem ocorrer de forma inesperada ou serem antecidos por sinais. Este tipo de evento também pode ser sazonal, como as inundações e escorregamentos de solos. Enxurradas, vendavais, terremotos, erupções vulcânicas e chuvas de granizo são alguns exemplos desse tipo de desastre.

Os desastres de evolução crônica ou gradual, ao contrário, caracterizam-se por serem insidiosos e evoluírem através de etapas de agravamento progressivo, como, por exemplo, seca, desertificação e perda de solo para a agricultura. Já os desastres por somação de efeitos parciais são, na realidade, caracterizados pela soma de numerosos acidentes ou ocorrências, com características semelhantes e frequência constante, os quais, quando ao término de um período, definem um grande desastre, produzindo danos significativos. Pode-se citar, como exemplo, a cólera, a malária, o tráfico de drogas, acidentes de trabalho e de trânsito, entre outros.

2.1.3 Classificação dos desastres quanto à intensidade do dano

A intensidade dos desastres também pode ser definida em termos absolutos ou a partir da proporção entre as necessidades de recursos e as possibilidades dos meios disponíveis na área afetada, para dar resposta cabal ao problema. Assim sendo, eles são classificados relativamente em função da importância e intensidade dos danos e prejuízos, diferenciando-se em quatro níveis: acidentes, desastres de médio porte, desastres de grande porte e desastres de muito grande porte.

Os acidentes são caracterizados quando os danos e prejuízos consequentes são de pouca importância para a coletividade como um todo, já que, na visão individual das vítimas, qualquer desastre é de extrema importância e gravidade. Em geral são facilmente suportados e superados pela comunidade afetada como um todo. Os desastres de médio porte são caracterizados quando os danos e prejuízos, embora importantes, podem ser recuperados com os recursos disponíveis na própria área sinistrada. Os desastres de grande porte exigem o reforço dos recursos disponíveis na área sinistrada, através do aporte de recursos regionais, estaduais e, até mesmo, federais. Os desastres de muito grande porte, para garantir uma resposta eficiente e cabal recuperação, exigem a intervenção coordenada dos três níveis do Sistema Nacional de Defesa Civil — SINDEC — e, até mesmo, de ajuda externa.

A proposta do presente trabalho visa proporcionar apoio em emergências de qualquer um dos quatro níveis de intensidade mencionados anteriormente, sempre que a instalação de um Posto de Comando (temporário ou móvel) for considerada necessária.

2.2 Processos da Gestão de Emergências

Geralmente, uma situação de emergência urbana inicia-se devido à existência de uma vulnerabilidade no local afetado. Soma-se a isso então a ocorrência de um evento adverso e a dificuldade de se combater ou mitigar completamente seus efeitos. Assim sendo, temos como resultado danos estruturais e perdas humanas, além de prejuízos econômicos e ao meio ambiente.

A Gestão de Emergências é o processo pelo qual todas as incertezas que existem em situações potencialmente perigosas e danosas podem ser minimizadas e a segurança pública maximizada. O objetivo é limitar os custos de emergências, crises ou desastres através da organização e aplicação de uma série de estratégias e táticas que refletem o ciclo de vida completo de um desastre, ou seja, preparação, resposta, recuperação e mitigação (DRABEK, 1996).

A Gestão de Emergências é um processo dinâmico e a etapa de planejamento, embora seja uma parte crítica, não é o único componente. Treinamento, a realização de simulações, o teste de equipamentos e a coordenação de atividades com a comunidade são outras funções importantes (FEMA, 1993). O ciclo de vida completo da gestão de emergências, com suas fases, papéis, atividades, recursos e processos envolvidos pode ser visto na Figura 3. A seguir descreveremos cada uma das fases desse ciclo de vida da gestão de emergências (FEMA, 2007; EIIF, 2009).

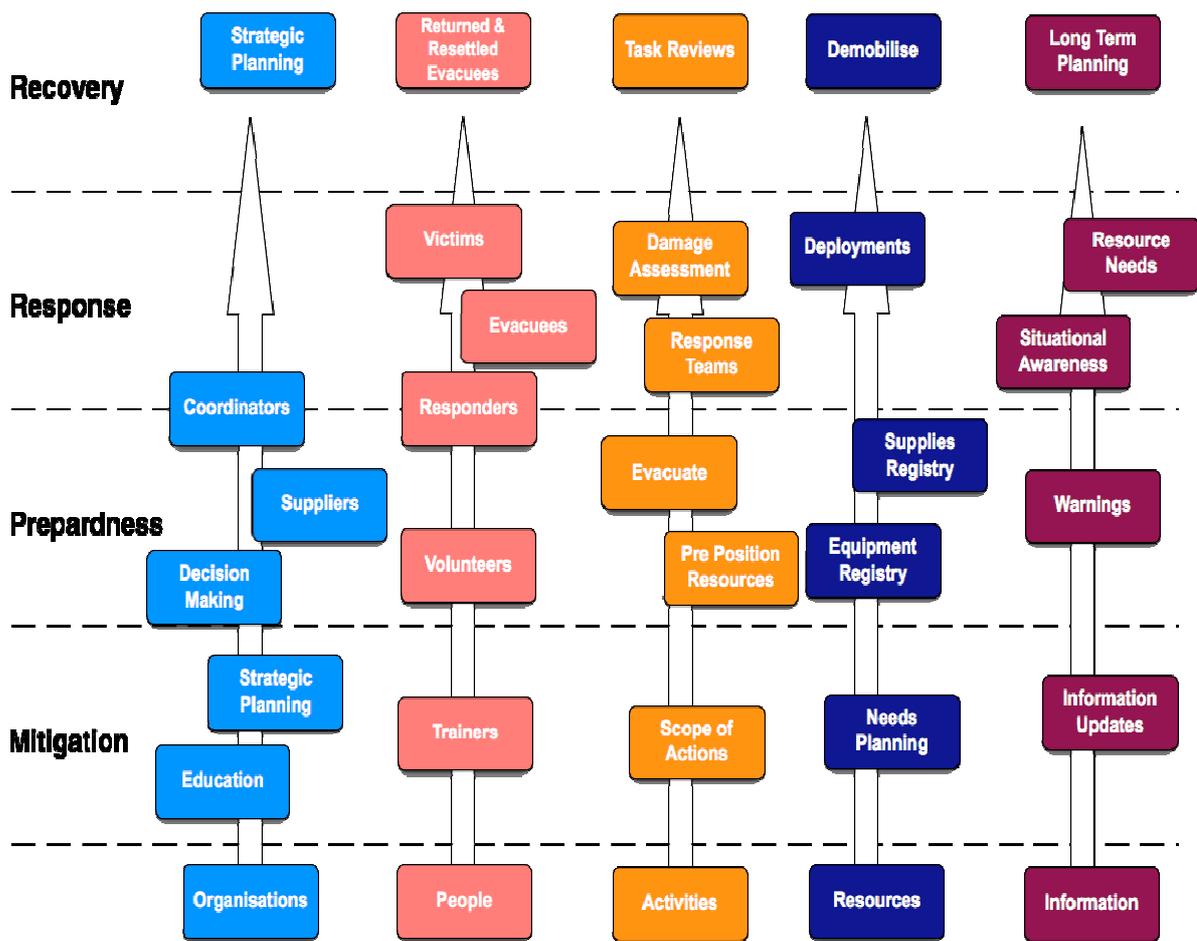


Figura 3. Esquema do Phased Emergency Framework Model (EIIF, 2009)

2.2.1 Mitigação

Esta fase é realizada muito antes da ocorrência de um evento adverso e é repetida após a sua conclusão. Essa etapa refere-se às políticas e ações destinadas a reduzir a vulnerabilidade de uma população ou minimizar os efeitos nocivos de futuros desastres inevitáveis. As ações dessa fase envolvem, dentre outras atividades, a análise de planejamento e risco de ameaças potenciais, incluindo as atividades para reduzir o risco e a parte de educação/treinamento sobre como lidar com potenciais incidentes.

2.2.2 Preparação

Esta fase deve ser constantemente realizada antes da ocorrência de um desastre e inclui ações que melhoram a capacidade de responder com sucesso às emergências. As ações dessa etapa envolvem, dentre outras atividades, a pré-implantação de serviços organizacionais, avisos para as pessoas e provisão de recursos para o potencial impacto de uma ameaça de emergência antecipada. Também pode conduzir para a realização de estudos e previsões sobre os impactos e necessidades gerados na ocorrência de uma catástrofe em particular.

2.2.3 Resposta

Esta fase é considerada a mais complexa e, por conseguinte, a mais estudada de todas. Alguns dos fatores que lidam com essa complexidade é a imprevisibilidade e a velocidade dos acontecimentos, o número de pessoas envolvidas, a escassez de tempo para tomar decisões e agir conforme o planejado, a indisponibilidade de recursos e alguma incerteza sobre a percepção situacional. As ações dessa etapa envolvem, dentre outras atividades, a implantação de serviços de resgate, organização de serviços de coordenação e recursos para lidar com necessidades imediatas após o incidente de emergência. Essa fase é iniciada imediatamente após a ocorrência de um evento adverso e termina na estabilização da emergência.

Como o presente trabalho se propõe a atuar na fase de resposta de uma emergência, vamos discutir aqui dois exemplos de cenários de emergência que ganharam notoriedade na mídia internacional e nacional, respectivamente, devido à proporção desses desastres. O primeiro deles foi o furacão Katrina, o pior e mais destrutivo furacão da temporada de furacões atlânticos em 2005. Ele foi o desastre natural mais caro, com valores acima de 81,2 bilhões de dólares, e um dos cinco furacões mais mortais na história dos Estados Unidos, com o número de mortos ultrapassando a marca de 1.836 pessoas. Dentre os furacões atlânticos registrados, ele foi o sexto mais forte no geral (KATRINA, 2012).

O número mais significativo de mortes ocorreu em New Orleans, Louisiana, que inundou quando o sistema de diques falhou catastroficamente horas depois que a tempestade tinha se movido para o interior. Eventualmente 80% da cidade e grandes setores de paróquias vizinhas foram inundados e as águas permaneceram por semanas. No entanto, o pior dano de propriedades ocorreu em áreas costeiras, como todas as cidades à beira-mar no Mississippi, que foram inundadas em mais de 90% em questão de horas, enquanto barcos e barças

cassino batiam em edifícios, empurrando carros e casas para o interior, e as águas atingiam 10-19 km de distância da praia.

Nos Estados Unidos, conforme delineado no plano de resposta nacional, a resposta aos desastres é, antes de tudo, uma responsabilidade do governo local. Quando o governo local exaure seus recursos, então pedidos específicos de recursos adicionais são solicitados a partir do nível do condado. Esse processo de solicitações procede da mesma forma a partir do município para o Estado e deste para o governo Federal, conforme a necessidade de recursos adicionais é identificada.

Algumas respostas para a recuperação do desastre causado pelo Katrina começou antes mesmo da tempestade, com as preparações da Agência Federal de Gestão de Emergências (FEMA) que variaram desde implementações logísticas de abastecimento até uma equipe funerária com caminhões refrigerados. Uma rede de voluntários começou a prestar assistência aos moradores locais e residentes emergentes de New Orleans e freguesias assim que a tempestade atingiu a costa, e continuou por mais de seis meses após a tempestade.

Uma força tarefa conjunta foi estabelecida para atuar como resposta militar na cena. Aproximadamente 58.000 funcionários da Guarda Nacional foram ativados para lidar com o rescaldo da tempestade, com tropas vindas de todos os 50 estados. O Departamento de Defesa americano também ativou os membros voluntários da Patrulha Aérea Civil. A FEMA prestou assistência habitacional (ajuda de aluguel, trailers, etc.) para mais de 700 mil candidatos — famílias e indivíduos. Para o fornecimento de novas habitações, a FEMA também pagou as despesas de hotel de 12 mil pessoas e famílias deslocadas pelo Katrina até o prazo final estabelecido para essa cobertura dos custos de hotéis. Após esse prazo, os evacuados ainda foram elegíveis para receber ajuda federal, podendo ser usada para aluguéis de apartamentos, estadias adicionais em hotéis, ou consertando as casas em ruínas.

Reforços da lei e agências de segurança pública de todo os Estados Unidos proveram respostas de ajuda mútua à Louisiana e New Orleans nas semanas após o desastre. Muitas agências responderam com mão de obra e equipamentos de lugares distantes. Além disso, grupos de resposta internacionais e organizações não governamentais partiram em auxílio às equipes de respostas a emergências governamentais dos EUA. Mais de 70 países prometeram doações em dinheiro ou outra forma de assistência. Notavelmente, Cuba e Venezuela (ambos considerados como hostis aos interesses do governo americano) foram os primeiros países a oferecer assistência, prometendo mais de um milhão de dólares, vários hospitais móveis, tratamento de água, alimentos enlatados, água mineral, óleo para aquecimento, 1.100 médicos e 26,4 toneladas de medicamentos, embora essa ajuda tenha sido rejeitada pelos EUA.

Diversas organizações de caridade providenciaram ajuda às vítimas da tempestade. Elas não foram autorizadas pela Guarda Nacional a entrar propriamente em New Orleans durante vários dias após a tempestade por causa de preocupações com a segurança. Essas organizações levantaram 4,25 bilhões de dólares em doações do público, com a Cruz Vermelha recebendo mais da metade dessas doações. Algumas organizações menores e indivíduos por conta própria ignoraram as restrições de acesso e proveram algum conforto e alívio para as primeiras vítimas da tempestade. Por exemplo, dois aviões fretados evacuaram 200 pacientes do Hospital Caridade. Voluntários do serviço de emergência de rádio amador proveram comunicações em áreas onde a infraestrutura de comunicações tinha sido danificada ou totalmente destruída. Muitas empresas também contribuíram para os esforços de socorro.

Já o segundo cenário de emergência, de repercussão nacional, refere-se à maior tragédia climática na história do Brasil, causada pelas fortes chuvas que ocorreram em Janeiro de 2011 na região serrana do estado do Rio de Janeiro, ocasionando grandes enchentes e deslizamentos de morros e encostas. Além das vidas humanas, essa região sofreu grandes perdas materiais com o desabamento de casas, ruas e vilas inteiras, e perdas na infraestrutura de serviços essenciais como abastecimento de água, esgoto, energia elétrica e telefonia. Os serviços governamentais contabilizaram 916 mortes e em torno de 345 desaparecidos, além de cerca de 35 mil desalojados em consequência dos desastres naturais (TRAGÉDIA, 2011).

Em resposta a esses acontecimentos, o Governo Federal destinou 780 milhões de reais para reconstruir as cidades afetadas. O Governo do estado do Rio de Janeiro concentrou seus recursos no resgate das vítimas soterradas e na desobstrução das estradas, e pediu o auxílio da Marinha e de órgãos da administração pública federal nesses trabalhos. As cidades serranas atingidas pelas chuvas receberam reforço de mais de 200 homens da Força Nacional de Segurança, equipes de limpeza pública, voluntários na área de saúde e resgate. O Exército também enviou cerca de 400 soldados, 6 helicópteros, 30 veículos e equipamentos de apoio.

Os municípios atingidos pela tragédia, porém, alegaram que as dimensões do caos instaurado durante a etapa de resposta foram de grandes proporções, chegando a faltar até itens de saúde, vacinas, remédios, materiais básicos de higiene e *kits* de alimentação. Grupos de socorro tentavam chegar até as localidades mais atingidas, porém a dificuldade de acesso complicava cada vez mais o andamento das atividades de resgates. Empresas, condomínios e personalidades também contribuíram com doações. Os usuários de mídias sociais como o Twitter, Orkut e Facebook demonstraram o poder da mobilização em redes sociais e organizaram um comboio para levar mantimentos, roupas e medicamentos para a área afetada.

Apesar de todos os esforços junto aos governos locais, o Governo Federal admitiu perante a ONU que o Brasil, até então, encontra-se despreparado para evitar mortes e mitigar efeitos trágicos em vários municípios próximos a rios, encostas sem monitoramento ambiental e moradias irregulares. Em comparação a países como EUA, Canadá e Austrália, onde o volume de chuvas tem sido maior e onde há a ocorrência de tornados e furacões, no Brasil ainda morre-se muito mais em decorrência das enchentes e das chuvas fortes.

2.2.4 Recuperação

Esta fase começa logo assim que o perigo está sob controle. Seu principal objetivo é reparar, reconstruir ou recuperar o que foi perdido pelo desastre, o que pode durar semanas, meses ou até mesmo anos para ser concluído. As ações dessa etapa envolvem, dentre outras atividades, a implantação de serviços organizacionais em longo prazo para restaurar a comunidade, empresas e áreas ambientalmente afetadas, incluindo uma revisão e análise da eficácia da mitigação e fases de preparação e feedback para melhorar os serviços para futuros incidentes.

2.3 Princípios da Gestão de Emergências

A chave para o sucesso de uma boa gestão de crises (ou gestão de emergências) é um diagnóstico preciso e oportuno da criticidade dos problemas e as dinâmicas de eventos que se seguem. Isso requer conhecimento, liderança corajosa cheia de habilidades para assumir os riscos e vigilância. Uma gestão de crises bem-sucedida também requer motivação, um senso de urgência, comprometimento e pensamento criativo, com uma visão estratégica de longo prazo. Quando se está gerindo uma crise, normas organizacionais pré-estabelecidas, culturas, regras e procedimentos se tornam grandes obstáculos: os administradores e burocratas tendem a proteger-se através de um jogo burocrático, escondendo-se atrás de defesas organizacionais e legais. Um senso de urgência dá lugar à inércia, às defesas organizacionais e à autoproteção por gerentes e funcionários. A gestão de crises bem sucedida requer: (1) sentir a urgência da questão, (2) pensar de forma criativa e estratégica para resolver a crise; (3) tomar ações ousadas e agir com coragem e sinceridade; (4) romper com a cultura organizacional de autoproteção, assumindo riscos e ações que possam produzir melhores soluções nas quais não haveria perdedores significativos; e (5) a manutenção de uma presença contínua na situação que está em rápida mudança, com possíveis desdobramentos de acontecimentos dramáticos (FARAZMAND, 2001).

O Instituto de Gestão de Emergência do FEMA (IAEM, 2007) convocou um grupo de trabalho constituído de praticantes e acadêmicos da área de gestão de emergência visando instituir os princípios da gestão de emergência. Esse grupo, após pesquisar a vasta literatura da área, concordou em oito princípios chave que devem ser usados para orientar o desenvolvimento de uma doutrina de gestão de emergência. Os princípios acordados pelo grupo ditam que a gestão de emergência deve ser:

1. **Abrangente:** os gestores de emergência devem considerar e levar em conta todos os perigos, fases, interessados e impactos relevantes aos desastres.
2. **Progressiva:** os gestores de emergência devem antecipar desastres futuros e tomar medidas preventivas e preparatórias para construir comunidades resistentes e resilientes a desastres.
3. **Dirigida a Riscos:** os gestores de emergência devem usar princípios de gerenciamento de riscos adequados (identificação de perigos, análise de risco e análise de impacto) ao atribuir prioridades e recursos.
4. **Integrada:** os gestores de emergência devem assegurar a unidade de esforços entre todos os níveis de governo e todos os elementos de uma comunidade.
5. **Colaborativa:** os gestores de emergência devem criar e sustentar amplas e sinceras relações entre indivíduos e organizações para fomentar confiança, defender uma atmosfera de equipe, construir um consenso e facilitar a comunicação.
6. **Coordenada:** os gestores de emergência devem sincronizar as atividades de todas as partes interessadas relevantes para alcançar um propósito comum.
7. **Flexível:** os gestores de emergência devem usar abordagens criativas e inovadoras para resolver os desafios impostos pelo desastre.
8. **Profissional:** os gestores de emergência devem valorizar uma abordagem baseada em ciência e conhecimento, constituída de educação, treinamento, experiência, práticas éticas, administração pública e melhoria contínua.

Com base nesses princípios, diversas abordagens sistemáticas de apoio à Gestão de Emergências são construídas. A seguir mostraremos alguns sistemas que são empregados durante as fases de uma emergência com a finalidade de auxiliar os membros de uma equipe de Gestão de Emergências.

2.4 Abordagens de apoio à Gestão de Emergências

A Gestão de Emergências é uma disciplina que pode se aproveitar muito da aplicação de ciência e tecnologia na prática para lidar com os eventos inesperados, incidentes, acidentes ou desastres, melhorando assim as avaliações de riscos, o planejamento de ações para reduzir sua probabilidade de incidência ou seus impactos, a coordenação dos recursos, a execução das operações de socorro e resgate, além da assistência e da reabilitação das áreas e populações afetadas. Dessa forma, as organizações que trabalham com Gestão de Emergências vêm empregando ferramentas computacionais, principalmente desde a década de 90, para apoiar suas atividades (JOHANSSON; TRNKA; GRANLUND, 2007).

Durante nossos estudos verificamos que o Brasil, particularmente, ainda tem um longo caminho a percorrer na área de gestão de emergências. A atuação das organizações responsáveis pelo processo de gestão de emergências está limitada, de certa forma, aos grandes centros urbanos e industrializados. Ainda assim, essas organizações não dispõem de verbas, pessoal suficiente ou aparelhagem avançada para o combate às emergências nesses locais. Por muitas vezes o trabalho de resgate é realizado contando-se com a ajuda de voluntários e pessoas presentes no local afetado pelo desastre.

Segundo um documento apresentado em 2010, pela Secretaria Nacional da Defesa Civil, a Defesa Civil brasileira e demais órgãos de socorro encontram-se despreparados para grandes catástrofes. O governo admite ainda que não avaliou nenhuma escola ou hospital para elaborar o documento. A falta de planejamento é apontada como principal fator para pôr em risco a população. No documento fica claro que o Brasil estaria economizando recursos financeiros se a prioridade fosse a prevenção (CHADE, 2011).

Um dos maiores desafios da equipe de resposta a emergências é obter informações confiáveis e íntegras sobre as vítimas de acidentes que estão desaparecidas ou inconscientes. Imagine um cenário de emergência em que ocorreu um deslizamento de terra em uma encosta onde se localizava uma comunidade. Como resultado desse acidente, várias casas na área afetada foram soterradas. Algumas das pessoas que viviam nessa comunidade foram encontradas, mas outras estão desaparecidas ou inconscientes. Como a equipe de emergência poderia agir para encontrar informações sobre essas pessoas desaparecidas ou inconscientes? O mais provável é que os integrantes da equipe operacional deveriam pessoalmente procurar as pessoas envolvidas nesse cenário de emergência para obter mais informações sobre essas vítimas, o que não seria um trabalho fácil, devido ao caos instaurado no local da emergência.

Para ilustrar as dificuldades com as quais as equipes de resposta a emergências devem lidar no seu trabalho, separamos algumas imagens de tragédias previamente citadas neste trabalho. As ilustrações mostradas na Figura 4, Figura 5 e Figura 6 são um comparativo do estado de algumas das localidades onde ocorreu a tragédia na região serrana do Rio de Janeiro, antes e depois desse evento catastrófico. As fotos do estado anterior ao evento são de 03/12/2009, enquanto as fotos do estado posterior ao evento são de 12/01/2011. Justamente em face dessa realidade exposta é que muitas abordagens sistemáticas na área de Gestão de Emergências visam trabalhar seu apoio às equipes operacionais e ao comando e controle nas fases de uma emergência.



Figura 4. Capela Santo Antônio em Nova Friburgo - RJ (TRAGÉDIA, 2011)



Figura 5. Acesso à rua Adelino Pereira Valente em Nova Friburgo - RJ (TRAGÉDIA, 2011)



Figura 6. Rua Adelino Pereira Valente em Nova Friburgo - RJ (TRAGÉDIA, 2011)

2.4.1 Abordagens de apoio à equipe de operação

O foco desse tipo de abordagem é transferir as informações sobre a situação em campo para o grupo de comando, localizado em uma sala ou em um posto avançado, e consultar os dados disponíveis na sede da organização. Portanto, as tecnologias mais exploradas para se alcançar esse objetivo têm sido as de dispositivos móveis, como *Personal Digital Assistants* (PDAs), *Smartphones*, *Tablet PCs* e demais dispositivos compactos que podem ser acoplados ou embutidos nos uniformes (*wearable devices*). Dessa forma, procura-se evitar que essas pessoas levem consigo grandes equipamentos ou computadores, facilitando sua locomoção e, portanto, seu trabalho. Os sistemas desse tipo devem possuir uma entrada de dados rápida e direta, e a visualização dos dados recebidos deve ser simples e focada.

Um sistema de informação baseado na percepção da situação compartilhada e na percepção da situação da equipe operacional, chamado *Situation Aware Vigilant Emergency Reasoner* — SAVER — (JAVED; NORRIS; JOHNSTON, 2012), tem a intenção de reunir informações de fontes relevantes para apresentá-las aos gestores de decisões de uma emergência. Ele também fornece recomendações para ação reforçadas por explicações fundamentadas. O gerente pode então avaliar essas recomendações e substituí-las, se ele achar que o SAVER não abordou todas as questões relevantes. Porém, é mais provável que o gerente concorde que as características do sistema pertinentes à memória, busca por padrões e orientação a metas vá preencher as lacunas no conhecimento do usuário e melhorar a percepção da situação, conforme sugerido pelas simulações executadas com o sistema.

Marterer, Moi e Koch (2012) apresentam uma arquitetura para sistemas distribuídos dirigidos a eventos para coletar e analisar dados em operações de emergência e treinamentos.

Com base nessa arquitetura, um sistema cliente servidor chamado PRONTO foi projetado para apoio à decisão nas operações de emergência através da integração de tecnologias como o processamento de eventos complexos. A novidade do PRONTO é que toda mudança no mundo real dentro de um contexto de operação de emergência é visto como um evento.

Esses eventos são recolhidos a partir de vários tipos de fontes, tais como dispositivos de *hardware* e interação do usuário com o sistema, além de fluxos de dados em áudio e vídeo. O sistema agrega eventos de todos estes tipos de fontes e filtra informações relevantes para os usuários. Ele também avalia a gestão de recursos em operações de resgate de emergência e fornece uma melhor visão disponível do quadro operacional durante as operações de emergência e exercícios de treinamento.

O projeto e-Triage (ADLER et al., 2011) apresenta um *framework* de TI para suporte à gestão de incidentes de casualidades em massa baseado em quatro conceitos: dispositivos portáteis para uso tanto em operações de resgate diárias quanto em desastres, infraestrutura de comunicação autônoma baseada em satélite (com células de rádio terrestres que podem ser instaladas na área de operação, combinando dispositivos com *software* dedicado ao registro de vítimas), banco de dados auto-organizável e auto-sincronizável distribuído, sem um ponto de falha, para disponibilidade máxima das informações e pesquisa de aceitação psicológica.

Tablet PCs com *software* de interface gráfica otimizada são usados em campo para registrar os pacientes, e os dados adquiridos são transmitidos automaticamente por meio de redes celulares GSM/UMTS terrestres existentes e disponíveis, ou por meio de células de rádio autônomas que estão ligadas a dados centrais e redes de telefonia com *links* de satélite. Dessa forma, os controladores de operação, tomadores de decisão e hospitais têm acesso em tempo útil ao mesmo nível de informação, de modo que um quadro operacional comum pode ser desenvolvido. Para diferentes usuários, distintas interfaces gráficas são desenvolvidas, o que ajuda a lidar com o trabalho sob estresse extremo.

2.4.2 Abordagens de apoio ao comando

O Comando e Controle é o responsável pelas decisões gerais e pelo gerenciamento de recursos durante a ocorrência de um desastre. Para essas decisões serem tomadas da melhor maneira possível, é preciso que elas sejam baseadas em conhecimento contextual atualizado, por meio da percepção da situação enfrentada pelas equipes de operação, e em conhecimento formal prévio, que, geralmente, costuma estar armazenado em bancos de dados disponíveis no Centro de Comando e Controle. Por isso, diversas abordagens que visam apoiar a decisão com base nestes dois tipos de conhecimento têm sido propostas.

Visando melhorar a identificação e a gestão de meios de resposta em um incidente com vítimas em massa, bem como para ajudar a coordenar uma resposta inicial, Kondaveti e Ganz (2009) propõem um sistema de apoio à decisão para a alocação de recursos na gestão de desastres baseado na rápida aquisição de informações e em funcionalidades de rastreamento de recursos. Usando informações coletadas de agências de resposta a emergências, técnicas de pesquisa de operação são usadas para encontrar uma solução ideal para a expedição e alocação de recursos para o local do desastre. Dessa forma, a ferramenta proposta pode ajudar as organizações de resposta a emergências não só a realizar atividades de resposta a emergências eficientemente, mas também a realizar o planejamento de recursos da emergência (quantidade e localização de recursos).

Através da disponibilização de informações contextuais correntes vindas diretamente do campo para ampliar e agilizar a construção da percepção situacional do comando, Ferreira (2011) elabora uma heurística baseada na utilidade dessas informações a cada momento, visando uma tomada mais rápida de decisões durante a resposta a situações de emergência urbana. Tal heurística se divide em dois níveis de atuação: a distribuição da “informação correta para a pessoa certa” e a indicação da “informação correta no momento apropriado”.

O desenvolvimento dessas ideias em um sistema para uma mesa digital multitoque resultou na ferramenta colaborativa WITS (What Is The Situation?) que recebe a informação de dispositivos externos e a inclui na representação do cenário real, relacionando-a ao conhecimento já existente, calculando sua pontuação e classificação segundo as regras da heurística, e aplicando as transformações visuais necessárias. Dessa forma, essa ferramenta favorece o trabalho coletivo síncrono e co-localizado dos integrantes do posto de comando, permitindo que eles interajam entre si e com as informações disponibilizadas na mesa, realizando inclusive operações de inclusão, alteração, exclusão e fusão dessas informações.

Também atuando na área de percepção situacional para apoiar a tomada de decisões dos gestores de emergência, o sistema MASAS — Multi-Agency Situational Awareness System — (PAGOTTO; O’DONNELL, 2012) está rapidamente se tornando o sistema nacional canadense para troca de informações relevantes a incidentes na gestão de emergência entre as várias agências e jurisdições do país. Utilizando informação estruturada alinhada a padrões abertos e uma arquitetura de gerenciamento centralizado aberto, esse sistema fornece uma comunidade de confiança virtual com a capacidade de trocar informações sobre gestão de emergências sem maiores problemas. Ele oferece uma arquitetura de troca de informações que é baseada em um sistema altamente resiliente de centros de agregação de dados que são

facilmente acessíveis por funcionários de gestão de emergências em todos os níveis, diretamente ou através de ferramentas comerciais de terceiros.

2.4.3 Abordagens de apoio ao comando e às equipes de operação

Abordagens desse tipo devem ser sistemas mais complexos e completos, que procuram abranger várias etapas da Gestão de Emergências, garantindo assim um suporte simultâneo a esses dois grupos distintos. Nesse sentido, Campbell et al. (2008) desenvolvem um ambiente de simulação assistido por computador com uma ferramenta de apoio a decisões chamado RimSim para auxiliar o planejamento de resposta a emergências e o treinamento da equipe operacional. Esse sistema modular fornece uma oportunidade para projetar de forma flexível exercícios de simulação de resposta a emergências individuais e em grupos.

Os módulos interativos do RimSim fornecem opções de uso multi-jogadores síncronos ou individual assíncrono com participantes simulados. Através desse *role-play* interativo com visualizações compartilhadas, pode-se estudar a cognição distribuída com uma meta de longo prazo de identificar oportunidades para melhorar a gestão da informação durante a fase de resposta a emergências. Ao identificar modelos mentais que suportam uma hipótese encarnada e criam uma arquitetura flexível para aconselhar o *role-play* de treinamento e planejamento, o RimSim trabalha para melhorar a participação socorrista na coleta de dados e as atividades de resposta com um *software* que apresenta um nível de complexidade útil de cenários de crise para o planejamento e treinamento de resposta a emergências.

Uma abordagem baseada em listas de verificação (KRÜGER; WUCHOLT; BECKSTEIN, 2012), um tema muito discutido e controverso entre os especialistas de organizações de resgate, mostra que *Intelligent Electronic Checklist Support Systems (IECSS)* podem ser especialmente úteis para a cooperação (inter-)organizacional em cenários de desastres. Esse estudo conclui que listas de verificação poderiam ser um apoio útil para dois diferentes grupos de usuários: (1) especialistas (ou seja, líderes treinados para missões específicas), servindo sobretudo como um nível de proteção redundante. (2) forças menos experientes, sendo uma ajuda útil para a coordenação e gerenciamento de tarefas baseadas em padrões. No cerne do sistema está um cliente de listas de verificação que inicializa as listas de verificação de nível superior para todas as funções relevantes para a missão.

Já o DACE — *Dependency-based Awareness and Coordination Environment* — (YU; CAI, 2012) é um sistema de coordenação suportado por computador que oferece uma solução escalável para a coordenação na resposta a emergências. O sistema serve como uma ajuda cognitiva para atores humanos, mantendo um grupo de modelos mentais das atividades gerais

de colaboração e suas dependências, e determinando os efeitos de eventos enquanto eles se propagam através da teia de dependências. O *framework* contribui para a meta de ampliação de percepção situacional baseada na coordenação de resposta a emergências e possui dois componentes principais: (1) compreensão e modelagem de atividades colaborativas e suas dependências, e (2) um mecanismo de representação e raciocínio que ajuda os usuários a determinar a relevância dos eventos que impactam os estados de atividades.

Capítulo 3 - Sistemas de Recomendação

Os sistemas de recomendação são uma subclasse dos sistemas de filtragem da informação que buscam prever a preferência que um usuário pode ter em relação a um item (como músicas, livros ou filmes) ou elemento social (por exemplo, pessoas ou grupos) que ainda não tenha considerado, usando um modelo construído a partir das características de um item (abordagens baseadas no conteúdo) ou ambiente social do usuário (abordagens de filtros colaborativos) (RICCI; ROKACH; SHAPIRA, 2011).

O objetivo de um sistema de recomendação é gerar recomendações significativas de itens ou produtos que possam interessar a uma determinada coleção de usuários. Os sistemas de recomendação diferem na maneira como eles analisam as fontes de dados para desenvolver noções de afinidade entre usuários e itens, que podem ser utilizados para identificar os pares bem combinados (MELVILLE; SINDHWANI, 2010). Eles comparam os dados coletados com dados semelhantes e diferentes dos outros recolhidos e geram uma lista de itens (que podem estar ordenados segundo determinadas preferências) recomendados para o usuário.

As primeiras ideias que surgiram para a construção de sistemas de recomendação foram baseadas em estatísticas de correlações simples e modelagem preditiva, não envolvendo o maior número de práticas usuais em estatísticas e aprendizagem de máquina da literatura contemporânea (BILLSUS; PAZZANI, 1998). Atualmente, os sistemas de recomendação são ferramentas populares por entregarem conteúdo relevante para os usuários de acordo com as informações de perfil obtidas durante suas interações com o sistema.

Com o advento das redes sociais e da Web 2.0, o rico conteúdo fornecido por interações com o usuário nesse tipo de ferramenta gera uma enorme quantidade de informações disponíveis na Web, tornando necessário o desenvolvimento de sistemas de recomendação de uma maneira personalizada aos usuários. Esses sistemas de recomendação mostram conteúdos relevantes para os usuários, filtrando o que poderia ser interessante para aquele perfil e tentando identificar a relação entre usuários e itens para desenvolver a recomendação e a personalização. O desafio é compreender como a Web 2.0 e as redes sociais em conjunto podem atingir todos os tipos de usuários (ZIESEMER; OLIVEIRA, 2011).

No nosso dia a dia encontramos vários exemplos comerciais e não comerciais de sistemas de recomendação. Podemos citar como exemplos os seguintes *sites*:

- **Amazon**⁴: a cada item que o usuário visita, a loja recomenda itens adicionais com base em uma matriz do que outros clientes compraram junto com o item atualmente selecionado (o famoso “clientes que compraram o item X também levaram os itens Y, Z, ...”).
- **Pandora Radio**⁵: este repositório de *streaming* de áudio *online* obtém como entrada inicial uma música ou músico e toca músicas com características semelhantes, com base em uma série de palavras-chave atribuídas ao artista ou pedaço da música.
- **Netflix**⁶: esta locadora virtual oferece um serviço com previsões de filmes que um usuário pode gostar de assistir com base nas avaliações anteriores do usuário e seus hábitos de assistir, levando também em conta as características do filme, dentre outros.

Sistemas de recomendação são uma alternativa útil para algoritmos de busca, uma vez que ajudam os usuários a descobrir itens que não poderiam encontrar por si mesmos. Curiosamente, sistemas de recomendação são frequentemente construídos utilizando motores de busca que indexam dados não tradicionais. Seu funcionamento pode ser mais bem entendido segundo um *framework* simplificado (BRIDGE et al., 2005) que contrasta o *modus operandi* desses sistemas:

- **Proativo X Reativo**: uma abordagem proativa permite a extração de dados durante a interação dos usuários com a aplicação (*online* e *real time*), associando as requisições aos usuários. A coleta desses dados é implícita, ou seja, sem a necessidade de um pedido explícito da opinião dos usuários. Os dados extraídos dessa forma são dados com qualidade, completos, “sem ruídos” e, portanto, “livre de erros”. Já as estratégias reativas procuram associar pedidos aos usuários, com base em *logs* da aplicação, depois de eles terem interagido com o sistema. Ainda, durante a interação com o sistema, é necessário que o usuário forneça uma consulta explícita para que o sistema reaja com uma recomendação em resposta à consulta.
- **Single-Shot X Conversational**: sistemas de recomendação que adotam uma estratégia de recomendação do tipo *single-shot* retornam um único conjunto de sugestões para um usuário em uma determinada sessão. Na vida real, os

⁴ <http://www.amazon.com/>.

⁵ <http://www.pandora.com/>.

⁶ <http://www.netflix.com/>.

cenários de recomendação raramente são de tão curta duração, principalmente porque os seres humanos são perfeitamente capazes de especificar plenamente suas necessidades com antecedência e raramente estão satisfeitos com as recomendações iniciais. Justamente por essas razões que os sistemas de recomendação conversacionais adotam uma abordagem iterativa para indicar uma recomendação. Dessa forma, os usuários podem elaborar as suas necessidades como parte de um diálogo de recomendação estendida. Sistemas conversacionais ainda podem ser subdivididos em *navigation-by-asking* (questionamento) ou *navigation-by-proposing* (proposição), de acordo com a maneira que eles eliciam os requisitos dos usuários.

- **Questionamento X Proposição:** sistemas de recomendação conversacionais por questionamento perguntam aos usuários uma série de questões acerca de suas necessidades (por exemplo, um sistema de recomendação para computadores pessoais poderia perguntar “Quanto de memória você deseja?” ao usuário). Sistemas desse tipo enfrentam o problema de decidir qual o melhor conjunto de questões para se perguntar em uma sessão, além da ordem dessas perguntas. Porém, fornecer recomendações levando-se em consideração a resposta dos usuários a perguntas explícitas pode colocar uma carga significativa de responsabilidade sobre os usuários. Na verdade, às vezes os usuários podem não ser capazes de responder a uma determinada pergunta (por exemplo, quando seu conhecimento sobre o domínio da recomendação é insuficiente). Isso é o que motiva a pesquisa em sistemas de recomendação conversacionais por proposição. Esses sistemas, alternativamente, apresentam determinados produtos aos usuários e eliciam os devidos requisitos na forma de *feedback* do usuário sobre os produtos propostos.

Independentemente do *modus operandi* de um sistema de recomendação, eles são divididos em três estratégias clássicas de recomendação. Na seção a seguir discutiremos os tipos de abordagens de recomendação existentes.

3.1 Tipos de Recomendação

Os sistemas de recomendação podem produzir uma lista de recomendações através de duas maneiras distintas: filtragem colaborativa ou filtragem baseada em conteúdo. A filtragem

colaborativa busca construir um modelo através de decisões semelhantes feitas por outros usuários, bem como analisando o comportamento passado de um usuário (itens adquiridos anteriormente e/ou avaliações dadas a esses itens). Em seguida, ela usa esse modelo para prever itens que o usuário possa ter algum tipo de interesse, ou avaliá-los de acordo com sua preferência (MELVILLE; SINDHWANI, 2010). Dessa forma, a filtragem é dita colaborativa, pois cada usuário, ao avaliar os itens adquiridos, está colaborando com as futuras recomendações desses itens para outros usuários com perfis semelhantes ao seu.

Abordagens baseadas em filtragem de conteúdo utilizam uma série de características discretas de um item para recomendar itens adicionais com propriedades semelhantes. Cada tipo de abordagem tem seus próprios pontos fortes e fracos. No caso de abordagens baseadas em filtros colaborativos, justamente por se balizarem em avaliações passadas de outros usuários, tais sistemas possuem o problema de *cold start*, ou seja, requerem uma grande quantidade de informações sobre um utilizador a fim de fazer recomendações precisas. Já os sistemas com filtros baseados em conteúdo precisam de muito pouca informação para começar, mas são muito mais limitados em escopo (por exemplo, só podendo fazer recomendações que são semelhantes à semente original).

De acordo com Adomavicius (2005), os sistemas de recomendação são divididos em três categorias: filtragem colaborativa, filtragem baseada em conteúdo e filtragem híbrida. Cada uma dessas categorias é discutida nas seções a seguir.

3.1.1 Filtros colaborativos — *Collaborative Filtering*

O termo “filtragem colaborativa” foi introduzido no contexto do primeiro sistema de recomendação comercial, chamado Tapestry (GOLDBERG et al., 1992). Ele foi concebido para recomendar documentos extraídos de grupos de notícias para determinados usuários. Sua motivação foi alavancar a colaboração social para evitar que esses usuários fossem inundados por um grande volume de documentos. Os métodos de filtragem colaborativa são baseados em coletar e analisar uma grande quantidade de informações sobre o comportamento dos usuários, atividades ou preferências e predizer o que os usuários irão gostar com base em sua similaridade com outros usuários. Filtragem colaborativa baseada em usuários tenta modelar o processo social de pedir uma recomendação para um amigo.

A filtragem colaborativa, que analisa dados de uso dos usuários para encontrar os melhores pares possíveis de usuário-item, tem sido justaposta à antiga metodologia de filtragem de conteúdo, que teve sua origem na área de recuperação da informação. Na filtragem de conteúdo, as recomendações não são “colaborativas” no sentido de que as

sugestões apresentadas para um usuário não utilizam explicitamente as informações de toda a base de usuários. O sistema GroupLens (RESNICK et al., 1994) pode ser citado como um dos sucessos iniciais da aplicação da filtragem colaborativa em domínios específicos.

Sistemas de filtros colaborativos funcionam através do *feedback* do usuário (através de avaliações dos itens de um domínio) e exploração das semelhanças no comportamento de avaliação entre os vários usuários na hora de recomendar um item. Eles podem atuar em domínios onde não há muito conteúdo associado a itens, ou onde o conteúdo é de difícil análise para um computador — tais como ideias, opiniões, etc. Eles possuem a capacidade de fornecer recomendações casuais, ou seja, podem recomendar itens que são relevantes para o usuário, mas não contêm conteúdo específico do perfil do usuário. Uma das principais vantagens da filtragem colaborativa é que ela não se baseia no conteúdo analisável pela máquina e, portanto, é capaz de obter grande precisão ao recomendar itens complexos, tais como filmes, sem a necessidade de um “entendimento” do item propriamente dito.

Um de seus subtipos mais famosos é a filtragem colaborativa item-a-item (a popular recomendação “pessoas que compraram o item X também compraram o item Y”, dos sites de venda). Filtros colaborativos podem ainda ser subdivididos em abordagens baseadas em vizinhanças (*neighborhood-based*) e baseadas em modelo (*model-based*). Métodos baseados em vizinhanças também são comumente referidos como abordagens baseadas em memória (*memory-based*) (BREESE; HECKERMAN; KADIE, 1998).

- **Filtragem Colaborativa Baseada em Vizinhança:** esses tipos de abordagens costumam analisar um subconjunto de usuários escolhidos com base em sua semelhança com o usuário ativo, e então uma combinação ponderada de suas avaliações é utilizada para produzir previsões para este usuário. Diversas extensões para esta técnica foram criadas visando uma melhoria significativa de desempenho, tais como: filtragem colaborativa baseada em item (*Item-based Collaborative Filtering*), ponderação de significância (*Significance Weighting*), votação padrão (*Default Voting*), frequência inversa do usuário (*Inverse User Frequency*) e amplificação de Caso (*Case Amplification*).
- **Filtragem Colaborativa Baseada em Modelo:** essas abordagens fornecem recomendações de itens estimando parâmetros de modelos estatísticos para as avaliações dos usuários. Recentemente, modelos de fator latente (*latent factor*) e fatoração de matriz (*matrix factorization*) têm emergido como uma metodologia do estado da arte nesta classe de técnicas (BELL; KOREN; VOLINSKY, 2009). Ao contrário dos métodos baseados em vizinhanças, que

geram recomendações baseadas em noções estatísticas de similaridade entre os usuários, ou entre itens, os modelos de fatores latentes supõem que a semelhança entre os usuários e itens é simultaneamente induzida por uma estrutura oculta de menor dimensão nos dados. As técnicas de fatoração de matriz são uma classe de modelos de fatores latentes amplamente bem sucedidos onde os usuários e itens são simultaneamente representados como vetores de características desconhecidas (vetores coluna) ao longo de N dimensões latentes.

3.1.2 Filtros baseados em conteúdo — *Content-based Filtering*

Métodos de filtragem baseados em conteúdo remetem às informações e características dos itens recomendados. Eles fornecem recomendações comparando as representações de conteúdo contidas em um item a representações de conteúdo que interessa ao usuário. Em outras palavras, estes algoritmos tentam recomendar itens que são semelhantes aos que um usuário gostou no passado ou está examinando no presente. Em particular, os itens candidatos são comparados com vários itens previamente avaliados pelo usuário, e então os itens mais congruentes são recomendados. Por exemplo, dado como tipo de informação o “gênero de música”, e sabendo que um usuário gostou de artistas como “Caetano Veloso” e “Gilberto Gil” pode-se inferir uma predileção pela Música Popular Brasileira (MPB) e, portanto, o sistema poderia recomendar “Chico Buarque”.

Esta abordagem tem suas raízes nas áreas de recuperação de informação e filtragem de informação. Grande parte das pesquisas nesta área se concentra em recomendação de itens com conteúdo textual associado, tais como páginas da *web*, livros e filmes (onde as páginas da *web* em si ou conteúdos associados, como descrições e revisões de usuários, estão disponíveis). Assim sendo, várias abordagens têm tratado este problema como uma tarefa de recuperação da informação, onde o conteúdo associado a preferências do usuário é tratado como uma consulta, e os documentos não avaliados são pontuados de acordo com os índices de relevância e similaridade com esta consulta (BALABANOVIC; SHOHAM, 1997).

Uma alternativa para essas abordagens de recuperação da informação é tratar a recomendação como uma tarefa de classificação, em que cada exemplo representa o conteúdo de um item, e as últimas avaliações de um usuário são utilizadas como etiquetas para estes exemplos. No domínio de recomendação de livros, Mooney e Roy (2000) utilizaram os textos de campos como título, autor, sinopses, críticas e assunto para treinar um classificador de Bayes multinomial ingênuo. Outros algoritmos de classificação também já foram usados para

a recomendação puramente baseada em conteúdo, incluindo K vizinhos mais próximos, árvores de decisão e redes neurais (PAZZANI; BILLSUS, 1997).

A principal questão da filtragem baseada em conteúdo é a capacidade de o sistema aprender as preferências do usuário a partir de suas ações sobre uma fonte de conteúdo, e usá-las em outros tipos de conteúdo. Quando o sistema está limitado a recomendar somente conteúdo do mesmo tipo que o usuário está utilizando, o valor agregado da recomendação é significativamente menor do que quando outros tipos de conteúdos de outros serviços são recomendados. Por exemplo, recomendar artigos de notícias baseado em navegação de notícias é útil, mas é muito mais útil quando músicas, vídeos, produtos, discussões, etc. de diferentes serviços são recomendados com base na navegação de notícias. Esse tipo de recomendação, justamente por ser mais complexa e de difícil inferência, agrega um maior valor de uso ao sistema de recomendação.

3.1.3 Filtros híbridos — *Hybrid Recommenders*

As abordagens de filtros híbridos visam alavancar os pontos fortes de cada um desses dois tipos de recomendadores, os baseados em conteúdo e colaboração, e, portanto, combinam os dois. Uma abordagem simples é permitir ambos os métodos de filtragem produzirem diferentes listas ordenadas de recomendação, e então mesclar os resultados para confeccionar uma lista final (COTTER; SMYTH, 2000). Claypool, Gokhale e Miranda (1999) combinam as duas previsões usando uma média ponderada de adaptação, onde o peso dos componentes de colaboração aumenta conforme o número de usuários que acessam um item aumenta.

Melville, Mooney e Nagarajan (2002) propuseram um *framework* geral para filtragem colaborativa impulsionada pelo conteúdo, onde previsões com base em conteúdo são aplicadas para converter uma matriz esparsa de avaliações de usuários em uma matriz de avaliação completa, e então um método de filtragem colaborativa é usado para fornecer as recomendações. Um classificador Bayes ingênuo é treinado sobre os itens avaliados de cada usuário para substituir os itens não cotados por previsões. A matriz resultante de pseudo avaliações é usada então para encontrar vizinhos semelhantes ao usuário ativo e produzir previsões utilizando a correlação de Pearson, devidamente ponderada para contabilizar a sobreposição de itens realmente avaliados e as previsões do conteúdo do usuário ativo. Esta abordagem demonstrou ter um melhor desempenho do que os sistemas exclusivamente de filtragem colaborativa ou puramente de filtragem baseada em conteúdo, e uma combinação linear desses dois tipos de sistemas.

Várias abordagens híbridas são baseadas em filtragens colaborativas tradicionais, mas também mantêm um perfil baseado em conteúdo para cada usuário. Estes perfis baseados em conteúdo são usados para encontrar usuários semelhantes, em vez de itens com avaliações parecidas. Além dessas, outras abordagens híbridas tratam a recomendação como uma tarefa de classificação, e incorporam elementos de colaboração nesta tarefa. Existem ainda algumas abordagens híbridas que tentam combinar diretamente os dados provenientes das filtragens baseadas em conteúdo e em colaboração no âmbito de um único *framework* probabilístico.

De acordo com Burke (2007), os sistemas híbridos podem ser classificados em:

- **Ponderado:** quando um sistema de recomendação híbrido funde resultados de diferentes técnicas, atribuindo pesos a cada uma delas. Esse é um dos tipos mais comuns de sistemas de recomendação híbridos;
- **Comutação:** quando sistemas de recomendação diferentes são utilizados de acordo com os critérios de recomendação. Dessa forma, temos um agregado de sistemas de recomendação funcionando como um único sistema;
- **Misto:** quando as recomendações de diferentes sistemas são apresentadas em conjunto. Essa é uma das mais simples implementações de sistemas de recomendação híbridos;
- **Cascata:** quando as recomendações de um sistema de recomendação são refinadas por outro. Dessa forma, o resultado final da recomendação tende a ser mais específico ao usuário, e com um maior grau de exatidão;
- **Combinação de Recursos:** quando recursos de diferentes fontes de dados de recomendação são combinados em um algoritmo de recomendação único. Semelhante aos sistemas de Comutação, porém, ao invés de subsistemas, temos a combinação de distintos recursos para formar um algoritmo de recomendação melhorado;
- **Aumento de Recursos:** quando um sistema de recomendação é usado para calcular um resultado e, em seguida, esse resultado é usado como valor de entrada para outro sistema de recomendação. Semelhante aos sistemas em Cascata, porém seu objetivo é gerar uma nova recomendação com base nos valores de saída anteriores, ao invés de simplesmente refiná-los.

3.2 Desafios e Limitações

Sistemas de recomendação introduzem dois problemas de incentivos interessantes. Em primeiro lugar, uma vez que se estabeleceu um perfil de interesses, é fácil se deixar levar pelo consumismo através das avaliações de produtos fornecidas por outros usuários. Além disso, esse problema não estaria totalmente resolvido ainda que as avaliações fossem reunidas implicitamente a partir de recursos existentes ou através da monitoração do comportamento dos usuários. Segundo, se alguém pode fornecer recomendações, os proprietários de conteúdo podem gerar montanhas de recomendações positivas para os seus próprios materiais e recomendações negativas para seus concorrentes (RESNICK; VARIAN, 1997).

Os sistemas de recomendação também levantam preocupações sobre a privacidade pessoal. Em geral, quanto mais informações os indivíduos têm sobre as recomendações, melhor eles serão capazes de avaliar essas recomendações. No entanto, as pessoas podem não querer os seus hábitos ou pontos de vista amplamente conhecidos. Além dessas implicações sociais, existem outros obstáculos comuns na implantação de sistemas de recomendação que apresentaremos a seguir, bem como algumas direções de pesquisas que lidam com eles.

3.2.1 *Cold-Start*

Novos itens e novos usuários representam um desafio significativo para os sistemas de recomendação. Coletivamente, esses problemas são referidos como o “problema de arranque a frio” (*cold-start problem*) (SCHEIN et al., 2002). O primeiro desses problemas surge em sistemas de filtragem colaborativa, onde um item não pode ser recomendado a menos que algum usuário o tenha avaliado antes. Esse problema não se aplica apenas a novos itens, mas também para itens mais obscuros, o que é particularmente prejudicial para usuários com gostos ecléticos. O problema do novo item é também muitas vezes referido como o problema do primeiro avaliador.

Uma vez que abordagens baseadas em conteúdo (MOONEY; ROY, 2000; PAZZANI; BILLSUS, 1997) não se baseiam em avaliações de outros usuários, eles podem ser utilizados para produzir as recomendações para todos os itens, desde que os atributos desses itens estejam disponíveis. De fato, as previsões baseadas no conteúdo dos usuários semelhantes também podem ser usadas para melhorar ainda mais as previsões para o usuário ativo (MELVILLE; MOONEY; NAGARAJAN, 2002).

O problema do usuário novo é difícil de resolver, pois sem preferências anteriores de um usuário não é possível encontrar usuários semelhantes ou construir um perfil baseado em

conteúdo. A investigação nessa área tem focado principalmente na eficiência e eficácia da seleção de itens a serem avaliados por um usuário, de modo a melhorar rapidamente o desempenho da recomendação com o mínimo de *feedback* fornecido pelo usuário. Nesse cenário, técnicas clássicas de aprendizagem ativa podem ser aproveitadas para abordar a tarefa de seleção de item (HARPALE; YANG, 2008; JIN; SI, 2004).

3.2.2 Escalabilidade

Um dos principais problemas encontrados na abordagem de filtragem colaborativa está relacionado com a escalabilidade desse tipo de sistema. Isso acontece devido à necessidade de se computar a similaridade entre os usuários cada vez que eles avaliam um novo item. A complexidade computacional do método cresce linearmente com o número de usuários (uma vez que o item pode ser avaliado por cada usuário do sistema), o que é típico em *sites* de redes sociais e comércio eletrônico, reduzindo o desempenho de um sistema que tem grande acesso diário ou mudanças frequentes. Uma das maneiras de melhorar a escalabilidade de sistemas com essa abordagem de recomendação é usando algoritmos baseados em itens para recomendar itens semelhantes que foram avaliados positivamente por outros usuários.

3.2.3 Esparsidade

Na prática, muitos sistemas de recomendação comerciais baseiam-se em grandes conjuntos de dados. Como resultado, a matriz usuário-item usada para filtragem colaborativa pode ser extremamente grande e esparsa, trazendo diversos desafios de desempenho das recomendações. Além disso, a maioria dos usuários não avalia a maioria dos itens e, portanto, a matriz usuário-avaliação é tipicamente muito esparsa. Isso é um problema para os sistemas de filtragem colaborativa, uma vez que diminui a probabilidade de se encontrar um conjunto de usuários com avaliações semelhantes.

No caso de filtragem baseada em conteúdo, esse problema é causado por itens raros (baixa amostragem de itens semelhantes). Este problema ocorre frequentemente quando um sistema tem uma alta taxa de item-a-usuário, ou o sistema está nas fases iniciais de utilização. Este problema pode ser atenuado usando-se informações adicionais de domínio (MELVILLE; MOONEY; NAGARAJAN, 2002; SU et al., 2007) ou fazendo suposições sobre o processo de geração de dados que permite uma alta qualidade de imputação (SU et al., 2008).

3.2.4 Fraude

À medida que os sistemas de recomendação foram sendo cada vez mais adotados por *sites* comerciais, eles começaram a desempenhar um papel muito importante em aumentar a rentabilidade desses vendedores. Assim sendo, muitos vendedores sem escrúpulos começaram a se dedicar a diferentes formas de fraude para usar os sistemas de recomendação em seu benefício próprio. Uma abordagem típica consiste de tentar inflar a desejabilidade percebida de seus próprios produtos (ataques do tipo *push*) ou então rebaixar as avaliações de seus concorrentes (ataques do tipo *nuke*).

Esses tipos de ataque têm sido amplamente estudados como ataques *shilling* (*shill* é um termo que se refere a alguém que propositadamente dá aos espectadores a impressão de que ele é um cliente independente entusiasmado com um vendedor ou comerciante de ideias, para quem ele está trabalhando secretamente) (LAM; RIEDL, 2004) ou ataques de injeção de perfil (BURKE et al., 2005.). Esses ataques normalmente envolvem a criação de perfis falsos e assumem diferentes quantidades de conhecimento sobre o referido sistema.

Por exemplo, o “ataque média” (LAM; RIEDL, 2004) pressupõe o conhecimento da avaliação média de cada item e o atacante atribui valores distribuídos aleatoriamente em torno dessa média, juntamente com uma nota alta para o item a ser empurrado para o usuário. Estudos têm demonstrado que tais ataques podem ser bastante prejudiciais para avaliações previstas, apesar da filtragem colaborativa baseada em item tender a ser mais robusta para esses ataques (LAM; RIEDL, 2004). Métodos baseados em conteúdo, que só dependem de avaliações passadas dos usuários, não são afetados por ataques de injeção de perfil.

Apesar dos métodos baseados em conteúdo evitarem algumas armadilhas discutidas acima, a abordagem de filtros colaborativos possui certas vantagens sobre esses métodos. Em primeiro lugar, a filtragem colaborativa pode atuar em domínios onde não há muitos conteúdos associados a itens, ou onde o conteúdo é de difícil análise para o computador, como ideias, opiniões, etc. Em segundo lugar, um sistema de filtragem colaborativa tem a capacidade de fornecer recomendações casuais, ou seja, ele pode recomendar itens que são relevantes para o usuário, mas que não contenham algum conteúdo do perfil do usuário.

3.2.5 Superespecialização

Na abordagem baseada em conteúdo, o perfil do usuário é baseado em suas interações com o sistema e não há relações com outros usuários. Além disso, quando um usuário avalia um item, a recomendação ocorrerá com base apenas na categoria de item que foi acessada.

Esse problema é conhecido como superespecialização, ou seja, o usuário é limitado a receber recomendações de itens que são sempre similares aos já avaliados porque não existe informação suficiente para produzir outros tipos de recomendação.

Por exemplo, um usuário em um ambiente tipo *news feed* que acessa conteúdo político receberá a maior parte do conteúdo sobre esta categoria. Isso acontece porque o sistema aprende que esse tipo de conteúdo é interessante para o usuário ativo. O problema é que o usuário pode ser sobrecarregado com sugestões repetidas dessa mesma categoria e talvez seja apenas um interesse momentâneo. Para se resolver esse problema existem implementações de sistemas híbridos que levam em consideração as características iniciais do usuário e recomendam conteúdos avaliados por usuários semelhantes.

3.3 Sistemas de Recomendação Social

Sistemas de Recomendação Social, também conhecidos como sistemas de recomendação para a *web social*, são sistemas de recomendação que visam trabalhar no domínio das mídias sociais. Sistemas desse paradigma lidam constantemente com o desafio de sobrecarga social (tanto de informações quanto de interações com outros usuários) apresentando somente o conteúdo mais atraente e relevante para o usuário naquele momento. Para isso, eles frequentemente utilizam técnicas de personalização adaptadas às necessidades e interesses de um usuário individual, ou de um conjunto de usuários. Os sistemas de recomendação social também têm o objetivo de crescente adoção, engajamento e participação dos usuários (novos e existentes) em *sites* de mídia social.

As mídias sociais e os sistemas de recomendação podem se beneficiar mutuamente, uma vez que as mídias sociais introduzem novos tipos de dados públicos e metadados que podem ser utilizados para melhorar recomendações (como *tags*, comentários, avaliações, relacionamentos sociais — explícitos, implícitos, diretos e indiretos —, etc.), e os sistemas de recomendação, por sua vez, podem significativamente impactar no sucesso das mídias sociais e da *web social* como um todo, assegurando que cada usuário seja apresentado aos itens mais atrativos e relevantes, que satisfaçam suas necessidades pessoais (GUY; CARMEL, 2011).

Exemplos de sistemas de recomendação social incluem o Digg⁷, um *site* de notícias sociais que permite aos usuários votarem no conteúdo da *web* de forma que a notícia seja elevada (*digging*), ganhando destaque no *site*, ou rebaixada, aparecendo no final da listagem de notícias (*burying*); o Google Reader⁸, um agregador baseado na *Web* capaz de ler *feeds*

⁷ <http://digg.com/>.

⁸ <http://www.google.com.br/reader/>.

Atom e RSS (*Web feeds* permitem que programas verifiquem se há atualizações de conteúdo publicadas em um *site*), permitindo também que usuários indiquem conteúdo a outros usuários ou verifiquem o que seus amigos estão lendo atualmente; o YouTube⁹, um *site* de compartilhamento de vídeos em que os usuários podem fazer *upload*, visualizar e compartilhar vídeos entre si, além de receberem recomendações de vídeos baseadas em suas atividades recentes; e além desses, os *sites* de redes sociais, como Facebook¹⁰ ou Orkut¹¹, e *sites* de *microblogging*, como o Twitter¹², também fazem recomendação de amizades (ou seguidores, no caso do Twitter) baseados no perfil e nos círculos sociais do usuário.

Dessa forma, vemos que a rede de amigos e pessoas de interesse do usuário torna-se mais acessível na era da *Web 2.0* através das mídias sociais, principalmente através dos *sites* de relacionamentos pessoais (redes sociais). Então, a utilização de tais relações sociais pode ser muito eficaz para o processo de recomendação, quando em comparação aos métodos de recomendação tradicionais previamente discutidos, uma vez que a recomendação viria de pessoas que o usuário conhece e, portanto, pode julgar; pouparia o trabalho de *feedbacks* explícitos, como em classificações; e seria eficiente para novos usuários, pois esses já entrariam familiarizados com as recomendações de sua própria rede social.

Vários trabalhos na literatura (SINHA; SWEARINGEN, 2001; GOLBECK, 2006; GROH; EHMIG, 2007) têm mostrado a eficácia e eficiência das recomendações sociais (baseadas no círculo social e amizades do usuário) sobre as abordagens de recomendação baseadas puramente em filtros colaborativos. Já trabalhos como o de Guy et al. (2009, 2010) procuram viabilizar a recomendação de itens variados baseados na exploração de relacionamentos nas redes sociais e utilizando pessoas e *tags*. Portanto, vemos que redes sociais articuladas desempenham um papel importante na filtragem colaborativa para as mídias sociais, pois elas podem melhorar o processo de recomendação de várias maneiras, seja através de *tagômetros* (*tags* + redes sociais) ou puramente explorando os relacionamentos contidos em uma rede social.

Um processo de recomendação social típica consiste de três etapas sequenciais: (1) identificação de fontes relevantes para um usuário, (2) a análise dos perfis das fontes de consumo dos itens (indicando que eles gostam), e (3) a geração de recomendações para o usuário, com base nestes perfis (KONSTAN, 2004). Indicadores adicionais de relações sociais, como laços de amizade em redes sociais *online*, são utilizados como fontes para

⁹ <https://www.youtube.com/>.

¹⁰ <http://www.facebook.com/>.

¹¹ <http://www.orkut.com.br/>.

¹² <https://twitter.com/>.

associar um usuário com pares relevantes a fim de que essas recomendações sociais sejam realizadas. Similarmente, o histórico de comunicação dos usuários, que pode ser extraído de aplicações de e-mail e mensagens instantâneas, poderia também ser usado como um indicador adicional de relacionamentos sociais (ZHENG; PROVOST; GHOSE, 2007).

Dos diversos tipos de sistemas de recomendação social existentes, aqueles que mais interessam para o presente trabalho são: (1) os que utilizam algoritmos de *social matching*, ou seja, sistemas de recomendação que utilizam algoritmos de combinação social para recomendar pessoas umas às outras (TERVEEN; MCDONALD, 2005; GUY; RONEN; WILCOX, 2009); e (2) os que utilizam algoritmos de *expertise location*, ou seja, sistemas de recomendação que utilizam algoritmos de detecção de habilidades para ajudar pessoas a encontrarem especialistas em determinado tópico ou área específica (MCDONALD; ACKERMAN, 2000).

Os sistemas do tipo (1) possuem a atração interpessoal como fator de maior importância e deve revelar certa quantidade de informação pessoal, o que acarreta problemas sobre privacidade, confiança e reputação das pessoas envolvidas na rede social em questão. Esses sistemas possuem uma sobrecarga de interação social, pois devem analisar elementos que podem indicar sinais de laços sociais como, por exemplo, participação em grupos, comentários, *tagging* de pessoas, conexões mútuas, compartilhamento de conteúdos, etc., além da usual sobrecarga de informação que os sistemas de recomendação clássicos possuem.

Já os sistemas do tipo (2) são às vezes referidos como sistemas de recomendação de especialistas e possuem como objetivo ajudar os usuários a encontrarem respostas a uma pergunta, receberem conselhos, ter uma conversa com especialistas de determinada área, promoverem ideias, etc. Esses sistemas diferem dos sistemas de recomendação de pessoas tradicionais já que são necessariamente iniciados pelo próprio usuário, que deve formular uma consulta dentro de um determinado contexto (área de especialidade desejada pelo usuário ou tópico/tema que o usuário possua alguma dúvida). Tipicamente, esses sistemas se aplicam somente às necessidades *ad-hoc* dos usuários e são mais semelhantes a máquinas de buscas do que a sistemas de recomendação propriamente ditos.

Ao longo deste capítulo, vimos que os sistemas de recomendação tradicionais ignoram as conexões sociais entre os usuários, desconsiderando as atitudes e o comportamento esperado dos outros membros da rede social do usuário. Para melhorar a precisão da recomendação a rede social dos usuários deve ser analisada, uma vez que as pessoas que estão socialmente conectadas são mais propensas a compartilhar os mesmos interesses ou interesses

similares. Os usuários podem ser facilmente influenciados pelos seus amigos de confiança, e preferem as recomendações de seu círculo de amizades.

Assim, a questão da confiança na recomendação também pode ser abordada através da análise dos relacionamentos sociais do usuário. Relações sociais de confiança, derivadas da rede social do usuário, são adequadas para reforçar a qualidade da informação *online*. A fusão da confiança das informações baseadas em redes sociais e dos sistemas de recomendação pode melhorar a precisão das recomendações, melhorando também a experiência do usuário.

Metodologias de pesquisas científicas que visam o projeto de sistemas de informação, como as de Walls, Widmeyer e El Sawy (1992, 2004), reconhecem a importância das “teorias do kernel”, definidas como teorias explicativas gerais das ciências naturais ou sociais. Dessa forma, o presente trabalho visa aplicar essas metodologias dirigidas à teoria de sistemas de recomendação social para, através de estudos comportamentais dos usuários, seus relacionamentos sociais e algoritmos de *social matching*, melhorar a precisão e confiança do processo de recomendação de pessoas.

Capítulo 4 - Sistemas Baseados em Contexto nos Dispositivos Móveis, as Redes Sociais e o Cenário de Emergência

Os seres humanos são bem sucedidos em transmitir e compreender ideias apropriadamente. Isso se deve a vários fatores: a riqueza da língua partilhada, o entendimento comum do funcionamento do mundo e uma compreensão implícita das situações cotidianas. Quando os seres humanos se comunicam, eles são capazes de utilizar as informações situacionais implícitas, ou contexto, para aumentar a compreensão de uma conversação. Infelizmente, essa capacidade de transmitir ideias não é plenamente transferível dos seres humanos para os computadores quando há interação entre eles, apesar de todo o avanço na área de inteligência artificial.

Em um ambiente computacional interativo tradicional, os usuários possuem um pobre mecanismo de entrada de dados para os computadores. Conseqüentemente, os computadores não são capazes de analisar o contexto do diálogo humano-computador. Dessa forma, vemos como o contexto é uma fonte de informações raramente utilizada em nossos ambientes computacionais. Como resultado, nós temos uma pobre compreensão do que é o contexto e de como ele pode ser usado para aumentar a riqueza da comunicação em interações homem-computador, além de produzir serviços computacionais mais úteis. De acordo com Dey (2001),

Contexto é qualquer informação que pode ser utilizada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, lugar ou objeto que é considerado relevante para a interação entre um usuário e uma aplicação, incluindo o próprio usuário e a própria aplicação. Um sistema é sensível ao contexto se ele usa o contexto para fornecer informações relevantes e/ou serviços para o usuário, onde a relevância depende da tarefa do usuário.

Já Brézillon (1999) apresenta um levantamento da literatura que lida direta e explicitamente com o contexto, independentemente do domínio de aplicação, onde define contexto vagamente como algo “que limita uma solução de um problema sem intervir nela explicitamente”. Seu trabalho deixa claro que o contexto desempenha um papel importante em todos os domínios com atividades como raciocínio e interpretação, e só pode ser capturado pela experiência. Ainda, conclui que o contexto parece possuir uma dupla natureza: estática ou dinâmica, discreta ou contínua, de conhecimento ou de processo; e que esse problema é agravado quando se considera que existem diferentes tipos de contextos interdependentes.

O uso de informações contextuais (por exemplo, localização, tempo, clima, estados mentais da pessoa, etc.) tem um potencial significativo para simplificar a interação do usuário com um sistema complexo (CHEVERST et al., 2000; CHEVERST et al., 2001). Nesse sentido de se utilizar informações contextuais visando produzir serviços computacionais mais úteis, ou até mesmo melhorando serviços computacionais existentes atualmente, encontramos diversos trabalhos interessantes na literatura.

O trabalho de Yap, Tan e Pang (2005) propõe um *framework* genérico que reforça os sistemas de recomendações existentes com um modelo de percepção contextual orientado a tarefas. Eles defendem uma distinção formal entre a instanciação do contexto e o processo de recomendação em si, utilizando uma otimização dinâmica de contextos para se fazer recomendações para alguns usuários em particular.

Já a pesquisa de Adomavicius e Tuzhilin (2001) conclui que em muitas aplicações, tais como para se recomendar pacotes de férias ou restaurantes aos clientes, não é suficiente apenas recomendar itens a usuários ou associar usuários a itens, mas devemos também levar em consideração o contexto para melhor recomendar. Por exemplo, as preferências dos clientes para pacotes de férias específicos podem depender significativamente da época do ano ou de outras considerações temporais, ou seja, informações contextuais.

Portanto, para se fornecer recomendações úteis em tais aplicações, o sistema deve suportar múltiplas dimensões, tais como usuários, itens, tempo, lugar e outras informações contextuais relevantes ao processo de recomendação. Dessa forma, a proposta seria adicionar capacidades de perfil em uma abordagem de recomendação multidimensional. Por exemplo, os usuários seriam definidos não só pelo seu nome, mas por perfis mais ricos que capturassem seu comportamento através de informações contextuais relevantes ao tipo de recomendação.

Ainda no âmbito de utilização de contexto no processo de recomendação podemos citar o Sightplanner (LUBERG et al., 2011), um sistema de recomendação semântico que sugere rota para turistas. Esse sistema possui um processo de extração e recuperação da informação semiautomático, onde os dados são pesquisados em diferentes portais na *internet* e normalizados em uma ontologia própria. O principal desafio nesse cenário é o fato de a informação ser coletada de várias fontes, que geralmente a armazena em formatos proprietários e a disponibiliza em diferentes línguas e graus de precisão. Daí a necessidade da aplicação de algoritmos semânticos para denotar se a informação coletada de diferentes fontes se refere ao mesmo local físico. Nesse sistema, um turista pode especificar sua localização, hora e duração da visita, além de suas preferências sobre diferentes tipos de objetos e eventos.

Com base no perfil de turismo criado, o recomendador identifica objetos de interesse para o usuário. Para cada objeto encontrado uma pontuação de classificação é criada, com os objetos que o usuário provavelmente gosta possuindo uma pontuação mais alta e vice-versa. Um mecanismo de planejamento organiza os objetos e eventos em um calendário de viagem. Ainda, em um ciclo de *feedback* interativo, o turista tem a opção de modificar a viagem sugerida. Para calcular a lista final de objetos recomendados, os seguintes processos estão envolvidos: (a) processo de verificação de objetos, (b) processo de *matching*, (c) processo de planejamento, (d) processo de representação de resultados, e (e) processo de *feedback*.

Cheverst et al. (2001) defende que o uso de contexto (situacional e ambiental) possui um potencial significativo para simplificar a compreensão de um usuário sobre sistemas interativos complexos e melhorar a interação entre eles. A adaptação ao contexto pode ser usada para: simplificar/reduzir a especificação da tarefa exigida do usuário a fim de alcançar seus objetivos desejados, reduzindo a necessidade de entrada de dados ou determinadas ações por parte do usuário; alterar a saída produzida pelo sistema, reduzindo a quantidade de informação que deve ser processada pelo usuário ou aumentando a qualidade da informação apresentada; reduzir a complexidade das regras que constituem o modelo mental do usuário sobre o sistema, migrando a computação do usuário para algum tipo de agente inteligente.

Este agente deve efetivamente antecipar o objetivo do usuário (com base no contexto atual) e então dificuldades podem surgir: o problema de não conseguir alcançar um estado estável; o *trade-off* entre a prescrição e a liberdade; a confiança no agente realizando a adaptação. Por exemplo, os usuários são obrigados a confiar no comportamento da inteligência do sistema, e isso requer que o sistema tenha um comportamento previsível e a capacidade de antecipar com sucesso e consistentemente o objetivo do usuário. Infelizmente, o agente poderia erroneamente presumir o objetivo do usuário, seja por causa de inteligência falha ou por causa de informação contextual incorreta ou fora da validade. Em tais circunstâncias, é bem provável que o usuário se sinta frustrado porque o sistema irá aparecer excessivamente prescritivo ou, pior ainda, apresentar resultados incorretos.

Já Dey, Abowd e Salber (2001) definem contexto, identificam as categorias de informação contextual e caracterizam o comportamento de aplicativos sensíveis ao contexto. Apesar de o impacto da computação sensível ao contexto requerer a compreensão de noções muito sutis e de altos níveis sobre contexto, eles concentram seus esforços em peças de contexto que podem ser inferidas automaticamente a partir de sensores em um ambiente físico. Eles apresentam um *framework* conceitual que separa a aquisição e a representação do contexto da entrega e da reação ao contexto em uma aplicação sensível ao contexto.

Este *framework* é a base para a construção de um conjunto de ferramentas, o Toolkit de Contexto, que apoia o desenvolvimento rápido de várias aplicações sensíveis ao contexto. Por fim, a utilidade desse *framework* é ilustrada através uma série de aplicações sensíveis ao contexto que foram prototipadas usando o Toolkit de Contexto, e também é demonstrado como esse *framework* pode apoiar a investigação de importantes desafios para a pesquisa na área de computação sensível ao contexto.

Outros trabalhos na literatura (STRANG; LINNHOFF-POPIEN, 2004; BALDAUF; DUSTDAR; ROSENBERG, 2007) descrevem os modelos existentes para modelagem de informação contextual e representação do contexto em sistemas sensíveis ao contexto:

- **Modelos Chave-Valor:** Esses modelos representam a mais simples estrutura de dados para a modelagem de informação contextual. Em particular, os pares chave-valor são fáceis de gerenciar, mas faltam recursos para estruturação sofisticada a fim de permitir algoritmos eficientes de recuperação do contexto. Eles são frequentemente usados em estruturas de serviços diversos, onde os pares chave-valor são usados para descrever as capacidades de um serviço. Serviços de descoberta são então aplicados por meio de algoritmos correspondentes, que utilizam esses pares chave-valor.
- **Modelos de Esquema de Marcação:** Comum a todas as abordagens de modelagem de esquema de marcação está uma estrutura hierárquica de dados composta por *tags* de marcação com atributos e conteúdos. Em particular, o conteúdo das *tags* de marcação é geralmente definido de forma recursiva por outras *tags* de marcação. Representantes típicos desse tipo de abordagem de modelagem de contexto são perfis. Eles geralmente são baseados em uma serialização de derivações do SGML, a superclasse de todas as linguagens de marcação como o popular XML.
- **Modelos Gráficos:** Um instrumento de modelagem de uso geral muito conhecido é a UML, que tem um forte componente gráfico (diagramas UML). Devido à sua estrutura genérica, o UML é também apropriado para modelar o contexto e existem várias abordagens que o utilizam.
- **Modelos Orientados a Objeto:** Abordagens desse tipo têm a intenção de empregar os principais benefícios de uma abordagem orientada a objetos — ou seja, o encapsulamento, a reutilização e a herança — para cobrir parte dos problemas decorrentes da dinâmica do contexto em ambientes ubíquos. Os

diferentes tipos de contexto, suas representações e detalhes de processamento são encapsulados em nível de objetos e, portanto, escondidos para outros componentes. O acesso à informação contextual e à lógica de processamento do contexto é fornecido somente através de interfaces especificadas.

- **Modelos Baseados em Lógica:** A lógica define as condições em que uma expressão de conclusão ou fato podem ser derivados (um processo conhecido como raciocínio ou inferência) de um conjunto de outras expressões ou fatos. Para descrever essas condições em um conjunto de regras, um sistema formal é aplicado. Em um modelo de contexto baseado em lógica, o contexto é consequentemente definido como fatos, expressões e regras. A informação contextual precisa ser representada de uma maneira formal como fatos, e geralmente é adicionada, atualizada e excluída de um sistema baseado em lógica em termos de fatos ou inferida a partir das regras do sistema. Comum a todos os modelos baseados em lógica está um alto grau de formalidade.
- **Modelos Baseados em Ontologias:** Ontologias representam uma descrição dos conceitos e relacionamentos. Elas são um instrumento muito promissor para a modelagem de informação contextual devido a sua alta e formal expressividade e às possibilidades de aplicação de técnicas de raciocínio em ontologias. Elas são particularmente adequadas para projetarem as partes da informação que descrevem e são usadas na vida diária de uma pessoa em uma estrutura de dados utilizável por computadores. Vários *frameworks* sensíveis ao contexto utilizam ontologias como modelos contextuais subjacentes.

Dessa forma, vemos que as aplicações sensíveis ao contexto se tornarão reais extensões autônomas com uma fonte central de informações que serão atualizadas em tempo real. Isso supõe a capacidade de gerenciar: informações que evoluem com o tempo (por exemplo, horário para visitar um castelo); informação vinda de fontes heterogêneas (por exemplo, condição climática, hotéis, imprensa); conhecimento sobre os usuários (incluindo suas preferências e perfis percebidos pelo sistema por meio das ações dos usuários); *software* com uma parte central e partes móveis conectadas a um usuário ou grupo de usuários; e *hardware* e a interação entre diversas peças de *hardware* (BRÉZILLON, 2003).

Com esse modelo em mente, a proposta do presente trabalho define em seu arcabouço de solução a utilização de informações contextuais provenientes tanto de dispositivos móveis quanto das mídias sociais a fim de melhorar o processo de recomendação de pessoas. Para que pudéssemos formalizar melhor esse arcabouço e construir uma ferramenta de recomendação

de pessoas baseada nele, foi de crucial importância o entendimento da utilização do contexto em sistemas complexos que tivessem como fonte as informações contextuais presentes nos dispositivos móveis e nas mídias sociais. Portanto, nas seções a seguir mostraremos alguns trabalhos que nos serviram de inspiração para a construção de nosso arcabouço de solução.

4.1 Sistemas Baseados em Contexto nos Dispositivos Móveis

Nos últimos anos, os dispositivos móveis (tablets, celulares, PDAs e dispositivos híbridos como Pocket PC, Blackberry e smartphones) têm cada vez mais penetrado tanto na vida profissional quanto na vida familiar das pessoas. A maioria desses dispositivos móveis é de natureza pessoal, ou seja, permanecem e viajam junto com uma pessoa a maior parte do tempo e, assim, compreendem os vários contextos sociais dessa pessoa (por exemplo, quando estão junto com os colegas de trabalho, os amigos, os membros da família, outros alunos, com outros membros de uma equipe de esporte, com outros visitantes em uma excursão, etc.).

A fim de explorar as possibilidades de adoção da tecnologia de dispositivos móveis em mercados emergentes, Lang, Oreglia e Thomas (2010) realizaram estudos etnográficos para compreender as práticas sociais e uso dessa tecnologia entre os jovens trabalhadores migrantes na China. Os celulares foram rapidamente adotados e frequentemente usados para iniciar, manter e até mesmo melhorar as conexões sociais (ou seja, para se comunicarem com a família e amigos, mantendo ou ampliando a rede social). Eles também demonstraram um papel importante no que se refere à qualidade das práticas sociais, mostrando-se valiosos pela capacidade de encontrar lugares interessantes para sair com os amigos e pelo suporte ao entretenimento compartilhado com os colegas de trabalho.

Portanto, os recentes avanços em miniaturização de tecnologias móveis e portáteis, que produzem dispositivos móveis a preços acessíveis com alta conectividade sem fio, poder computacional e capacidade de identificação posicional, e a crescente adoção dessas tecnologias pela população proporcionam várias oportunidades para se capturar informações contextuais de uma forma não intrusiva. Para a finalidade desta pesquisa, separamos essas informações contextuais em três tipos distintos:

- **Proximidade Física:** os padrões dinâmicos de proximidade física em uma população podem ser capturados através de dispositivos portáteis simples, explicitamente projetados para este fim, como o dispositivo chamado de Sociômetro (CHOUDHURY; PENTLAND, 2002) que incorpora elementos baseados em infravermelho e acelerômetros para rastrear interações sociais dentro de uma organização, ou também se utilizando dispositivos móveis, já

que atualmente esses tipos de aparelho já possuem sensores infravermelhos e acelerômetros embutidos até mesmo em modelos mais comuns. Além disso, torna-se cada vez mais viável a utilização de técnicas de curto alcance baseadas em ondas de rádio, presentes em dispositivos móveis contemporâneos, para esta finalidade. Exemplos incluem o sistema baseado em Bluetooth chamado BlueAware (EAGLE; PENTLAND, 2004), que pode ser executado em um smartphone contemporâneo e sistemas baseados em WLAN, como o PLIM (PEDDEMORS; LANKHORST; DE HEER, 2003) e Reality Mining (EAGLE; PENTLAND, 2003), que funcionam em PDAs. Outro exemplo de sistema sensível à proximidade física é o MEK – Mobile Exchange of Knowledge (MONCLAR et al., 2009; SOUZA et al., 2011), que é uma abordagem móvel de troca de conhecimento entre pessoas que compartilham os mesmos interesses. Nele, os objetos são transmitidos de forma pró-ativa via frequência de rádio de curto alcance. A disseminação dos dados realizada entre os dispositivos móveis que possuem esse sistema é baseada em informações de contexto e características do perfil do usuário.

- **Localização Física:** através do registro da localização física e posterior correlacionamento desses registros podemos obter indiretamente a proximidade física entre as pessoas. Por exemplo, se o usuário A estiver no Rio de Janeiro e o usuário B estiver em São Paulo, basta calcularmos a distância entre esses dois pontos (Rio de Janeiro – São Paulo) e assim obteremos a proximidade física entre os usuários A e B. Nos últimos anos, várias técnicas de sensoriamento local tornaram-se disponíveis, com base em receptores de infravermelho, identificação por radiofrequência, pequenos receptores GPS e, recentemente, serviços de localização fornecidos por redes de telefonia móvel (HIGHTOWER; BORRIELLO, 2001). O registro da localização física em si pode fornecer o contexto em que a pessoa se encontra e auxiliar no planejamento de rotas, ou ainda expor uma relação dinâmica entre a localização das pessoas e sua rede social. Apesar de atualmente a grande maioria dos dispositivos móveis possuírem GPS integrado, nem todas as questões de pesquisas requerem o registro de localizações físicas com precisão de GPS. Em algumas situações, pode ser suficiente apenas registrar o Cell-ID de uma rede de telefonia celular em que um celular está conectado para se

identificar a região onde o usuário se encontra, ou ainda reduzir essa região de localização até uma precisão pretendida através de técnicas de triangulação.

- **Interação Social:** talvez ainda mais importante do que registrar a proximidade física é a capacidade de registrar quando as pessoas estão se comunicando, com quem e por quanto tempo. Os padrões de interação social por meios eletrônicos de comunicação, por exemplo, e-mail, telefonia móvel (ligações e troca de mensagens SMS/MMS) e mensagens instantâneas são relativamente fáceis de detectar e capturar (GARTON; HAYTHORNTHWAITE; WELLMAN, 1997; DE VOS; HOFTE; DE POOT, 2004; FISHER; DOURISH, 2004). Em sistemas do tipo mensageiros instantâneos, ainda é tecnicamente viável se registrar quais os grupos de contatos estão abertos e quais estão fechados, dando a impressão de que partes de uma rede social são atualmente relevantes.

Com a melhora da tecnologia de conectividade sem fio, os usuários podem acessar as informações em ambientes móveis mais rapidamente. Com uma capacidade computacional mais poderosa, os usuários podem processar mais dados nesses pequenos dispositivos. E, finalmente, com capacidade de identificação posicional, os usuários são capazes de acessar informações e processar dados que estão relacionados com suas localizações físicas.

Dessa forma, podemos construir agentes recomendadores em ambientes móveis sensíveis ao contexto, para auxiliar os usuários enquanto eles estão se movendo. Muitos tipos diferentes de contexto podem ser usados por um agente recomendador, tais como o clima, as condições de rotas, tempo (data/hora) e local, etc. Nesse sentido, Tung e Soo (2004) discutem como um agente recomenda informação de acordo com os contextos do usuário. Eles criaram um agente recomendador residente em um pequeno dispositivo móvel que pode recomendar restaurantes ao usuário com base no contexto.

O processo de recomendação é estabelecido em cinco fases distintas: aquisição de requisitos do usuário, identificação do contexto do usuário, busca de informações adequadas, resolução das violações de restrição, e a recomendação em si. O contexto identificado e utilizado pelo agente contém as preferências do usuário, o tempo (data/hora) e contextos espaciais. Porém, os autores reconhecem que existem vários outros contextos que poderiam ser utilizados para melhorar o processo de recomendação. Por exemplo, eles citam que o agente poderia medir o clima através de um termômetro, ou ainda procurar sites que forneçam informações sobre o clima.

Outro agente recomendador para dispositivos móveis é o MobyRek (RICCI; NGUYEN, 2007), que ajuda os usuários na busca por produtos em uma determinada área. Ele suporta perguntas e respostas de questões limitadas, baseadas principalmente em críticas. Para assegurar que as recomendações sejam relevantes às necessidades do usuário, e que a interação do usuário seja simplificada, o modelo integra preferências de longo prazo e específicas de uma sessão, integra as preferências coletadas através de um sistema recomendador baseado na *web*, e explora uma abordagem de conversação baseada em críticas.

O sistema deriva as preferências estáveis ou de longo prazo do usuário, tais como a preferência por ambientes para não fumantes, a partir de várias sessões de interação recentes com o usuário. Estas preferências estáveis permanecem fiéis ao longo das sessões. Em contraste, as preferências específicas de sessões, tais como o desejo de comer pizza, são transitórias. As preferências específicas de sessão incluem tanto as preferências contextuais (tais como restrições de tempo-espaço) e preferências de características do produto.

As preferências contextuais caracterizam o contexto do pedido do usuário, enquanto as preferências de características do produto expressam o gosto do usuário. Na questão de recomendação de restaurantes, por exemplo, as restrições de espaço-tempo garantem que o restaurante recomendado está aberto no dia do pedido e não está muito longe do usuário. As preferências de características do produto podem indicar que o usuário, por exemplo, busca restaurantes de baixo custo, ou então prefere comer pizza.

No cálculo geral do processo de recomendação desse sistema, as características que são mais frequentes ou mais recentemente criticadas têm pesos maiores. Além disso, para fazer as sugestões mais úteis, o sistema ainda possui uma função de roteamento mostrando o caminho a partir da posição do usuário para o restaurante selecionado.

Diversas pesquisas também têm sido desenvolvidas ultimamente para se aproveitarem as informações contextuais coletadas através de dispositivos móveis no cenário de gestão de emergências. Uma aplicação que merece destaque nesse campo é o MobileMap (MONARES et al., 2009), que permite a troca de informações textuais e gráficas usando dispositivos móveis. Essa aplicação lida com vários dos problemas principais de projeto que foram identificados como sendo cruciais no domínio da gestão de emergência: *accountability*, avaliação, alocação de recursos e comunicação. A *accountability* é implementada ao fornecer informações sobre a localização de caminhões de incêndio e de equipamentos, auxiliando também na tarefa de alocação de recursos. O sistema melhora o suporte à comunicação acrescentando redundância e reduzindo as comunicações via rádio.

Outra abordagem interessante utilizando-se dispositivos móveis no auxílio à gestão de emergências é uma proposta de arquitetura de *software* feita por Neyem, Ochoa e Pino (2008). Essa arquitetura é chamada de *Service-Oriented Mobile Units* (SOMU) e possui a finalidade de descentralizar os dados e serviços em redes móveis, as *Mobile Ad-hoc Networks* (MANETs). Cada unidade dessa rede móvel funciona autonomamente, provendo e consumindo serviços de baixo acoplamento com outras unidades, de forma a contribuir para a colaboração sob demanda entre pessoas dispersas, inclusive de equipes distintas. A plataforma SOMU fornece soluções para lidar com a maioria dos requisitos das aplicações de *groupware* móveis em ambientes *ad-hoc*: autonomia, interoperabilidade, disponibilidade de informações compartilhadas, variabilidade do contexto de trabalho, uso de recursos de *hardware*, colaboração discricionária, compartilhamento de informações sob demanda, sincronização de informações sob demanda, baixo custo de coordenação e percepção do alcance dos usuários.

Por fim, Bello et al. (2007) julga a comunicação na resposta a emergências como um recurso vital, mas frágil em sua organização e tecnologia. Em seu trabalho é apresentado um fluxo de comunicação para organizar a troca de informações e garantir que as infraestruturas móveis ubíquas possam funcionar corretamente. A partir desse fluxo de comunicação um sistema de informações para situações de emergência baseado nas diretrizes do projeto *Aplicación en Red para Casos de Emergencia* (ARCE) e utilizando o *framework* para computação ubíqua UILE é construído. Com isso, pode-se organizar e garantir o fluxo de informações entre a equipe de resposta, equipada com PDAs, e os escritórios móveis, que possuem infraestrutura de redes e são montados em áreas próximas aos desastres, de modo a atuar como servidores para as equipes móveis distribuídas.

4.2 Sistemas Baseados em Contexto nas Mídias Sociais

As redes sociais representam grupos de pessoas e as conexões entre elas. As vantagens oferecidas pela análise de redes sociais têm resultado em uma crescente utilização das mídias sociais para a compreensão de uma série de pequenas ou até mesmo grandes interações em grupo. A modelagem e análise de uma rede social foca em padrões estruturais de laços entre atores, que são tipicamente pessoas, mas também podem ser organizações. As unidades básicas de análise são os laços (ou relações) e as conexões transitivas entre esses atores.

A análise de redes sociais é baseada na noção intuitiva de que esses padrões de interação social são características importantes da vida das pessoas que os possuem. Os analistas de redes sociais acreditam que o modo como um indivíduo vive depende em grande parte de como esse indivíduo está ligado nessa rede de conexões sociais. Além disso, muitos

acreditam que o sucesso ou o fracasso das sociedades e organizações muitas vezes dependem da padronização de suas estruturas internas.

A estrutura de uma rede social pode ser expressa em várias medidas quantitativas, incluindo o alcance, a densidade, a centralidade, os grupos e posições. A maioria dos modelos de redes sociais visualiza a estrutura de rede como uma propriedade estática de uma rede social. As redes sociais podem ser obtidas de duas maneiras: abordagens sociocêntricas (focadas na sociedade como um todo) e egocêntricas (focadas em um indivíduo específico).

Primeiramente, a abordagem sociocêntrica considera toda uma rede com base em algum critério específico de limite populacional, como uma organização formal. Uma rede inteira descreve os laços que todos os membros de uma população mantêm com todos os outros no grupo. Embora esse método esteja disponível para lidar com conjuntos de dados incompletos, essa exigência limita o tamanho das redes sociais que podem ser examinadas. Em segundo lugar, a abordagem egocêntrica considera as relações relatadas por um indivíduo local. Esta abordagem é particularmente útil quando a população é grande.

Parte do sucesso das redes sociais pode ser atribuída ao fenômeno dos “seis graus de separação”, o que significa que a distância entre quaisquer dois indivíduos em termos de relações pessoais diretas é relativamente pequena. Além disso, as redes sociais fornecem um meio de visualizar a interação existente e potencial em diversos tipos de ambientes. Apesar das redes sociais se originarem de uma disciplina descritiva e analítica, atualmente há uma tendência a incorporar as redes sociais em sistemas computacionais para fins diversos. Particularmente, os projetistas e desenvolvedores de groupware estão usando as redes sociais para tornar os sistemas mais sensíveis a situações sociais, visualizar as conexões de pequenos ou grandes grupos, e orientar os usuários para colaborações eficientes e efetivas.

McDonald (2003) mostra que diversos estudos de comportamento colaborativo muitas vezes acham que a rede social de um indivíduo influencia fortemente a busca por informações e o comportamento de colaboração com o grupo. Nesses estudos, as redes sociais descrevem uma ampla e complexa gama de interação social. Esses resultados demonstram a importância da sociabilidade, do compartilhamento de contexto e da localização física, dentre outras propriedades, como uma influência sobre a atividade colaborativa.

Ainda nesse trabalho, McDonald examina como os usuários interpretam um sistema de recomendação socialmente sensível. Seu trabalho leva em consideração que as redes sociais embutidas em sistemas computacionais requerem uma avaliação baseada no usuário, de forma semelhante a qualquer outro sistema ou componente de interface com o usuário. Por fim, seus resultados levantam uma série de questões para o uso de redes sociais nos sistemas de

groupware. Parece claro que as pessoas têm sentimentos mistos sobre redes sociais como uma ferramenta para encontrar colaboração.

Dentre os sistemas que usam as redes sociais como um mecanismo para a recomendação de pessoas específicas para a colaboração, podemos citar o ReferralWeb (KAUTZ; SELMAN; SHAH, 1997), onde os relacionamentos de coautoria e co-citação são extraídos para criar uma rede social que é usada para encontrar um possível especialista. Essas evidências de relacionamento são retiradas de forma incremental através de documentos publicamente disponíveis na *internet*, como *links* em páginas pessoais, por exemplo. Após um usuário se registrar no sistema, uma busca na *web* é realizada para recuperar documentos que o mencionam e, então, nomes de outros indivíduos são extraídos desses documentos.

Esse processo é realizado com elevado grau de precisão (acima de 90%), utilizando técnicas específicas. Ele é aplicado de forma recursiva para um ou dois níveis, e o resultado é combinado em um modelo de rede global. Essa cadeia de referências serve para duas funções principais: fornecer um motivo para o especialista concordar em responder ao solicitante, tornando sua relação explícita (por exemplo, eles possuem um colaborador em comum), e fornecer um critério para o pesquisador utilizar na avaliação da confiabilidade do especialista. A rede social criada através do ReferralWeb também pode responder a consultas sobre a que distância (em conexões pessoa-pessoa) um pesquisador está de outro e quem está entre eles.

Ogata et al. (2001) também usa redes sociais para facilitar a busca por pessoas com quem colaborar. Seu trabalho foca na mediação de conexões pessoais (*Personal Connection — PeCo*) através da mineração e análise de trocas de e-mails entre indivíduos. A mineração e construção da rede social se baseiam em um tipo de modelo de comunicação por atos de discurso (declarações que contêm as informações necessárias para fazer valer e realizar ações). Um usuário pode encontrar outro ao realizar diferentes consultas, mas ele não possui uma visualização completa da rede social. O sistema construído, PeCo-Mediator-II, é um sistema *web* distribuído constituído de um banco de dados pessoal (PeCo-Collector) e um agente de *software* (PeCo-Agent).

O PeCo-Collector incrementalmente reúne informações sobre os conhecidos do usuário e suas conexões analisando trocas de e-mail. Já o PeCo-Agent conecta-se aos *sites* dos colegas e negocia com outros agentes e usuários para encontrar colaboradores. O sistema foi avaliado através de um experimento de campo seguido de um questionário. No experimento, estudantes de pós-graduação foram disponibilizados (através de e-mail) para ajudar estudantes de graduação matriculados em um curso sobre programação em C. Resumidamente, os resultados da pesquisa descobriram que os estudantes estimaram o sistema por ajudá-los a

encontrarem ajuda, mas não sentiram que o sistema refletia com exatidão a proximidade social dos seus amigos no curso de programação.

No âmbito de utilização das mídias sociais para se extrair informações contextuais e melhorar os processos de recomendação, vemos que ferramentas desse tipo, como o Facebook, por exemplo, permitem acesso às informações pessoais do usuário, tais como sexo, idade e país, além de sua lista de amigos. Nesse sentido de se explorar as informações contextuais e relações sociais disponíveis em sites de redes sociais, foi criada uma aplicação chamada PersonalTV (DE PESSEMIER; DERYCKERE; MARTENS, 2009) que utiliza tais informações para realizar uma filtragem adicional do conteúdo recomendado (programas de televisão) e explora as relações sociais, transformando o processo de consumo em uma atividade social onde os usuários obtêm um papel mais interativo.

Os autores citam que os sistemas de recomendação tradicionais usam um perfil de usuário que reflete seu gosto pessoal, mas não levam em consideração outros critérios como características temporais, geoespaciais ou emocionais. Essas características, que podem ser capturadas através das informações contextuais, são mais dinâmicas, mudando durante o dia. Por exemplo, na parte da manhã, um usuário pode não querer assistir a filmes, mas, em vez disso, notícias ou a previsão do tempo.

Além disso, os hábitos de uma pessoa no fim de semana podem diferir daqueles durante a semana, e as pessoas também preferem uma oferta de conteúdo diversificado quando estão no trabalho, em casa, etc. Ao investigar os padrões de comportamento do usuário e o contexto de consumo associado pode-se ter um *insight* dos hábitos dependentes do contexto do usuário e seu comportamento de consumo específico. Portanto, os autores, movidos por essa visão, armazenaram esse contexto como uma fonte de informação extra de seu sistema de recomendação.

O trabalho de Hofte e Mulder (2003) mostra por que é relevante ampliar a perspectiva de rede social pessoal no trabalho colaborativo para uma visão dependente de contexto mais dinâmica que eles chamam de “perspectiva de rede social pessoal dinâmica”. Na visão dos autores, o objetivo de elaborar ferramentas de redes sociais pessoais dinâmicas deve ser o de melhorar os processos em sistemas sócio-técnicos (compostos de seres humanos e tecnologia) como um todo em vez de projetar o aplicativo de computação sensível ao contexto mais inteligente e preciso que existe (ou seja, otimizando somente a tecnologia). As aplicações de redes sociais pessoais dinâmicas são aquelas que sentem, mediam, visualizam e, em alguns casos, interpretam as informações contextuais das redes sociais pessoais dinâmicas como parte do serviço que é oferecido ao usuário final.

Os métodos para medir redes sociais pessoais dinâmicas podem ajudar a selecionar quais informações de contexto devem ser transmitidas ou agregadas a outros usuários humanos, e quais informações contextuais podem ser interpretadas de forma confiável o suficiente por computadores. Para os usuários humanos da aplicação, as informações de contexto, por um lado, precisam fornecer um valor agregado (satisfazendo também requisitos de privacidade, confiança, segurança e custo), mas, por outro lado, essas informações contextuais fornecidas pelo aplicativo podem ser apenas uma fonte de informações de contexto que os seres humanos usam para tomar decisões em um determinado cenário.

O *microblogging* é uma forma de mídia social que está sendo adotada rapidamente como um meio de comunicação por oferecer maneiras de recuperar, produzir e difundir informações facilmente. A natureza do compartilhamento dessas informações tem um ciclo de vida de produção e consumo de informação rápido e repetitivo (STARBIRD et al., 2010). O Twitter é um serviço de *microblogging* popular, através do qual os usuários enviam mensagens curtas de até 140 caracteres, chamados de *tweets*, a partir de clientes *web* ou móvel. Os usuários possuem perfis pessoais públicos ou privados que podem incluir dados básicos como nome, localização e informações biográficas.

Os perfis públicos e mensagens enviadas por esses perfis podem ser acessadas livremente e são legíveis por qualquer pessoa com acesso à *internet*, enquanto os perfis privados podem ser acessados apenas por aqueles com permissão de visualização. Os usuários desse serviço estabelecem uma rede social ao “seguirem” outros usuários do Twitter e possuem usuários que “os sigam” também. Cada vez mais, o *microblogging* está sendo considerado como um meio de comunicação para emergências, por causa de sua onipresença crescente, rapidez de comunicação e acessibilidade multi-plataforma.

Nesse sentido, duas pesquisas merecem destaque por avaliar o papel de atuação do Twitter em cenários de emergências. Em uma delas, Vieweg et al. (2010) apresentam uma análise de dados do Twitter relativos a geolocalização, referenciamento de localidades e informações de atualização da situação em dois conjuntos de dados distintos, baseados em desastres naturais ocorridos durante a primavera de 2009 nos Estados Unidos da América.

O fato de os *twitteiros* (usuários do Twitter) nos dois eventos estudados transmitirem tipos semelhantes de informação, mas em diferentes graus dependendo do tipo de emergência, chama a atenção. Essa análise de dados do Twitter durante os eventos identificou os recursos de informação gerados durante emergências, levando ao desenvolvimento de um esboço de *framework* que identifica características adicionais de *tweets* para informar a concepção e

desenvolvimento de sistemas que empregam estratégias de extração de informações com a finalidade de melhorar a percepção situacional de uma emergência.

Já Sutton (2010) foca seu estudo nos participantes dessa rede social e na análise do conteúdo que eles disponibilizaram, ligado a um incidente específico que chamou a atenção de um grupo emergente de atores geograficamente distribuídos. Todas as buscas e recuperações de dados foram baseadas em informações disponíveis publicamente no Twitter. Esse tipo de dado permite pesquisa por palavras-chave, incluindo aquelas que são organizadas por *hashtags* (#), e por usuários individuais. A pesquisa demonstrou que a comunicação via redes sociais em emergências suporta atividades distribuídas e coordenadas que podem trazer benefícios diretos para a comunidade geograficamente baseada em torno do local do acidente.

O Twitter serviu como um meio para alertar sobre o desastre, para melhorar a percepção situacional do problema e contabilizar os efeitos após o evento de desastre. Os participantes se identificaram como especialistas e proveram comentários sobre os efeitos nocivos à saúde e ao meio ambiente causados pelo acidente. Também foi possível questionar e proporcionar a correção de informações enganosas que vieram de fontes oficiais (como agentes públicos e representantes da indústria). Enquanto os indivíduos e organizações diretamente afetadas pelo desastre, bem como os responsáveis pelo evento, pouco participaram dessa discussão *online*, a comunidade virtual convergiu para oferecer suporte através dessa mobilização de informações em tempo real.

Por fim, é sugerido que pesquisas adicionais são necessárias para se avaliar a extensão em que essas novas mídias estão sendo adotadas nas comunidades locais e como elas são utilizadas em respostas a eventos de crise. Além disso, uma investigação comparativa também deve ser conduzida, procurando por tipo de risco, localização geográfica, o alcance do impacto, e a ocorrência do evento, a fim de identificar padrões de troca de informações e o efeito que a comunicação oficial exerce no diálogo *online*. Também é apontado que a automatização do alerta e captura de dados sobre desastres em tempo real é um importante mecanismo que pode auxiliar os pesquisadores e profissionais a identificar os fluxos de dados relevantes e realizar uma rápida análise de resposta no combate a emergências.

Essas pesquisas sobre informações contextuais nos forneceram importantes ferramentas para trabalharmos um *framework* sobre o qual nossa proposta de solução foi construída. No capítulo a seguir formalizaremos essa nossa proposta de solução, mostrando o *framework* idealizado para a construção de nossa ferramenta como prova de conceito deste trabalho.

Capítulo 5 - Proposta

A proposta de solução que visamos atingir no presente trabalho é a criação de um processo efetivo e eficiente que busque a disponibilidade de informações confiáveis e íntegras sobre as vítimas de um determinado cenário-problema no domínio de emergências. O foco dessa proposta é fornecer um modo de minimizar o trabalho da equipe de resposta a emergências na coleta de informações sobre as vítimas de um acidente, ao mesmo tempo em que tenta maximizar a confiabilidade dessas informações (recomendando apenas aquelas pessoas que são mais relevantes para esse processo de coleta de informação, ou seja, as pessoas mais próximas da vítima), acelerando assim o processo de resgate da vítima. O objetivo principal deste trabalho é o processo de recomendação de pessoas, que é baseado na relevância social entre elas e a vítima. Para tanto, esse processo deverá ser capaz de receber como entrada o nome de uma vítima de emergência e fornecer como saída um conjunto de pessoas que possam contribuir com informações úteis e de qualidade sobre a vítima. A visão geral deste processo que procuramos criar é mostrada na Figura 7.

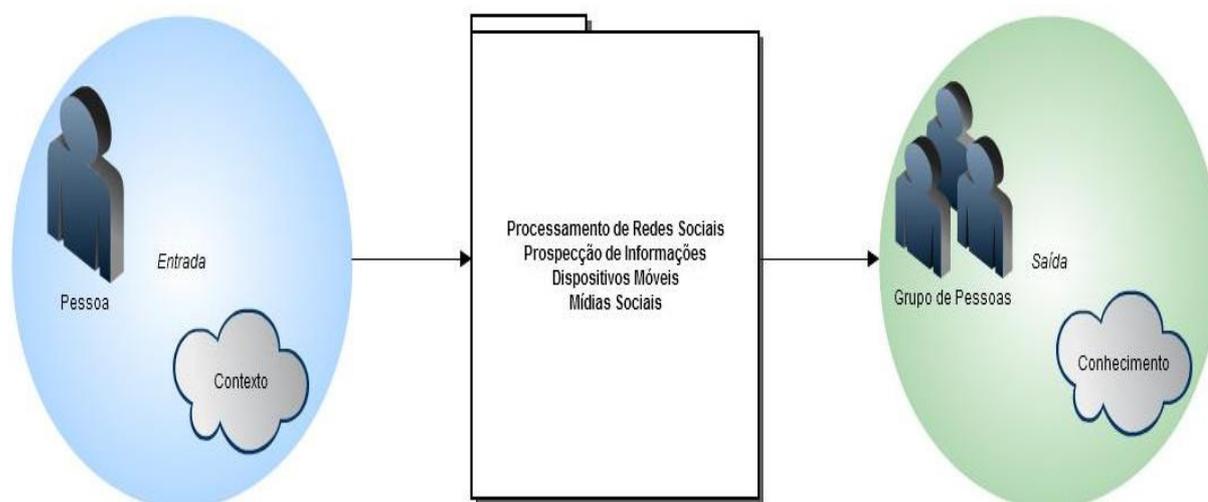


Figura 7. Visão geral do processo da proposta de solução

Esse processo, de posse do nome da vítima, começará a procurar nos arredores da emergência por pessoas que conheçam a vítima e que possam prestar maiores informações sobre ela. Assim, esperamos que através da análise contextual da rede social de uma vítima de emergência consigamos recomendar pessoas próximas a ela e que possam contribuir com informações úteis para a equipe de resposta à emergência. Com base nesse processo, construiremos uma ferramenta móvel que será capaz de auxiliar as equipes de resposta em um determinado cenário-problema. As funcionalidades dessa ferramenta estarão voltadas para a importância da informação contextual na fase de resposta a uma emergência. Na próxima

seção focaremos no cenário de uso de nossa proposta de solução, para então poder descrevê-la em maiores detalhes.

5.1 Organizações e o Cenário de Emergência

Antes de explicarmos o cenário de uso de nossa proposta de solução, faz-se necessário o entendimento da estrutura das organizações de gestão de emergência. No Brasil, devido ao momento e à forma em que foram instauradas, as organizações de resposta a emergências normalmente se baseiam em princípios, normas e estruturas militares. Por exemplo, os primeiros responsáveis pelo combate a incêndios surgiram durante o Império de D. Pedro II e pertenciam ao Arsenal de Marinha. Em 1880, já com um Corpo de Bombeiros formado, foi promulgado o decreto que concedia condição militar aos oficiais do Corpo (CBMERJ, 2012).

A Defesa Civil também apresenta uma estrutura muito semelhante à militar. Os países envolvidos com a II Guerra Mundial foram os responsáveis por criarem as primeiras organizações desse tipo. O Brasil também estabeleceu, em 1942, o Serviço de Defesa Passiva Antiaérea que, no ano seguinte, teria sua denominação alterada para Serviço de Defesa Civil (BRASIL, 2012b). Dessa forma, princípios como o de hierarquia e cadeia de comando, ordem e disciplina, centralização e poder são encontrados nas estruturas dessas organizações. Esses princípios são alguns dos principais a formarem a base da estrutura de Comando e Controle. Essa estrutura atribui o poder e a responsabilidade do comando a um só indivíduo e determina suas responsabilidades na direção e controle de um grupo de trabalho.

Seja no cenário militar, em operações de guerra, ou no urbano, em atividades de resposta a emergências, essa estrutura prevê, de forma simplificada, a divisão dos indivíduos em duas categorias ou grupos: o Comando e a Operação. Essas equipes precisam colaborar entre si e compartilhar informações a fim de obterem êxito em suas atividades. A equipe de Operação é formada por indivíduos treinados, responsáveis pela realização do socorro. Esse grupo trabalha na linha de frente da emergência, possuindo tarefas como o resgate de vítimas, a realização de atendimentos médicos, a orientação de pessoas, entre outras.

Suas atividades são intensamente colaborativas e precisam de alto grau de coordenação a fim de se obter os resultados almejados sem comprometer a segurança própria. As principais características desse grupo são: alta mobilidade, uma vez que carregam consigo apenas os equipamentos básicos necessários ao combate da emergência; capacidade restrita de comunicação, usando normalmente rádios ou celulares como principais formas de comunicação; visão local da emergência, com um acesso extremamente restrito aos dados (somente informações locais); baixa disponibilidade de tecnologias, recursos e equipamentos.

Próximos ao local da ocorrência estão os Comandantes de Operação (ou Comandantes de Incidente). De acordo com o Sistema de Gestão de Incidentes Nacionais dos Estados Unidos — National Incident Management System — (NIMS, 2008) eles são responsáveis pelo desenvolvimento de estratégias e táticas, pela distribuição de ordens às equipes e por gerir todas as operações realizadas no local da emergência. Seu trabalho é realizado no Posto de Comando, que pode ser uma instalação móvel montada ou um veículo estacionado na parte frontal da ocorrência. Essas unidades móveis possuem acesso restrito aos dados da ocorrência e certa disponibilidade de tecnologias de comunicação, recursos e equipamentos.

Ainda nos Postos de Comando pode haver a presença de especialistas para auxiliar nas decisões tomadas por um comandante de operação, que são mais relacionadas à dinâmica dos acontecimentos e devem ser tomadas muito rapidamente para salvar vidas ou evitar o agravamento dos problemas. Essas decisões são geralmente baseadas em poucas informações, que apresentam alto grau de incerteza. Dessa forma, pode-se dizer que esse grupo possui uma visão geral da emergência.

Acompanhando todo este trabalho de combate à emergência estão os indivíduos da Sala ou Centro de Comando. Esse grupo é formado por autoridades, oficiais e comandantes das organizações envolvidas e especialistas nas áreas necessárias ao combate à emergência. Sendo assim, suas avaliações e escolhas são mais focadas no planejamento de atividades para a estabilização da situação como um todo e em um prazo ligeiramente mais extenso.

Além de informar as ordens a serem executadas e fornecer informações que possam ser úteis ou que tenham sido solicitadas, o comando monitora o desdobramento dos fatos, avaliando as mudanças ocorridas na situação com o intuito de planejar os próximos passos. Devido ao seu distanciamento do local da ocorrência, sua percepção situacional é construída a partir dos relatos que lhe são enviados pelas equipes de operação. As principais características desse grupo são: alta disponibilidade de equipamentos de comunicação e recursos tecnológicos; vasta quantidade de dados e acesso rápido e ilimitado às informações das ocorrências; visão holística da emergência e seu desdobramento.

O cenário de uso planejado considera que as equipes de trabalho precisam obter informações confiáveis e íntegras sobre as vítimas da emergência a fim de planejar melhor suas ações e realizar atividades de socorro, resgate ou requisitar assistência especializada, que requeira algum material ou conhecimento específico. Além disso, Silva, Oliveira e Borges (2012) argumentam que a mobilidade é um requisito fundamental nesse tipo de situação e defendem o uso de *smartphones* pelos membros das equipes de trabalho.

Esses dispositivos móveis possuem a capacidade de, em poucos segundos, identificar, processar requisições e trocar informações com outros dispositivos móveis ou servidores. Assim, o uso dessa solução poderia proporcionar meios de minimizar o trabalho da equipe de resposta de emergência na coleta de informações sobre as vítimas de um acidente, ao tentar maximizar a confiabilidade dessas informações (recomendando apenas aquelas pessoas que são mais relevantes para este processo de coleta de informação, ou seja, as pessoas socialmente mais próximas à vítima), acelerando assim o processo de resgate da vítima.

Para auxiliar no entendimento do cenário de uso de nosso processo proposto como solução ao problema visado por este trabalho, vejamos uma pequena narração do fluxo de acontecimentos que culminam com a utilização de nossa proposta. Tudo se inicia quando acontece uma emergência e a equipe de resposta a emergências é acionada. Ela então se dirige ao local onde a emergência ocorreu e começa a realizar o seu trabalho. Durante a realização deste trabalho de resposta à emergência, a equipe tem notícias de uma vítima que morava em uma das casas no local e que está desaparecida.

Porém, a equipe de emergência nada mais sabe a respeito da vítima além de seu nome, e, portanto, precisa agora de maiores informações a respeito dela para que a equipe possa realizar o processo de tomada de decisões que culminará com os resultados positivos esperados e a resolução da emergência. É exatamente neste ponto em que nossa proposta de solução se encaixa. A equipe de resposta à emergência poderá usar nossa ferramenta móvel em seu auxílio, utilizando-a para descobrir pessoas próximas ao local onde ocorreu a emergência que possam contribuir com o trabalho da equipe, fornecendo informações úteis e de qualidade sobre a vítima, ajudando dessa maneira a equipe de resposta a emergências no combate ao cenário-problema.

Vale lembrar que, atualmente, todo esse processo é feito de forma pessoal pelos integrantes da equipe de operação, possuindo apenas o auxílio de rádios e outras formas simplificadas de comunicação. Então, perante o cenário de emergência, a equipe operacional tem que percorrer toda a extensão do local onde ocorreu o acidente em busca de pessoas que possuam informações sobre possíveis vítimas. O fluxograma de acontecimentos no exemplo acima descrito que levam à utilização de nossa abordagem no cenário de emergência pode ser visualizado na Figura 8.



Figura 8. Fluxograma de atuação da abordagem no cenário de emergência

Todos os aspectos inerentes ao cenário de uso do presente trabalho, entretanto, precisam ser considerados com relação aos requisitos funcionais do sistema, que deve ser, primordialmente, capaz de:

- a) Enviar requisições de busca pelo nome da vítima a outros dispositivos móveis e ao servidor central;
- b) Compilar e exibir em uma listagem as respostas de outros dispositivos móveis e do servidor central a essas requisições;
- c) Calcular o valor do índice de relevância social dos contatos para o usuário, de acordo com o algoritmo multicritério da heurística;
- d) Aplicar os padrões de coleta e compilação das informações dos contatos do usuário;
- e) Permitir ao usuário controlar as configurações de privacidade e segurança;
- f) Permitir a disponibilização de informações sobre a localização do usuário;
- g) Permitir o contato da equipe de emergência com o usuário escolhido para prestar informações sobre a vítima.

Os requisitos *a* e *b* são triviais para qualquer sistema que vise compartilhar informações entre dispositivos móveis. Já o requisito *c* é resultado da premissa de que informações confiáveis e íntegras podem provir de pessoas socialmente próximas à vítima,

apresentada na hipótese desta dissertação. Assim, a obtenção dessas informações é facilitada pelo uso da tecnologia móvel atual.

Através da combinação dos requisitos *d* e *e* é oferecido ao usuário um controle maior sobre o aplicativo que está em seu dispositivo móvel, de modo que ele possa realizar configurações de atividade, privacidade e segurança no sistema. Por fim, os requisitos *f* e *g* permitem que o usuário escolhido para disponibilizar as informações sobre a vítima seja localizado e contatado. A seguir, descreveremos com maiores detalhes tanto o processo criado como base para a solução, quanto a ferramenta móvel construída sobre esse arcabouço.

5.2 Descrição

O processo se inicia quando o usuário instala o nosso aplicativo em seu dispositivo móvel (**cliente móvel**). Após a instalação do sistema, o aplicativo se registra no *site* (**servidor web**), fornecendo algumas informações do usuário, como nome, data de nascimento, número do celular, perfis nas mídias sociais e seus contatos no celular (Figura 9a). Então, o **servidor web** guarda essas informações (Figura 9b) e pesquisa informações complementares sobre a rede social do usuário na *internet* e nas mídias sociais, como o Facebook e o Twitter (Figura 9c), formando assim um *cache* de informações dos usuários registrados, com os dados passados pelo aplicativo e as informações da rede social desses usuários (Figura 9d).

Durante a resolução de uma emergência, a equipe responsável (**servidor comando e controle**) procura informações sobre as vítimas (**cliente móvel / servidor web**) (Figura 9e). Assim sendo, o **servidor comando e controle** dispara requisições para os **clientes móveis** dentro de um raio de ação pré-definido (Figura 9f). Dessa forma, os **clientes móveis** pesquisam a rede social do usuário atrás do *matching* da busca (nome da vítima) (Figura 9g). Em caso de *matching* positivo, o **cliente móvel** responde a requisição do **servidor comando e controle** indicando que seu usuário conhece a vítima e o grau de importância desse relacionamento. Em caso de *missmatch* total, ou seja, nenhum dos **clientes móveis** nos arredores da emergência acusa conhecimento da vítima, o **servidor web** é acionado em busca da análise contextual da rede social da vítima (Figura 9h).

O **servidor web** então verifica as informações sobre a vítima, primeiramente procurando os dados necessários no *cache* previamente construído (Figura 9i). Caso a ferramenta não consiga encontrar nenhuma entrada útil no *cache*, então ela faz uma pesquisa na *web* (Figura 9c) e, em caso de sucesso na busca, atualiza o *cache* (Figura 9d). Dessa forma o **servidor web** retorna uma resposta positiva ao **servidor comando e controle**, com a rede social da vítima identificada. Os usuários mais íntimos e próximos à vítima são então

recomendados para auxiliarem a equipe de emergência, seguindo os critérios de seleção pré-estabelecidos.

Pensando na quase remota possibilidade de o **servidor web** não encontrar nenhum registro sobre a vítima na *internet*, criamos então um fluxo de controle para esse caso de retorno de resposta negativa ao **servidor comando e controle**. Assim, o **servidor comando e controle** possui a opção de permitir que os **clientes móveis** próximos ao local da emergência realizem o *broadcast* da consulta para sua lista de contatos. Portanto, os **clientes móveis** de nível um assumem um papel temporário de servidor, disparando requisições com a consulta ao nome da vítima para os usuários do serviço contidos na micro rede social do dispositivo móvel em questão. Então os **clientes móveis** acionados (nível dois) pesquisam sobre a vítima em suas respectivas redes sociais. Dessa forma, os **clientes móveis** de nível dois retornam a resposta (positiva ou negativa) para o **cliente móvel** de nível um em questão, e este, por sua vez, repassa a resposta recebida para o **servidor comando e controle**.

A arquitetura da proposta de solução, bem como o funcionamento da ferramenta móvel construída, pode ser visualizada na Figura 9.

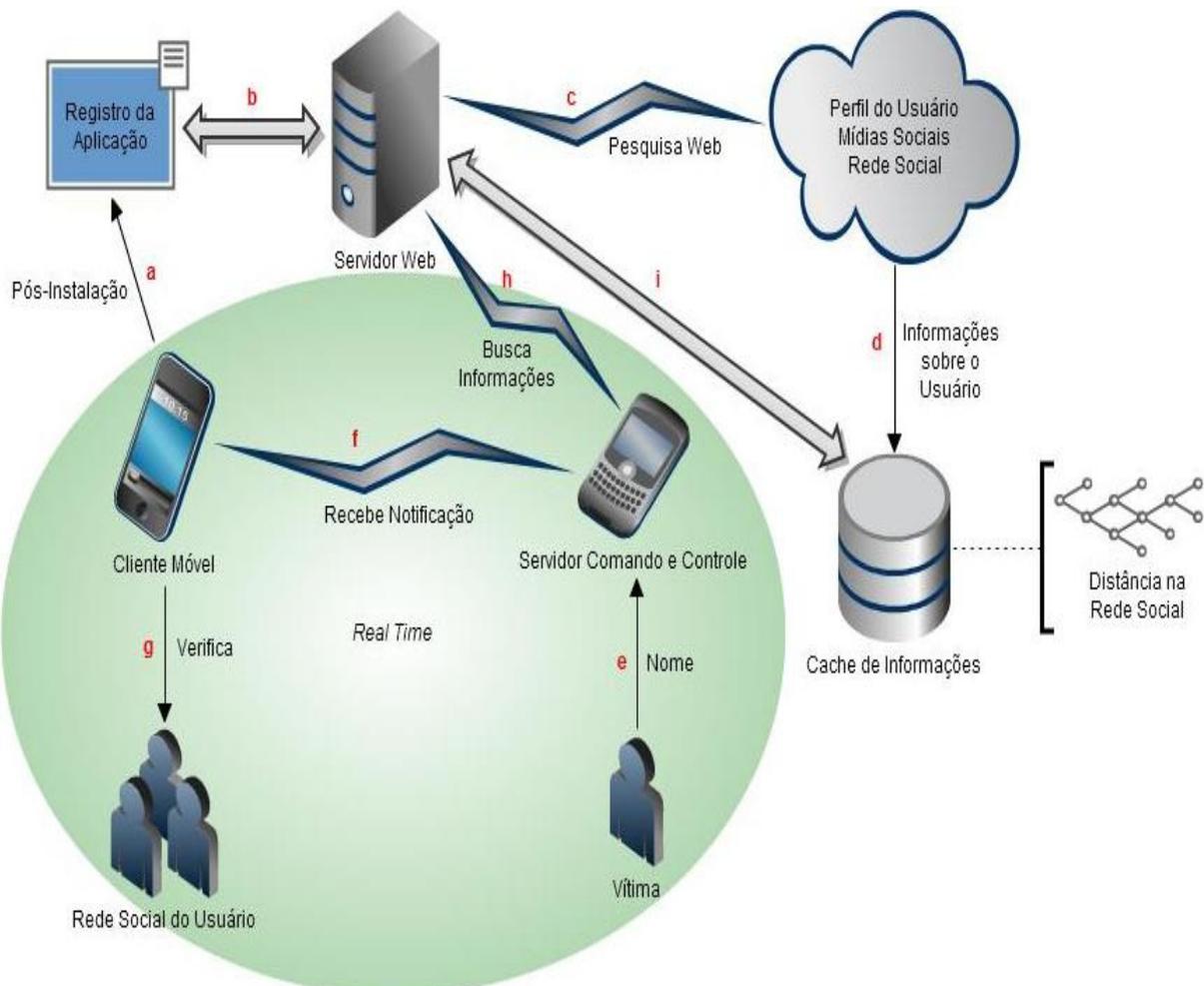


Figura 9. Arquitetura da solução proposta

O cerne desse processo de recomendação de pessoas para auxiliar a equipe de resposta a emergências com informações de qualidade sobre a vítima baseia-se em i) a distância social entre essas pessoas e a vítima, e ii) a distância física entre essas pessoas e o local onde a emergência ocorreu. Esse processo de recomendação de pessoas verifica a rede social da vítima da emergência em busca de pessoas próximas a ela o suficiente para fornecerem informações confiáveis e íntegras sobre a vítima. Além da análise contextual da rede social da vítima de emergência, nós buscamos também informações contidas nos dispositivos móveis das pessoas próximas ao local da emergência e as informações disponíveis através de mídias sociais na internet que podem nos conduzir a um modelo de recomendação melhor.

5.2.1 Distância Social

A principal atividade deste trabalho é a identificação da relevância social das pessoas contidas na rede social da vítima de uma emergência. Para isso, desenvolvemos e seguimos o processo descrito na Figura 10. Como fonte de dados são utilizados as interações nas mídias sociais [como o Facebook (2004) e o Twitter (2006), por exemplo] e os telefonemas, e-mails e mensagens de texto (SMS/MMS) contidos nos dispositivos móveis de cada usuário.

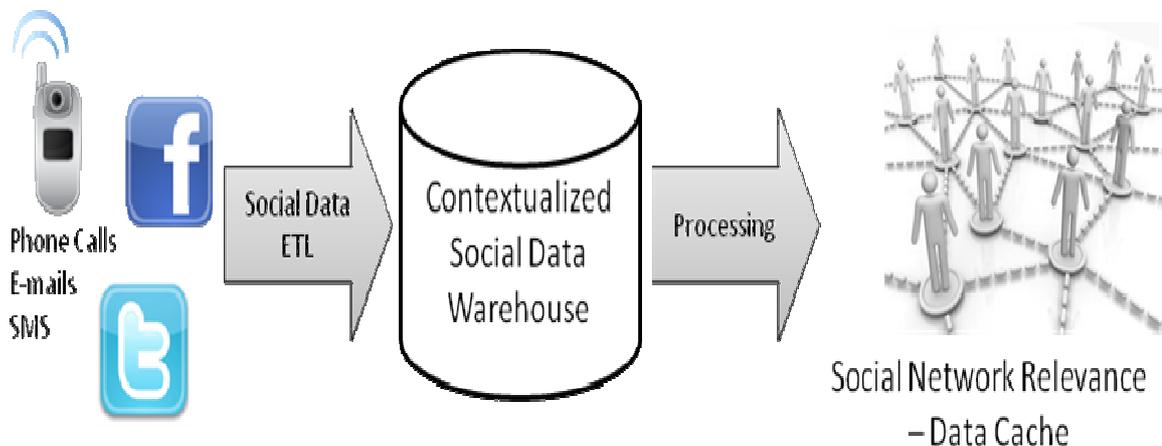


Figura 10. Tratamento de dados e processamento da relevância social

Inicialmente, o processo de busca pelas pessoas conhecidas da vítima se dará através do processamento de informações contextuais armazenadas dentro dos dispositivos móveis das pessoas que estão nos arredores do local onde a emergência ocorreu, como, por exemplo, telefonemas, e-mails, mensagens de texto, etc. Através da análise da rede social da vítima e das informações contextuais disponíveis nos dispositivos móveis das pessoas conseguiremos inferir a distância social entre aquela pessoa e a vítima, para sabermos se essa pessoa é próxima o suficiente da vítima para auxiliar a equipe de resposta a emergências com informações confiáveis e íntegras sobre a vítima.

Para inferirmos corretamente a distância social entre as pessoas ao redor da localidade da emergência e a vítima, precisaremos analisar certas características da rede social da vítima, tais como: o tipo de relação entre as pessoas (amigo, colega, conhecido, parente, parceiro, etc.); o peso dessa relação (importância, grau de afinidade, proximidade, etc.); a quantidade de conexões distintas entre elas, ou seja, a quantidade de tipos de relações entre duas pessoas (por exemplo, João é professor de Maria e trabalham juntos em um projeto — dois tipos de relação identificados, a saber, professor/aluno e chefe/subordinado).

Calculamos a relevância social usando alguns parâmetros importantes que refletem as interações e as estruturas sociais entre os indivíduos de uma determinada rede social. Esses parâmetros são apresentados no Quadro 2.

Variável	Significado
Distância da Relação (DR)	É o número de conexões de amizade necessárias para conectarmos duas pessoas distintas dentro de uma rede social, formando-se uma cadeia do tipo “amigo de amigo”. Pode ser obtido calculando-se o número de pessoas que conectam a vítima à pessoa recomendada e somando-se uma unidade. Ou seja, a distância é 1 para os amigos diretos, 2 para os amigos dos amigos, 3 para os amigos dos amigos dos amigos, e assim por diante.
Tipo de Relação (TR)	O tipo de uma relação indica qual é o tipo de vínculo ou ligação entre duas pessoas. No presente trabalho, os tipos de relação são divididos em: RLD (Relação de Longa Duração, como os pais ou cônjuges), Amigo, Família (ser membro da), Colega de Trabalho e Conhecido.
Peso da Relação (PR)	Durante nossos estudos formulamos a crença de que alguns tipos são mais fortes do que outros. Dessa forma, definimos arbitrariamente alguns pesos para refletirem a importância de cada um desses tipos: <ul style="list-style-type: none"> • RLD – 5 • Amigo – 4 • Família (ser membro da) – 3 • Colega de Trabalho – 2 • Conhecido – 1
Número de Conexões (NC)	A frequência de comunicação entre as pessoas. Podemos obtê-la através do cálculo dos totais de números de SMS/MMS trocados, e-mails enviados e recebidos, chamadas telefônicas realizadas e recebidas (em dispositivos móveis), e das interações nas mídias sociais (como comentários, citações, <i>likes</i> , compartilhamento de conteúdo, troca de mensagens, <i>tweets</i> , dentre outros).

Quadro 2. Variáveis relacionadas à distância social

O parâmetro Distância da Relação (DR) é baseado no experimento do mundo pequeno (MILGRAM, 1967), que analisou o comprimento médio do caminho nas redes sociais das pessoas nos Estados Unidos. Essa pesquisa foi baseada na sugestão de que a sociedade humana é uma rede típica de mundo pequeno, caracterizado por conexões de comprimento

curto. Esse trabalho mostrou que o mundo está cada vez mais interligado, afirmando que apenas cinco intermediários (em média) seriam suficientes para conectar quaisquer dois indivíduos escolhidos aleatoriamente, independentemente de onde eles viviam.

O parâmetro Tipo de Relação (TR) representa os laços interpessoais, que são definidos pela sociologia matemática¹³ como conexões que transmitem informações entre as pessoas. Kapferer (1969) postulou a existência de laços multiplex, caracterizada por múltiplos contextos em um relacionamento. Multiplexidade é a sobreposição de papéis ou filiações em uma relação social (VERBRUGGE, 1979). Portanto, o parâmetro Peso da Relação (PR) nestes casos de multiplexidade é ajustado para utilizar o maior peso dos laços multiplex.

O parâmetro Número de Conexões (NC) baseia-se na “força” de um laço interpessoal, ou seja, em uma combinação linear da quantidade de tempo, da intensidade emocional, da intimidade (ou confiança mútua), e dos serviços recíprocos que caracterizam cada laço (GRANOVETTER, 1973). Nós o calculamos atribuindo pesos diferentes para cada tipo de interação (conexão) entre duas pessoas: *likes* e citações (1), comentários e compartilhamento de conteúdo (2), mensagens (3), SMS/MMS (4), e-mails (5) e ligações (6).

Nós fazemos uma média aritmética ponderada com esses dados, mas também levamos em consideração o total de duração das chamadas (efetuadas ou recebidas) nessa análise, uma vez que o tempo de conversação pode ser um forte indicador de intimidade entre as pessoas.

Nosso índice de Relevância Social (RS) é calculado conforme mostra a equação (1).

$$RS = PR * NC \quad (1)$$

Se $DR = 1$, então nós usamos PR conforme está descrito na Tabela 1. Caso contrário, $PR = 1/DR$.

O parâmetro NC é calculado de acordo com a média aritmética ponderada de cada tipo de interação social entre a vítima e a pessoa a ser recomendada, conforme mostrado na equação (2) — onde c_i é a quantidade de interações do tipo de conexão i e p_i é o peso do tipo de conexão i . Inicialmente, para efeitos de demonstração e análise deste estudo, atribuímos valores previamente fixados para cada um desses pesos. Porém, nada impede que no futuro esses pesos sejam atribuídos dinamicamente, de acordo com o comportamento do usuário.

$$NC = \frac{\sum_{i=1}^n p_i * c_i}{\sum_{i=1}^n p_i} \quad (2)$$

¹³ Sociologia matemática é o campo da sociologia que utiliza modelos altamente formalizados para entender os processos sociais e estruturas sociais, utilizando assim a matemática para construir teorias sociais.

A Distância Social (DS) é inversamente proporcional à Relevância Social (RS). Isto é, quanto maior a relevância social (de uma pessoa em relação à outra), mais próximas elas são (menor distância social). O cálculo da distância social tenta identificar a afinidade da vítima com outra pessoa de sua rede social. No processo de recomendação de nossa proposta, o próximo passo é ordenar e identificar qual a pessoa com maior relevância social (ou menor distância social) está mais próxima, fisicamente, do local da emergência.

5.2.2 Distância Física

Nem sempre estamos rodeados daqueles que conhecemos. Podemos viajar sozinhos, irmos sozinhos a um restaurante novo ou assistir ao jogo do seu time preferido em um estádio, novamente “desacompanhado”. Então, como nossa solução proposta deverá se comportar nesses casos onde somos vítimas de uma emergência em um local onde quase nenhuma pessoa, ou até mesmo ninguém, nos conhece a ponto de poder auxiliar a equipe de emergência com informações úteis a nosso respeito?

Para esses casos, nossa solução proposta deverá analisar os aspectos contextuais que nos cercam e novamente estão disponíveis através dos nossos dispositivos móveis, seja através de informações contextuais presentes nos aparelhos ou através de serviços facilmente acessáveis através dos mesmos.

Como contextos mais dinâmicos a serem analisados pela ferramenta podemos citar a localização geográfica (coordenadas GPS), a condição climática (clima, temperatura, umidade relativa do ar, etc.) de onde nos encontramos ou do local da emergência, a localização temporal (qual a hora do dia no momento de interesse), a distância a que a pessoa se encontra ao local da emergência, o trânsito que a pessoa poderá enfrentar dependendo da via de acesso que ela for pegar até o local da emergência, dentre outros.

Portanto, a distância física baseia-se no caminho que a pessoa recomendada deverá percorrer até chegar à localidade da emergência, incluindo a maneira com que cada pessoa recomendada deverá se locomover até a chegada ao local da emergência. Dessa forma, se faz necessária a análise das informações contextuais anteriormente citadas para que o processo de recomendação possa sugerir as pessoas que, além de estarem mais perto do local da emergência, possam chegar lá mais rapidamente.

As variáveis de contexto analisadas que podem nos fornecer as informações desejadas sobre as pessoas que serão recomendadas podem ser vistas no Quadro 3. Tais informações referem-se à distância física entre essas pessoas e o local onde ocorreu a emergência, bem como à forma que elas poderão se deslocar até a emergência ou quais caminhos percorrerão.

Variável	Significado
Localização Geográfica	Coordenadas GPS que indicam onde a pessoa está no momento. É importante para o cálculo da distância física entre a pessoa e o local da emergência, além de outras coisas como para indicar se a pessoa está perto de uma forma alternativa de transporte que poderia levá-la mais rapidamente para o local de emergência (por exemplo, uma estação de metrô ou trem).
Condições Climáticas	Está chovendo? Está nevando? Qualquer informação útil sobre a condição climática que possa influenciar significativamente na locomoção da pessoa recomendada.
Tempo	Tempo estimado de chegada da pessoa recomendada ao local da emergência ou qualquer outra informação relacionada ao tempo do usuário, como se ele estava em um determinado local há uma hora.
Distância	A distância física entre a pessoa recomendada e o local da emergência.
Condições de Tráfego	Qualquer informação útil sobre as condições da estrada, como se qualquer rua que o usuário possa utilizar em seu caminho até a localidade da emergência está fechada, ou então qualquer outra condição de tráfego no percurso até o local da emergência.

Quadro 3. Variáveis relacionadas à distância física

5.2.3 Informação Contextual

Todos esses contextos dinâmicos citados nas seções anteriores devem ser analisados para que a ferramenta construída obedeça aos princípios de eficiência e efetividade na recomendação das pessoas para prestarem informações confiáveis e íntegras sobre uma vítima da emergência. Primeiramente devemos analisar a informação contextual contida na rede social do usuário para que possamos inferir a distância social entre ele e a vítima, como pode ser visto na seção 5.2.1. Após isso, nós teremos que levar em consideração a informação contextual física, conforme descrito na seção 5.2.2. Estas variáveis de distância física serão utilizadas de uma maneira que possamos recomendar a pessoa mais importante para a vítima, de modo que ela possa chegar ao local da emergência o mais rapidamente possível, porque neste tipo de situação o tempo é o pior inimigo.

Outro tipo de informação contextual a ser analisado são as informações cujo contexto se enquadra em um perfil mais estático, ou seja, informações que não mudam (ou raramente mudam) com o passar do tempo. As informações que se encaixam nesse perfil são: nome (identificador da vítima ou pessoa recomendada); tipo sanguíneo e fator RH (para possíveis emergências onde haja a necessidade de transfusão de sangue); altura, peso e idade (informações importantes para o processo de tomada de decisão em resgates onde a pessoa se encontra soterrada, por exemplo); profissão (caso a equipe de emergências necessite da ajuda de um determinado especialista, por exemplo), dentre outras.

Todas essas informações são adquiridas em diferentes veículos para análise em nossa ferramenta móvel proposta: através de pesquisas genéricas na *internet* por informações sobre a vítima; buscas nas mídias sociais para extrair informações diretamente do perfil do usuário, de sua lista de amigos (rede social) ou analisando-se a troca de mensagens entre os usuários dessas mídias; verificando nos dispositivos móveis as ligações recentes (realizadas ou recebidas) e suas durações, a intensidade da troca de mensagens (SMS, MMS e e-mail) entre as pessoas, a localização geoespacial da pessoa (através das coordenadas GPS), os metadados embutidos em arquivos (fotos, vídeos e áudio), a lista de contatos do usuário (agenda de telefone e e-mail), etc.

5.3 Privacidade e Segurança

A questão da privacidade e segurança, principalmente quando se trata de serviços móveis ou baseados em localização, é na maioria dos casos considerada essencial e possui um conceito geral de que “quanto mais, melhor”, ou seja, quanto mais privacidade e segurança o projeto puder oferecer, melhor é para o usuário. Em muitos casos, um elevado nível de privacidade e segurança é visto pelo provedor do serviço como sendo um fator assegurador para o usuário e que é aplicado para impedir que os usuários sintam-se preocupados com a divulgação de seus dados pessoais.

Barkhuus (2004) apresenta um experimento onde dois grupos de usuários são confrontados com serviços móveis baseados em localização capazes de identificar a localização geográfica de outros usuários contidos em sua lista de contatos. Muitas pessoas de ambos os grupos mostraram-se realmente preocupadas com relação à privacidade, apesar de algumas mostrarem-se satisfeitas com o uso de tais serviços, enquanto outras desconfiavam de sua real utilidade.

Dos resultados desses dois estudos de caso, Barkhuus (2004) extrai então três questões de projeto a serem abordadas por serviços baseados em localização do usuário. Uma das implicações óbvias para qualquer projeto desse tipo é a necessidade essencial de informações em tempo real sobre o nível de privacidade do usuário em uma situação específica. Para os usuários aceitarem esses serviços, eles precisam saber quem pode e, principalmente, quem não pode ver a sua localização.

Além disso, os serviços baseados em localização devem permitir a desativação de curto prazo. O que também preocupou os participantes desse experimento foram situações específicas de curto espaço de tempo. Eles refletiram sobre instantes em que não gostariam de ser rastreados, mesmo pelos familiares ou amigos próximos. Alguns também solicitaram um

serviço que estivesse disponível apenas em determinadas situações, como quando tivessem que ir para casa sozinhos à noite. Uma forma de ter certeza que os usuários aceitarão esses serviços é se certificar de que seja possível desligá-lo e informar o usuário quando ele está ligado ou desligado.

Finalmente, os estudos de caso indicaram que serviços baseados em localização têm maior potencial para adoção em ambientes fechados ou, no mínimo, limitados. Os serviços no segundo estudo de caso foram percebidos menos ameaçadores para os participantes, muito provavelmente devido ao fato de eles saberem que há um limite. Outro fator que também pode ajudar na confiança do usuário é ser um serviço não comercial e, portanto, não ser provável que os dados acabem em ambientes comerciais. Portanto, serviços dependentes de localização poderiam ser restritos a definições fechadas, sejam físicas (como determinados locais, bairros ou cidades) ou grupos de usuários (por exemplo, amigos ou colegas de trabalho). Dessa forma, os usuários teriam um maior senso de controle e conhecimento de quem potencialmente irá saber a sua localização.

Outro elemento importante na gestão de privacidade em computação ubíqua é o nível de precisão das informações. É importante para o usuário que a precisão das informações possa ser ajustável, permitindo uma divulgação diferenciada, aproximando-se da maneira como o usuário lida com essas informações na vida social tradicional. Um estudo (LEDERER; MANKOFF; DEY, 2003) revela que a identidade do solicitante da informação é um determinante de preferências de privacidade mais forte do que a situação em que o usuário se encontra no instante da solicitação.

Em outras palavras, as preferências de privacidade variam mais dependendo do inquiridor do que da situação. Segundo os resultados desse estudo, os indivíduos eram mais propensos a aplicar as mesmas preferências de privacidade ao mesmo inquiridor em situações diferentes do que aplicar as mesmas preferências de privacidade a inquiridores diferentes em uma mesma situação. Portanto, a relação essencial é que a identidade do solicitante e a situação do usuário no momento do inquérito determinam quais as preferências do usuário na hora de moderar a precisão das informações divulgadas.

Langheinrich (2001) desenvolveu seis princípios para orientar projetos de sistemas ubíquos, com base em um conjunto de práticas de informação justas que são comuns à maioria das legislações de privacidade em uso atualmente: aviso prévio, escolha e consentimento, proximidade e localidade, anonimato e pseudonimato, segurança, e acesso e recurso. Seu trabalho tenta avaliar a privacidade sob três ângulos complementares (sua

história, seu estatuto jurídico e sua utilidade) e apresenta algumas implicações sociais da computação ubíqua que podem conduzir a questões de privacidade e segurança:

- Ubiquidade: A computação ubíqua está em toda parte;
- Invisibilidade: Não só os computadores deveriam estar em todo lugar, mas também deveriam realmente desaparecer de nossa vista;
- Sensoriamento: O avanço na tecnologia permite a criação de micro sensores com alta capacidade de processamento, de modo que eles possam avaliar com precisão certos aspectos do ambiente, até mesmo aspectos emocionais de nossas vidas.
- Amplificação de Memória: Avanços no campo de processamento de áudio e vídeo, combinado com equipamentos sensoriais de última geração, podem continuamente e discretamente gravar cada ação, permitindo navegar e pesquisar o nosso passado.

Tendo em mente essas implicações sociais da computação ubíqua e os princípios discutidos por Langheinrich (2001) em seu trabalho, tivemos o cuidado de seguir essas orientações para abordar as questões inerentes à privacidade e segurança neste trabalho. Também seguimos outro guia de privacidade para construção de serviços móveis que precisem disponibilizar informações dos usuários, inclusive sua localização. Esse guia (IACHELLO et al., 2005) foi formulado com base nos resultados de três experimentos de campo: um estudo formativo com método de amostragem de experiência (*Experience Sampling Method* — ESM), uma implementação piloto e um estudo de usuários estendido.

Todos esses experimentos foram orientados para as preocupações pessoais significativas com a privacidade, segurança e proteção de dados causados por este tipo de serviço. O guia é então um pequeno conjunto de diretrizes para projetistas de aplicações sociais móveis poderem melhor atender às necessidades muitas vezes conflitantes de segurança dos usuários, preservação de sua privacidade e criação de aplicativos atraentes e úteis, além de mostrar como essas diretrizes podem ser aplicadas a serviços dessa natureza.

Nas seções a seguir mostraremos como foram abordadas as questões de privacidade e segurança na transformação de nosso arcabouço de solução para uma arquitetura segura. Também mostraremos as abordagens de privacidade e segurança para cada uma das funcionalidades de nossa proposta de solução, como, por exemplo, a construção de um mecanismo de “lista negra” que permita ao usuário identificar quais contatos nunca deverão

ser indicados como resposta a uma requisição da equipe de resposta a emergências, ou ainda as configurações de horários de coleta das informações e seu modo de disponibilização.

5.4 Arquitetura

A ferramenta construída possui uma arquitetura cliente-servidor em três nós. O primeiro nó é o **servidor web** (Figura 11a), um repositório para troca de informações e pesquisa de dados dos usuários da ferramenta. O segundo nó é o **servidor comando e controle** (Figura 11b), um servidor móvel que é representado pelo sistema que está instalado nos dispositivos móveis que são utilizados pela equipe de resposta a emergências. Este nó é responsável por disparar consultas à procura de pessoas que possam auxiliar na emergência como, por exemplo, cidadãos para disponibilizar informações sobre uma vítima. O terceiro e último nó é o **cliente móvel** (Figura 11c), representado pelos sistemas instalados nos dispositivos móveis utilizados pela população. Este nó fornecerá informações importantes para o *matching* das consultas, tais como se o usuário faz parte da rede social da vítima e poderá auxiliar a equipe de resposta a emergências.

Esses três nós irão interagir entre si de maneira a realizar a análise contextual da rede social da vítima (Figuras 11d, 11e, 11f e 11g). Os componentes e módulos dessa arquitetura podem ser visualizados na Figura 11. A arquitetura e o funcionamento de cada um desses módulos da ferramenta móvel construída como prova de conceito de nossa proposta de solução para o problema (Servidor Web, Servidor Comando e Controle, e Cliente Móvel) são descritos a seguir, nas seções 5.4.1, 5.4.2 e 5.4.3, respectivamente.

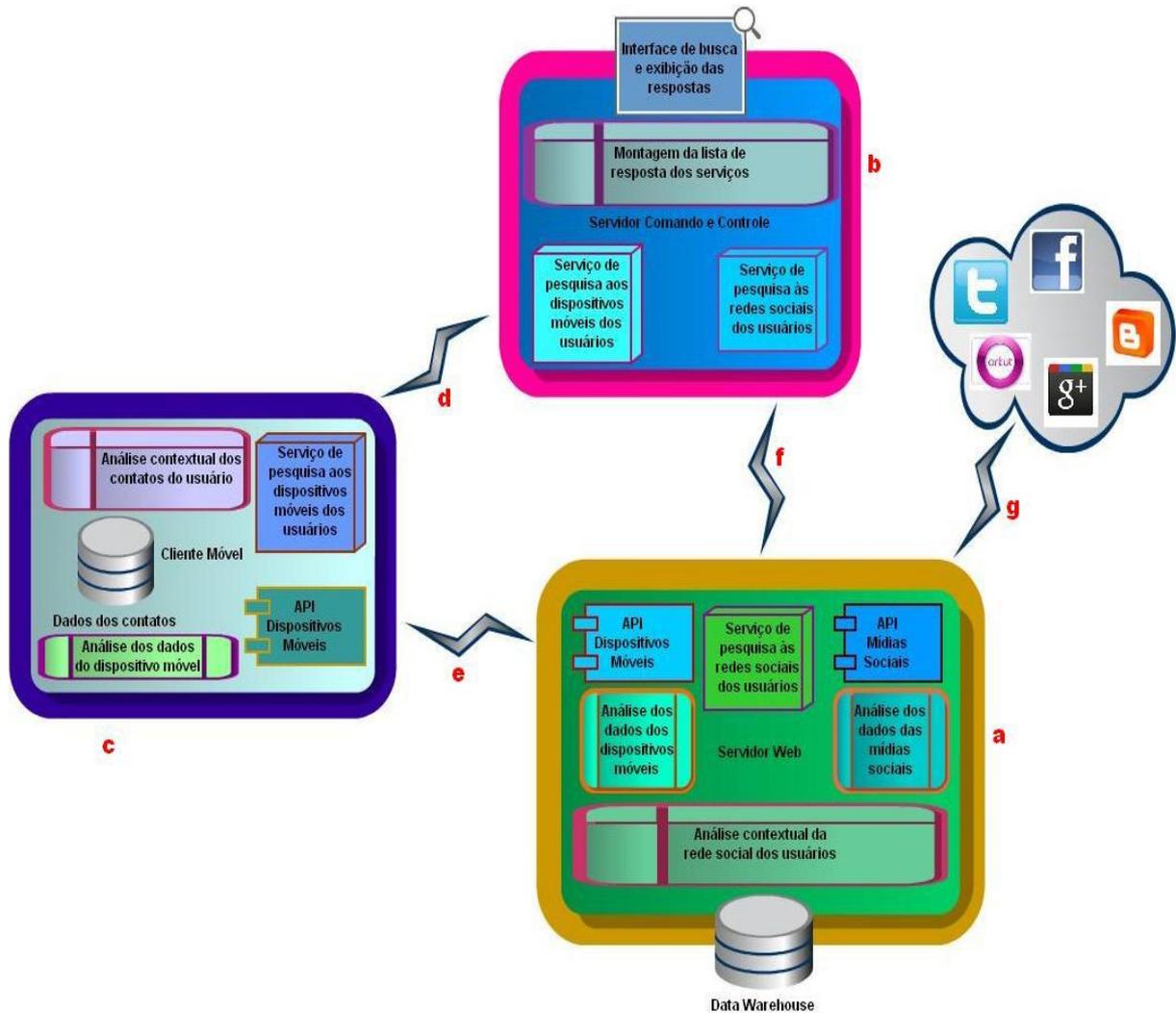


Figura 11. Componentes e módulos da ferramenta de solução proposta

5.4.1 Servidor Web

Não cabe a este trabalho entrar nas minúcias envolvendo uma possível especificação técnica do servidor web que executará o módulo de análise contextual da rede social dos usuários registrados. No entanto, basta-nos deixar claro que o servidor deverá possuir uma grande capacidade de processamento (CPU e memória RAM) e armazenamento de dados (HDD).

A necessidade de uma grande capacidade de processamento justifica-se pelo fato da complexidade do processo de análise contextual da rede social dos usuários, que envolve a busca pelas informações acerca dos usuários e o tratamento desses dados, aliado às informações pertinentes da rede social desses usuários. À medida que os usuários aumentam, também aumenta o trabalho de processamento durante a análise contextual das redes sociais de cada usuário.

Já a necessidade de uma grande capacidade de armazenamento de dados é justificada pelo fato de guardarmos as informações de cada usuário e de cada pessoa contida em sua rede social, para futuro processamento durante a etapa de análise contextual da rede social. A intenção é que seja montado um *Data Warehouse* das informações contextuais dos usuários e suas redes sociais. Novamente, a tendência é que o consumo de espaço no disco rígido aumente à medida que o número de usuários da aplicação vá aumentando, ou até mesmo no caso das redes sociais dos usuários aumentarem de tamanho.

O módulo abrigado pelo servidor web é construído em Java (versão 7), por ser uma linguagem de programação orientada a objetos de fácil utilização e aprendizado, além de possuir ótimos recursos e bibliotecas já prontos para uso e de licença aberta (*open source* ou simplesmente gratuita), sem maiores restrições. Outras vantagens de se construir o módulo web em Java incluem, mas não se limitam a: portabilidade (independência de plataforma – “*write once, run anywhere*”), segurança, desempenho, estabilidade e suporte a integração com outras linguagens Não-Java.

O servidor web ficará responsável por receber os dados dos usuários provenientes dos clientes móveis e buscar informações contextuais pertinentes à rede social desses usuários nas mídias sociais. Essas informações coletadas dos dispositivos móveis dos usuários (troca de mensagens SMS/MMS, e-mails, quantidade de ligações e duração total das chamadas, etc.) serão agregadas às informações coletadas das mídias sociais (compartilhamento de conteúdo, comentários, citações em *posts*, *tags* em fotos e vídeos, *likes*, *tweets*, mensagens, etc.) para a formação de um *Data Warehouse* de onde serão extraídas as informações de distância e relevância social entre os usuários e as pessoas contidas em suas redes sociais. Este *cache* de informações contextuais das redes sociais dos usuários é periodicamente atualizado para futuras consultas durante uma situação de emergência.

Para isso, o servidor web deverá possuir duas APIs básicas: uma API para a troca de informações contextuais provenientes do dispositivo móvel dos usuários e outra API para a coleta das informações contextuais da rede social do usuário provenientes das mídias sociais. A primeira API, que permite a troca de informações entre os dispositivos móveis do usuário e o servidor web, deverá ser criada segundo os padrões e arquiteturas definidos pela presente proposta de trabalho. Já a segunda API, que coleta informações das mídias sociais, na verdade poderá ser um conjunto das APIs proprietárias de cada mídia social disponível, como as do Facebook (FACEBOOK DEVELOPERS, 2012a) e Twitter (TWIDEV, 2012). Para a pesquisa nas mídias sociais que não disponibilizam APIs para consulta aos seus dados, existe a

possibilidade de construir um *crawler* que ficará responsável por indexar as informações dessas mídias sociais.

Além dessas APIs, deverá existir ainda a disponibilização de um serviço que permita ao servidor comando e controle realizar a análise contextual da rede social de uma vítima de emergência, retornando como resposta as pessoas mais próximas à vítima, para que elas possam ser contatadas imediatamente e auxiliarem a equipe de resposta a emergências com informações confiáveis e íntegras, que possam ser úteis no processo de resgate da vítima. Esses processos e atividades do servidor web podem ser visualizados na Figura 12.

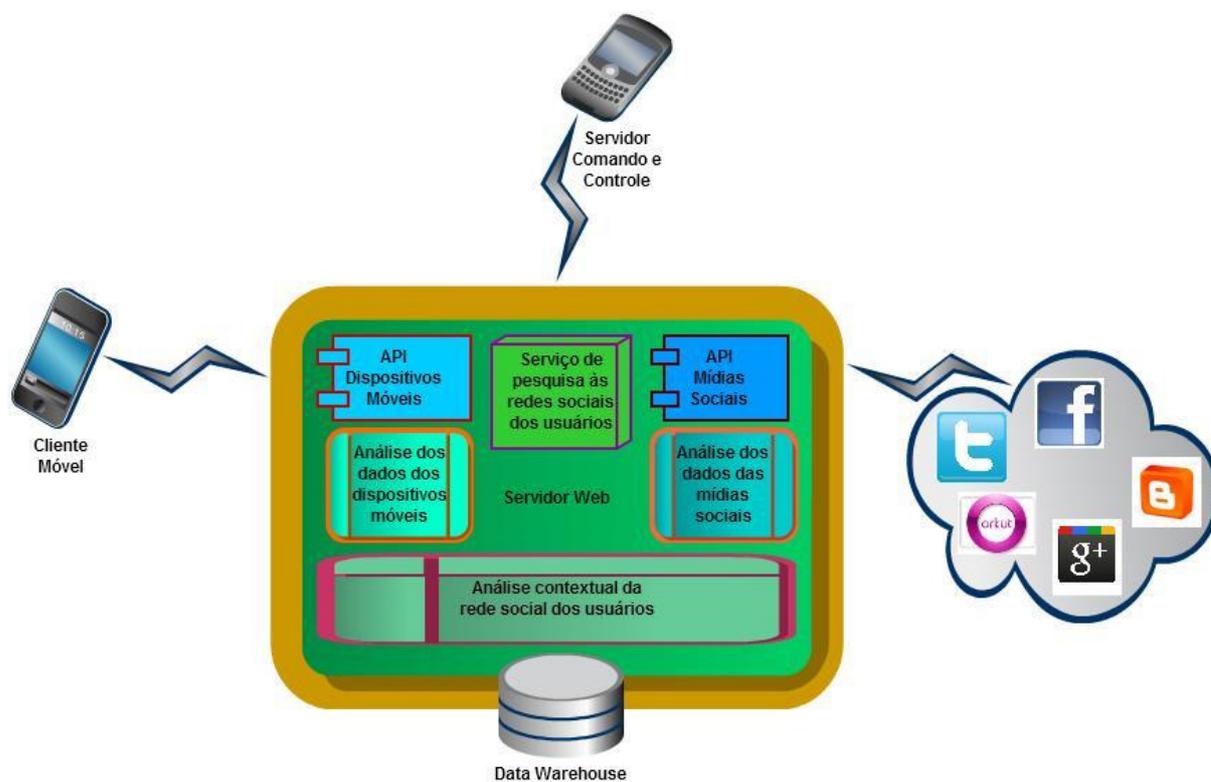


Figura 12. Processos e atividades do Servidor Web

Para a realização desses processos e atividades faz-se necessária a construção de alguns módulos e componentes, conforme descrito a seguir:

- **Serviço de pesquisa às redes sociais:** é o serviço que ficará responsável por i) invocar a API das mídias sociais para atualizar os dados da rede social dos usuários cadastrados e ii) intermediar a comunicação entre o Servidor Web e o Servidor Comando e Controle, recebendo desse o nome da vítima a ser pesquisado e retornando o resultado da análise contextual da rede social da vítima como resposta.
- **API Mídias Sociais:** é o conjunto de APIs (um para cada mídia social a ser analisada) responsável por coletar os dados dos usuários nas mídias sociais,

como informações pessoais e interações com os participantes de sua rede social.

- **Gerente de dados das mídias sociais:** é o componente responsável por manipular os dados provenientes das mídias sociais, efetivamente mantendo-os atualizados.
- **API Dispositivos Móveis:** é o módulo responsável por i) coletar os dados do usuário provenientes de seu dispositivo móvel, como as interações com os contatos do celular, seja através de e-mails, mensagens de texto, ligações telefônicas ou outras formas e ii) intermediar a comunicação entre o Servidor Web e o Cliente Móvel, recebendo desse o nome da vítima a ser pesquisado e retornando o resultado da análise contextual da rede social do usuário como resposta.
- **Gerente de dados dos dispositivos móveis:** é o componente responsável por manipular os dados provenientes dos dispositivos móveis dos usuários, efetivamente mantendo-os atualizados.
- **Analisador de Contextos:** é o módulo responsável por realizar a análise contextual da rede social dos usuários, retornando como resposta o índice de relevância social dos contatos com o nome pesquisado para o usuário.
- **Base da rede social dos usuários:** é um *cache* de informações (Data Warehouse) sobre o usuário e sua rede social, contendo dados básicos sobre o usuário e os dados de interação social com os participantes de sua rede social (provenientes tanto das mídias sociais quanto dos dispositivos móveis do usuário). É sobre essas informações que a análise contextual da rede social dos usuários será realizada.

A arquitetura do Servidor Web, contendo os módulos e componentes acima descritos, pode ser vista na Figura 13.

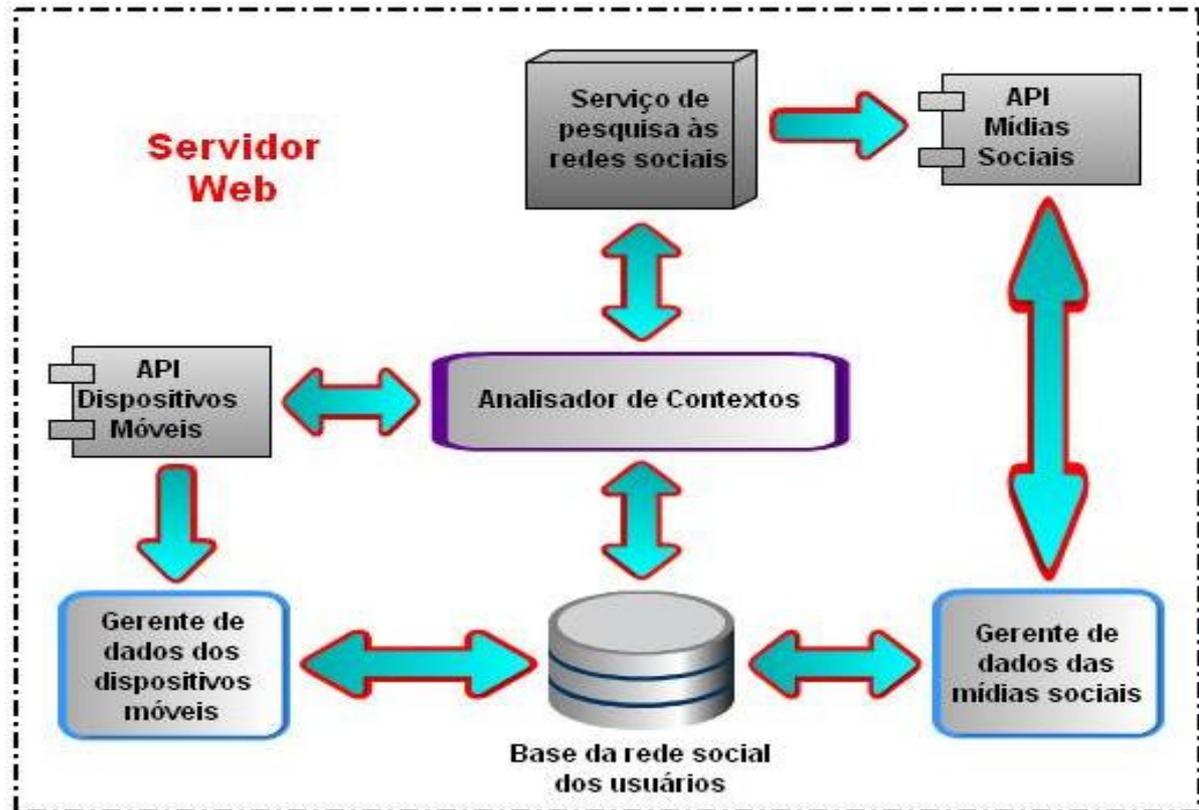


Figura 13. Arquitetura do Servidor Web

5.4.2 Servidor Comando e Controle

O servidor comando e controle é um módulo da nossa proposta de solução que fica instalado nos dispositivos móveis utilizados pela equipe de resposta a emergências. Este módulo deverá ser construído para ser usado na plataforma Android (2008). O sistema Android se destaca por possuir um apoio tão forte oferecido pelo Google (1998) e pela sua natureza de programação aberta, acessível a todos os interessados. Baseado em Linux (1991), ele traz consigo suporte para todo tipo de conexão sem fio — 3G, EDGE, Wi-Fi e Bluetooth —, para multimídia e promete ser extremamente versátil, facilmente adaptado a PDAs ou aos tradicionais telefones em barra, com suas telas menores.

De acordo com dados levantados pela empresa Gartner (2012), Symbian (1997) e Android (2008) estão dominando o setor de sistemas operacionais móveis até o momento. Segundo essas estimativas, Symbian e Android são juntos responsáveis por 62,8% do mercado mundial de sistema operacional para dispositivos móveis, sendo que dessa parcela a Symbian possui atualmente 32,3% e o Android vem logo em seguida com 30,5%. O sistema Android tem crescido rapidamente no mercado, estando cada vez mais próximo do líder de vendas, e a expectativa é que até 2014 o Android se torne líder no mercado de dispositivos móveis.

Sua principal responsabilidade é disparar requisições aos clientes móveis (dispositivos móveis utilizados pela população) próximos ao local onde ocorreu a emergência, para verificar se esses usuários conhecem a vítima da emergência. Ao receber as respostas de todos os clientes móveis próximos ao local da emergência, o servidor comando e controle exibirá a listagem de todos os usuários, contendo cada lista de contatos que possuem o mesmo nome da vítima, para avaliação da equipe de resposta a emergências.

O protótipo da tela que será exibida pelo servidor comando e controle para a equipe de resposta a emergências, contendo a junção das respostas dos dispositivos móveis próximos ao local da emergência, pode ser visto na Figura 14. Note que a tela de resposta exibe a listagem dos contatos que combinam com o nome da busca realizada pelo servidor comando e controle, contidos em cada dispositivo móvel dos usuários próximos ao local da emergência.

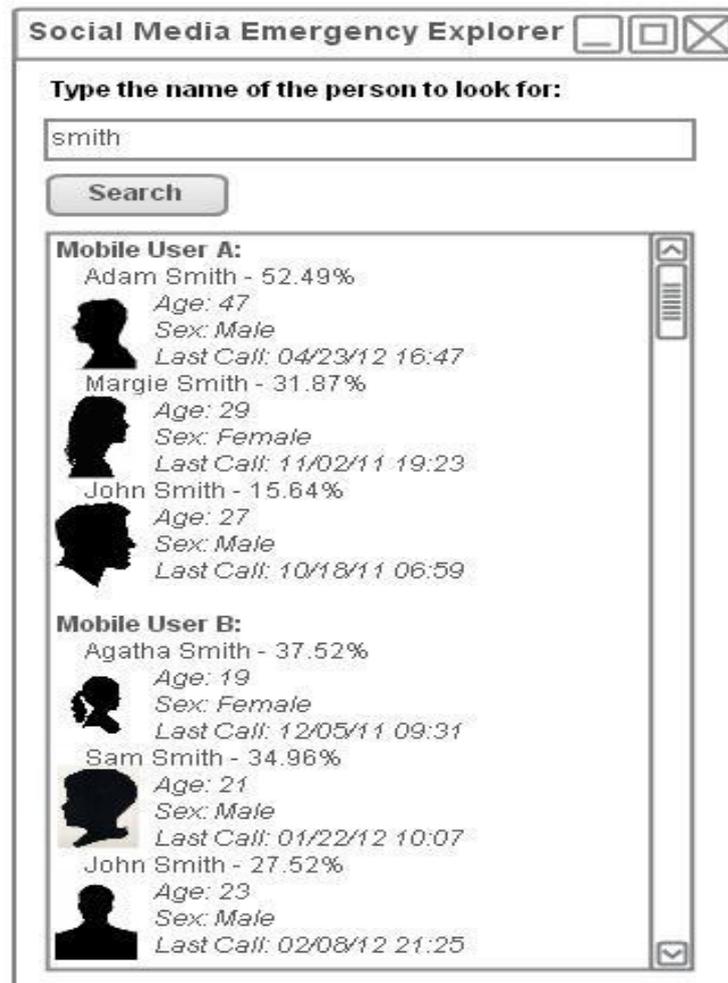


Figura 14. Protótipo da tela de resposta do servidor comando e controle

A lista de resposta de cada dispositivo móvel dos usuários (clientes móveis) é ordenada de acordo com a ordem de chegada das respostas no servidor comando e controle. Já a lista de contatos de cada usuário é ordenada segundo a proximidade social dos contatos com o usuário, indicada como a porcentagem (%) de relevância social de cada contato em relação

ao total de contatos nessa listagem. Ou seja, a proximidade social de cada contato com o usuário é dada relativamente aos contatos retornados pela busca, e não em relação a todos os contatos do usuário presentes no dispositivo móvel.

Além do nome dos contatos que conferem com a busca realizada de cada usuário que está próximo ao local da emergência, o servidor comando e controle exibe também alguns dados desses contatos, como a idade, o sexo e a data de última ligação entre o usuário e aquele contato. Esses dados servem para evitar o problema dos homônimos, auxiliando dessa forma a equipe de resposta a emergências a identificar quais desses usuários conhecem realmente a vítima da emergência, e não apenas um homônimo. O detalhe desses dados auxiliares e do cálculo da relevância social dos contatos não é o foco desta seção, e são tratados apropriadamente no Capítulo 6.

Portanto, basicamente, o servidor comando e controle é responsável somente pelas requisições de pesquisas ao nome da vítima de emergência que está desaparecida ou inconsciente e pela exibição dos resultados dessa pesquisa. A partir da exibição do resultado agregado dessa busca pelo nome da vítima, a equipe de resposta a emergências poderá analisar a listagem de usuários que possivelmente conhecem a vítima e entrar em contato com eles, para coletar informações úteis sobre a vítima que possam auxiliar no processo de resgate. As atividades deste módulo de nossa proposta de solução são:

1. Disparar requisições de busca pelo nome da vítima do acidente para os clientes móveis pesquisarem em sua lista de contatos.
2. Realizar a junção de todas as respostas recebidas dos clientes móveis (contendo a listagem de contatos que possuem o mesmo nome da vítima, ordenados segundo suas relevâncias sociais para aquele usuário) e exibi-las numa listagem única de usuários e suas listas de contatos que conferem com o nome anteriormente pesquisado.
3. Consultar o servidor web para realizar a análise contextual da rede social da vítima, em busca de pessoas socialmente próximas a ela para fornecerem informações íntegras e confiáveis para auxiliar a equipe de resposta a emergências no resgate dessas vítimas.
4. Localizar o(s) usuário(s) escolhido(s) pela equipe de resposta a emergências para prestar informações sobre a vítima.

A atividade 3 (consulta ao servidor web) só ocorre caso não haja nenhuma resposta dos dispositivos móveis próximos ao local da emergência. A atividade 4, que localiza o usuário escolhido pela equipe de resposta a emergências para prestar informações úteis sobre

a vítima, pode ser implementada através da utilização dos dados posicionais do usuário (coordenadas GPS), que é submetido pelo cliente móvel juntamente com a resposta à requisição de pesquisa pelo nome da vítima (uma vez habilitada essa opção no cliente móvel, para preservar a privacidade do usuário).

De posse das coordenadas GPS do(s) usuário(s) escolhido(s), o servidor comando e controle poderá facilmente acessar serviços já existentes como o Google Latitude (GOOGLE, 2009), Google Maps (GOOGLE, 2005a), Google Weather (GOOGLE, 2008), Climatempo (1997), dentre outros, para repassar essas informações contextuais (distância, clima, tráfego, etc.) e auxiliar no processo de tomada de decisão da equipe de resposta a emergências sobre qual o usuário mais indicado a ser convocado para prestar informações sobre a vítima, observando-se as condições de urgência do resgate.

O Google Latitude (GOOGLE, 2010) é um serviço que permite encontrar pessoas e visualizar suas localizações através dos mapas do Google Maps. Ele apresenta um modo fácil de compartilhamento da sua localização (exata ou próxima) com seus amigos, caso eles também aceitem compartilhar essas informações. O serviço é construído em um conceito básico, a localização do usuário. Uma localização é um registro de onde o usuário estava em um determinado momento. Ela sempre contém uma latitude, longitude e um *timestamp*, além de opcionalmente conter mais detalhes sobre a posição do usuário na época, tais como altitude ou metadados sobre a posição, como sua precisão, por exemplo.

Um usuário possui uma localização atual, que representa sua posição mais recente conhecida. Essa localização atual vai mudando conforme o usuário se move no mundo e sua nova localização é relatada. Em alguns casos, no entanto, um usuário pode optar por se ocultar temporariamente, tornando a sua posição invisível ao excluir sua localização atual. Portanto, um usuário oculto não tem localização atual. Um usuário pode ter um histórico de localizações, que é um conjunto de todos os seus locais armazenados como uma série ao longo do tempo. O local mais recente do histórico de localizações geralmente contém os mesmos dados que a localidade atual do usuário. Porém, nem todos os usuários possuem um histórico de localizações, já que esta é uma característica opcional do Google Latitude.

O Google Maps (GOOGLE, 2005b) é um serviço de pesquisa e visualização de mapas e imagens de satélite da Terra. Com esse serviço também é possível encontrar empresas cadastradas em qualquer lugar, ou visualizar empresas ou locais públicos próximos a um determinado ponto especificado pelo usuário. O usuário que possuir uma conta Google, também pode destacar as suas próprias rotas, pontos e áreas, além de gerar comentários sobre determinados lugares e compartilhar seus respectivos links de acesso a esses mapas criados.

Dessa forma, é possível construir facilmente aplicativos baseados em localidades, conectando pessoas a lugares e impulsionando a visualização e consequente exploração do espaço em torno do usuário. Sua poderosa base de dados possui atualmente mais de 50 milhões de negócios e pontos de interesse ao redor do mundo.

O serviço ainda oferece diversas maneiras de incorporar as direções do Google Maps em suas aplicações. Os usuários podem encontrar as melhores rotas de condução e o tempo que leva para chegarem ao seu destino. Além disso, eles também podem utilizar o Street View para terem uma visão abrangente de onde estão indo. Tudo isso pode ser acessado via dispositivos móveis, através da SDK nativa para Android.

O Google Weather é um serviço criado para uso em conjunto com o iGoogle, uma página inicial de pesquisa personalizável oferecida pelo Google, onde o usuário pode adicionar notícias, fotos, previsão do tempo e tudo que desejar da *web* na sua própria página. Apesar desse serviço não possuir uma API documentada, sua interface é facilmente detectável e compreende um punhado de outras características, além do clima. Quando comparada a outras APIs de clima, ela se mostra superior, apesar de não possuir documentação ou suporte próprio da empresa. Uma das razões pelas quais este serviço se mostra bastante útil e superior aos demais é que o Google deixou pouco trabalho para o desenvolvedor. Ao contrário de outras APIs, a resposta XML para as chamadas da API do Google Weather é bastante explícita. Não há a necessidade de se descobrir o que significa uma abreviação específica ou qualquer outro código. Ou existe uma resposta positiva (com as condições climáticas), ou então uma resposta notificando de um erro durante o processamento da requisição.

De fato, a chamada à API é muito simples, necessitando apenas passar o nome de uma cidade ou então seu código postal, como mostra a seguinte consulta <http://www.google.com/ig/api?weather=Rio+Janeiro>. A resposta XML a essa requisição é exibida na Figura 15. Como pode ser visto na Figura 15, a resposta XML do Google Weather é facilmente compreensível pelos seres humanos (*human-readable XML*). De nossa parte, cabe novamente ressaltar que apesar desse serviço estar disponível e acessível dessa maneira desde o seu lançamento, a API em si não é oficialmente pública ainda e seu termo de uso declara que o serviço é para utilização somente como *gadget* do iGoogle. Existem algumas documentações extraoficiais *online* espalhadas pela *web* e criadas pelos usuários do serviço, mas não há uma documentação oficial ainda. Essa API deve ser usada com cautela e devemos esperar mudanças no futuro. Então, antes de lançar uma versão oficial dessa proposta de solução, o termo de uso desse serviço deveria ser estudado com mais calma para saber se sua utilização neste trabalho infringiria de alguma maneira o propósito de uso do serviço.

```

- <xml_api_reply version="1">
- <weather module_id="0" tab_id="0" mobile_row="0" mobile_zipped="1" row="0" section="0">
- <forecast_information>
  <city data="Rio de Janeiro, Rio de Janeiro"/>
  <postal_code data="Rio Janeiro"/>
  <latitude_e6 data=""/>
  <longitude_e6 data=""/>
  <forecast_date data="2012-06-24"/>
  <current_date_time data="1970-01-01 00:00:00 +0000"/>
  <unit_system data="SI"/>
</forecast_information>
- <current_conditions>
  <condition data="Nublado"/>
  <temp_f data="75"/>
  <temp_c data="24"/>
  <humidity data="Umidade: 73%"/>
  <icon data="fig/images/weather/cloudy.gif"/>
  <wind_condition data="Vento: S a 16 km/h"/>
</current_conditions>
- <forecast_conditions>
  <day_of_week data="dom"/>
  <low data="18"/>
  <high data="25"/>
  <icon data="fig/images/weather/sunny.gif"/>
  <condition data="Sol"/>
</forecast_conditions>
- <forecast_conditions>
  <day_of_week data="seg"/>
  <low data="19"/>
  <high data="24"/>
  <icon data="fig/images/weather/mostly_sunny.gif"/>
  <condition data="Parcialmente ensolarado"/>
</forecast_conditions>
- <forecast_conditions>
  <day_of_week data="ter"/>
  <low data="18"/>
  <high data="24"/>
  <icon data="fig/images/weather/mostly_sunny.gif"/>
  <condition data="Ensolarado na maioria"/>
</forecast_conditions>
- <forecast_conditions>
  <day_of_week data="qua"/>
  <low data="19"/>
  <high data="25"/>
  <icon data="fig/images/weather/sunny.gif"/>
  <condition data="Sol"/>
</forecast_conditions>
</weather>
</xml_api_reply>

```

[Continuação da Resposta]

Figura 15. Resposta XML da API do Google Weather

O Climatedo (2009) é um serviço gratuito de informações climáticas, semelhante ao Google Weather, e fornecido pela Climatedo, a maior empresa de Meteorologia da América Latina. Sua utilização é significativamente pior do que a API do Google Weather. A API do Climatedo fornece dados como temperaturas mínima e máxima, a probabilidade de chuvas e pluviometria (quantidade de chuvas que cai numa determinada região). A exemplo do Google Weather, sua API também não é documentada, mas a resposta XML enviada pelo serviço não é tão clara quanto a do Google Weather.

A requisição ao serviço deve ser feita passando como parâmetro o código da cidade, que deve ser retirado diretamente do site do Climatedo. Esse fato de ter que usar um código próprio do serviço dificultaria uma possível integração do sistema, uma vez que sempre que esses códigos alterarem, o aplicativo teria que ser atualizado para refletir essas alterações dos códigos das cidades cadastradas no serviço do Climatedo. Um exemplo de requisição feita ao serviço é o seguinte <http://selos.climatedo.com.br/selos/selo.php?CODCIDADE=321>. A resposta a essa requisição pode ser visualizada na Figura 16.

```

- <selos>
  <video urlprev="http://videos.climatempo.com.br/selos/video.swf" urlhkt="http://videos.climatempo.com.br/mkt/video1.swf"/>
  <parametro qtd="4" video="se"/>
  <cidade nome="RJ - Rio de Janeiro" data="24/06 Dom" low="15" high="28" prob="0" mm="0" ico="2" frase="Sol com
  muitas nuvens durante o dia e períodos de céu nublado. Noite com muitas nuvens."/>
  <cidade nome="RJ - Rio de Janeiro" data="25/06 Seg" low="17" high="27" prob="0" mm="1" ico="2" frase="Sol com
  muitas nuvens durante o dia e períodos de céu nublado. Noite nublada com possibilidade de garoa."/>
  <cidade nome="RJ - Rio de Janeiro" data="26/06 Ter" low="17" high="25" prob="0" mm="1" ico="2" frase="Nublado pela
  manhã, com possibilidade de garoa. Tarde de sol com diminuição de nuvens. Noite com muita nebulosidade."/>
  <cidade nome="RJ - Rio de Janeiro" data="27/06 Qua" low="16" high="27" prob="0" mm="0" ico="2" frase="Sol com
  algumas nuvens. Não chove."/>
</selos>

```

Figura 16. Resposta XML de uma requisição ao serviço do Climatempo

Existem duas possibilidades de implementação da atividade de análise dessas informações contextuais referentes à distância física dos usuários. A mais simples delas é apenas exibir para a equipe de resposta a emergências as coordenadas GPS do usuário, uma vez autorizado pelo próprio, e deixar que os próprios integrantes da equipe operacional acessem manualmente esses serviços citados anteriormente e tirem suas próprias conclusões através da utilização desses serviços. A outra opção seria realizar a integração desses serviços com nossa proposta de solução através de suas APIs.

Dessa forma, ao selecionar um usuário para disponibilizar informações úteis sobre a vítima da emergência, o servidor comando e controle realizaria a pesquisa nessas APIs, utilizando os dados de coordenadas GPS informados pelo usuário, e exibiria todas as informações contextuais relativas à distância física, para que a equipe de resposta a emergências pudesse tomar a decisão se esse usuário poderia ser convocado para prestar informações sobre a vítima ou se outro usuário estaria mais facilmente acessível para poder disponibilizar essas informações sobre a vítima da emergência. Mais à frente explicaremos melhor o funcionamento de nossa proposta de solução, mostrando um workflow dos eventos e atividades envolvidas no uso de nosso trabalho.

As atividades e serviços do servidor comando e controle, que foram descritos nesta seção, estão mostrados na Figura 17.

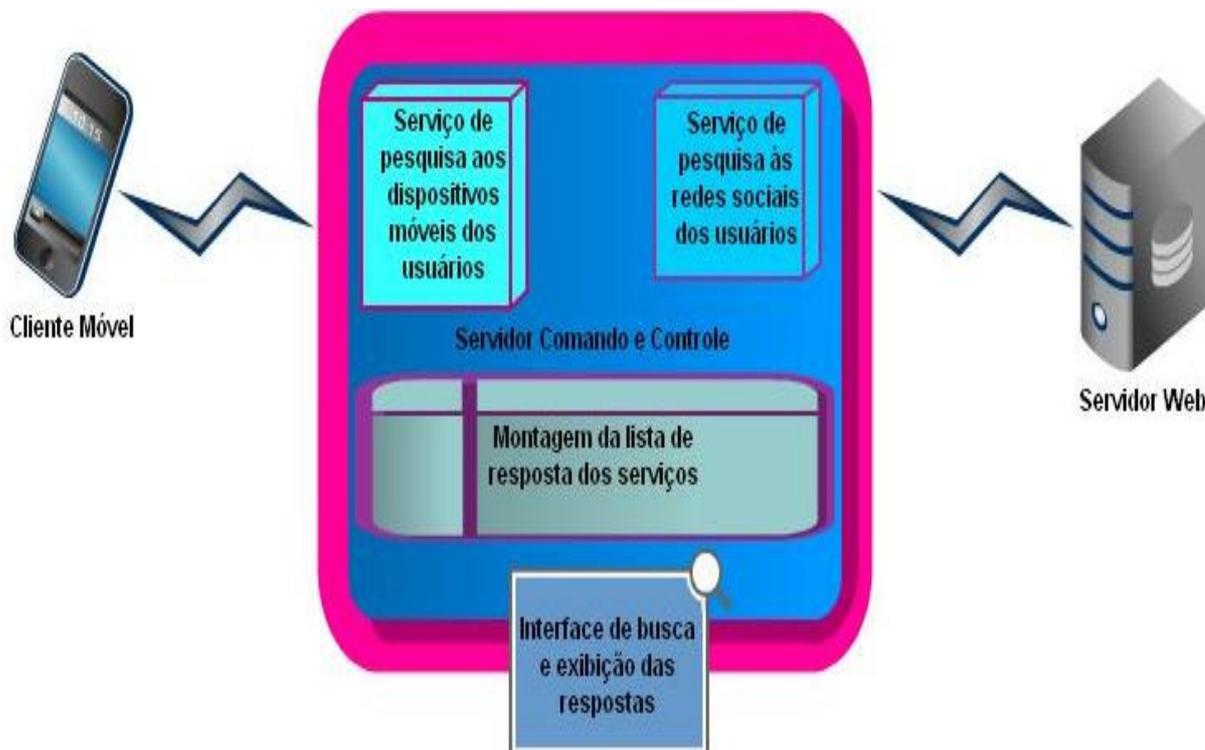


Figura 17. Atividades e serviços do Servidor Comando e Controle

Para a realização dessas atividades e serviços faz-se necessária a construção de alguns módulos e componentes, conforme descrito a seguir:

- **Serviço de pesquisa às redes sociais:** é o serviço que ficará responsável por intermediar a comunicação entre o Servidor Web e o Servidor Comando e Controle, enviando para aquele o nome da vítima a ser pesquisado e recebendo o resultado da análise contextual da rede social da vítima como resposta.
- **Serviço de pesquisa aos dispositivos móveis:** é o serviço responsável por intermediar a comunicação entre o Cliente Móvel e o Servidor Comando e Controle, enviando para aquele o nome da vítima a ser pesquisado e recebendo o resultado da análise contextual da rede social do usuário como resposta.
- **Gerente de Respostas:** é o componente responsável aguardar e sincronizar todas as respostas, montando assim uma listagem única de contatos e suas relevâncias sociais agrupados por usuários (donos dos dispositivos móveis).
- **Interface:** é a interface de entrada de dados para a pesquisa (nome da vítima) e a saída de dados dessa pesquisa (lista de contatos e suas relevâncias sociais, agrupados por usuários — Clientes Móveis).

A arquitetura do Servidor Comando e Controle, contendo os módulos e componentes acima descritos, pode ser vista na Figura 18.

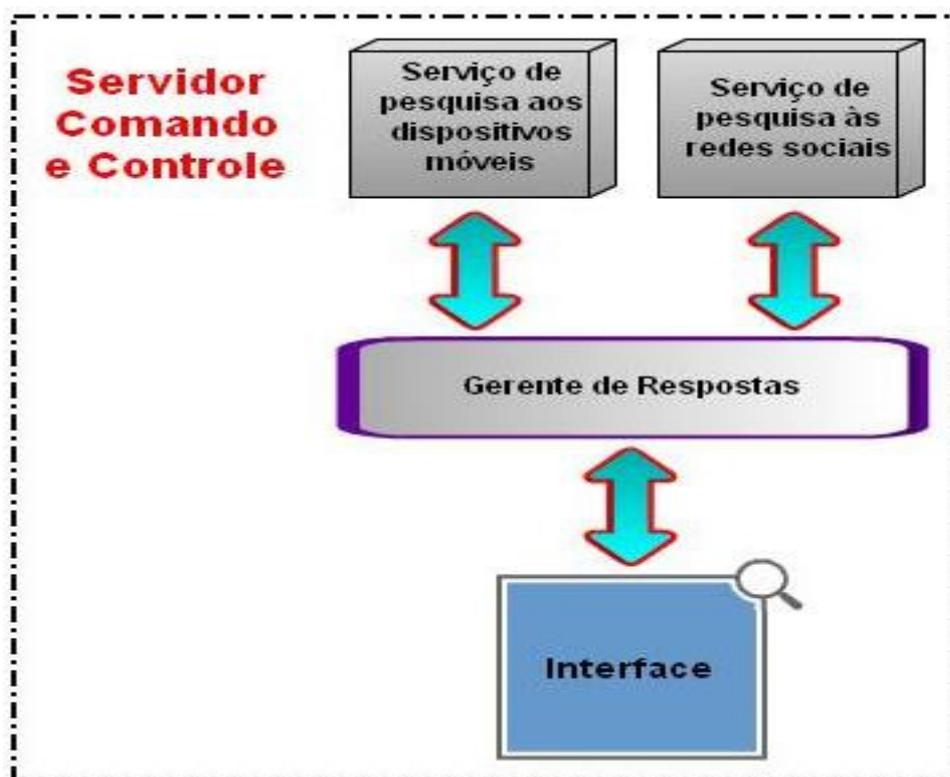


Figura 18. Arquitetura do Servidor Comando e Controle

5.4.3 Cliente Móvel

O cliente móvel é um módulo da nossa proposta de solução que reside nos dispositivos móveis dos usuários, os cidadãos ordinários de nossa sociedade. Ele é construído especificamente para utilização na plataforma Android, assim como o servidor móvel. Sua principal responsabilidade é extrair os dados da lista de contatos do usuário e enviar para o servidor web, para que esses dados possam ser combinados com aqueles provenientes das mídias sociais, formando o *cache* de informações contextuais da rede social do usuário. Este módulo também é responsável por receber as requisições do servidor comando e controle, com a pesquisa pelo nome da vítima, e retornar uma resposta com a lista de contatos que conferem com a busca pelo nome da vítima.

A funcionalidade de coleta dos dados da lista de contatos dos dispositivos móveis (troca de mensagens SMS/MMS, e-mails, ligações realizadas e recebidas, duração total dessas chamadas, etc.) é uma atividade executada periodicamente em segundo plano — *background* —, de forma configurável pelo usuário, e que envia esses dados para processamento no servidor web, atualizando dessa forma o *Data Warehouse* que funcionará como um *cache* das informações contextuais dos contatos dos usuários do sistema.

Já a atividade de pesquisa pelo nome da vítima é um serviço que fica executando permanentemente no dispositivo móvel (que pode ser desativado e reativado pelo usuário a

qualquer momento), aguardando por uma requisição do servidor comando e controle. Quando esta requisição é corretamente identificada, o serviço do cliente móvel então entra em contato com o servidor web, requisitando a análise contextual da rede social do usuário em busca da vítima em questão. Quando o servidor web retorna uma resposta com a lista dos contatos identificados, ordenados pela ordem de relevância social com o usuário, ela é devidamente repassada para o servidor comando e controle, juntamente com algumas informações contextuais do usuário em questão, como resposta à requisição original.

Caso haja alguma falha na conexão ao servidor web, o cliente móvel então analisa somente os dados disponíveis no dispositivo móvel do usuário e calcula o índice de relevância social de cada contato que confere com a busca original (nome da vítima). Dessa forma, ao invés de não retornar resposta alguma ao servidor de comando e controle, é retornada a lista dos contatos socialmente relevantes ao usuário, calculando-se o índice de relevância social baseado somente nas informações contidas no dispositivo móvel. Essa é uma medida adotada no projeto desta proposta de trabalho, para amenizar os pontos de falhas existentes em cada etapa do processo de funcionamento desta proposta de solução. As atividades e os processos do cliente móvel podem ser visualizados na Figura 19.

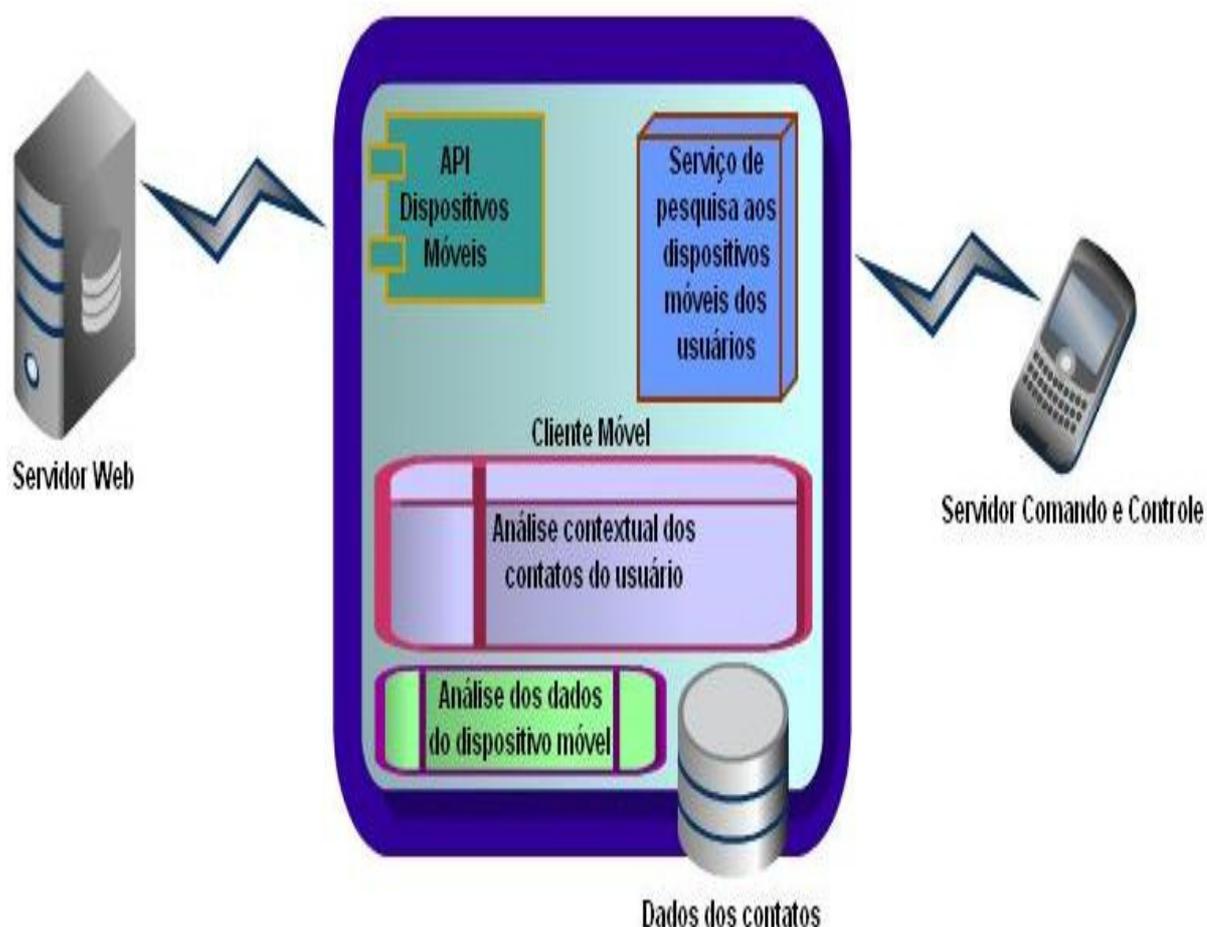


Figura 19. Atividades e processos do Cliente Móvel

As informações contextuais do usuário que são enviadas para o servidor comando e controle serão utilizadas no processo de escolha de qual usuário é o mais indicado para compartilhar as informações sobre a vítima, tal como se o usuário pode chegar rapidamente ao local de emergência. Para a realização dessas atividades e processos faz-se necessária a construção de alguns módulos e componentes, conforme descrito a seguir:

- **Serviço de pesquisa ao dispositivo móvel:** é o serviço que ficará responsável por i) periodicamente executar a rotina de atualização dos dados de interação entre o usuário e seus contatos, seja através de e-mails, mensagens de texto, ligações telefônicas ou outras formas, enviando-os para o Servidor Web através da API Dispositivo Móvel, e ii) intermediar a comunicação entre o Cliente Móvel e o Servidor Comando e Controle, recebendo desse o nome da vítima a ser pesquisado e enviando o resultado da análise contextual da rede social da vítima como resposta.
- **API Dispositivo Móvel:** é o módulo responsável por i) enviar para o Servidor Web os dados do usuário coletados, como suas interações com os contatos do celular e ii) intermediar a comunicação entre o Cliente Móvel e o Servidor Web, enviando para esse o nome da vítima a ser pesquisado e recebendo o resultado da análise contextual da rede social do usuário como resposta.
- **Analizador de Contextos:** é o componente responsável por realizar a análise contextual dos dados presentes no dispositivo móvel do usuário.
- **Gerente de dados do dispositivo móvel:** é o módulo responsável por manipular os dados presentes no dispositivo móvel do usuário, efetivamente coletando as principais informações de interação do usuário com seus contatos e outros dados relativos ao próprio usuário e seus contatos.
- **Base de Contatos:** é o conjunto de dados pertinentes aos contatos do usuário no dispositivo móvel, incluindo as informações de interação desses contatos com o usuário (e-mails, mensagens de texto, telefonemas, etc.).

A arquitetura do Cliente Móvel, contendo os módulos e componentes acima descritos, pode ser vista na Figura 20. Na seção seguinte descreveremos como foram construídos alguns desses módulos, componentes, APIs e serviços de nossa proposta de solução, a fim de que criássemos uma prova de conceito para realizar uma análise inicial do presente trabalho.

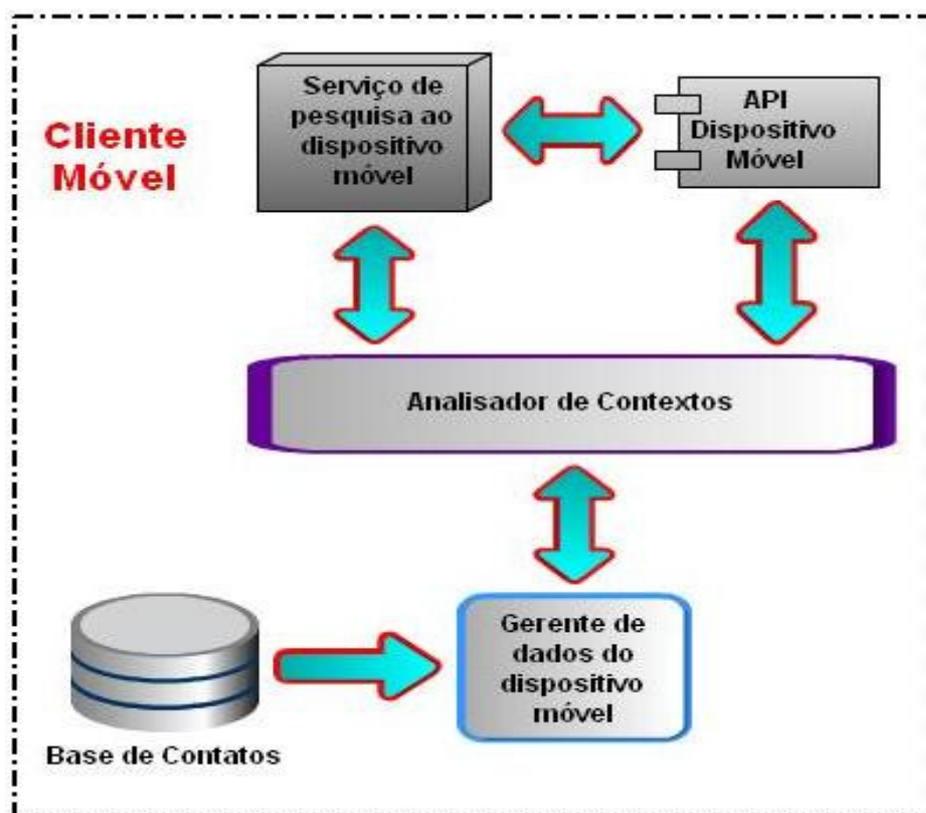


Figura 20. Arquitetura do Cliente Móvel

5.5 Implementação

A ferramenta móvel construída como prova de conceito para nossa proposta de solução do problema possui as seguintes funcionalidades descritas a seguir. Localização de usuários (vítimas desaparecidas ou pessoas que possam prestar informações importantes para ajudar no processo de tomada de decisão para a solução da emergência). A fim de respeitar a privacidade do usuário, disponibilizamos três opções para essa funcionalidade: localizador automático, que permite a disponibilização da localização do usuário a qualquer requisição; permissão manual para enviar a localização do usuário, que solicita a permissão do usuário antes de divulgar sua localização para a requisição; nunca divulgar localização, que desabilita essa funcionalidade de localização do usuário.

Consultas sobre pessoas que possuam informações sobre as vítimas da emergência para ajudar em eventuais necessidades que porventura a equipe de resposta a emergências possa ter. Como já explicado anteriormente, essas consultas levam em consideração a distância social da vítima em relação a essas pessoas recomendadas e também a distância física que essas pessoas se encontram do local da emergência, dentre outras informações contextuais já citadas.

Por fim, para se viabilizar todo esse trabalho de busca por pessoas próximas à vítima de uma emergência, se faz necessário um registro prévio dos usuários do sistema, para a formação de um *cache* com informações provenientes da *internet*, das mídias sociais e da rede social do usuário, visando diminuir o custoso processamento em tempo real em busca das informações da rede social das pessoas envolvidas no processo.

Para a construção de uma prova de conceito de nossa proposta, optamos por utilizar a plataforma Android para dispositivos móveis, por usar licença *open-source* e ser um dos sistemas líderes do mercado de tecnologia móvel, além de possuir uma considerável gama de documentação disponível através da *internet*. O Android é um sistema operacional baseado em Linux para dispositivos móveis (como, por exemplo, *smartphones* e *tablets*) e desenvolvido pela *Open Handset Alliance*, liderada pelo Google e outras companhias.

O Android e seu código fonte são distribuídos pelo Google, sob a Licença Apache. Suas principais características e especificações são:

- **Visor:** A plataforma é adaptável a *layouts* tradicionais de *smartphones* e maiores, tais como VGA, bibliotecas de gráficos 2D e gráficos 3D com base em especificações OpenGL ES 2.0.
- **Armazenamento:** o SQLite, um banco de dados relacional leve, é usado para fins de armazenamento de dados.
- **Conectividade:** o Android suporta várias tecnologias de conectividade, incluindo GSM, EDGE, IDEN, CDMA, EVDO, UMTS, Bluetooth, Wi-Fi, LTE, WiMAX e NFC.
- **Mensagens:** SMS e MMS são formas disponíveis de troca de mensagens. As mensagens de texto são geralmente agrupadas em threads, por padrão. Atualmente, o *Android Cloud To Device Messaging* (C2DM) é também uma parte do *Android Push Messaging service*.
- **Suporte a Java:** Apesar da maioria dos aplicativos do Android serem escritos em Java, não há nenhuma máquina virtual Java na plataforma e o código Java Byte não é executado. Classes Java são compilados em executáveis Dalvik e executados em Dalvik, uma máquina virtual especial projetada especificamente para o Android e otimizada para dispositivos móveis alimentados por bateria e com limitações de memória e CPU. Suporte a J2ME pode ser prestado através de aplicativos de terceiros.

- **Suporte de hardware adicional:** o Android pode usar câmeras de vídeo ou fotos, telas sensíveis ao toque, GPS, acelerômetros, giroscópios, barômetros, magnetômetros, controladores de jogos dedicados, sensores de proximidade e de pressão, termômetros, aceleradores blits bit 2D (com orientação de *hardware*, escala, conversão de formato *pixel*) e aceleradores gráficos 3D.
- **Características gerais:** suporte a múltiplos idiomas, *web browser* baseado no *open-source WebKit layout engine*, suporte de mídia e *streaming media*, suporte nativo para *multi-touch*, *multitasking* de aplicações (com um gerenciador exclusivo de alocação de memória), recursos baseados em voz, captura de tela e armazenamento externo. O Android também suporta *tethering* (permitindo que um telefone seja usado como um *hotspot Wi-Fi* com ou sem fios), mas não suporta chamadas de vídeo nativamente. Porém, alguns aparelhos possuem uma versão personalizada do sistema operacional que as suporta, seja através da rede UMTS (como o Samsung Galaxy S) ou VoIP.

Das diversas versões do Android existentes, escolhemos trabalhar com a 2.3.X (codinome Gingerbread) que possui uma API nível 9-10, de acordo com a nomenclatura padrão Android. Essa escolha deveu-se ao fato dessa versão ser a mais utilizada atualmente. A seguir, mostramos no Quadro 4 as versões do Android com suas distribuições de uso e na Figura 21 um gráfico de uso dessas versões do Android baseado em dados coletados durante um período de 14 dias encerrado em 01 de junho de 2012 (DEVELOPERS, 2012).

Versão	Codinome	Lançamento	Nível de API	Distribuição
1.5	Cupcake	30/04/2009	3	0,3%
1.6	Donut	15/09/2009	4	0,6%
2.1	Eclair	26/10/2009	7	5,2%
2.2	Froyo	20/05/2010	8	19,1%
2.3.X	Gingerbread	06/12/2010	9 – 10	65,0%
3.X	Honeycomb	22/02/2011	11 – 13	2,7%
4.0.X	Ice Cream Sandwich	19/10/2011	14 – 15	7,1%
4.1.X	Jelly Bean	28/06/2012	16	N/A

Quadro 4. Diferentes versões do Android e distribuição de uso (DEVELOPERS, 2012)

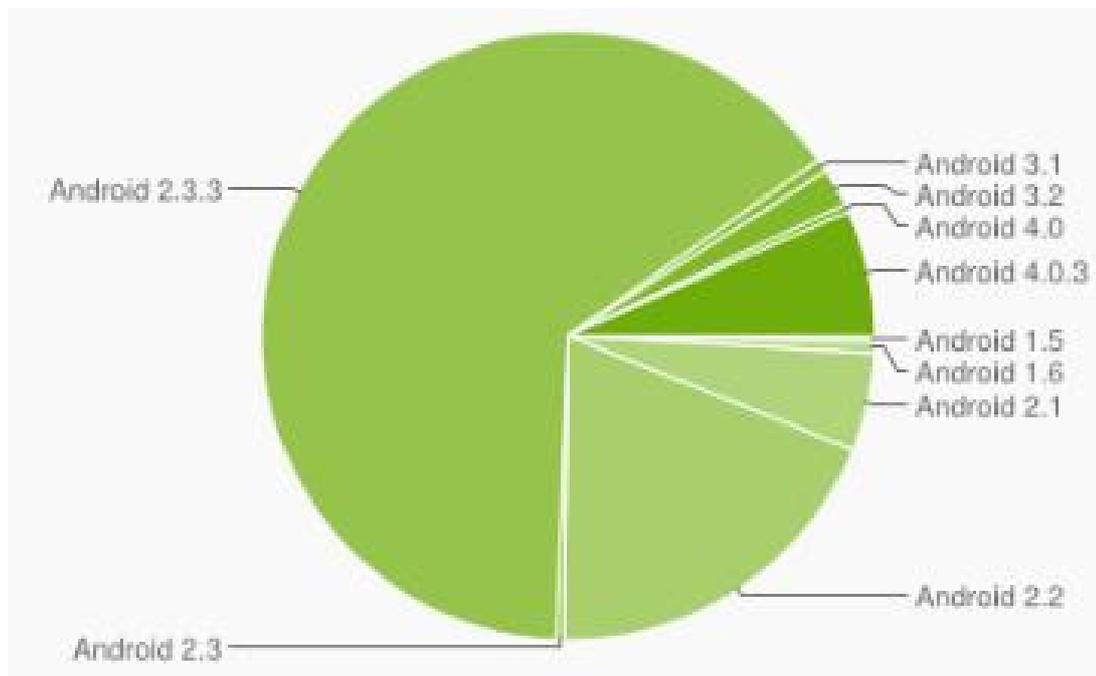


Figura 21. Distribuição de uso das diferentes versões, 01 de junho de 2012 (DEVELOPERS, 2012)

Portanto, devido a essas suas características e especificações, além de sua participação no mercado de dispositivos móveis, a plataforma Android (especialmente na versão 2.3.X – codinome Gingerbread) mostrou-se a mais viável para fornecer esses dados relativos ao cálculo e análise da distância social e da distância física dos usuários. Nas seções a seguir descrevemos os SDKs utilizados na construção do protótipo, bem como explicamos todas as atividades envolvidas na construção do protótipo, inclusive debatendo as maiores dificuldades encontradas.

5.5.1 Android SDK

O Android SDK é utilizado para se desenvolver aplicativos para o sistema móvel Android, podendo-se inclusive testá-los tanto em um dispositivo móvel real ou em um emulador (uma máquina virtual que simula a versão do sistema Android desejado). Ele é composto de pacotes modulares que podem ser obtidos separadamente utilizando seu gerenciador. Existem vários pacotes distintos disponíveis para o Android SDK, dentre os quais nos vale citar (ANDROID, 2012b):

- **SDK Tools:** contém ferramentas para depuração e testes, além de outras utilidades que são necessárias para desenvolver um aplicativo.
- **SDK Platform-tools:** contém ferramentas dependentes de plataforma para desenvolvimento e depuração das aplicações. Essas ferramentas suportam os mais recentes recursos da plataforma Android e normalmente são atualizadas

somente quando uma nova plataforma se torna disponível, mantendo sempre a compatibilidade com plataformas mais antigas.

- **SDK Platform:** há uma plataforma SDK disponível para cada versão do Android. Ela inclui um arquivo chamado “android.jar” contendo uma biblioteca totalmente compatível com o sistema Android. É utilizada para criar um aplicativo Android (*build target*).
- **Imagens do Sistema:** cada versão da plataforma oferece uma ou mais imagens diferentes do sistema (tal como para as arquiteturas ARM e x86). O emulador Android requer uma imagem do sistema para funcionar. É recomendável sempre testar o aplicativo na versão mais recente do Android, utilizando primeiro um emulador com a imagem mais recente do sistema.
- **Google APIs:** ferramenta extra do SDK que fornece uma plataforma que pode ser utilizada para desenvolver um aplicativo usando as APIs especiais do Google e uma imagem do sistema para o emulador para que se possa testar o aplicativo usando essas APIs do Google.
- **Suporte Android:** uma biblioteca estática que pode ser incluída nos aplicativos para usarem APIs poderosas que não estão disponíveis na plataforma padrão. Por exemplo, a biblioteca de suporte contém versões da classe Fragment que é compatível com as versões do Android 1.6 e superiores (a classe foi originalmente introduzida no Android 3.0) e as APIs ViewPager que permitem a construção de uma interface de usuário *side-swipeable* de uma maneira muito simples e fácil.
- **Google Play Billing:** fornece bibliotecas estáticas e exemplos que permitem a integração dos serviços de faturamento no aplicativo com o Google Play (plataforma oficial de conteúdos do Google, incluindo a loja virtual do Android, onde estão disponíveis todos os aplicativos destinados ao sistema, conhecida anteriormente como Android Market).
- **Google Play Licensing:** fornece bibliotecas estáticas e exemplos que permitem a realização de verificação de licença para a aplicação ao distribuir com o Google Play.

Além dessas ferramentas já citadas, ainda existe a possibilidade de utilização do Android NDK, que é um conjunto de ferramentas que permite implementar partes do aplicativo usando código nativo em linguagens como C e C++. Para determinados tipos de

aplicativos isso pode ser útil, ao reutilizar bibliotecas de código existentes escritas nessas linguagens e com desempenho possivelmente maior.

Tipicamente, utilizar código nativo no Android não necessariamente resulta em uma melhoria de desempenho perceptível, mas sempre aumenta a complexidade do aplicativo. Portanto, essa ferramenta deve ser utilizada com muito cuidado e somente caso não haja alternativas viáveis utilizando o próprio Android SDK ou bibliotecas Android já disponíveis. Bons candidatos típicos para o uso do NDK são operações intensivas e autocontidas de CPU que não alocam muita memória, tais como processamento de sinais, simulação de física, e assim por diante.

As aplicações Android são escritas na linguagem de programação Java. As ferramentas do Android SDK compilam o código — juntamente com todos os dados e arquivos de recursos — em um pacote Android, representado por um arquivo com o sufixo “.apk”. Todo o código em um arquivo “.apk” único é considerado como uma aplicação. Este é o arquivo que os dispositivos Android usam para instalar o aplicativo.

Uma vez instalado em um dispositivo, cada aplicativo Android vive em sua própria área de segurança (ANDROID, 2012a):

- O sistema operacional Android é um sistema multi-usuário do Linux, onde cada aplicação é um usuário diferente.
- Por padrão, o sistema atribui a cada aplicação um identificador único do usuário Linux (o ID é usado apenas pelo sistema e é desconhecido para a aplicação). O sistema define as permissões para todos os arquivos em um aplicativo de modo que apenas o ID do usuário atribuído a esse aplicativo pode acessá-los.
- Cada processo tem sua própria máquina virtual, para que o código de um aplicativo seja executado isoladamente de outras aplicações.
- Por padrão, cada aplicativo é executado em seu próprio processo Linux. O Android inicia o processo quando qualquer um dos componentes do aplicativo precisa ser executado, e depois encerra o processo quando ele não é mais necessário ou quando o sistema precisa recuperar o espaço na memória para outras aplicações.

Dessa maneira, o sistema Android implementa o princípio de privilégio mínimo. Ou seja, cada aplicação, por padrão, tem acesso somente para os componentes que ela necessita para fazer seu trabalho e nada mais. Isso cria um ambiente muito seguro onde um aplicativo

não pode acessar partes do sistema para as quais não são dadas as devidas permissões. No entanto, existem formas de um aplicativo compartilhar dados com outras aplicações e de um aplicativo acessar determinados serviços do sistema:

- É possível permitir que dois aplicativos compartilhem o mesmo ID do usuário Linux, caso em que eles serão capazes de acessar os arquivos um do outro. Para conservar os recursos do sistema, aplicações com o mesmo ID do usuário também podem ser executadas no mesmo processo Linux e compartilharem a mesma máquina virtual (as aplicações também devem ser assinadas com o mesmo certificado).
- Um aplicativo pode solicitar permissão para acessar dados do dispositivo, como contatos do usuário, mensagens SMS, armazenamento montável (cartão SD), câmera, Bluetooth e muito mais. Todas as permissões de aplicativos devem ser concedidas pelo usuário no momento da instalação.

O desenvolvimento em Android é basicamente modular. Uma aplicação em Android é tipicamente dividida em blocos de construção essenciais (ou módulos), chamados de componentes. Cada componente é um ponto de acesso diferente através do qual o sistema pode interagir com a aplicação. Nem todos os componentes são pontos de entrada reais para o usuário e ainda alguns podem depender de outros, mas cada um existe como uma entidade própria e desempenha um papel específico — cada um é um bloco de construção singular que ajuda a definir o comportamento geral do aplicativo.

No Android SDK há quatro tipos diferentes de componentes de aplicação. Cada tipo possui uma finalidade distinta e tem um ciclo de vida distinto que define como o componente é criado e destruído. A seguir são listados os quatro tipos de componentes de aplicação:

- **Atividades:** uma atividade representa uma única tela com uma interface de usuário. Por exemplo, um aplicativo de e-mail pode ter uma atividade que mostra uma lista de e-mails novos, outra atividade para compor um e-mail, e uma outra atividade para ler e-mails. Embora as atividades trabalhem juntas para formar uma experiência de usuário coesa na aplicação de e-mail, cada uma é independente das outras. Assim sendo, um aplicativo diferente pode começar qualquer uma dessas atividades, se o aplicativo de e-mail assim o permitir. Por exemplo, uma aplicação da câmera pode iniciar a atividade na aplicação de e-mail que compõe uma nova mensagem, para que o usuário compartilhe a foto com seus colegas. Uma atividade é implementada como uma subclasse de Activity.

- **Serviços:** um serviço é um componente que roda em segundo plano para executar operações de longa duração ou executar trabalhos através de processos remotos. Um serviço não fornece uma interface de usuário. Por exemplo, um serviço pode reproduzir música em segundo plano enquanto o usuário estiver em um aplicativo diferente, ou pode buscar dados em rede sem bloquear a interação do usuário com uma atividade. Outro componente, como uma atividade, por exemplo, pode iniciar o serviço e deixá-lo executando direto ou se conectar a ele para interagir durante todo o processo. Um serviço é implementado como uma subclasse de `Service`.
- **Provedores de Conteúdo:** um provedor de conteúdo gerencia um conjunto compartilhado de dados do aplicativo. Os dados podem ser armazenados no sistema de arquivos, em um banco de dados SQLite, na *web*, ou qualquer outro local de armazenamento persistente que o aplicativo possa acessar. Através do provedor de conteúdo, outras aplicações podem consultar ou até mesmo modificar os dados, se o provedor de conteúdo assim o permitir. Por exemplo, o sistema Android fornece um provedor de conteúdo que gerencia informações de contatos do usuário. Dessa forma, qualquer aplicativo com as permissões adequadas pode consultar parte do provedor de conteúdo (como, por exemplo, `ContactsContract.Data`) para ler e gravar informações sobre uma pessoa em particular. Os provedores de conteúdo também são úteis para leitura e escrita de dados privados à aplicação e não compartilhados. Um provedor de conteúdo é implementado como uma subclasse de `ContentProvider` e deve implementar um conjunto padrão de APIs que permitem outras aplicações realizarem transações.
- **Receptores Broadcast:** um receptor *broadcast* é um componente que responde a anúncios de transmissão de todo o sistema (*broadcast* ou *broadcasting*). Muitas transmissões são provenientes do próprio sistema, por exemplo, uma transmissão anunciando que a tela tenha desligado, que a bateria está fraca, ou que uma imagem foi capturada. Os aplicativos também podem iniciar transmissões, por exemplo, deixando outras aplicações saberem que alguns dados foram baixados para o dispositivo e estão disponíveis para elas usarem. Embora receptores *broadcast* não exibam uma interface de usuário, eles podem criar uma notificação na barra de status para alertar o usuário quando um evento de transmissão ocorre. Mais comumente, porém, um receptor de

broadcast é apenas uma “porta” para outros componentes e destina-se a fazer uma quantidade muito reduzida de trabalho. Por exemplo, iniciar um serviço para executar algum trabalho com base no evento ocorrido. Um receptor *broadcast* é implementado como uma subclasse de `BroadcastReceiver` e cada transmissão é entregue como um objeto `Intent`.

Uma característica peculiarmente única do projeto do sistema Android é que qualquer aplicação pode iniciar um componente de outro aplicativo. Por exemplo, se uma aplicação permite que o usuário capture uma foto com a câmera do dispositivo, provavelmente outro aplicativo poderá usar essa funcionalidade em vez de desenvolver uma atividade própria para capturar uma foto. Não será necessário incorporar nem mesmo executar o código do aplicativo da câmera. Em vez disso, pode-se simplesmente começar a atividade no aplicativo da câmera que captura uma foto. Quando terminada, a foto é retornada para que a aplicação original possa usá-la. Para o usuário é tudo transparente, parecendo como se a câmera fosse realmente uma parte daquela aplicação.

Quando o sistema inicia um componente, ele inicia o processo para aquela aplicação (se ela já não estiver em execução) e instancia as classes necessárias para o componente. Por exemplo, se um aplicativo inicia a atividade que captura uma foto na aplicação da câmera, essa atividade é executada no processo que pertence à aplicação da câmera, não no processo do aplicativo original. Portanto, ao contrário de aplicações na maioria dos outros sistemas, os aplicativos Android não possuem um único ponto de entrada, mas sim múltiplos pontos de entrada, desde que disponibilizados pelo próprio aplicativo.

Um aplicativo não pode ativar diretamente um componente de outra aplicação, porque o sistema executa cada aplicação em processos separados com permissões de arquivos que restringem o acesso a outros aplicativos. O sistema Android, no entanto, pode. Então, para ativar um componente em outro aplicativo, uma aplicação deve entregar uma mensagem ao sistema que especifica sua intenção de iniciar um componente em particular. O sistema, então, ativa o componente para a aplicação. Dessa forma, o sistema Android está sempre no controle, já que ele tem que servir de intermediário para todos os aplicativos e é ele quem verifica se os componentes podem ou não ser acessados por outras aplicações.

No sistema Android, existem duas formas distintas de se ativar esses componentes dos aplicativos: mensagens assíncronas ou requisições síncronas. Três dos quatro tipos de componentes — atividades, serviços e receptores *broadcast* — são ativados por uma mensagem assíncrona chamada `intent`. `Intents` vinculam componentes individuais uns aos outros em tempo de execução (eles são como mensageiros que solicitam uma ação a outros

componentes), independentemente do componente pertencer a uma aplicação ou à outra. Um intent é criado com um objeto Intent, que define uma mensagem para ativar um componente específico ou um tipo específico de componente — um intent pode ser explícito ou implícito, respectivamente.

Para atividades e serviços, um intent define a ação a ser executada (por exemplo, “ver” ou “enviar” alguma coisa) e pode especificar o URI dos dados que serão utilizados (dentre outras coisas que o componente a ser iniciado pode precisar saber). Por exemplo, um intent pode transmitir um pedido para uma atividade mostrar uma imagem ou abrir uma página *web*. Em alguns casos, você pode iniciar uma atividade para receber um resultado, caso em que a atividade também retorna o resultado em um intent (por exemplo, emitir um intent para deixar o usuário escolher um contato pessoal e recebê-lo na aplicação — o intent de retorno inclui um URI apontando para o contato escolhido). Para os receptores *broadcast*, o intent simplesmente define o anúncio sendo transmitido (por exemplo, uma transmissão para indicar que a bateria do dispositivo está baixa inclui apenas uma string de ação conhecida que indica “bateria está baixa”).

O outro tipo de componente, o provedor de conteúdo, não é ativado por intents. Em vez disso, ele é ativado quando é alvo de um pedido através de um ContentResolver. O *content resolver* trata todas as operações diretamente com o provedor de conteúdo, para que o componente que está realizando essas operações não precise conhecer o provedor e, em vez disso, chame os métodos do objeto ContentResolver. Isso deixa uma camada de abstração entre o provedor de conteúdo e o componente requerendo as informações, por motivos de segurança.

Todos esses componentes e permissões de uma aplicação devem ser declarados em um arquivo de manifesto chamado “AndroidManifest.xml”, que deve estar no diretório raiz do projeto do aplicativo. Esse arquivo serve justamente para que o sistema Android saiba quais componentes existem no aplicativo e quais são as suas permissões de acesso, antes de executar a aplicação. O arquivo de manifesto faz uma série de coisas além de declarar os componentes do aplicativo, das quais podemos destacar:

- Identificar todas as permissões de usuário que o aplicativo requer, tais como acesso à *internet* ou acesso de leitura aos contatos do usuário.
- Declarar o nível mínimo de API exigido pelo aplicativo, com base em quais APIs o aplicativo usa.
- Declarar recursos de *hardware* e *software* usados ou exigidos pelo aplicativo, como uma câmera, serviços de *bluetooth* ou tela *multitouch*.

- Bibliotecas de API que a aplicação precisa usar (exceto as bibliotecas nativas da API Android), como, por exemplo, a biblioteca do Google Maps.

Atividades, serviços e provedores de conteúdo incluídos no código fonte do aplicativo, mas não declarados no arquivo de manifesto, não são visíveis para o sistema e, conseqüentemente, nunca poderão ser executados. No entanto, receptores *broadcast* podem ser declarados no arquivo manifesto ou criados dinamicamente no código (como objetos `BroadcastReceiver`) e registrados *on-the-fly* no sistema, executando uma função de registro própria do Android.

Conforme discutido anteriormente, pode-se usar intents para iniciar atividades, serviços e receptores *broadcast*. Uma das maneiras é explicitamente, nomeando o componente de destino (usando o nome da classe do componente) no intent. No entanto, o poder real dos intents reside no conceito de intent actions. Com intent actions, basta descrever o tipo de ação que se deseja realizar (e, opcionalmente, os dados sobre os quais a ação será realizada) e permitir que o sistema encontre um componente no dispositivo que possa executar essa ação e iniciá-la. Se existirem vários componentes que possam executar a ação descrita pelo intent, então o usuário poderá selecionar qual usar.

A forma como o sistema identifica os componentes que podem responder a um intent é comparando o intent recebido aos filtros de intent fornecidos no arquivo de manifesto de outros aplicativos no dispositivo. Quando um componente é declarado no arquivo de manifesto do aplicativo, pode-se, opcionalmente, incluir filtros de intent que declaram os recursos do componente para que ele possa responder aos intents de outras aplicações.

Por exemplo, um aplicativo de e-mail com uma atividade para compor um novo e-mail poderia declarar um filtro de intent no seu arquivo de manifesto para responder a intents do tipo “enviar” (para enviar um e-mail). Uma atividade em outra aplicação poderia, então, criar um intent com a ação “enviar” (`ACTION_SEND`), que no sistema corresponderia à atividade de “enviar” no aplicativo de e-mail. Portanto, o sistema poderia executar essa atividade quando a aplicação original invocasse o intent.

Outra utilidade do arquivo de manifesto é declarar os requisitos da aplicação. Há uma variedade de dispositivos que usam Android e nem todos oferecem os mesmos recursos e capacidades. A fim de impedir que o aplicativo seja instalado em dispositivos que não possuem recursos necessários para seu uso, é importante definir claramente um perfil para os tipos de dispositivos que o aplicativo suporta, declarando requisitos de *software* e *hardware* no arquivo de manifesto do aplicativo. A maioria dessas declarações são apenas informativas e o sistema não as lê. Mas os serviços externos, tais como o Google Play, as utilizam para

fornecer uma filtragem para os usuários, quando eles buscam as aplicações através do seu dispositivo.

Por exemplo, se uma aplicação requer a utilização de uma câmera e usa APIs introduzidas no Android 2.1 (API nível 7), ela deve declará-los como requisitos no seu arquivo de manifesto. Dessa forma, os dispositivos que não possuem uma câmera e tem uma versão do Android inferior a 2.1 não podem instalar o aplicativo através do Google Play. No entanto, pode-se também declarar que o aplicativo usa a câmera, mas não exige isso. Nesse caso, o aplicativo deve executar uma verificação em tempo de execução para determinar se o dispositivo possui uma câmera e desabilitar os recursos que usam a câmera, se ela não está disponível.

Existem ainda muitos outros recursos no Android SDK, mas não nos cabe aqui neste espaço discuti-los. A seguir veremos em maior detalhamento alguns desses componentes e recursos do Android SDK, que são necessários ao entendimento da construção e utilização de nosso protótipo.

5.5.1.1 Atividades

Uma atividade é um componente do aplicativo que fornece uma tela com a qual os usuários podem interagir a fim de fazer algo, como discar um número de telefone, tirar uma foto, enviar um e-mail ou visualizar um mapa. Uma aplicação consiste geralmente de múltiplas atividades que são fracamente acopladas umas às outras. Normalmente, uma atividade em um aplicativo é especificada como a atividade “principal”, que é apresentada ao usuário ao iniciar o aplicativo pela primeira vez. Cada atividade pode, em seguida, iniciar outra atividade a fim de executar diferentes ações. Cada vez que uma nova atividade começa, a atividade anterior é interrompida, mas o sistema preserva a atividade em uma pilha (a “pilha de retorno”). Quando uma nova atividade é iniciada, ela é empurrada na pilha de retorno e assume o foco do aplicativo. A pilha de retorno segue o mecanismo básico de pilha, “last in, first out”. Assim, quando o usuário termina a atividade atual e pressiona o botão “Voltar”, ela é removida da pilha (e destruída) e a atividade anterior é reiniciada. A Figura 22 mostra o funcionamento da pilha de retorno (*Back Stack*).

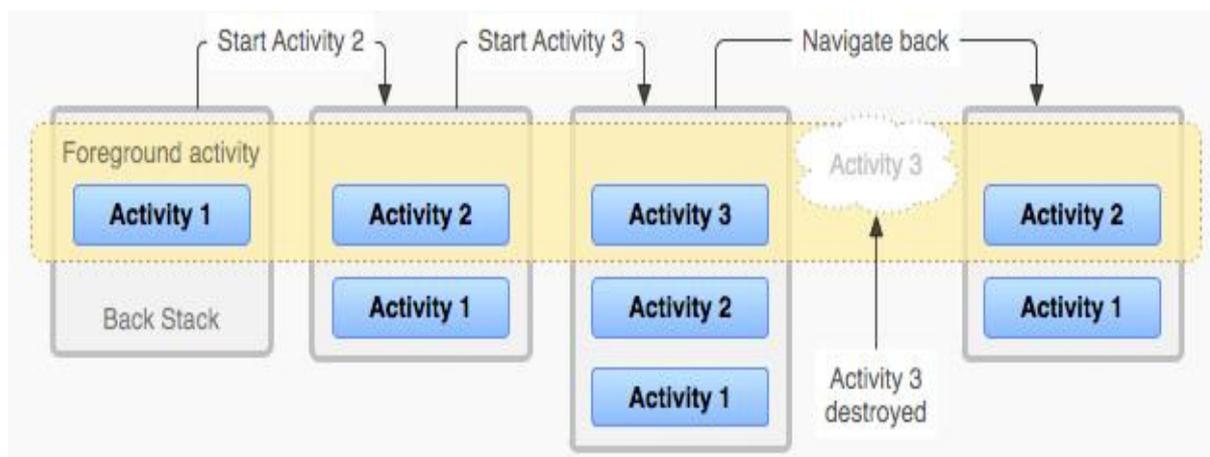


Figura 22. O funcionamento da pilha de retorno – *Back Stack* (ANDROID, 2012a)

Quando uma atividade é interrompida porque uma nova atividade é iniciada, ela é notificada dessa mudança de estado através de métodos de retorno (*callback methods*) do ciclo de vida da atividade. Existem vários métodos de retorno que uma atividade pode receber devido a uma mudança em seu estado — seja o sistema criando-a, interrompendo-a, reiniciando-a ou destruindo-a — e cada *callback* fornece uma oportunidade de realizar um trabalho específico que é apropriado à mudança de estado. Por exemplo, quando parada, uma atividade deve liberar todos os objetos grandes, como conexões de rede ou banco de dados. Quando a atividade recomeça, ela pode readquirir os recursos necessários e retomar as ações que foram interrompidas. Essas transições de estado são todas partes do ciclo de vida da atividade.

Para que uma atividade seja acessível através do sistema Android, ela deve ser declarada no arquivo de manifesto do aplicativo. Uma atividade também pode especificar vários filtros de intent no arquivo de manifesto, a fim de declarar como outros componentes de aplicativos do sistema podem ativá-la. A atividade “principal” do aplicativo deve incluir um filtro de intent que declara que aquela atividade responde à ação “principal” e deve ser colocada na categoria “lançador”, para poder ser inicializada pelo sistema. Na Figura 23 podemos ver um exemplo de trecho do arquivo de manifesto, onde a atividade “principal” de um aplicativo é declarada. O elemento `<action>` especifica que este é o ponto de entrada “principal” da aplicação. O elemento `<category>` especifica que esta atividade deve ser listada no lançador de aplicações do sistema (para permitir aos usuários iniciarem esta atividade).

```
<activity android:name=".ExampleActivity" android:icon="@drawable/app_icon">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

Figura 23. Trecho do arquivo de manifesto onde é declarada a atividade principal de uma aplicação (ANDROID, 2012a)

Se uma aplicação deve ser autocontida, não permitindo que outros aplicativos executem suas atividades, então ela não precisa de quaisquer outros filtros de intent, além da ação “principal” citada anteriormente. Apenas uma atividade da aplicação deve possuir a ação “principal” e a categoria “lançador”, como no exemplo da Figura 23. Atividades não disponíveis para outros aplicativos não devem ter filtros de intent e podem ser iniciadas usando intents explicitamente.

No entanto, se uma atividade puder responder aos intents implícitos que são entregues a partir de outras aplicações (e da própria aplicação), então ela deve definir filtros de intent adicionais. Para cada tipo de intent ao qual a atividade pode responder, ela deve incluir um *<intent-filter>* que inclui um elemento *<action>* e, opcionalmente, um elemento *<category>* e/ou um elemento *<data>*. Esses elementos especificam o tipo de intent que uma atividade pode responder.

Gerenciar o ciclo de vida das atividades através da implementação dos métodos de retorno (*callback methods*) é crucial para o desenvolvimento de uma aplicação forte e flexível. O ciclo de vida de uma atividade é diretamente afetado por sua associação com outras atividades, sua tarefa e a pilha de retorno. Uma atividade pode existir em três estados essencialmente:

- **Reiniciada:** a atividade está no primeiro plano da tela e tem o foco do aplicativo. Este estado é também referido geralmente como “executando”.
- **Pausada:** outra atividade está em primeiro plano e tem o foco, mas esta ainda é visível. Ou seja, outra atividade é visível sobre esta e aquela atividade é parcialmente transparente ou não cobre toda a tela. A atividade pausada está completamente viva (o objeto Activity é retido na memória, mantendo todas as informações de estado e seus membros, e continua anexado ao gerenciador de

janelas), mas pode ser morta pelo sistema em situações de memória extremamente baixa.

- **Parada:** a atividade é completamente obscurecida por outra atividade (a atividade está agora em segundo plano — “*background*”). A atividade parada também ainda está viva (o objeto Activity é retido na memória, mantendo todas as informações de estado e seus membros, mas não está anexado ao gerenciador de janelas). No entanto, já não é mais visível para o usuário e pode ser morta pelo sistema quando a memória é necessária em outro lugar.

Se uma atividade está pausada ou parada, o sistema pode retirá-la da memória de duas formas: pedindo para ela terminar (chamando seu método de finalização), ou simplesmente matando seu processo. Quando a atividade é aberta novamente (depois de ter sido terminada ou morta), ela deve ser criada por inteiro novamente.

Quando uma atividade realiza transições para dentro e para fora dos diferentes estados descritos anteriormente, ela é notificada através de vários métodos de retorno distintos. Todos os métodos de retorno (*callback methods*) podem ser sobrescritos para fazer o trabalho adequado quando o estado das atividades muda. A Figura 24 mostra um esqueleto de atividade com cada um dos métodos fundamentais do ciclo de vida.

```

public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // The activity is being created.
    }
    @Override
    protected void onStart() {
        super.onStart();
        // The activity is about to become visible.
    }
    @Override
    protected void onResume() {
        super.onResume();
        // The activity has become visible (it is now "resumed").
    }
    @Override
    protected void onPause() {
        super.onPause();
        // Another activity is taking focus (this activity is about to be "paused").
    }
    @Override
    protected void onStop() {
        super.onStop();
        // The activity is no longer visible (it is now "stopped")
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // The activity is about to be destroyed.
    }
}

```

Figura 24. Métodos de retorno fundamentais do ciclo de vida de uma atividade (ANDROID, 2012a)

Juntos, esses métodos definem o ciclo de vida de uma atividade. Ao implementar esses métodos, pode-se monitorar três *loops* aninhados no ciclo de vida de uma atividade:

- A vida inteira de uma atividade acontece entre a chamada do método **onCreate()** e a chamada ao método **onDestroy()**. Uma atividade deve executar a preparação do estado “global” (tais como a definição de *layout* de tela, por exemplo) em **onCreate()**, e liberar todos os recursos remanescentes em **onDestroy()**. Por exemplo, se uma atividade possui uma *thread* que executa

em segundo plano para baixar dados da rede, ela deve criar essa *thread* no **onCreate()** e depois pará-la em **onDestroy()**.

- O tempo de vida visível de uma atividade acontece entre a chamada de **onStart()** e a chamada de **onStop()**. Durante este tempo, o usuário pode ver a atividade na tela e interagir com ela. Por exemplo, **onStop()** é chamado quando uma nova atividade começa e esta não é mais visível. Entre esses dois métodos, pode-se manter os recursos que são necessários para mostrar a atividade para o usuário. Por exemplo, pode-se registrar um `BroadcastReceiver` em **onStart()** para monitorar as mudanças que impactam a interface do usuário, e cancelar o registro em **onStop()** quando o usuário não pode mais ver o que está sendo exibindo. O sistema pode chamar **onStart()** e **onStop()** várias vezes durante toda a vida útil da atividade, enquanto ela alterna entre ser visível e oculta ao usuário.
- A vida em primeiro plano de uma atividade acontece entre a chamada de **onResume()** e a chamada de **onPause()**. Durante este tempo, a atividade está na frente de todas as outras atividades na tela e tem foco de entrada de dados do usuário. Uma atividade pode frequentemente transitar para dentro e para fora do primeiro plano — por exemplo, **onPause()** é chamado quando o dispositivo vai entrar em estado de espera (*standby/sleep*) ou quando uma caixa de diálogo aparece. Uma vez que esse estado de transição pode ocorrer diversas vezes, o código nesses dois métodos deve ser bastante leve, a fim de evitar transições lentas que fazem o usuário esperar muito.

A Figura 25 ilustra esses *loops* e os caminhos que uma atividade pode tomar entre estados. Os retângulos representam os métodos de retorno que podem executar operações quando a atividade transita entre os estados.

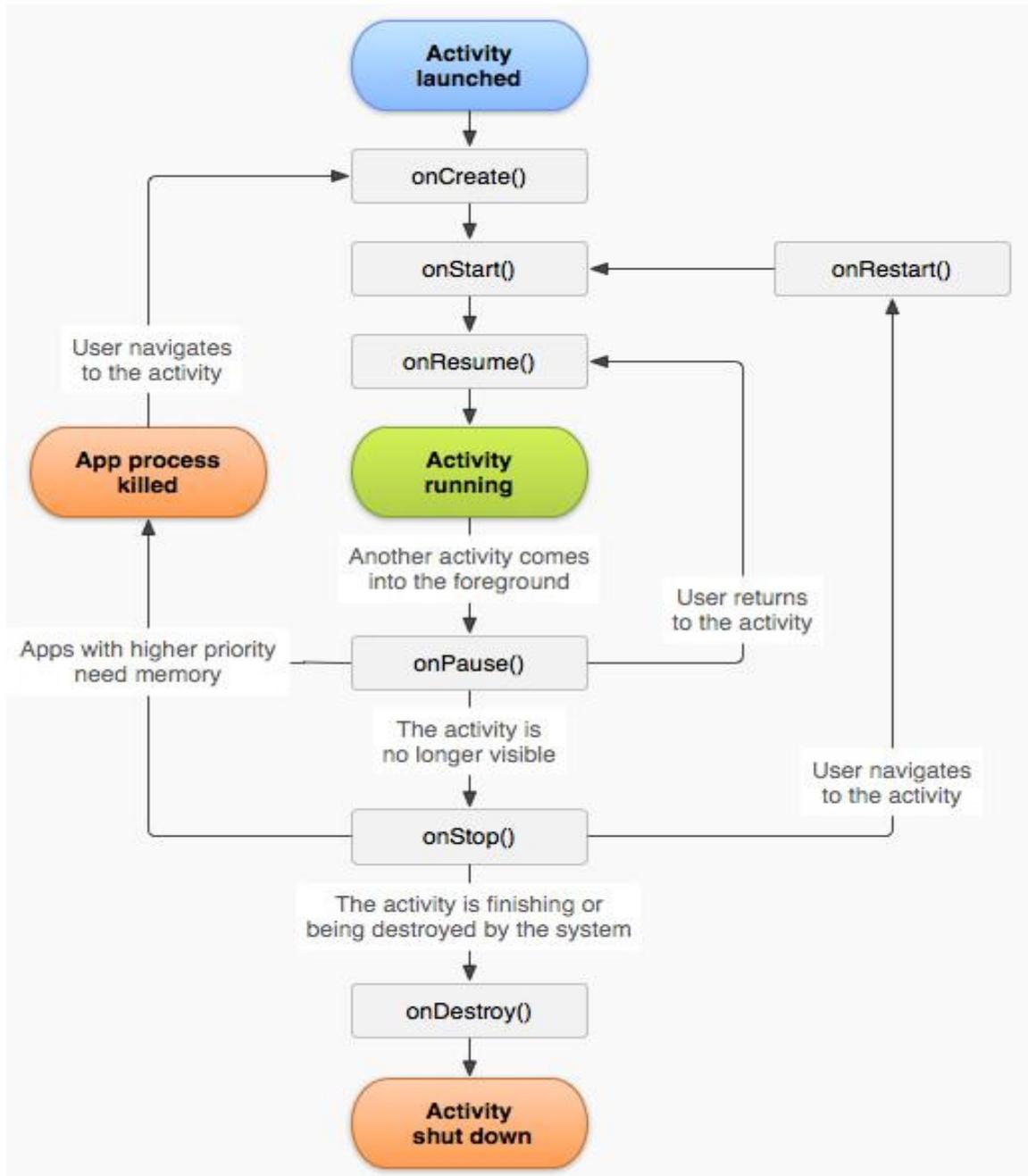


Figura 25. Ciclo de vida de uma atividade (ANDROID, 2012a)

Existe ainda muito mais a se falar sobre Atividades das aplicações no sistema Android. Porém, como este não é o foco do nosso trabalho, decidimos apenas cobrir o básico necessário para se entender como funciona o nosso protótipo. A seguir, discutiremos um pouco mais a fundo outro componente importante das aplicações Android.

5.5.1.2 Serviços

Um serviço é um componente de aplicação que pode executar operações de longa duração em segundo plano e não fornece uma interface de usuário. Outro componente de aplicativo pode iniciar um serviço e ele continuará a ser executado em segundo plano, mesmo

se o usuário mudar para outro aplicativo. Além disso, um componente pode se conectar a um serviço para interagir com ele e até mesmo realizar comunicação entre processos (IPC). Por exemplo, um serviço pode lidar com transações de rede, tocar música, executar operações de entrada e saída de arquivo, ou interagir com um provedor de conteúdo, tudo em segundo plano. Um serviço possui essencialmente duas formas:

- **Iniciado:** um serviço é “iniciado” quando um componente do aplicativo (como uma atividade, por exemplo) o inicializa chamando sua função de início. Uma vez iniciado, um serviço pode ser executado em segundo plano por tempo indeterminado, mesmo que o componente que o iniciou seja destruído. Normalmente, um serviço iniciado executa uma única operação e não retorna um resultado para quem o iniciou. Por exemplo, um serviço pode fazer o *download* ou *upload* de um arquivo através da rede. Quando a operação é completada, o serviço deve parar sozinho.
- **Conectado:** um serviço está “conectado” quando um componente da aplicação conecta-se a ele chamando uma função de vinculação. Um serviço conectado oferece uma interface cliente-servidor que permite aos componentes interagirem com o serviço, enviar requisições, obter resultados, e até mesmo fazê-los através de processos com a comunicação entre processos (IPC). Um serviço conectado funciona somente enquanto outro componente de aplicação está vinculado a ele. Vários componentes podem se ligar a um único serviço de uma só vez, mas quando todos eles se desvinculam, o serviço é destruído.

Essa separação de formas é apenas teórica, uma vez que serviços podem trabalhar dessas duas maneiras ao mesmo tempo: eles podem ser iniciados para executar indefinidamente e, ainda assim, permitir vinculação de outros componentes. Para isto, basta apenas implementar ambos os métodos de retorno, um que permite aos componentes iniciarem o serviço e outro que permite aos componentes conectarem-se ao serviço.

Independentemente de uma aplicação estar iniciada, conectada, ou ambas, qualquer componente de aplicação pode utilizar o serviço (mesmo sendo de outra aplicação distinta), da mesma maneira que qualquer componente pode usar uma atividade — iniciando-o através de um intent. No entanto, o serviço pode ser declarado como privado no arquivo de manifesto do aplicativo, bloqueando assim o acesso de outras aplicações.

Um serviço é uma subclasse de `Service` que precisa sobrescrever alguns métodos de retorno (*callback methods*) que lidam com aspectos chave do ciclo de vida do serviço e

fornecem um mecanismo para outros componentes se conectarem ao serviço, se for o caso. Os métodos de retorno mais importantes são:

- **onStartCommand():** o sistema chama este método quando um outro componente, como uma atividade, solicita que o serviço seja iniciado, executando o método **startService()**. Uma vez que este método é executado, o serviço é iniciado e pode ser executado em segundo plano por tempo indeterminado. Quando seu trabalho está terminado, o serviço deve ser parado, chamando **stopSelf()** ou **stopService()**. Se o serviço funcionar apenas através de vinculação (serviço conectado), não é necessário implementar esses últimos métodos.
- **onBind():** o sistema executa este método quando outro componente quer se conectar ao serviço (tal como para executar RPC), chamando **bindService()**. Na implementação deste método, deve ser fornecida uma interface que os clientes usam para se comunicar com o serviço, retornando um **IBinder**. Este método deve ser implementado sempre, mas se o serviço não permitir uma vinculação, então este método deve retornar *null*.
- **onCreate():** o sistema chama este método quando o serviço é criado pela primeira vez, para realizar os procedimentos iniciais de configuração — antes de executar os comandos **onStartCommand()** ou **onBind()**. Se o serviço já está rodando, este método não é chamado.
- **onDestroy():** o sistema chama este método quando o serviço não é mais utilizado e está sendo destruído. O serviço deve implementar este método para limpar quaisquer recursos tais como *threads*, *listeners* cadastrados, receptores, etc. Este é o último método de retorno que o serviço recebe.

Se um serviço é iniciado por outro componente através do método **startService()** — o que resulta em uma chamada para **onStartCommand()** —, então o serviço permanece em execução até terminar com **stopSelf()** ou outro componente pará-lo através do método **stopService()**. Já se um componente executa o método **bindService()** para criar um novo serviço — e **onStartCommand()** não é chamado —, então o serviço é executado apenas enquanto o componente está conectado a ele. Uma vez que o serviço seja desvinculado de todos os clientes, o sistema então o destrói.

O sistema Android somente irá parar um serviço forçosamente quando a memória do dispositivo está baixa e ele deve recuperar recursos do sistema para a atividade que tem o foco

do usuário. Se o serviço está vinculado a uma atividade que tem o foco do usuário, então é menos provável que ele seja morto, e se o serviço é declarado para ser executado em primeiro plano, então ele quase nunca será morto. Caso contrário, se o serviço foi iniciado e é de longa duração, então o sistema irá baixar a sua posição de prioridade na lista de tarefas em segundo plano ao longo do tempo e o serviço se tornará altamente suscetível à morte — se o serviço é iniciado, então ele deve ser projetado para elegantemente se recuperar de reinicializações causadas pelo sistema. Se o sistema mata um serviço, ele é reiniciado tão logo os recursos estejam disponíveis novamente.

Assim como as atividades (e quaisquer outros componentes), todos os serviços devem ser declarados no arquivo de manifesto do aplicativo. Para declarar um serviço, um elemento `<service>` deve ser adicionado como um filho do elemento `<application>`, como mostra a Figura 26.

```

<manifest ... >
  ...
  <application ... >
    <service android:name=".ExampleService" />
    ...
  </application>
</manifest>

```

Figura 26. Declaração de um serviço no arquivo de manifesto do aplicativo (ANDROID, 2012a)

Existem outros atributos que podem ser incluídos no elemento `<service>` para definir suas propriedades, tais como as permissões necessárias para iniciar o serviço e o processo no qual o serviço deve ser executado. Assim como uma atividade, um serviço pode definir filtros de intent para permitir que outros componentes invoquem o serviço usando intents implícitos. Ao declarar filtros de intent, componentes de qualquer aplicativo instalado no aparelho do usuário podem potencialmente iniciar o serviço, se o filtro de intent declarado pelo serviço coincide com o intent que o aplicativo passa no método `startService()`.

Para criar um serviço privado, e acessá-lo apenas localmente (ou seja, as outras aplicações não poderão usá-lo), então não é preciso (e não se deve) fornecer quaisquer filtros de intents. Sem qualquer filtro de intent, o serviço deve ser iniciado usando um intent explícito (nomeando a classe do serviço desejado). Além disso, pode-se garantir que um serviço seja privado para uma aplicação somente ao se incluir o atributo `android:exported` configurado para `“false”` no arquivo de manifesto do aplicativo. Essa forma se mostra eficaz mesmo se o serviço fornece filtros de intents.

O ciclo de vida de um serviço é muito mais simples do que o de uma atividade. No entanto, é importante prestar atenção à forma como o serviço é criado e destruído, porque um serviço pode ser executado em segundo plano, sem que o usuário note. O ciclo de vida de um serviço (desde quando ele é criado até quando ele é destruído) pode seguir dois caminhos diferentes:

- **Serviço Iniciado:** o serviço é criado quando outro componente chama **startService()**. O serviço é executado indefinidamente e deve finalizar chamando **stopSelf()**, quando toda a tarefa tiver completada. Outro componente também pode parar o serviço chamando **stopService()**. Quando o serviço é interrompido, o sistema o destrói.
- **Serviço Conectado:** o serviço é criado quando outro componente (um cliente) chama **bindService()**. O cliente se comunica com o serviço através de uma interface **IBinder**. O cliente pode fechar a conexão chamando **unbindService()**. Vários clientes podem se conectar ao mesmo serviço e quando todos eles se desvinculam, o sistema destrói o serviço automaticamente (não há a necessidade do serviço se finalizar).

Estes dois caminhos não são totalmente separados. Ou seja, um novo componente pode se conectar a um serviço que já foi iniciado anteriormente com **startService()**. Por exemplo, um serviço de música pode ser iniciado em segundo plano chamando **startService()** com um intent que identifica a música a ser tocada. Mais tarde, possivelmente quando o usuário desejar exercer algum controle sobre o tocador ou obter informações sobre a música atual, uma atividade pode se conectar ao serviço chamando **bindService()**. Em casos como esse, **stopService()** ou **stopSelf()** não interrompe o serviço até que todos os clientes estejam desconectados.

Um serviço também possui métodos de retorno do ciclo de vida que podem ser implementados para monitorar mudanças no estado do serviço e realizar determinados trabalhos em momentos apropriados. A Figura 27 exibe um esqueleto de serviço demonstrando cada um dos métodos do ciclo de vida.

```

public class ExampleService extends Service {
    int mStartMode;        // indicates how to behave if the service is killed
    IBinder mBinder;      // interface for clients that bind
    boolean mAllowRebind; // indicates whether onRebind should be used

    @Override
    public void onCreate() {
        // The service is being created
    }
    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        // The service is starting, due to a call to startService()
        return mStartMode;
    }
    @Override
    public IBinder onBind(Intent intent) {
        // A client is binding to the service with bindService()
        return mBinder;
    }
    @Override
    public boolean onUnbind(Intent intent) {
        // All clients have unbound with unbindService()
        return mAllowRebind;
    }
    @Override
    public void onRebind(Intent intent) {
        // A client is binding to the service with bindService(),
        // after onUnbind() has already been called
    }
    @Override
    public void onDestroy() {
        // The service is no longer used and is being destroyed
    }
}

```

Figura 27. Métodos de retorno do ciclo de vida de um serviço (ANDROID, 2012a)

Ao implementar esses métodos, pode-se monitorar dois *loops* aninhados do ciclo de vida de um serviço:

- A vida inteira de um serviço acontece entre o momento em que **onCreate()** é chamado e o momento em que **onDestroy()** retorna. Como em uma atividade, um serviço faz a sua configuração inicial em **onCreate()** e libera todos os recursos restantes em **onDestroy()**. Por exemplo, um serviço de reprodução de música poderia criar a *thread* onde a música será tocada em **onCreate()** e então parar essa *thread* em **onDestroy()**. Os métodos **onCreate()** e **onDestroy()** são chamados para todos os serviços, sejam eles criados por **startService()** ou **bindService()**.

- A vida ativa de um serviço começa com uma chamada para **onStartCommand()** ou **onBind()**. Para cada método é entregue o intent que foi passado na chamada a **startService()** ou **bindService()**, respectivamente. Se o serviço é iniciado, o tempo de vida ativa termina ao mesmo tempo em que a vida inteira — o serviço ainda está ativo mesmo após o retorno de **onStartCommand()**. Se o serviço é conectado, a vida ativa termina quando o método **onUnbind()** retorna.

A Figura 28 ilustra os métodos de retorno típicos para o ciclo de vida de um serviço. Embora a figura separe serviços que são criados por **startService()** daqueles criados por **bindService()**, qualquer serviço, não importando como ele é iniciado, pode potencialmente permitir que clientes se conectem a ele. Assim, um serviço que foi inicialmente criado com **onStartCommand()** — por um cliente executando **startService()** — ainda pode receber uma chamada para **onBind()** — quando um cliente executa **bindService()**.

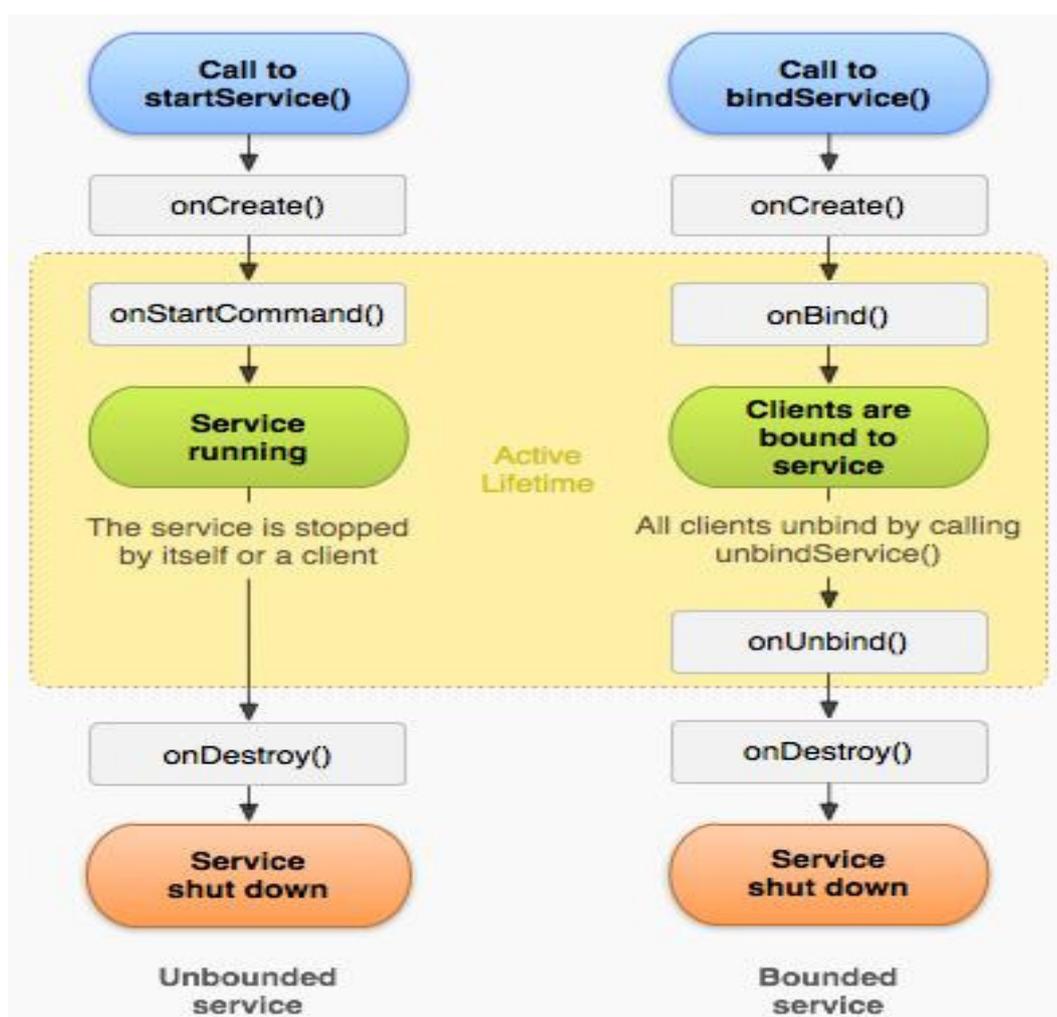


Figura 28. Ciclo de vida de um serviço (ANDROID, 2012a)

Ainda há muito assunto para ser discutido sobre os serviços de aplicativos do Android, como, por exemplo, criação e execução de serviços, definição de conexões e interações com serviços com o AIDL (*Android Interface Definition Language*), dentre outras coisas. Mas como este não é o foco do nosso trabalho, decidimos apenas explicar o básico sobre serviços, de modo que a compreensão do desenvolvimento de nosso protótipo fique mais clara para o leitor. A seguir, vamos falar sobre outro componente importante no desenvolvimento de aplicativos para Android, os provedores de conteúdos.

5.5.1.3 *Provedores de Conteúdo*

Os provedores de conteúdo gerenciam o acesso a um conjunto estruturado de dados. Eles encapsulam os dados e fornecem mecanismos para definir a segurança dos dados. Os provedores de conteúdo são a interface padrão que conecta os dados em um processo com o código sendo executado em outro processo.

Para acessar dados em um provedor de conteúdo, usa-se o objeto **ContentResolver** no contexto do aplicativo para se comunicar com o provedor, como se fosse um cliente. O objeto **ContentResolver** se comunica com o objeto provedor, uma instância de uma classe que implementa **ContentProvider**. O objeto provedor recebe requisições de dados de clientes, executa a ação solicitada, e retorna os resultados.

Se um aplicativo não pretende compartilhar seus dados com outras aplicações, então ele não precisa ter seu próprio provedor de conteúdo. No entanto, para fornecer sugestões de pesquisas personalizadas em um aplicativo próprio, então esse aplicativo precisa ter seu próprio provedor de conteúdo. Um provedor de conteúdo também é necessário caso se deseje transferir dados complexos ou arquivos entre aplicações.

O sistema Android em si inclui provedores de conteúdo próprios que gerenciam dados, tais como áudio, vídeo, imagens e informações de contato pessoal. Com algumas restrições, esses provedores de conteúdo são acessíveis a qualquer aplicativo Android.

Um provedor de conteúdo é a parte de um aplicativo Android que gerencia o acesso a um repositório central de dados, muitas vezes fornecendo sua própria interface do usuário para trabalhar com os dados. No entanto, os provedores de conteúdo são principalmente destinados a serem utilizados por outros aplicativos, que acessam o provedor usando um objeto cliente de provedor. Juntos, os provedores e clientes de provedor oferecem uma interface padrão de dados consistente que também lida com comunicação entre processos e acesso seguro aos dados.

Um provedor de conteúdo apresenta dados para aplicações externas como uma ou mais tabelas que são semelhantes às tabelas encontrados numa base de dados relacionais. A linha representa uma instância de algum tipo de dado que o provedor coleta, e cada coluna representa uma peça individual de dados coletados para uma instância.

Por exemplo, um dos provedores de conteúdo integrados na plataforma Android é o dicionário do usuário, que armazena as grafias de palavras fora do padrão que o usuário deseja manter. O Quadro 5 mostra como os dados podem parecer na tabela deste provedor.

Word	App Id	Frequency	Locale	_ID
mapreduce	user1	100	en_US	1
precompiler	user14	200	fr_FR	2
applet	user2	225	fr_CA	3
const	user1	255	pt_BR	4
Int	user5	100	en_UK	5

Quadro 5. Amostra da tabela do dicionário do usuário (ANDROID, 2012a)

No Quadro 5, cada linha representa uma instância de uma palavra que não pode ser encontrada em um dicionário padrão. Cada coluna representa alguns dados para essa palavra, tais como a língua local (coluna **Locale**) em que foi encontrada pela primeira vez. Os cabeçalhos de coluna são os nomes das colunas que estão armazenadas no provedor de conteúdo. Para se referir à língua local de uma linha, deve se referir à coluna **Locale**. Para este provedor de conteúdo, a coluna **_ID** serve como uma coluna de “chaves primárias” que o provedor mantém automaticamente.

Um aplicativo acessa os dados de um provedor de conteúdo com um objeto cliente `ContentResolver`. Este objeto tem métodos que chamam métodos com nomes idênticos no objeto provedor, uma instância de uma das subclasses concretas de `ContentProvider`. Os métodos de `ContentResolver` fornecem as funções básicas “CRUD” (*create, retrieve, update e delete* — criar, recuperar, atualizar e excluir, em português) de armazenamento persistente.

O objeto `ContentResolver` no processo do aplicativo cliente e o objeto `ContentProvider` na aplicação que possui o provedor de conteúdo automaticamente manipulam a comunicação entre os processos. O `ContentProvider` também atua como uma camada de abstração entre seu repositório de dados e a aparência externa dos dados como tabelas.

A aplicação de um provedor de conteúdo pode especificar permissões que outros aplicativos devem ter para acessar os dados do provedor. Essas permissões garantem que o usuário saiba exatamente quais dados um aplicativo tenta acessar. Com base nos requisitos do

provedor de conteúdo, outras aplicações solicitam as permissões necessárias para acessar o provedor. Os usuários finais veem as permissões solicitadas ao instalar o aplicativo.

Se a aplicação de um provedor de conteúdo não especificar as permissões, então os outros aplicativos não têm acesso aos dados do provedor. No entanto, os componentes na aplicação do provedor de conteúdo sempre têm pleno acesso de leitura e escrita, independentemente das permissões especificadas.

Para obter as permissões necessárias para acessar um provedor de conteúdo, um aplicativo deve solicitá-los com um elemento `<uses-permission>` em seu arquivo de manifesto. Quando o Gerenciador de Pacotes Android instala o aplicativo, o usuário deve aprovar todas as permissões que esse aplicativo requisita. Se o usuário aprovar todos eles, o Gerenciador de Pacotes continua a instalação; se o usuário não aprová-los, o Gerenciador de Pacotes aborta a instalação.

A Figura 29 exibe um trecho do arquivo de manifesto de uma aplicação, contendo um elemento `<uses-permission>` que solicita acesso de leitura ao provedor de conteúdo do Dicionário do Usuário, conforme descrito anteriormente.

```
<uses-permission android:name="android.permission.READ_USER_DICTIONARY">
```

Figura 29. Trecho do arquivo de manifesto de uma aplicação solicitando acesso de leitura ao provedor de conteúdo do dicionário do usuário (ANDROID, 2012a)

Tudo o acima exposto é somente o básico sobre os provedores de conteúdo, mas já é o necessário para que se entenda melhor o funcionamento de nosso protótipo. A partir de agora, iremos focar em um provedor de conteúdo nativo do Android e muito utilizado em nosso protótipo, pois é justamente a partir dele que conseguimos acessar os dados básicos para a análise contextual da rede social do usuário. Trata-se do provedor de conteúdo dos contatos do usuário, o **Contacts Provider**.

O **Contacts Provider** é um componente Android poderoso e flexível que gerencia o repositório central do dispositivo de dados sobre pessoas. O provedor de conteúdo dos contatos é a fonte de dados que pode ser vista na aplicação de contatos do dispositivo Android, e também se pode acessar seus dados via uma aplicação própria e transferir dados entre o dispositivo e serviços *online*. Esse provedor de conteúdo acomoda uma grande variedade de fontes de dados e tenta gerenciar tantos dados quanto possível para cada pessoa, portanto, sua organização é complexa. Devido a essa complexidade, a API desse provedor de conteúdo inclui um amplo conjunto de classes de contrato e interfaces que facilitam tanto a recuperação de dados quanto sua modificação.

O provedor de conteúdo de contatos mantém três tipos de dados sobre uma pessoa, cada um deles correspondendo a uma tabela fornecida pelo provedor, conforme ilustrado na Figura 30. As três tabelas são comumente referidas através de seus nomes das classes no contrato. As classes definem constantes para os URIs do conteúdo, os nomes das colunas, e os valores das colunas utilizadas pelas tabelas:

- **Tabela `ContactsContract.Contacts`:** linhas que representam as diferentes pessoas, com base em agregações de um “contato puro” (linhas da tabela `ContactsContract.RawContacts`).
- **Tabela `ContactsContract.RawContacts`:** linhas contendo um resumo dos dados de uma pessoa, específicos a uma conta de usuário e tipo.
- **Tabela `ContactsContract.Data`:** linhas contendo os detalhes de um “contato puro” (linhas da tabela `ContactsContract.RawContacts`), tais como os endereços de e-mail ou os números de telefone desse contato.

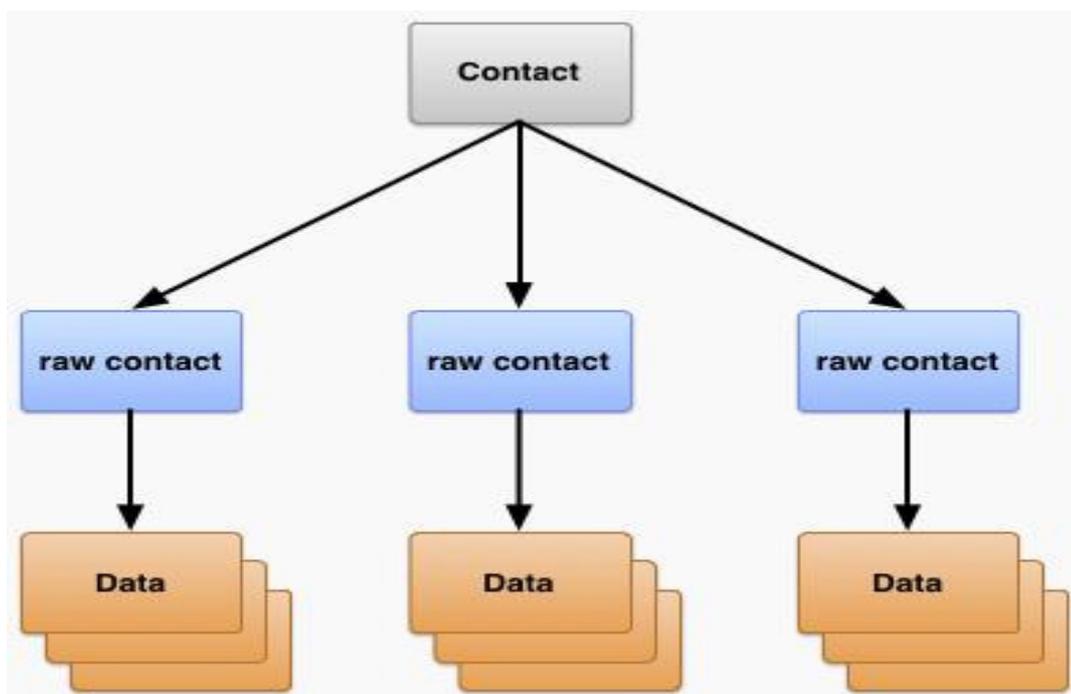


Figura 30. Estrutura de tabelas do provedor de conteúdo dos contatos (ANDROID, 2012a)

As outras tabelas representadas pelas demais classes de contrato em `ContactsContract` são tabelas auxiliares que o provedor de conteúdo dos contatos utiliza para gerenciar suas operações ou apoiar funções específicas nos contatos do dispositivo ou aplicações de telefonia.

Um “contato puro” representa os dados de uma pessoa proveniente de um único tipo de conta e nome de conta. Pelo simples motivo de o provedor de conteúdo dos contatos permitir mais de um serviço *online* como fonte de dados para uma pessoa, ele também

permite vários “contatos puros” para a mesma pessoa. Múltiplos “contatos puros” também permitem que um usuário combine dados de uma pessoa provenientes de mais de uma conta do mesmo tipo de conta.

A maior parte dos dados de um “contato puro” não é armazenado na tabela `ContactsContract.RawContacts`. Em vez disso, eles são armazenados em uma ou mais linhas na tabela `ContactsContract.Data`. Cada linha de dados tem uma coluna `Data.RAW_CONTACT_ID` que contém o valor da coluna `RawContacts._ID` de sua linha pai na tabela `ContactsContract.RawContacts`.

Para entender como os “contatos puros” funcionam, considere a usuária “Emily Dickinson”, que possui as três contas de usuário a seguir definidas em seu dispositivo: uma conta de e-mail no Gmail com usuário “emily.dickinson@gmail.com”, outra conta de e-mail no Gmail com usuário “emilyd@gmail.com”, e uma conta no Twitter com usuário “amherstbelle”. Considere também que esta usuária permitiu a sincronização de contatos para todas essas três contas na opção configurações de contas de seu dispositivo Android.

Suponha agora que a Emily Dickinson abre uma janela do navegador (no seu dispositivo móvel ou em um computador pessoal), faz o *login* no Gmail como “emily.dickinson@gmail.com”, abre seus contatos de e-mail e acrescenta seu amigo “Thomas Higginson”. Mais tarde, ela faz *login* novamente no Gmail, mas dessa vez como “emilyd@gmail.com” e envia um e-mail para “Thomas Higginson”, que automaticamente é adicionado como um contato pelo Gmail. Ela também segue o “colonel_tom” (Thomas Higginson ID do Twitter) no Twitter.

Como resultado dessas operações, o provedor de conteúdo dos contatos cria três “contatos puros”:

1. Um “contato puro” para “Thomas Higginson” associado ao nome de conta “emily.dickinson@gmail.com”. O tipo de conta de usuário é o Google.
2. Um segundo “contato puro” para “Thomas Higginson” associado ao nome de conta “emilyd@gmail.com”. O tipo de conta de usuário também é o Google. Há um segundo “contato puro” mesmo o nome sendo idêntico ao nome anterior, porque a pessoa foi adicionada através de uma conta de usuário diferente.
3. Um terceiro “contato puro” para “Thomas Higginson” associado ao nome de conta “amherstbelle”. O tipo de conta de usuário é o Twitter.

Conforme discutido anteriormente, os dados de um “contato puro” são armazenados em uma ou mais linhas da tabela `ContactsContract.Data`, que estão ligadas ao valor da coluna

_ID do “contato puro” (tabela `ContactsContract.RawContacts`). Isso permite que um único “contato puro” possua várias instâncias de um mesmo tipo de dados, como endereços de e-mail ou números de telefone. Por exemplo, se “Thomas Higginson” para o nome de conta “emily.dickinson@gmail.com” (a linha de “contato puro” para a Thomas Higginson associado à conta do Google “emily.dickinson@gmail.com”) tem um endereço de e-mail pessoal “thigg@gmail.com” e um endereço de e-mail de trabalho “thomas.higginson@gmail.com”, o provedor de conteúdo dos contatos armazena esses dois endereços de e-mail em linhas diferentes na tabela `ContactsContract.Data` e as referencia ao mesmo “contato puro” na tabela `ContactsContract.RawContacts`.

Diferentes tipos de dados são armazenados na tabela `ContactsContract.Data`. Linhas de detalhamento dos contatos como nome de exibição, número de telefone, e-mail, endereço postal, foto e *website* são encontradas nessa tabela. Para ajudar na gestão desses diferentes tipos de dados, a tabela `ContactsContract.Data` possui algumas colunas com nomes descritivos e outras com nomes genéricos. O conteúdo de uma coluna de nome descritivo tem o mesmo significado, independentemente do tipo de dado nas linhas, enquanto que os conteúdos de uma coluna de nome genérico possuem significados diferentes, dependendo dos tipos de dados armazenados nela.

O provedor de conteúdo dos contatos combina as linhas de “contatos puros” em todos os tipos de contas e nomes de contas para formar um contato. Isto facilita a exibição e alteração de todos os dados recolhidos para uma pessoa. O provedor de conteúdo dos contatos gerencia a criação de novas linhas de contato e a agregação de “contatos puros” em uma linha de contato já existente. Aplicações ou adaptadores de sincronização não são permitidos a adicionar contatos, e algumas colunas em uma linha de contato são somente para leitura.

O provedor de conteúdo de contatos cria um novo contato em resposta à adição de um novo “contato puro” que não coincide com contatos existentes. O provedor de conteúdo também faz isso se os dados em um “contato puro” existente alterarem de tal forma que já não mais coincidem com o contato ao qual ele foi anexado anteriormente. Se uma aplicação ou adaptador de sincronização criar um novo “contato puro” que corresponde a um contato existente, esse novo “contato puro” é então agregado ao contato existente.

O Contacts Provider liga uma linha de contato a suas linhas de “contatos puros” através da coluna `_ID` da linha de contato na tabela de contatos (`ContactsContract.Contacts`). A coluna `CONTACT_ID` da tabela de “contatos puros” (`ContactsContract.RawContacts`) contém os valores da coluna `_ID` para a linha da tabela de contatos associada a cada linha de “contatos puros”.

A tabela `ContactsContract.Contacts` também possui uma coluna chamada `LOOKUP_KEY` que é um *link* “permanente” para a linha de contato. Como o provedor de conteúdo dos contatos mantém todos os contatos automaticamente, ele pode alterar o valor da coluna `_ID` em uma linha de contato, como resposta a uma agregação ou sincronização (uma forma de atualização automática do contato na tabela). Mesmo se isso acontecer, o conteúdo do URI `CONTENT_LOOKUP_URI` combinado com o `LOOKUP_KEY` do contato ainda irá apontar para a linha de contato. Assim pode-se usar a coluna `LOOKUP_KEY` para manter *links* para contatos “favoritos” e outras utilidades mais. Esta coluna tem o seu próprio formato que não está relacionado ao formato da coluna `_ID`.

A Figura 31 mostra como essas três principais tabelas se relacionam.

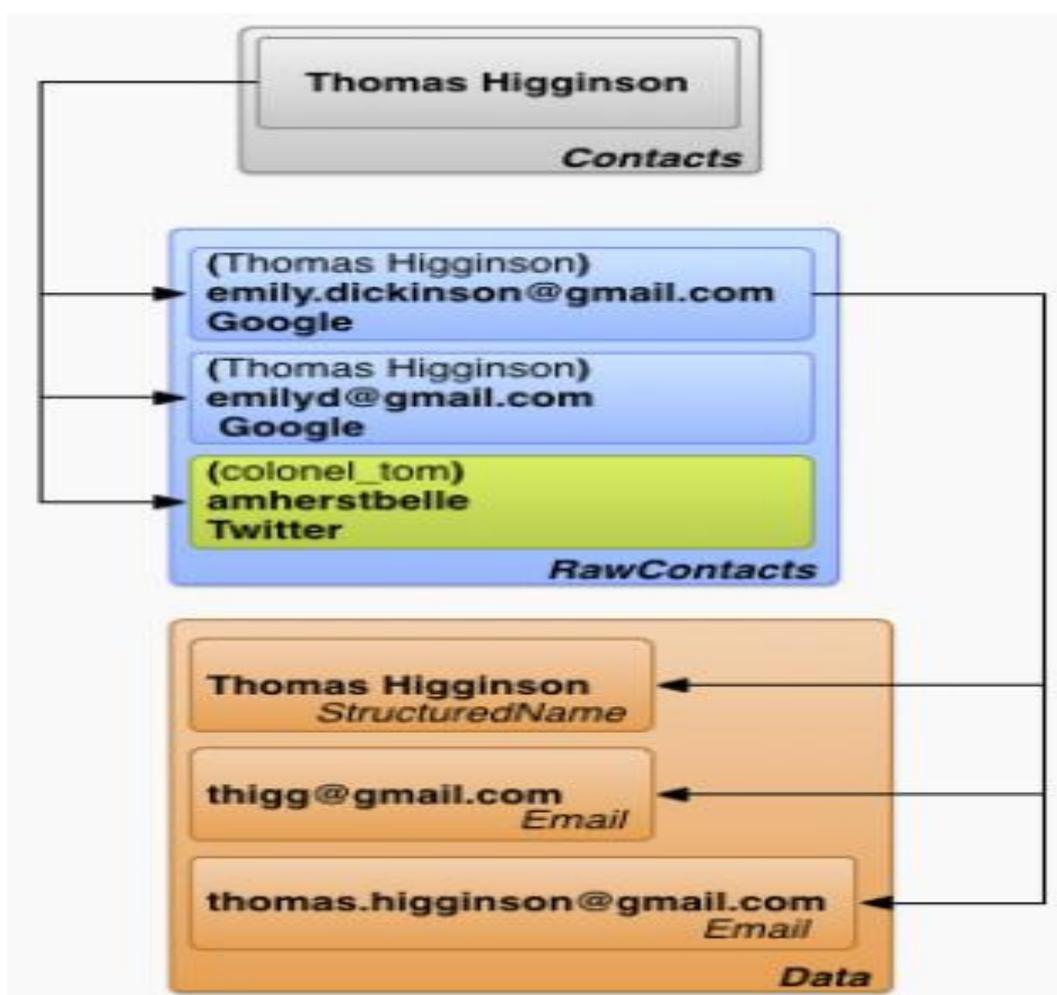


Figura 31. Relações entre as tabelas de contatos, “contatos puros” e detalhes (ANDROID, 2012a)

Os usuários fazem a entrada de dados dos contatos diretamente no dispositivo, mas os dados também fluem dos serviços *web* para o provedor de conteúdo dos contatos através de adaptadores de sincronização, que automatizam a transferência de dados entre o dispositivo e esses serviços. Adaptadores de sincronização são executados no plano de fundo sob o controle do sistema e executam os métodos do `ContentResolver` para gerenciar os dados.

No Android, o serviço *web* que um adaptador de sincronização trabalha é identificado por um tipo de conta. Cada adaptador de sincronização funciona com um tipo de conta somente, mas ele pode suportar múltiplos nomes de contas para esse tipo. A seguir oferecemos mais detalhes sobre tipo e nome de conta, e descrevemos como eles estão relacionados a adaptadores de sincronização e serviços:

- **Tipo de conta:** identifica um serviço no qual o usuário possua dados armazenados. Na maioria dos casos, o usuário tem que se autenticar com o serviço. Por exemplo, o Google Contacts é um tipo de conta, identificado pelo código “google.com”. Este valor corresponde ao tipo de conta utilizado pelo AccountManager.
- **Nome da conta:** identifica uma determinada conta ou o *login* para um tipo de conta. Contas do Google Contacts são as mesmas que as contas do Google, que possuem um endereço de e-mail como nome de conta. Outros serviços podem usar como nome de usuário uma única palavra ou um identificador numérico.

Os tipos de conta não devem ser obrigatoriamente únicos. Um usuário pode configurar várias contas do Google Contacts e fazer o *download* de seus dados para o provedor de conteúdo dos contatos, o que pode acontecer caso o usuário tenha um conjunto de contatos pessoais para um nome de conta pessoal, e outro conjunto especificamente para o trabalho. Nomes de contas geralmente são únicos. Juntos, eles identificam um fluxo de dados específico entre o provedor de conteúdo dos contatos e um serviço externo.

A Figura 32 mostra como o Contacts Provider se encaixa no fluxo de dados sobre as pessoas. Na caixa marcada com “*sync adapters*”, cada adaptador de sincronização é marcado com seu tipo de conta.

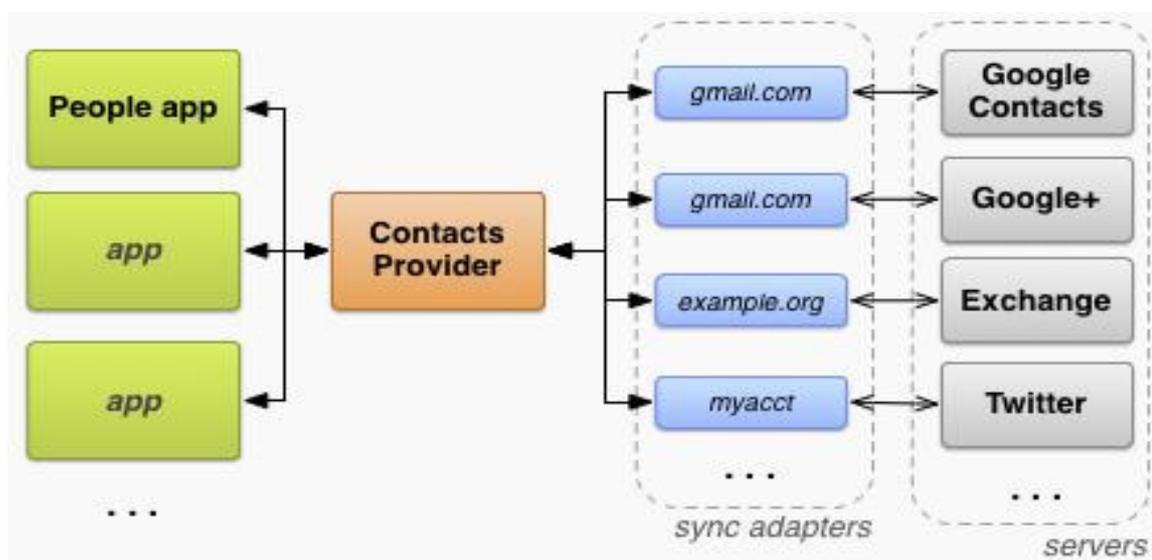


Figura 32. Fluxo de dados do provedor de conteúdo dos contatos (ANDROID, 2012a)

O provedor de conteúdo dos contatos é projetado especificamente para o tratamento de sincronização de dados dos contatos entre um dispositivo e um serviço *online*. Isso permite que os usuários façam *download* de dados existentes para um novo dispositivo e *upload* de dados existentes para uma nova conta. A sincronização também garante que os usuários tenham os dados mais recentes à mão, independentemente da fonte de adições e mudanças. Outra vantagem da sincronização é que torna os dados dos contatos sempre disponíveis, até mesmo quando o dispositivo não está conectado à rede.

Além disso, o **Contacts Provider** provê integração nativa com *streaming* de dados das redes sociais do usuário. Ele possui tabelas apropriadas para se gerenciar dados provenientes de redes sociais. Dessa maneira, pode-se usar um adaptador de sincronização que agregue o fluxo de dados a partir dessas redes sociais para essas tabelas próprias, ou pode-se ler o fluxo de dados a partir dessas tabelas e exibi-lo em um aplicativo próprio, ou ambos. Com esses recursos, os serviços de redes sociais e as aplicações podem ser facilmente integrados em uma experiência única de redes sociais no sistema Android.

Itens de *streaming* provenientes das redes sociais são sempre associados a um “contato puro”. A coluna RAW_CONTACT_ID referencia o valor da coluna _ID do “contato puro”. O tipo de conta e o nome da conta do “contato puro” também são armazenados na linha de um item de *streaming*. Existe também uma tabela para armazenar fotos associadas a um item de *streaming*. A coluna STREAM_ITEM_ID dessa tabela referencia os valores na coluna _ID da tabela de itens de *streaming*.

O fluxo de dados das redes sociais gerenciado pelo provedor de conteúdo dos contatos, em conjunto com a aplicação de contatos do dispositivo Android, oferece uma forma poderosa de se conectar os aplicativos de redes sociais com os contatos existentes. É possível sincronizar o serviço de rede social com o **Contacts Provider** através de um adaptador de sincronização, recuperando dessa forma a atividade recente dos contatos de um usuário e armazená-los nas tabelas de itens de *streaming* e fotos relacionadas a esses itens para uso posterior.

Além dessa sincronização regular realizada periodicamente, pode-se acionar o adaptador de sincronização para recuperar dados adicionais quando o usuário seleciona um contato para ver. Isso permite que o adaptador de sincronização recupere fotos de alta resolução e os itens de *streaming* mais recentes para aquele contato. Além disso, essa abordagem pode ser mais rápida e usar menos largura de banda da rede do que fazer uma sincronização completa com o adaptador de sincronização.

Diante de tudo o explanado nesta seção, verificamos que o sistema Android é a plataforma móvel correta para o desenvolvimento de nossa prova de conceito, por todas essas facilidades disponibilizadas pelo sistema e também pela integração nativa com serviços de redes sociais, além das APIs nativas para se conectar ao *hardware* dos dispositivos e utilizar os serviços de localização, como o sistema GPS do dispositivo móvel e o Google Maps.

5.5.2 Facebook SDK

Inicialmente, para criarmos uma prova de conceito para nosso trabalho, decidimos trabalhar somente com uma única rede social. A rede social escolhida para fornecer os dados de interação social entre os contatos e o usuário do dispositivo móvel foi o Facebook, por ser atualmente a rede social com mais usuários em todo o mundo, contabilizando um bilhão de usuários ativos mensalmente em outubro de 2012 (FACEBOOK NEWSROOM, 2012).

O Facebook SDK é extenso e possui muitas ferramentas que facilitam o uso e a integração das aplicações com o Facebook. Como nossa intenção não é analisar o Facebook SDK criticamente, nesta seção procuraremos nos ater somente às funcionalidades básicas para um bom entendimento sobre o funcionamento de nosso protótipo.

O Facebook SDK permite a criação de três tipos básicos de aplicativos: aplicações nativas do Facebook (ou seja, aplicações que executam diretamente dentro do Facebook e permitem a interação entre pessoas), aplicações *web* (ou seja, aplicações externas ao Facebook disponíveis como *sites* na internet que se comunicam com o Facebook de alguma maneira, como, por exemplo, através da conta do usuário ou de uma conta própria do aplicativo/site) e aplicações móveis (ou seja, aplicações externas ao Facebook disponíveis através de aplicativos de dispositivos móveis que se comunicam com o Facebook de alguma maneira, como, por exemplo, através da conta do usuário ou de uma conta própria do aplicativo).

Como o nossa proposta de solução utilizará um módulo *web* e um módulo móvel, iremos nos focar somente nos elementos do Facebook SDK que podem ser utilizados na construção de aplicações *web* e aplicações móveis. No caso do módulo *web*, existem duas possibilidades de SDK para implementação de suas funcionalidades: o JavaScript SDK e o PHP SDK. O JavaScript SDK permite o acesso a todos os recursos da API do Facebook através de JavaScript. Ele fornece um conjunto rico de funcionalidades do lado do cliente para autenticação e interação com o Facebook. Já o PHP SDK oferece suporte à plataforma do Facebook para aplicativos *web* baseados em PHP. Esta biblioteca ajuda a adicionar funcionalidades como *login* no Facebook e suporte à rede social do usuário no *website*.

Já para o módulo móvel, o Facebook oferece também dois tipos distintos de SDK: o iOS SDK (para iPhone e iPad) e o Android SDK. O iOS SDK fornece suporte de primeira classe para a plataforma Facebook em aplicativos do iPad, iPhone e iPod Touch escritos em Objective-C. Pode-se utilizar uma ferramenta de *single-sign-on*, executar chamadas à API do Facebook e exibir caixas de diálogo da plataforma Facebook. Este SDK é *open source* e está disponível no GitHub¹⁴. Já o Android SDK traz a plataforma Facebook para a plataforma Android (todos os dispositivos móveis). Pode-se usar este SDK para adicionar *single-sign-on* em aplicativos Android, invocar a API do Facebook e muito mais. Este SDK também é *open source* e está disponível no GitHub¹⁵.

Independentemente do tipo de aplicativo que será desenvolvido usando-se o Facebook SDK, ele possui seis conceitos básicos que são comuns a todos. Esses seis conceitos básicos do Facebook SDK são explicados a seguir (FACEBOOK DEVELOPERS, 2012b):

1. **Social Design:** o guia de Social Design ajuda a entender por que devem ser construídas grandes experiências sociais, bem como a forma de usar a plataforma do Facebook de maneira eficaz para criar essas experiências sociais. O Social Design é uma estratégia de produto que se baseia em comunidades de confiança dos usuários, favorece a conversa entre eles e, finalmente, cria um forte senso de identidade para todos.
2. **Social Plugins:** os *plugins* sociais permitem fornecer experiências sociais envolventes para os usuários com apenas uma linha de HTML. Como os *plugins* são servidos pelo Facebook, o conteúdo é personalizado para os usuários, quer eles tenham ou não feito o *login* em um *site*.
3. **Open Graph:** o protocolo Open Graph permite a integração de páginas na *internet* com o grafo social fornecido pelo Facebook. Essas páginas ganham a funcionalidade de outros objetos contidos no grafo, incluindo *links* do perfil e atualizações de *stream* para usuários conectados.
4. **Social Channels:** a plataforma do Facebook permite integração com canais sociais, como o News Feed e requisições no Facebook para ajudar a impulsionar o crescimento e envolvimento com o aplicativo, *site* ou conteúdo.
5. **Authentication:** a autenticação do Facebook permite que um aplicativo interaja com a Graph API em nome de usuários do Facebook e fornece um

¹⁴ O código fonte e demais informações sobre o SDK podem ser encontrados em <https://github.com/facebook/facebook-ios-sdk>.

¹⁵ O código fonte e demais informações sobre o SDK podem ser encontrados em <https://github.com/facebook/facebook-android-sdk>.

poderoso mecanismo de *login* único (*single-sign-on*) através de aplicativos *web*, *móveis* e *desktop*.

6. **Graph API:** A Graph API é o núcleo da plataforma Facebook, permitindo a leitura e gravação de dados do Facebook. Ele fornece uma visão simples e consistente do grafo social do usuário, uniformemente representando objetos (como pessoas, fotos, eventos e páginas) e as conexões entre eles (amizades, *likes* e *tags* de fotos).

O **Social Design** é uma forma de se pensar sobre o *design* do produto, colocando as experiências sociais dos usuários no centro das atenções. Essas experiências sociais podem ser facilmente criadas com os recursos disponíveis na plataforma Facebook. O Social Design define como entendemos a nós mesmos e uns aos outros. Ele pode ser dividido em três elementos principais, descritos a seguir:

- **Comunidade:** refere-se às pessoas que conhecemos e confiamos e que nos ajudam a tomar decisões.
- **Conversação:** refere-se às várias interações que temos com nossas comunidades.
- **Identidade:** refere-se ao nosso próprio senso de ser e como somos vistos pelas nossas comunidades.

A Figura 33 mostra as interações entre esses três elementos principais do Social Design:

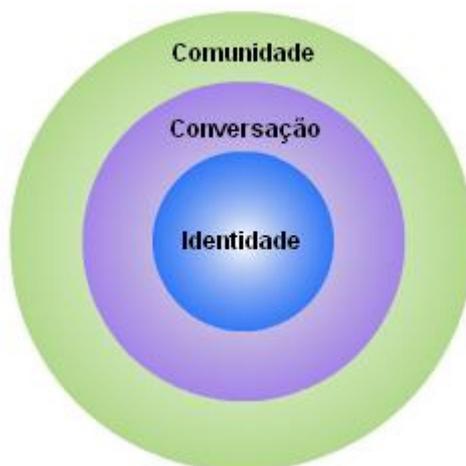


Figura 33. Interação entre os três elementos principais do Social Design

Os **Social Plugins** permitem que o usuário veja o que seus amigos tenham gostado, comentado ou compartilhado em *sites* por toda a *web*. Eles formam um conjunto de ferramentas que permitem uma maior interação do usuário com o Facebook, através de aplicativos ou *sites* na *internet*.

No núcleo do Facebook está o grafo social: as pessoas e as conexões que elas têm com tudo o que elas se importam. Historicamente, o Facebook conseguiu gerenciar este grafo e expandi-lo ao longo do tempo com o lançamento de novos produtos (ex: fotos, lugares). Em 2010 foi lançada uma versão inicial do **Open Graph**, uma extensão do grafo social através do protocolo Open Graph, para incluir *sites* de terceiros e as páginas que as pessoas gostavam por toda a *web*. Agora o Facebook está ampliando o **Open Graph** para incluir ações arbitrárias e objetos criados por aplicativos de terceiros e permitindo que esses aplicativos se integrem profundamente à experiência proporcionada pelo Facebook.

Depois que um usuário adiciona um aplicativo a sua linha do tempo no Facebook, as ações desses aplicativos específicos são compartilhadas no Facebook através do **Open Graph**. Como a aplicação torna-se uma parte importante de como os usuários se expressam, essas ações ficam visíveis em toda a linha do tempo, o *feed* de notícias e o *ticker* do usuário. Isso permite que um aplicativo se torne uma parte essencial da experiência do Facebook para o usuário e seus amigos. A Figura 34 ilustra o conceito do **Open Graph**.

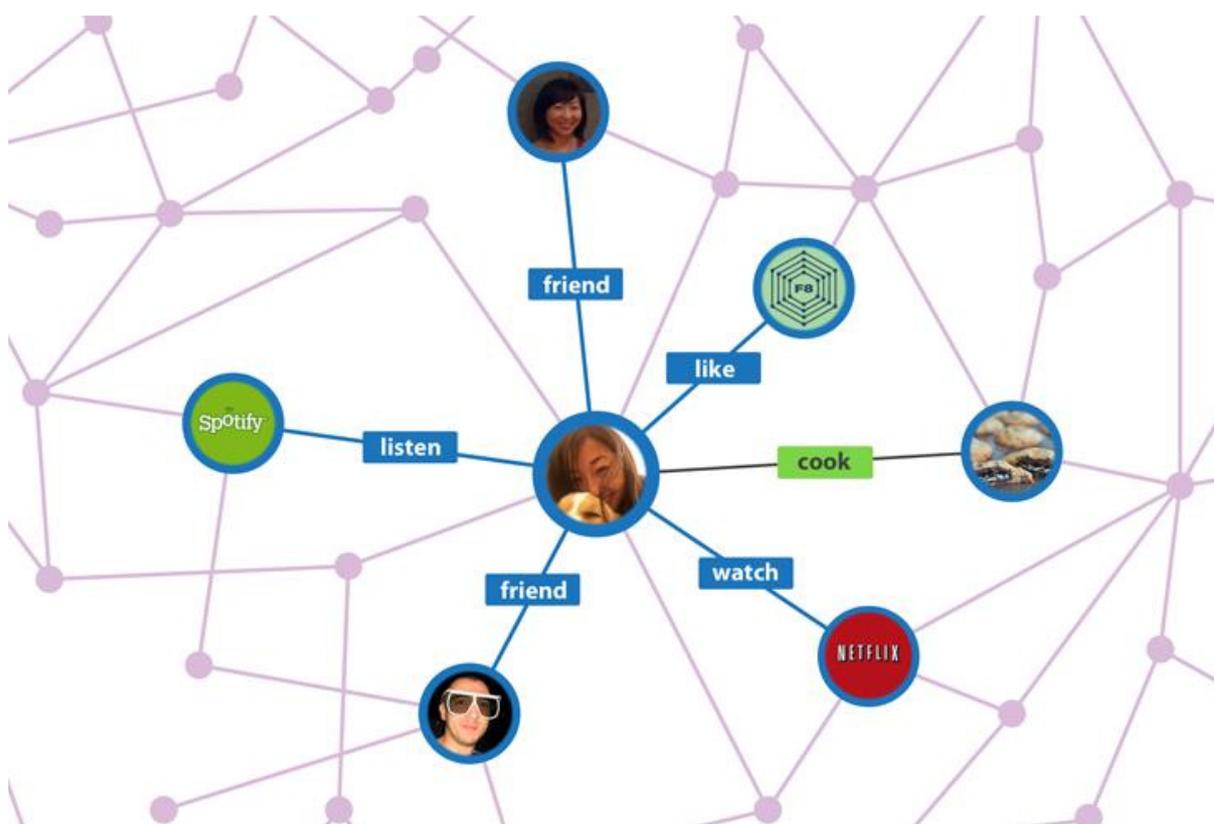


Figura 34. Conceito do Open Graph (FACEBOOK DEVELOPERS, 2012d)

Uma vantagem de usar a plataforma Facebook é o potencial alcance que se tem quando usuários do Facebook compartilham algum conteúdo de um aplicativo ou *site* com seus amigos. Por causa da força do endosso de um amigo, a comunicação através da plataforma do Facebook pode ajudar os produtos de alta qualidade a crescerem

tremendamente. É justamente nisso que se baseia o conceito de **Social Channels**, que são todos os canais disponíveis para comunicação entre os usuários do Facebook: *feed* de notícias, botão de Like, requisições de amizade, notificações, etc.

A autenticação (**Authentication**) dá a um aplicativo a capacidade de conhecer a identidade de um usuário do Facebook, e a permissão para ler e gravar dados através das APIs do Facebook. A plataforma Facebook usa OAuth 2.0 para autenticação e autorização. Um fluxo de autenticação bem sucedido resultada em um aplicativo obtendo um *token* de acesso do usuário que pode ser usado para fazer pedidos às APIs do Facebook e interagir com o Facebook em nome daquele usuário. Esse *token* de acesso tem um prazo de validade, e depois de expirado esse prazo, é necessário que a aplicação realize outro fluxo de autenticação.

Na autorização padrão de uma aplicação, o usuário só concede acesso a suas informações básicas. Para uma aplicação poder ler dados adicionais ou escrever dados no Facebook, é preciso solicitar permissões adicionais ao usuário.

Conforme mencionado anteriormente, o grafo social está no núcleo do Facebook e é a representação das pessoas e suas conexões com tudo o que lhes importa. O **Graph API** apresenta uma visão simples e consistente do grafo social do Facebook, uniformemente representando os objetos contidos nesse grafo (por exemplo, pessoas, fotos, eventos e páginas) e todas as conexões entre elas (por exemplo, as relações de amigos, o conteúdo compartilhado, *tags* de fotos, etc.).

Cada objeto no grafo social tem um identificador único e suas propriedades podem ser acessadas através de uma requisição ao **Graph API** na forma de uma requisição HTTPS do tipo <https://graph.facebook.com/ID>, onde ID é o identificador único do objeto, que pode ser um identificador numérico ou, alternativamente, um nome de usuário (no caso de pessoas e páginas que possuem nomes de usuário). Todas as respostas da **Graph API** são objetos JSON. Todos os objetos no Facebook podem ser acessados dessa mesma maneira: usuários, páginas, eventos, grupos, aplicações, mensagens de *status*, fotos, álbuns de fotos, fotos de perfil, vídeos, notas, *checkins*, etc.

Todos os objetos no grafo social do Facebook estão conectados uns aos outros através de relacionamentos. Esses relacionamentos são chamados de conexões na API do Facebook e podem ser examinados através de uma requisição ao **Graph API** com a seguinte estrutura: https://graph.facebook.com/ID/CONNECTION_TYPE. São suportados diferentes tipos de conexão para objetos diferentes. As conexões suportadas para pessoas e páginas incluem: amigos, *feed* de notícias (uma visão desatualizada, não refletindo o *feed* de notícias atual em facebook.com), *feed* do perfil (*Wall* ou mural do usuário), *likes*, filmes, música, livros, notas,

permissões, *tags* de foto, álbuns de fotos, *tags* de vídeo, *uploads* de vídeo, eventos, grupos, *checkins*, objetos com localização.

Por padrão, a maioria das propriedades de um objeto é retornada quando se faz uma consulta à **Graph API**. Porém, pode-se também escolher quais campos (ou conexões) serão retornados com o parâmetro de consulta "*fields*". Ao consultar as conexões de um usuário, há vários parâmetros úteis que permitem filtrar e paginar esses dados de conexão. Para realizar testes e entender melhor a **Graph API**, o Facebook disponibiliza uma ferramenta chamada Graph API Explorer¹⁶, que é a maneira mais fácil de começar a utilizar e aprender como funciona a **Graph API**.

A versão padrão da **Graph API** foi concebida para realmente facilitar a obtenção de dados de um determinado objeto e para explorar as conexões entre os objetos. Ela também inclui uma capacidade limitada de recuperar dados para alguns objetos ao mesmo tempo. Se um aplicativo precisa acessar grandes quantidades de dados em uma única vez, ou se é necessário fazer alterações em vários objetos de uma só vez, é muito mais eficiente realizar consultas em lote, em vez de fazer várias solicitações HTTP individuais.

Para permitir essas consultas em lote, a Graph API suporta *batching*. *Batching* (processamento em lotes) permite que sejam passadas instruções para várias operações em uma única requisição HTTP. Também se podem especificar dependências entre operações afins. O Facebook API irá processar cada uma dessas operações independentes em paralelo e irá processar as operações dependentes sequencialmente. Uma vez que todas as operações estejam concluídas, uma resposta consolidada será passada de volta e a conexão HTTP será fechada. Atualmente, o Facebook API limita essa operação em lotes a 50 consultas.

O Facebook SDK também oferece o Facebook Query Language (FQL), uma interface estilo SQL para consultar os dados expostos pela Graph API. O objeto FQL permite a execução de consultas FQL usando a Graph API. Ele fornece algumas funcionalidades avançadas que não estão disponíveis na Graph API padrão, incluindo várias consultas em lote numa única chamada.

As consultas em FQL são da forma *SELECT [fields] FROM [table] WHERE [conditions]*. Ao contrário de SQL, a cláusula *FROM* do FQL pode conter apenas uma única tabela. Pode-se usar a palavra-chave *IN* nas cláusulas *SELECT* ou *WHERE* para fazer *subqueries*, mas as elas não podem referenciar variáveis no escopo da consulta externa. Uma consulta também deve ser indexável, significando que pelo menos uma das propriedades da

¹⁶ Essa ferramenta pode ser acessada em <http://developers.facebook.com/tools/explorer>.

consulta dentro da cláusula *WHERE* deve ser uma coluna indexada na tabela que se deseja consultar. O FQL pode lidar com matemática simples, operadores booleanos básicos, operadores lógicos *AND*, *OR* ou *NOT*, e cláusulas *ORDER BY* e *LIMIT*. Para qualquer consulta que possua um identificador de usuário, pode-se passar a função *me()*, que retorna o usuário logado. Outras funções também disponíveis são *now()*, *strlen()*, *substr()* e *strpos()*.

O FQL também suporta *multi-query*, um processamento em lotes que avalia uma série de consultas FQL em uma única chamada e retorna todos os dados compilados ao mesmo tempo. Este método utiliza um dicionário codificado em JSON chamado “*queries*”, onde cada consulta individual usa a mesma sintaxe exata que uma consulta simples. No entanto, este método permite realizar buscas mais complexas. Pode-se buscar dados em uma consulta e usá-los em outra consulta da mesma chamada. A cláusula *WHERE* é opcional na última consulta, pois os dados que ela referencia já foram obtidos na consulta anterior. Para se fazer referência aos resultados de uma consulta em outra consulta dentro da mesma chamada, basta especificar seu nome na cláusula *FROM* precedido por um #.

Por exemplo, imagine que desejamos obter alguns dados sobre um usuário que irá participar de um evento. Normalmente, teríamos que realizar duas consultas em seguida, aguardando os resultados da primeira consulta antes de executar a segunda consulta, já que a segunda consulta depende dos dados da primeira. Mas com o método **fql.multiquery**, podemos executá-las ao mesmo tempo, obtendo assim todos os resultados que precisamos, e ainda por cima com um melhor desempenho do que executando uma série de chamadas ao método **fql.query**. Primeiro, precisamos obter o ID do usuário e o estado de RSVP de cada participante, formulando a primeira consulta — *query1* — como mostra a Figura 35a.

Então, para obtermos os dados do perfil de cada participante (nome, URL e foto), podemos fazer uma segunda consulta — *query2* — que faz uma referência aos resultados de *query1*. Pode-se formular a *query2* como mostra a Figura 35b. No final, estamos fazendo duas consultas em uma única chamada à API do Facebook, como mostra a Figura 35.

```
"query1":"SELECT uid, rsvp_status FROM event_member WHERE eid=12345678" a
"query2":"SELECT name, url, pic FROM profile WHERE id IN (SELECT uid FROM #query1)" b
```

Figura 35. Multi-query em FQL (FACEBOOK DEVELOPERS, 2012c)

Isso conclui nossa descrição básica do Facebook SDK para o entendimento da construção do protótipo e seu funcionamento. Obviamente, ainda há muito assunto para se falar sobre o Facebook SDK, mas como este não é o foco do nosso trabalho, decidimos nos ater a uma explicação simplificada dele. Na seção a seguir discutiremos como o Android SDK

e o Facebook SDK foram utilizados em conjunto na construção de uma prova de conceito da nossa proposta de solução. Também explicaremos as maiores dificuldades encontradas ao se criar e testar o protótipo.

5.5.3 Protótipo

Conforme dito anteriormente, muitas das ferramentas oferecidas pelo Android SDK e Facebook SDK foram utilizadas na construção de uma prova de conceito de nossa proposta de solução. O Android SDK, especialmente, mostrou-se muito útil na construção deste protótipo, principalmente por fornecer as facilidades de integração entre aplicativos e serviços do sistema, e também na comunicação com o *hardware* do dispositivo. Todas essas facilidades de integração do sistema fornecida pelo Android SDK já foram explicadas anteriormente, e além de terem sido úteis na construção do protótipo, podem muito bem serem exploradas mais profundamente para a construção de um sistema mais sólido proposto como solução para o problema estudado neste trabalho.

Já o Facebook SDK nos proporcionou uma integração com os dados da rede social dos usuários no Facebook. Através dele, pudemos coletar dados importantes para a análise contextual da rede social dos usuários, e desta maneira inferir as pessoas socialmente mais próximas a um usuário. Apesar do Facebook SDK ser de fácil utilização, sua API ainda é um pouco limitada no que tange às consultas feitas à rede social do usuário. Conforme mencionado anteriormente, o fato das consultas FQL terem que ser indexáveis não nos proporcionou a coleta de certos dados que poderiam melhorar a indicação de pessoas socialmente mais próximas a um usuário, e ainda nos dificultou a coleta de outros dados que poderiam ser facilmente obtidos caso a ferramenta FQL permitisse consultas não indexáveis.

Nesta seção explicaremos como nosso protótipo foi construindo utilizando-se essas ferramentas citadas anteriormente, e também discutiremos os maiores problemas encontrados durante a construção de nosso protótipo. Primeiramente, devemos explicar que nosso protótipo foi construído inteiramente sobre o cliente móvel da arquitetura de nossa solução proposta, explicada na seção 5.4. Os motivos que nos levaram a tomar essa decisão foram basicamente dois. O primeiro deles foi a facilidade de integração entre aplicativos, serviços e *hardware* fornecida pelo Android SDK, conforme já explicado anteriormente, além da simplicidade na coleta e a capacidade de armazenamento de dados oferecidos pelo Android SDK e Facebook SDK. O segundo motivo foi o prazo para a apresentação deste trabalho.

Dessa forma, nosso protótipo recebe como entrada um nome de uma pessoa, pesquisa no dispositivo móvel do usuário seus contatos que possuem esse nome pesquisado, coletando

os dados de interação entre eles via dispositivo móvel (através do Android SDK) e os dados de interação via redes sociais (através do Facebook SDK), além de outras informações pertinentes a cada um desses contatos. Depois disso, com todas essas informações em mãos, o protótipo realiza a análise contextual da rede social do usuário e monta um *ranking*, classificando esses contatos encontrados de acordo com seus índices de proximidade social com o usuário, em ordem decrescente — do contato com maior índice de proximidade social com o usuário para o contato com menor índice de proximidade social com o usuário.

Para isso, nosso protótipo possui basicamente três módulos: o **Social Emergency Analytics**, o **Mobile Device Data Colector** e o **Facebook Data Colector**. Cada um deles possuem seus próprios propósitos: o **Social Emergency Analytics** é o responsável por realizar a análise contextual na rede social do usuário, montando assim o *ranking* com os contatos socialmente mais próximos ao usuário; o **Mobile Device Data Colector** é o responsável por compilar os dados e informações dos contatos do dispositivo móvel do usuário; e o **Facebook Data Colector** é o responsável por compilar os dados e informações dos contatos provenientes do Facebook.

O **Social Emergency Analytics** recebe como entrada o nome da pessoa e o repassa aos outros dois módulos, para que eles façam o seu trabalho. Com o resultado da execução dos outros dois módulos, o **Social Emergency Analytics** então deve realizar a análise contextual da rede social do usuário, montando o *ranking* de proximidade social entre os contatos e o usuário e exibindo uma listagem dos contatos que possuem o nome pesquisado, ordenados em ordem decrescente de proximidade social com o usuário.

O **Mobile Device Data Colector** recebe como entrada o nome de pessoa a ser pesquisado. A partir desse nome, ele pesquisa então nos contatos do dispositivo móvel do usuário por pessoas que realizem o *match* com essa busca. A partir dessa lista de contatos, o **Mobile Device Data Colector** passa então a coletar todas as informações no dispositivo móvel sobre essas pessoas: número total de ligações (realizadas e recebidas), data e hora da última ligação, duração total das ligações (realizadas e recebidas) e total de mensagens trocadas (SMS e MMS), além do nome completo do contato e a foto de perfil (caso haja alguma disponível). Além disso, ele também verifica para cada contato se o mesmo possui perfil no Facebook. Dessa forma, o **Mobile Device Data Colector** retorna uma listagem de contatos com todas essas informações compiladas e prontas para análise.

O **Facebook Data Colector** recebe como entrada essa listagem de contatos com os dados compilados do dispositivo móvel e, para cada contato que possui um perfil no Facebook, ele realiza a coleta de dados e informações sobre esses contatos. Os dados

coletados de cada contato no Facebook são aqueles referentes à interação social do usuário com esses contatos: citações em *posts*, compartilhamento de *posts*, mensagens no mural, comentários e *likes*. Outros dados, como *tags* de fotos ou vídeos, também foram coletados, mas não obtiveram um tratamento específico na compilação de dados para posterior análise contextual na rede social do usuário (ou seja, eles foram compilados dentro de uma dessas categorias anteriormente citadas, conforme cada caso).

Esses três módulos básicos do protótipo com suas atividades próprias e as interações entre eles podem ser vistos na Figura 36.

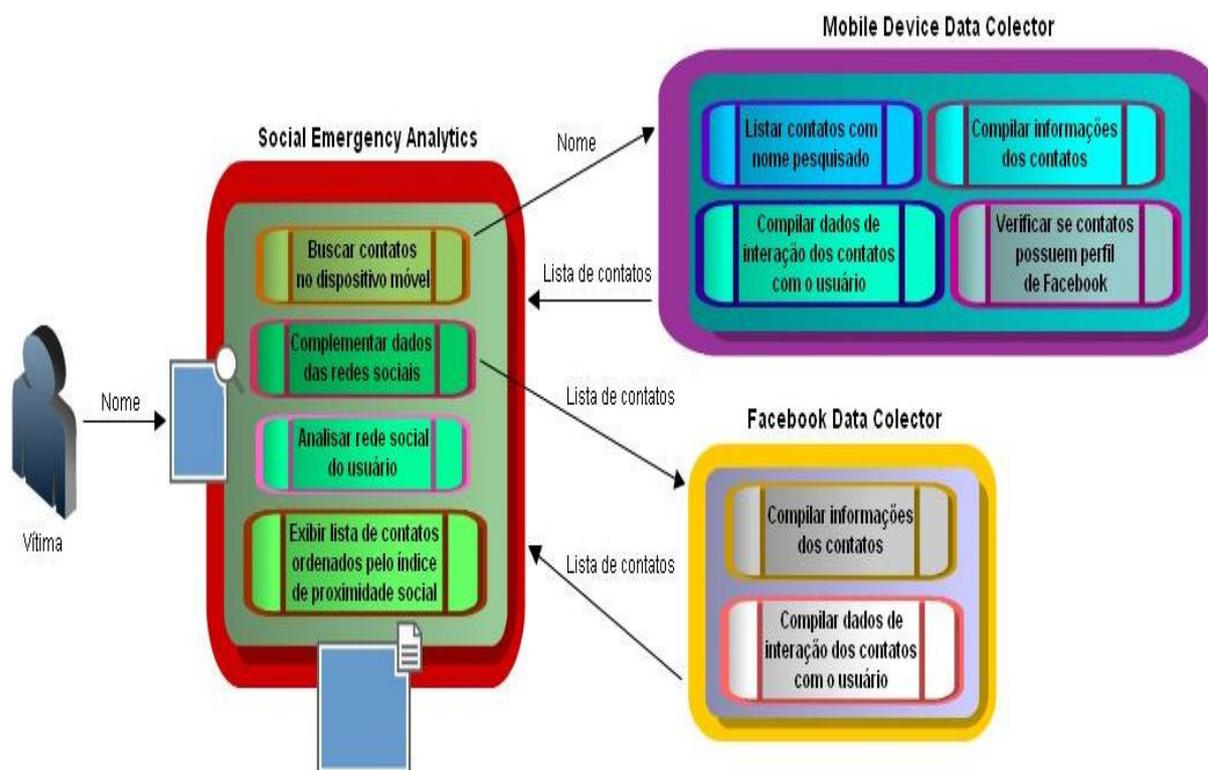


Figura 36. Módulos do protótipo e as interações entre eles

Nas seções a seguir explicaremos como foi construído cada módulo e citaremos quais as maiores dificuldades encontradas durante a implementação deles.

5.5.3.1 Social Emergency Analytics

Este módulo é construído através de uma Atividade do Android SDK. Sua função é unicamente receber como entrada o nome de uma pessoa e retornar uma lista de contatos do usuário dono do dispositivo móvel, que possuem o mesmo nome recebido como parâmetro de entrada. Ele serve como uma porta de entrada/saída do cliente móvel da arquitetura de nossa proposta de solução, fazendo a ligação entre os módulos de compilação de dados do dispositivo móvel e das mídias sociais. Dessa forma, sua principal atividade é a análise

contextual da rede social do usuário através dos dados compilados retornados pelos dois outros módulos do protótipo.

Com o resultado dessa análise em mãos, este módulo então organiza a lista de contatos que possuem o nome pesquisado criando um *ranking* baseado no índice de proximidade social de cada contato com o usuário. Este *ranking* é ordenado do contato mais relevante para o usuário (maior índice de proximidade social) até o contato menos relevante para o usuário (menor índice de proximidade social). As atividades deste módulo e suas interações são exibidas na Figura 37.

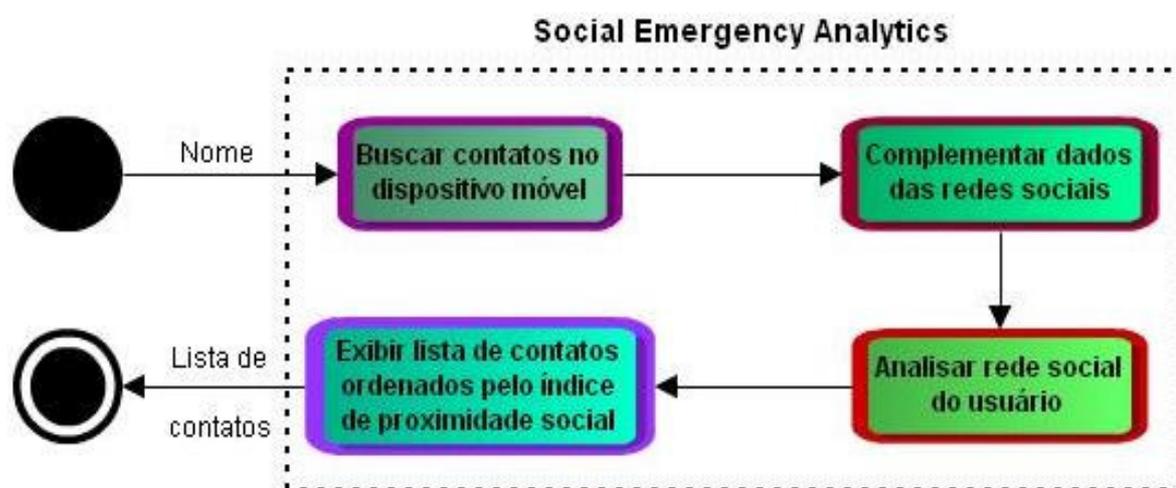


Figura 37. O módulo Social Emergency Analytics e suas atividades

A análise contextual da rede social do usuário é feita com base nas interações sociais do usuário com cada um desses contatos que possuem o mesmo nome pesquisado. É realizada uma média aritmética ponderada com cada uma dessas interações sociais coletadas através do dispositivo móvel e das mídias sociais, no caso do protótipo, o Facebook. Para tanto, cada interação social recebe um peso de acordo com sua relevância e pessoalidade entre as distintas formas de comunicação existentes analisadas: *likes* e citações em *posts* (peso 1), comentários em *posts* e compartilhamentos de conteúdo (peso 2), mensagens diretas (peso 3), SMS/MMS (peso 4), ligações (peso 5).

Além dessas interações acima citadas, analisamos também a duração total das ligações (recebidas e realizadas) em minutos, por considerarmos que a duração da conversação seja um forte indicador de intimidade entre os envolvidos. Dessa forma, para cada contato, é criado um índice de relevância social para o usuário. Com esses índices individuais de relevância social para o usuário, criamos o *ranking* da lista de contatos que possuem o nome pesquisado, de forma que cada contato seja avaliado em relação ao todo, ou seja, o fator de proximidade social de cada contato com o usuário é calculado em relação à totalidade de interações que o usuário teve com todos os contatos da lista.

Dessa forma, cada contato da lista é comparado com todos os outros contatos e o *ranking* é formado de acordo com a “porcentagem de relevância social” que aquele contato tem para o usuário, quando comparado com os outros contatos da lista. Com este *ranking* finalizado, ordenado de forma decrescente de relevância social, este módulo exibe então as informações de cada contato na lista: nome, foto, sexo, idade e data e hora do último contato realizado com o usuário.

Essas informações auxiliares sobre os contatos poderão ser úteis para que a equipe de resposta a emergências identifique se algum daqueles contatos do usuário é realmente a vítima, e não um homônimo. Este módulo não contou com nenhuma adversidade ou dificuldade na construção e sua maior contribuição para o protótipo é justamente a implementação o algoritmo de análise contextual da rede social do usuário e identificação dos contatos do usuário que correspondem ao nome pesquisado.

5.5.3.2 *Mobile Device Data Colector*

Este módulo também é construído através de uma Atividade do Android SDK, porém conta com o auxílio do Provedor de Conteúdo dos Contatos (*Contacts Provider*) para pesquisar os contatos do dispositivo móvel do usuário. Sua função é basicamente compilar as informações dos contatos contidos no dispositivo móvel do usuário que correspondem ao nome pesquisado e os dados de interação social entre esses contatos e o usuário no dispositivo móvel (SMS/MMS e ligações). Para isso, ele recebe como entrada o nome de uma pessoa e retorna como saída uma lista de contatos cujos nomes correspondem ao nome pesquisado, contendo as informações de cada contato e os dados de interação social entre eles e o usuário.

A partir do nome de uma pessoa a ser pesquisado, este módulo utiliza o *Contacts Provider* para recuperar a lista de contatos que correspondem ao nome pesquisado e suas informações básicas. Com essas informações básicas dos contatos em mãos, o módulo então utiliza a estrutura do *Contacts Provider* para recuperar e compilar os dados de interação social de cada contato com o usuário, como o total de mensagens SMS/MMS trocadas, número de ligações realizadas e recebidas, duração total dessas ligações e data e hora da última ligação (realizada ou recebida).

Como a estrutura do *Contacts Provider* (explicada na seção 5.5.1.3) permite o acesso a vários tipos de dados de um contato, agregados das mais diversas fontes, então é possível compilar os dados de interação do contato com o usuário através de todos os telefones cadastrados para aquele contato. Dessa forma, conseguimos obter a maior precisão possível

em relação à relevância social dos contatos para o usuário, considerando-se os dados contidos no dispositivo móvel somente.

Portanto, este módulo recebe como entrada um nome de pessoa a ser pesquisado do módulo **Social Emergency Analytics** e retorna para o mesmo módulo uma lista de contatos que correspondem ao nome pesquisado, contendo as informações básicas de cada contato (nome, foto e data e hora do último contato realizado com o usuário) e os dados de interação social do contato com o usuário via dispositivo móvel (total de mensagens SMS/MMS trocadas, número de ligações realizadas e recebidas e duração total dessas ligações).

Além disso, este módulo também verifica se cada contato na lista de retorno possui um perfil de Facebook atribuído a ele. Em caso positivo, o contato é marcado com seu perfil de Facebook para a análise das informações e dos dados provenientes das mídias sociais, realizada pelo próximo módulo. As atividades deste módulo e suas interações são exibidas na Figura 38.

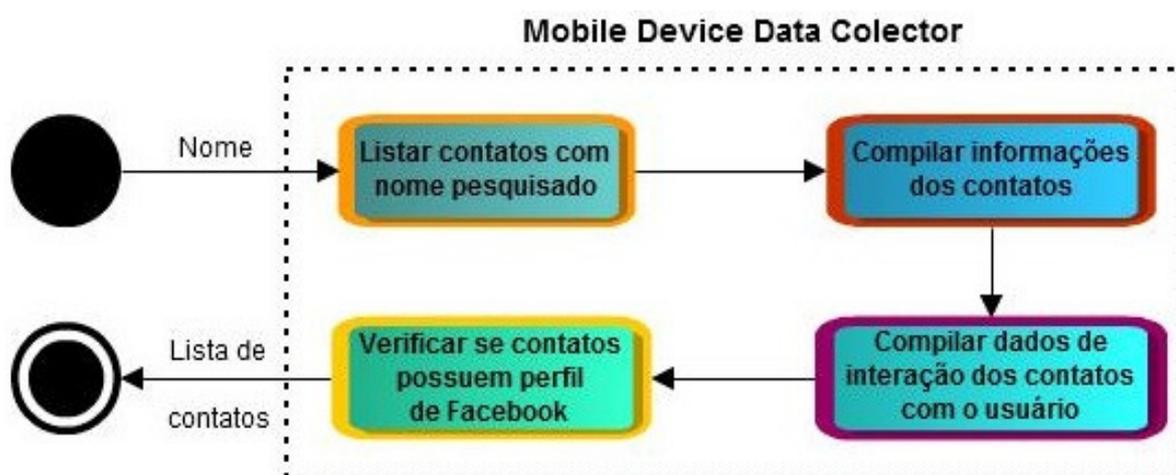


Figura 38. O módulo Mobile Device Data Colector e suas atividades

Com todas as funcionalidades oferecidas pelo *Contacts Provider*, não foi muito difícil obter-se esses dados de interação social entre os contatos e o usuário via dispositivo móvel. Para cada contato, pesquisamos em todos os números de telefones cadastrados essas possíveis interações sociais e também verificamos se ele possui um perfil de Facebook associado. Além do *Contacts Provider* para pesquisa das informações básicas dos contatos, também utilizamos o *CallLog Provider* para compilar os dados de ligações realizadas e recebidas.

Apesar do *CallLog Provider* prover fácil acesso a informações sobre as ligações realizadas e recebidas no dispositivo móvel, as informações sobre trocas de SMS/MMS entre os contatos e o usuário tiveram que ser compiladas através de um provedor de conteúdo não documentado. Este provedor de conteúdo para as mensagens SMS/MMS foi oficialmente

removido da documentação na versão 1.5 do Android SDK. Isso não quer dizer que não é possível mais utilizar o provedor de conteúdo SMS/MMS.

Como o sistema Android é *open source*, nada impede que o desenvolvedor procure na arquitetura do próprio sistema alguns provedores de conteúdo não documentados oficialmente pelo Google. Porém, apesar de esses provedores de conteúdo não documentados estarem disponíveis para uso, isso não quer dizer que os desenvolvedores devam usá-los. Deve-se considerar que a equipe de desenvolvedores responsáveis pelo sistema Android não garante suporte a esses provedores de conteúdo não documentados. Com isso, o acesso a um provedor de conteúdo não documentado usado em uma determinada versão do sistema Android pode mudar na versão seguinte, ou pior ainda, o provedor de conteúdo pode ser extinto de uma versão para a outra¹⁷.

Além dessas considerações sobre este módulo, cabe também ressaltar que a análise e compilação de outros dados de interação social como através de *e-mails*, mensageiros e outras aplicações para comunicação via dispositivo móvel só não foram realizadas neste protótipo, pois não havia uma forma nativa de realizá-las usando-se o Android SDK (como no caso das ligações e SMS/MMS, por exemplo). Isso não quer dizer que seja impossível acessar essas informações. Como já explicado anteriormente, o sistema Android foi construído especificamente para se facilitar a interação entre suas mais diversas aplicações.

Portanto, para se acessar esses tipos de informações, bastaria verificar para cada caso se o aplicativo próprio para cada tipo de comunicação utilizada permite o acesso a essas informações através de um provedor de conteúdo próprio. Essa compilação de dados poderia ser realizada para cada aplicação do mesmo tipo de comunicação entre o contato e o usuário, como, por exemplo, no caso de um usuário utilizar dois clientes de e-mails distintos ou vários aplicativos de mensageiro diferentes.

Apenas para se ilustrar um exemplo do que foi falado anteriormente sobre os perigos de se utilizar provedores de conteúdo não documentados no Android SDK, havia um provedor de conteúdo nativo de *e-mails* do Gmail que nas versões mais recentes do Android foi fechada a acessos externos, significando que agora somente a própria aplicação de *e-mails* do Gmail possui acesso a essas informações.

¹⁷ Mais informações sobre provedores de conteúdo não documentados podem ser encontradas em <http://android-developers.blogspot.com.br/2010/05/be-careful-with-content-providers.html>.

5.5.3.3 Facebook Data Colector

Este módulo é construído através de uma Atividade do Android SDK, porém conta com o auxílio do Facebook SDK para pesquisar as informações dos contatos do dispositivo móvel do usuário no Facebook, além das interações sociais entre esses contatos e o usuário nessa mídia social. Sua função é basicamente compilar as informações desses contatos (sexo e data de nascimento) e os dados de interação social entre eles e o usuário no Facebook (citações, compartilhamento de conteúdo, mensagens, comentários e *likes*).

Para isso, ele recebe como entrada uma lista de contatos cujos nomes correspondem ao nome pesquisado (vinda do módulo **Social Emergency Analytics**) e, para cada um desses contatos que possuem um perfil de Facebook associado, este módulo então complementa as informações do contato e os dados de interação social entre ele e o usuário no Facebook, retornando para o módulo **Social Emergency Analytics** a mesma lista, porém agora com informações complementares vindas do Facebook preenchidas.

Apesar da aparente simplicidade deste módulo, ele é um dos módulos mais importantes do protótipo, pois é justamente através dele que conseguimos obter as informações complementares sobre os contatos e também suas interações sociais com o usuário no Facebook. A atividade de compilação de dados provenientes das mídias sociais é uma das mais complexas de todo este trabalho e será explicada com mais detalhes a seguir. As atividades deste módulo e suas interações são exibidas na Figura 39.



Figura 39. O módulo Facebook Data Colector e suas atividades

Assim que este módulo é invocado, seu mecanismo de autenticação verifica se o usuário possui um *token* de acesso ao Facebook ainda válido. Como o Facebook SDK utiliza um mecanismo de *single-sign-on*, este módulo procura reaproveitar *tokens* de acesso ao Facebook provenientes de autenticações anteriores, tenham elas sido feitas através de nosso próprio aplicativo ou de outros aplicativos que compartilhem o mecanismo de autenticação do

Facebook SDK. Dessa forma, o usuário só precisa se autenticar com o Facebook através do nosso aplicativo caso não exista nenhum *token* de acesso ao Facebook válido, ou seja, caso o usuário não tenha acessado o Facebook de alguma forma via seu dispositivo móvel.

Esse mecanismo de autenticação do Facebook SDK mostra-se bastante versátil, uma vez que fornece dois tipos básicos de *tokens* de acesso: o *short-lived* (com validade mínima de uma hora e máxima de duas horas) e o *long-lived* (validade de 60 dias). A plataforma do Facebook oferece ainda a opção de estender o período de validade do *token* de acesso uma vez por dia. Dessa forma, caso o usuário esteja usando um *token short-lived* válido, por exemplo, a primeira vez no dia que ele acessar o Facebook usando esse *token*, a plataforma do Facebook estenderá o prazo de expiração para esse *token*.

Assim sendo, muito raramente o fluxo de execução do nosso protótipo é interrompido para autenticar um usuário, por causa de um *token* de acesso inválido. Porém, há também eventos que podem fazer um *token* de acesso se tornar inválido antes de esgotar seu prazo de validade. Tais eventos incluem o usuário alterar sua senha, o aplicativo atualizar sua chave secreta do Facebook, ou o usuário desautorizar a aplicação a acessar seus dados no Facebook.

No caso de o usuário desautorizar a aplicação a acessar seus dados no Facebook, deve-se considerar esse usuário como sendo um novo usuário efetivamente e, nesse caso, deve-se implementar o fluxo de autorização padrão da aplicação. Esse fluxo de autorização padrão da aplicação inclui solicitar ao usuário as permissões de acesso a seus dados no Facebook. Normalmente, esse fluxo de autorização padrão da aplicação é executado somente uma única vez, quando se instala o aplicativo no dispositivo móvel e ele é executado pela primeira vez. Nos demais casos, uma simples autenticação de usuário é efetuada para se obter um novo *token* de acesso ao Facebook válido.

Após a verificação da existência de um *token* de acesso válido, ou a autenticação do usuário, caso seja necessária, é realizada a compilação dos dados e informações coletadas via Facebook API. Primeiramente, o módulo compila as informações básicas (sexo e data de nascimento) de cada contato que possui um perfil no Facebook. Em seguida, ele começa a verificar as interações sociais (citações, compartilhamento de conteúdo, mensagens, comentários e *likes*) entre esses contatos e o usuário no Facebook.

Devido à restrição de o FQL permitir somente consultas indexáveis, nosso protótipo compilou apenas os dados de citações e *likes* em postagens diretas, ou seja, citações e *likes* realizados em comentários (sejam eles feitos nas próprias postagens do usuário, do contato ou de terceiros) não puderam ser analisados. Porém, deixamos claro aqui que a análise desses dados não é impossível. Devido à análise desses dados pelo protótipo ser realizada em *real*

time, realizando requisições ao Facebook API enquanto o usuário espera por uma resposta, ficou inviável analisarmos esses tipos de dados por conta dessa restrição a consultas indexáveis somente feita pelo FQL.

Para se analisar esses tipos de dados, é necessário trabalhar com as colunas indexadas de cada uma das tabelas consultadas para se obter esses dados. Com isso, para analisarmos as citações e *likes* feitos em comentários nas postagens do usuário, do contato ou de terceiros, precisaríamos obter primeiramente todas essas postagens, para depois coletar cada comentário feito em cada uma dessas postagens. Esse tipo de operação demanda muito processamento e armazenamento de dados para ser realizada em *real time*. Novamente, frisamos que não é impossível de se realizar essas operações e analisar esses tipos de dados, mas que somente teríamos uma performance sofrível em *real time*.

Para evitarmos esse tipo de problema, foi sugerida justamente a arquitetura em três nós citada na seção 5.4. Outra peculiaridade de nosso protótipo é que as *tags* em fotos ou vídeos são tratadas simplesmente como citações e, portanto, padecem do mesmo problema explicado anteriormente. Novamente, não que seja impossível tratar *tags* de fotos ou vídeos de maneira diferente, mas essa foi a nossa decisão de implementação do protótipo visando simplificar o cálculo do índice de relevância social. Mas, para todos os efeitos, o Facebook API fornece modos de se diferenciar e consultar *tags* de fotos e vídeos separadamente, permitindo assim analisarmos esses tipos de dados de maneira diferente e dando a eles pesos distintos no nosso algoritmo de cálculo da relevância social do contato em relação ao usuário.

Outros problemas e inconsistências também foram encontrados utilizando o Facebook SDK, quando testávamos nosso protótipo fazendo requisições ao API do Facebook. Uma das inconsistências encontradas foi a existência de “*posts fantasmas*”, ou seja, postagens do usuário que apareciam em seu perfil do Facebook, mas que não conseguiam ser recuperadas via Facebook API. Apesar de o Facebook SDK deixar claro que as consultas realizadas via API do Facebook não refletem os dados mais recentes da plataforma Facebook, encontramos algumas postagens que não eram recuperadas via Facebook API, mas que já haviam sido publicadas na plataforma do Facebook há um tempo considerável, como dias ou semanas.

Outra dificuldade encontrada foi com o esquema de paginação oferecido pela API do Facebook. A paginação do Facebook API explicita um parâmetro de limitação da quantidade de resultados retornados. Porém, o número de resultados retornados nem sempre é igual a esse limite especificado. Esse é um comportamento esperado do Facebook API. Os parâmetros da consulta são aplicados no lado servidor antes da verificação de visibilidade dos resultados, ou seja, antes de ver se os resultados retornados são visíveis para o usuário. Por isso, é possível

obter menos resultados do que o esperado. Isso pode tornar a paginação difícil e confusa, pois o fato de se obter menos resultados do que o esperado não necessariamente significa que não haja mais resultados a serem obtidos.

Dessa forma, dependendo do tamanho do parâmetro limite da resposta configurado é possível se obter resultados finais diferentes do esperado. Especificamente, quanto menor o tamanho da página, maior a probabilidade de se obter um resultado final incoerente com os dados reais armazenados na plataforma Facebook. Por exemplo, digamos que um contato possua 15 postagens que possam ser recuperadas via Facebook API, para o usuário que está utilizando nosso protótipo. Se ajustarmos o parâmetro limite da resposta para 10 itens e realizarmos a busca, o Facebook API irá pegar os 10 primeiros resultados que conferem com nossa consulta e então realizará o teste de visibilidade desses resultados (em relação ao usuário utilizando o protótipo) antes de nos enviar a resposta.

Digamos que desses 10 primeiros resultados, três não são visíveis ao usuário que realizou a consulta através de nosso protótipo. Então, somente sete resultados serão retornados. Se considerarmos que o tamanho do resultado recebido é menor do que o tamanho limite do resultado simplesmente porque não existem mais dados que confirmam com nossa consulta, então obteremos somente uma parte do total do resultado esperado. Assim, teremos obtido somente sete resultados e estaremos ignorando os oito resultados restantes que não foram solicitados ao Facebook API. A Figura 40 ilustra bem esse problema da paginação.

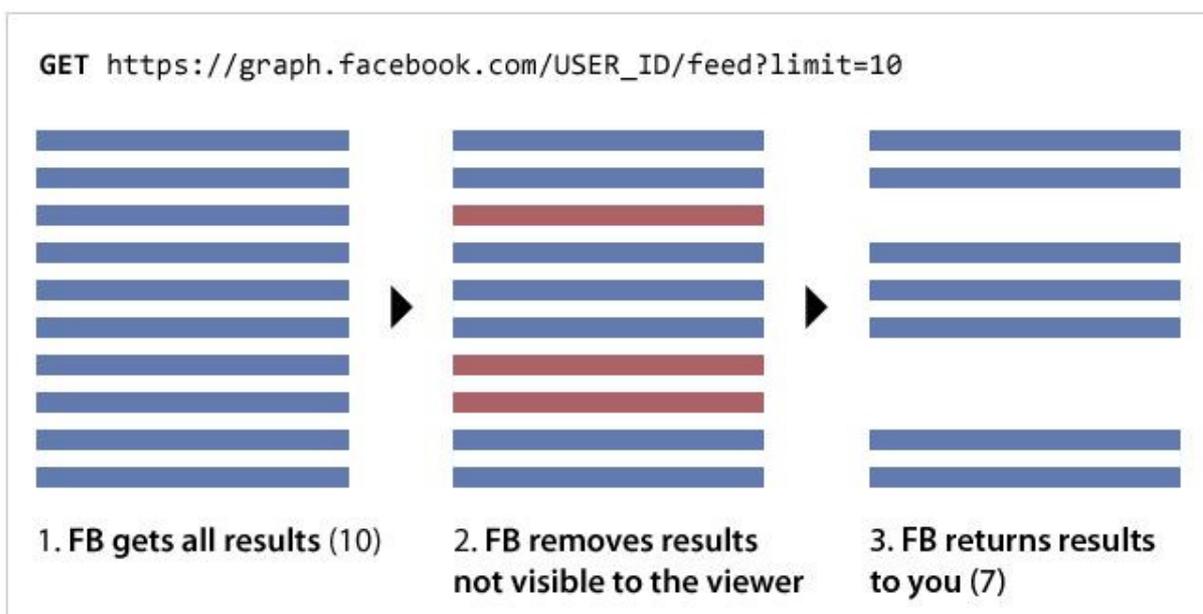


Figura 40. Paginação e verificação de visibilidade dos resultados no Facebook API (FACEBOOK DEVELOPERS, 2011)

Com isso, é possível até mesmo obter um resultado final de tamanho zero (apesar de na plataforma Facebook existir resultados a serem coletados) se configurarmos um tamanho

limite de página suficientemente pequeno, ou se coincidir de existirem múltiplos resultados não visíveis para o usuário em sequência na resposta da consulta. Para se tentar mitigar esse problema, consideramos que somente não existem mais resultados a serem buscados quando o tamanho da resposta é zero, e configuramos nosso protótipo para utilizar o tamanho máximo de página permitido pelo Facebook API, que são 5000 resultados (antes de o Facebook API realizar a verificação de visibilidade desses resultados). Esse esquema de paginação do Facebook API também permite uma incoerência de resultados finais diferentes para consultas idênticas realizadas com limites de páginas de tamanhos diferentes.

Outro problema encontrado remete ao tipo de recurso do Facebook SDK utilizado para se coletar os dados do Facebook. A Graph API permite uma interação mais facilitada com os objetos do Facebook e suas conexões, inclusive retornando alguns tipos de dados exclusivos desse recurso. Já o FQL permite maior liberdade e mais flexibilidade nas consultas realizadas, possibilitando ao usuário explorar esses mesmos objetos e conexões de diversas maneiras distintas. Ambos os recursos possuem seus prós e contras que devem ser avaliados na hora de se decidir de que maneira iremos recuperar os dados do usuário no Facebook.

Por fim, como nosso protótipo realiza a compilação dessas informações provenientes do Facebook em tempo real, esbarramos também no problema da dependência do Facebook API e das limitações dos recursos de seu *cluster* de servidores (como processamento, armazenamento, memória, etc.). Por vezes, durante nossos testes, o Facebook API acusou erro de processamento nas requisições de consultas enviadas pelo Facebook SDK. Na seção 7.2 discutimos um pouco mais todos esses problemas e limitações encontrados durante a construção de nosso protótipo, e na seção 7.3 indicamos melhorias a serem realizadas em nossa proposta de solução para se evitar ou tratar cada um desses problemas.

5.6 Trabalhos Correlatos

Durante nossos estudos encontramos alguns trabalhos difundidos na área de busca por vítimas de emergências e outros trabalhos que aproveitavam os dispositivos móveis e as redes sociais para recomendação de pessoas. Porém, não conseguimos encontrar um trabalho que combinasse esses dois aspectos: recomendação de pessoas através da análise contextual da rede social e aquisição de informações sobre vítimas de emergência.

Portanto, inspirados por esse nicho encontrado e se balizando por esses trabalhos estudados, decidimos criar esta proposta que combina a análise contextual da rede social, apoiada pelos dispositivos móveis e pelas mídias sociais, e a busca por informações

confiáveis e seguras sobre as vítimas de determinada emergência. Nas seções a seguir descreveremos os trabalhos estudados em ambas essas áreas.

5.6.1 Área de Busca por Informações sobre Vítimas de Emergência

Em todos os trabalhos descritos a seguir, o foco é se obter qualquer informação da vítima, como por exemplo sua localização, seu estado de saúde ou se ela está viva. Isso sugere que não há padronização ou verificação alguma das informações disponibilizadas, uma vez que qualquer pessoa pode anonimamente acessar esses sistemas. Conforme veremos no estudo desses trabalhos, todos se comportam da mesma maneira: oferecem um repositório unificado para disponibilização de informações sobre as vítimas de uma determinada emergência. O foco de nosso trabalho é construído sob outro prisma. Não o da disponibilização da informação em si, qualquer que seja, mas sim o da recomendação de pessoas para prestar informações confiáveis e íntegras sobre alguém.

Na última semana de Agosto de 2005, os Estados Unidos da América foram atingidos por uma tempestade tropical, o furacão Katrina, um grande furacão que destruiu uma parte do país e que alcançou a categoria 5 da escala de furacões de Saffir-Simpson (regredindo a 4 pouco antes de chegar à costa sudeste). Este desastre natural afetou mais de um milhão de pessoas na região litorânea do sul dos Estados Unidos, causando aproximadamente mil mortes. Com o propósito de auxiliar na fase de resposta a esta emergência, foi criado no início de Setembro de 2005 o Katrina PeopleFinder Project (MURPHY; JENNEX, 2006) em resposta às dezenas de grupos coletando listas de “perdidos e seguros” para as pessoas afetadas pelo furacão Katrina.

Assim como aconteceu no rescaldo do 11 de Setembro, uma infinidade de *sites* tinha reunido listas de sobreviventes ou de pessoas desaparecidas independentemente. O Katrina PeopleFinder Project criou um sistema para inserir dados de acordo com um formato padrão (PeopleFinder Interchange Format, também conhecido como PFIF), e visou os outros *sites* recolhendo esta informação, encorajando-os a usar o mesmo banco de dados e evitar a duplicação de esforço. Então, esse projeto procura criar um único banco de dados que combina tantas fontes quanto possível for de toda a *web* sem interromper o ímpeto existente da necessidade de combinar todos os dados de refugiados em grandes bancos de dados, como a Cruz Vermelha, e grandes fóruns, como o Craigslist.

Outra iniciativa nesse mesmo sentido é o People Finder Interchange Format (PFIF, 2005), que é um formato XML usado para troca de informações sobre indivíduos identificados ou encontrados na sequência de uma catástrofe. Foi criado rapidamente após a

devastação do furacão Katrina, como parte do Katrina PeopleFinder Project, em setembro de 2005. O PFIF é um padrão de dados público amplamente utilizado para obter informações sobre pessoas desaparecidas ou desabrigadas e foi projetado para permitir o compartilhamento de informações entre governos, organizações de socorro, e os registros de outros sobreviventes para ajudar as pessoas a encontrarem e contatarem seus familiares e amigos depois de um desastre.

O PFIF é uma extensão do XML, composto de registros de pessoas, que contêm informações de identificação sobre uma pessoa, e registros de nota, que contêm comentários e atualizações sobre o estado e a localização de uma pessoa. Cada nota é anexada a uma pessoa. Sua especificação define o conjunto de campos nesses registros e um formato baseado em XML para armazenar ou transferi-los. Registros PFIF XML podem ser incorporados em Atom feeds ou RSS feeds. Então, ele permite diferentes repositórios de dados de pessoas desaparecidas trocarem e agregarem seus registros. Cada registro tem um identificador único, que indica o nome do domínio do repositório original onde o registro foi criado. O identificador único é preservado enquanto o registro é copiado de um repositório para outro. Por exemplo, qualquer repositório que recebe uma cópia de uma determinada pessoa pode publicar uma nota anexa a essa pessoa, e até mesmo enquanto a nota e a pessoa são copiados para outros repositórios, eles permanecem rastreáveis às suas respectivas fontes originais.

O Google Person Finder (GOOGLE, 2010a), por sua vez, é uma aplicação *web* de código aberto¹⁸ que fornece registros e quadro de mensagens para os sobreviventes, familiares e entes queridos afetados por um desastre natural para disponibilizar e procurar informações sobre o estado e localização das pessoas. Ele foi criado por engenheiros do Google, em resposta ao terremoto no Haiti de 2010. O sistema é escrito em Python e hospedado no Google App Engine. Seu banco de dados e API são baseados no People Finder Interchange Format (PFIF) desenvolvido em 2005 para o Katrina PeopleFinder Project.

Geralmente, o Google Person Finder é embutido em uma página multilíngue de *Crisis Response* no *site* do Google, que também contém várias outras ferramentas de desastres, tais como fotografias de satélite, locais de abrigo, condições da estrada e informações sobre queda de energia. Para o terremoto e tsunami Tōhoku em 2011, o Google também configurou uma conta do Picasa para permitir que as pessoas enviassem fotos das listas de nome afixadas em abrigos de emergência, para serem manualmente transcritas.

¹⁸ O código fonte do sistema e demais informações sobre o mesmo podem ser encontrados em <http://code.google.com/p/googlepersonfinder/>.

Para a coleta das informações sobre as pessoas desaparecidas, os *sites* que adotam PFIF podem interligar-se uns com os outros através da exportação e transmissão de dados ou permitindo que seu site seja vistoriado automaticamente; *sites* como *blogs* e contas de usuários que não são compatíveis, são revisados por voluntários que procuram as informações pessoais que faltam em formato PFIF. O *widget* usado para disponibilizar informações sobre as pessoas diretamente tem dois botões (“eu estou procurando por alguém” e “eu tenho informações sobre alguém”), e pode ser incorporado diretamente em outras páginas da *web*.

Dessa forma, o Google Person Finder ajuda as pessoas a se reconectarem com amigos e entes queridos no rescaldo das catástrofes naturais e humanitárias: Uma crise acontece e as pessoas se separam. Elas deixam que o mundo saiba que estão procurando alguém. Indivíduos e organizações fornecem informações. As pessoas encontram informações sobre seus amigos e familiares (GOOGLE, 2010b). Esse fluxo de acontecimentos pode ser visualizado abaixo, na Figura 41.



Figura 41. Esquema de utilização do Google Person Finder (GOOGLE, 2010b)

Já o Crisis' Search (CRISIS, 2005) é um motor de busca na *internet* criado pela Gigahertz Inc. e fundamentado em um simples conceito útil e humanitário — compilar um banco de dados pesquisável de *links* relacionados com desastres que vão desde programas de governo até *weblogs* dando dicas de evacuação e outras dicas para que qualquer pessoa possa pesquisar e encontrar a ajuda que ela precisa em tempos de crise.

Esse portal *web* foi criado por conta da existência de muitos *sites* criados para ajudar durante as crises e que precisavam ser reunidos em um único lugar para facilitar o acesso a essas informações e, especialmente, atender a necessidade das vítimas do furacão Katrina a encontrar maneiras de conectar-se com entes queridos perdidos ou solicitar auxílio financeiro. Em razão da temporada de tempestades nos Estados Unidos, as pessoas precisam encontrar maneiras de se preparar para essas tempestades, verificando rotas de evacuação, procurando abrigos, analisando mapas e imagens de satélite ou procurando formas de deixar suas casas preparadas contra os efeitos das tempestades tropicais.

Então, o lançamento do *site* CrisisSearch.com foi especialmente oportuno, pois além de ajudar as organizações focadas na emergência do furacão Katrina a promover livremente seus grupos, também serviu para causas menores que muitas vezes são prejudicadas pela falta de ajuda ou informações durante esses tempos difíceis. O portal conta com a assistência da comunidade através da contribuição de *sites* bons e confiáveis para alimentar o banco de dados de *web sites* relacionados a crises. Os tipos de *sites* cadastrados são aqueles relacionados com evacuação, desastres e resposta a emergências.

Cada página da *web* submetida ao Crisis' Search é analisada em busca de *meta tags* e todo o texto da página principal é “quebrado” em palavras-chave para a formação do *cache* de dados. Todo *site* submetido deve ser aprovado antes de estar disponível nos resultados de pesquisa, a fim de evitar *spams* ou conteúdos e dados não relacionados.

Podemos verificar que os trabalhos correlatos citados não visam à confiabilidade das informações prestadas sobre a pessoa em questão, ou de suas fontes. Qualquer um (organização ou pessoa) pode disponibilizar informações sobre alguém e não há nenhum mecanismo de verificação da veracidade dessas informações. Tudo é baseado na confiança de que quem está disponibilizando as informações está falando a verdade. O próprio PFIF não fornece meios de se analisar a confiabilidade das informações. No nosso trabalho, estamos preocupados com a confiabilidade e a integridade das informações fornecidas. Para tanto, analisamos a rede social dos usuários verificando determinadas interações e tipos de relações entre esses usuários e a vítima da emergência. A partir dessa análise, inferimos um determinado grau de confiabilidade para os usuários que serão recomendados por nossa ferramenta móvel para auxiliar a equipe de resposta à emergência com informações que serão utilizadas durante o processo de tomada de decisão para a solução da emergência.

Por fim, os trabalhos correlatos citados também são completamente dependentes da pró-atividade dos usuários em se disporem a acessar o *site* e disponibilizar informações sobre uma determinada pessoa, enquanto que em nossa solução a busca por usuários que possam prestar informações sobre a vítima da emergência é feita de modo passivo, através da análise da rede social do usuário disponibilizada nos dispositivos móveis e na *web*. Assim, nossa solução será capaz de tornar mais ágil o processo de fornecimento de informações confiáveis e íntegras, através da indicação de um grupo de usuários (locais preferivelmente, mas também poderemos indicar usuários remotos) que estão fortemente relacionados à vítima.

5.6.2 Área de Dispositivos Móveis, Redes Sociais e Colaboração

Redes sociais móveis é um novo termo cunhado pela comunidade de computação pervasiva para descrever o papel de um número crescente de usuários móveis que utilizam os seus dispositivos principalmente para exercer funções sociais, como aquelas fornecidas tipicamente nas redes sociais. Elas permitem que os indivíduos se conectem socialmente uns aos outros usando um telefone celular.

Nesse sentido, Smith (2005) mostra casos de algumas empresas pequenas que estavam começando a explorar a crescente demanda de aplicações sociais móveis, também conhecidas como *mobile social-software services* (MoSoSos). Ele apresenta algumas aplicações MoSoSos, como o Dodgeball¹⁹, o Plazes²⁰, o Playtxt²¹ e o Jambo Networks²², este último sendo a única aplicação que persiste no mercado até hoje.

Na primeira aplicação, quando um usuário registrado realizava um *check-in*, seus amigos recebiam uma mensagem de texto indicando o local e a hora desse *check-in*, caso eles quisessem se encontrar. O serviço também notificava o usuário se um amigo, o amigo de um amigo, ou ainda sua “paixão” estivesse dentro de um perímetro de 10 quarteirões. Além disso, os usuários também podiam transmitir mensagens ou “gritar” para aqueles em sua rede.

O Plazes era um sistema de interação sensível à localização que ajudava os usuários móveis a conectarem-se com amigos ou outras pessoas semelhantes em qualquer lugar do globo. Os usuários disponibilizavam a sua localização e atividades atuais, compartilhando essas informações com outros usuários Plazes via computadores ou telefones celulares. Já o Playtxt ajudava os usuários móveis a localizarem amigos, amigos de conhecidos mútuos, ou desconhecidos com preferências correspondentes, que estivessem nas proximidades. Ele também permitia mensagens instantâneas entre os usuários sem revelar o número do celular.

Por fim, o Jambo Networks usa laptops, telefones celulares e PDAs que possuem Wi-Fi para encontrar pessoas a uma curta distância que possuem interesses similares e gostariam de se encontrar face a face. Ele fornece um diretório de membros móveis que funciona como um radar para oferecer a qualquer grupo de pessoas uma capacidade de detectar, pesquisar,

¹⁹ O Dodgeball foi fundado em 2000 pelos estudantes Dennis Crowley e Alex Rainerte da New York University. O serviço consistia de uma rede social móvel nova-iorquina com milhares de usuários em 22 cidades por todo os EUA. O Google adquiriu esse serviço em 2005 e o interrompeu em 2009, substituindo-o pelo Google Latitude.

²⁰ O Plazes era uma rede geosocial alemã fundada em 2004. Em 2008 a Nokia adquiriu o serviço e finalizou-o em 2012, migrando seus usuários e históricos para o Nokia Maps.

²¹ O Playtxt era uma rede social móvel baseada em localização que permitia aos usuários enviarem suas localizações por SMS, para receberem de volta quem estava nas proximidades e quais locais estavam cheios. O serviço foi criado em 2001 por um empreendedor britânico chamado Andrew J. Scott e funcionou até 2006.

²² O Jambo (<http://jambo.net/>) é uma aplicação *business-to-business* privada para grupos de afinidade, associações, conferências e comunidades *online*. A empresa foi fundada em 2003 e é sediada em Dallas, Texas.

navegar e se conectar uns aos outros, quando eles estão na mesma cidade e apenas a alguns quarteirões de distância.

Em seu trabalho, Smith apresenta um aplicativo para celulares desenvolvido no centro de pesquisas da Intel em Seattle, chamado Reno. Nesse aplicativo, os usuários podem consultar outros usuários sobre suas localizações e também divulgarem sua própria localização, quer seja em resposta a outra consulta ou espontaneamente. O Reno calcula a localização aproximada do aparelho localmente usando tecnologia GSM e, em seguida, apresenta ao usuário uma pequena lista de locais próximos classificados por proximidade.

Além da percepção local, o Reno incorpora três fatores de projeto que são críticos para o sucesso de aplicações sociais móveis: privacidade, praticidade e valor específico. Em relação à privacidade, as aplicações MoSoSos devem dar aos usuários controle suficiente de seus dados pessoais ou correrem o risco de serem rejeitadas como agentes do “*Big Brother*”. Portanto, o Reno incorpora uma série de funcionalidades de privacidade, incluindo controle do usuário sobre a divulgação das informações de localização.

Quanto à praticidade, as aplicações sociais móveis devem ser simples de usar e rápidas de operar, ou então as pessoas irão escolher outras formas de comunicação, como o telefone. Reno foi projetado para explorar a tradicional metáfora da caixa de entrada para deixar as pessoas olharem rapidamente para pedidos de sua localização e divulgá-la aos outros. Já no quesito valor específico, um aplicativo social móvel deve oferecer algum benefício chave para convencer os usuários a escolhê-lo repetidamente em detrimento a outras técnicas de comunicação. Em experimentos realizados pelos pesquisadores da Intel, as pessoas utilizavam o Reno em média 2,4 vezes por dia para marcarem encontros, o que foi encorajador dada à dificuldade inerente em usar um novo aplicativo em celulares desconhecidos.

Outros MoSoSos sensíveis à localização que são populares entre os usuários de redes sociais móveis atualmente são o Google Latitude²³ e o Foursquare²⁴. O Google Latitude (GOOGLE, 2009) é um aplicativo móvel sensível à localização desenvolvido pelo Google como sucessor ao serviço baseado em SMS anterior, o Dodgeball. Ele permite que um usuário de celular disponibilize sua localização atual para algumas pessoas.

Através da própria conta do usuário no Google, sua localização baseada no celular é mapeada no Google Maps²⁵. O usuário pode controlar a precisão e os detalhes do que cada um dos outros usuários podem ver (desde a localização exata até simplesmente a identificação da

²³ <https://www.google.com/latitude>

²⁴ O serviço foi criado em 2009 por Dennis Crowley (cofundador do Dodgeball) e Naveen Selvadurai. Ele pode ser acessado em <https://foursquare.com/>.

²⁵ <https://maps.google.com.br/>

cidade). Devido a questões de privacidade, a localização automática pode ser desligada pelo usuário, ou então um local pode ser inserido manualmente. Além disso, os usuários só podem ver a localização dos amigos que decidiram compartilhar sua localização com eles.

Já o Foursquare (2009) é uma rede social móvel sensível à localização onde os usuários realizam o *check-in* em locais usando um celular, via mensagens de texto ou um aplicativo específico. O local é selecionado de uma lista de locais que a aplicação localiza nas proximidades do usuário. A localização é baseada no *hardware* de GPS do dispositivo móvel ou no local da rede fornecido pela aplicação. Cada *check-in* dado pelo usuário é premiado com pontos (uma forma de *marketing* pessoal das empresas que podem render aos usuários premiações em certos estabelecimentos) e, às vezes, emblemas (como um jogo entre os usuários para saber quem frequenta mais determinados locais ao redor do mundo).

Os usuários são encorajados a serem extremamente específicos e detalhados em suas localizações, podendo fazer *check-ins* em determinados andares ou áreas de um edifício, ou ainda indicar uma atividade que estejam fazendo no local. Os usuários podem optar por terem seus *check-ins* publicados em suas contas no Twitter²⁶, Facebook²⁷, ou ambos. Em algumas versões do aplicativo é possível visualizar atualizações dos amigos, chamados de “Pings”. Os usuários também podem criar listas de afazeres para uso privado e adicionar dicas para locais que outros usuários possam ler, servindo como sugestões para coisas legais a se fazer, ver ou comer no local. O sistema também disponibiliza recomendações personalizadas e promoções baseadas em onde o usuário, seus amigos, e pessoas com gostos semelhantes estiveram.

Em relação à utilização de dispositivos móveis em trabalhos que visam à colaboração, vemos que apesar de bilhões de usuários levarem consigo aparelhos celulares nos dias de hoje, poucos deles usam os telefones de maneiras mais interessantes além de um simples telefonema. Interessado nesse fenômeno contraditório, SANTOS et al. (2008) apresenta um projeto de sistema e uma prova de conceito de um arcabouço que permite a colaboração generalizada entre os usuários de telefone celular através de uma rede social móvel. A ideia é usar os novos recursos disponíveis em telefones inteligentes para explorar os princípios de localidade espacial (área geográfica), localidade temporal (aviso imediato) e correspondência de interesse (interesse pessoal no assunto), a fim de contatar um número de usuários para colaborar em mini-tarefas, também chamadas de mini-campanhas que se formam espontaneamente como os já famosos e mundialmente difundidos “*flash mobs*”.

²⁶ <https://twitter.com/>

²⁷ <http://www.facebook.com/>

Em outro trabalho, ANKOLEKAR et al. (2009) apresenta Friendlee, uma ferramenta móvel que, através da análise das ligações e das trocas de mensagens do celular, forma uma rede íntima dos contatos socialmente mais próximos do usuário e proporciona uma percepção ambiental de sua rede social de uma forma convincente e não intrusiva. Esses tipos de redes sociais íntimas com os laços mais próximos e mais significativos, como entre amigos, família, parentes e colegas mais próximos, são caracterizados por uma alta frequência de interação, mas também pela grande necessidade de se sentir conectado, de estar em contato, e uma necessidade de compartilhar atividades detalhadas e informações contextuais.

Dessa forma, Friendlee é completamente centralizado no telefone celular e na sub-rede social que o medeia, amalgamando a maioria dos indicadores de percepção úteis, tais como a localização atual e o tempo gasto lá, a hora e clima locais, etc. Ele não somente acessa uma rede social potencialmente diferente, como é capaz de construir automaticamente um rico retrato da vida social do usuário com uma mínima participação do mesmo.

Já o trabalho de Woerndl, Schueller e Wojtech (2007) trata da incorporação do contexto em sistemas de recomendação no domínio de aplicações móveis. Assim como um sistema de recomendação tenta prever a relevância de determinadas informações para um usuário, este é baseado em informações sobre o usuário, metadados associados a itens e/ou avaliações implícitas ou explícitas de artigos realizadas por um grupo de usuários. Os usuários desse sistema podem selecionar entre vários componentes baseados em conteúdo ou filtragem colaborativa, incluindo um módulo baseado em regras que utiliza informações sobre ponto de interesses nas proximidades do usuário, e um componente para a integração da tradicional filtragem colaborativa. Desse modo, a ideia dos autores é combinar a filtragem colaborativa com outros modos de recomendação para lidar com a complexidade adicional do contexto, criando assim o que eles chamaram de *Context-Aware Collaborative Filtering*.

Portanto, vimos nos trabalhos apresentados a importância dessas aplicações no auxílio à colaboração entre os usuários e no acesso às suas informações contextuais, que podem ser utilizadas para um levantamento de perfil dos usuários. Todos os trabalhos anteriores nos serviram de embasamento para a criação de um *framework* de solução que analise as informações contextuais existentes nas redes sociais e nos dispositivos móveis e as utilize para melhorar o processo de recomendação de pessoas, baseando-se também nas relações sociais do círculo de amigos do usuário.

Capítulo 6 – Ilustração de Uso

Neste capítulo iremos ilustrar a utilização de nossa proposta de solução em um cenário de emergência típico. Usaremos como base o protótipo construído a partir do arcabouço de solução proposto, mas também indicaremos os demais elementos da arquitetura discutida para nossa proposta de solução. Com isso pretendemos esclarecer como se daria a utilização de nossa proposta de solução no dia a dia da população e, principalmente, as interações realizadas com a equipe de resposta a emergências durante um cenário de desastre qualquer.

Primeiramente, começaremos discutindo como é realizado atualmente esse trabalho de coleta de informações sobre vítimas de acidentes que estão desaparecidas ou inconscientes. Então, na próxima seção, introduziremos nossa proposta de solução nesse cenário para verificarmos como se daria o fluxo de trabalho da equipe de resposta a emergências com o auxílio de nossa proposta de solução.

Em um cenário típico de emergência urbana, as atividades de resposta a essas emergências são coordenadas segundo uma estrutura simplificada, com a divisão dos indivíduos envolvidos nessas atividades em duas categorias ou grupos: o Comando e a Operação. Essas duas equipes distintas precisam colaborar entre si e compartilhar o conhecimento a fim de obterem êxito em suas atividades.

A equipe de Operação é formada por indivíduos responsáveis pela realização do socorro, trabalhando na linha de frente da emergência, seja resgatando vítimas, realizando atendimentos médicos, orientando as pessoas ou coletando informações sobre o acidente e sobre as vítimas. As atividades desempenhadas pelos indivíduos desse grupo são intensamente colaborativas e eles precisam de um alto grau de coordenação para se obter os resultados almejados, sem comprometer a segurança de todos os envolvidos no processo.

Acompanhando e orientando o trabalho da equipe de Operação estão os indivíduos da Sala ou Centro de Comando. Eles são os responsáveis por informar as ordens a serem executadas e fornecer informações que possam ser úteis ou que tenham sido solicitadas, visando à coordenação do trabalho desenvolvido pela equipe de Operação. O Centro de Comando monitora o desdobramento dos fatos, avaliando as mudanças ocorridas na situação e planejando os próximos passos. Devido ao fato dessa equipe ficar distante do local do cenário de emergência, sua percepção situacional é construída a partir dos relatos que lhe são enviados pela equipe de operação.

6.1 Cenário de Emergência Típico Atual

Atualmente, quando uma emergência ocorre, o órgão responsável é notificado, como, por exemplo, os Bombeiros ou a Defesa Civil, e então uma equipe de resposta a emergências é enviada ao local do cenário do acidente. Ao chegar ao local do acidente, a atuação dessa equipe tem o objetivo de reduzir os efeitos do desastre (minimizando assim as perdas materiais e humanas) através de ações de resposta aos desastres, conforme o planejamento prévio dessas instituições.

Os trabalhos realizados no cenário de emergência atuam direto nos efeitos da ocorrência e são desenvolvidos com o emprego coordenado de pessoal treinado dos vários órgãos envolvidos, conforme planos preestabelecidos. Algumas das atividades desempenhadas pelos integrantes da equipe de resposta a emergências, mais precisamente pela equipe de Operação, são os resgates de vítimas, evacuação de pessoas em áreas de risco e o encaminhamento dos flagelados a locais e abrigos seguros, atendendo-os com medicamentos, agasalhos, alimentos e conforto moral. Todas essas medidas assistenciais devem ser estudadas e planejadas antecipadamente, para que não ocorram precipitações, desencontros, superposição de providências, atrasos e desperdícios no seu desencadeamento.

Dessa forma, durante todo esse trabalho desempenhado pela equipe operacional no cenário de emergência, se faz necessária a obtenção de informações confiáveis sobre as vítimas do acidente, principalmente aquelas que estão desaparecidas ou inconscientes. Esse trabalho de recuperação de informações sobre as vítimas da emergência é realizado sob regime de cooperação entre a equipe de resposta a emergências e a comunidade, com aproveitamento máximo dos recursos disponíveis.

Portanto, em meio ao cenário caótico de um desastre, a equipe de resposta a emergências deve ainda procurar informações sobre as vítimas com os demais envolvidos no acidente. Hoje, esse processo de busca por informações sobre vítimas da emergência é realizado pessoalmente pelos integrantes da equipe operacional, que devem percorrer toda a extensão do local da emergência em busca de pessoas que possam auxiliar a equipe de resposta a emergências com valiosas informações sobre as vítimas desaparecidas ou inconscientes. A Figura 42 ilustra todo esse fluxo de trabalho acima descrito.

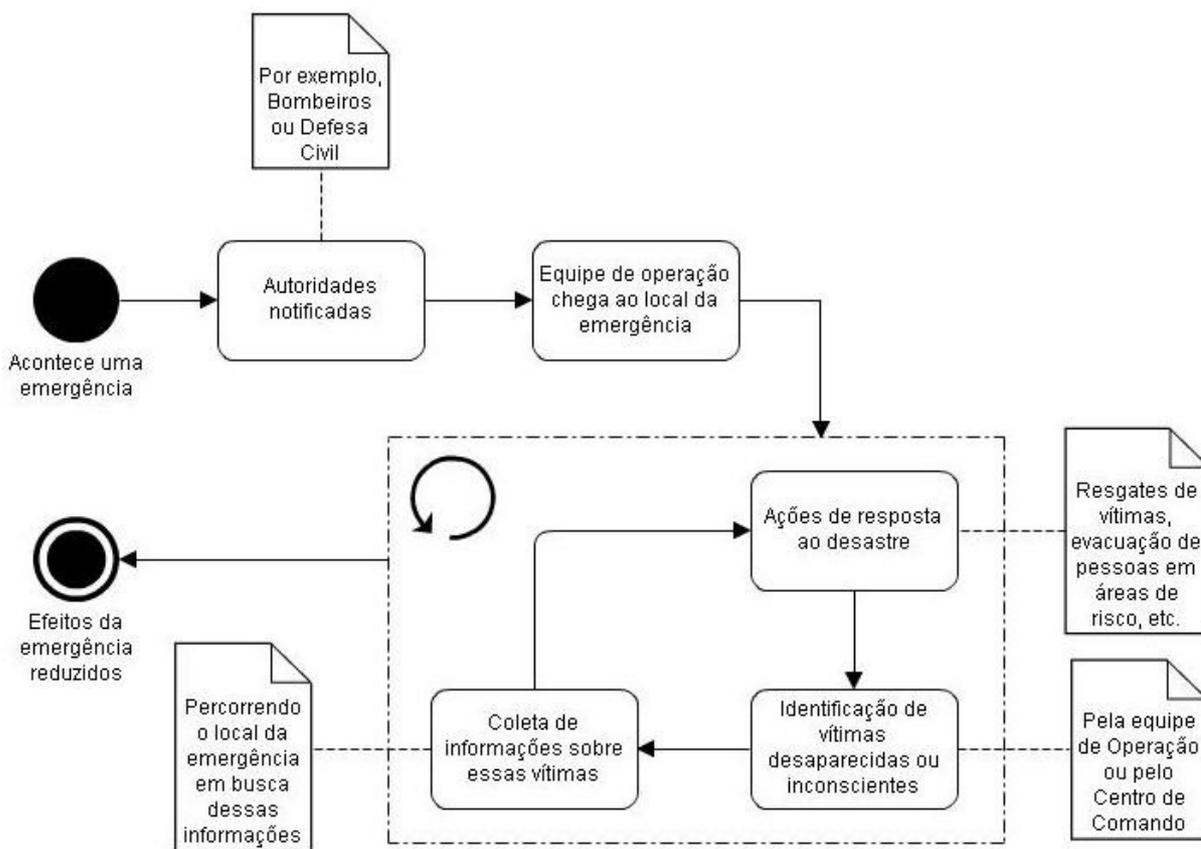


Figura 42. Fluxo de trabalho em um cenário de emergência atual

Vemos que esse fluxo de trabalho se inicia quando uma emergência ocorre. A partir desse momento, é necessário que as autoridades responsáveis pelo processo de combate à emergência sejam notificadas. Após a chegada da equipe de operação ao local da emergência é iniciado um ciclo de atividades visando à resolução da emergência. Esse ciclo é formado essencialmente por três atividades macro: (1) Ações de resposta ao desastre; (2) Identificação de vítimas desaparecidas ou inconscientes; e (3) Coleta de informações sobre essas vítimas. O trabalho da equipe operacional na fase de resposta de uma emergência só é dado por encerrado quando os efeitos da emergência foram reduzidos ou eliminados por completo.

6.2 Utilização da Solução Proposta no Cenário de Emergência

Nossa proposta de solução visa facilitar esse trabalho de coleta de informações confiáveis sobre vítimas desaparecidas ou inconscientes que a equipe de resposta a emergências realiza atualmente em um cenário de crise. Através de uma ferramenta computacional móvel construída sobre o arcabouço de nossa proposta de solução, a equipe de resposta a emergências obterá indicação de pessoas que fazem parte da rede social da vítima e possuem um determinado grau de conhecimento sobre ela para fornecerem informações

confiáveis sobre a vítima. Tais pessoas estariam preferencialmente próximas ao local do acidente, ou então teriam condições de chegar rapidamente ao local da emergência.

Dessa forma, o fluxo de utilização de nossa proposta de solução se inicia quando os cidadãos comuns (os usuários do sistema) instalam nossa ferramenta em seus dispositivos móveis. Durante a instalação, essa ferramenta é registrada no servidor central da arquitetura proposta (Servidor Web) fornecendo informações básicas sobre o usuário (o cidadão que acaba de instalar a ferramenta em seu dispositivo móvel) e obtendo o número de registro do usuário — seu identificador único no sistema.

É importante frisarmos que o usuário terá total conhecimento sobre as informações que serão enviadas e saberá que elas serão acessadas somente pela equipe de resposta a emergências durante um cenário de desastre onde ela precisará obter informações sobre vítimas desaparecidas ou inconscientes. Caso o usuário não concorde com o envio dessas informações, a ferramenta não será instalada em seu dispositivo móvel. Essas informações são criptografadas e somente a equipe de resposta a emergências poderá utilizá-las.

Após a instalação de nossa ferramenta no dispositivo móvel do usuário (Cliente Móvel), ele poderá configurar os aspectos comportamentais da ferramenta, tais como:

- Quantidade de vezes por dia que a ferramenta coletará as informações de interação social com os contatos do dispositivo móvel: seis (a cada 4 horas), quatro (a cada 6 horas), três (a cada 8 horas), duas (a cada 12 horas) ou uma (a cada 24 horas).
- Disponibilização da localização pessoal (envio das coordenadas GPS do dispositivo móvel): enviar automaticamente quando solicitado (ligado e habilitado), perguntar ao usuário antes de enviar (ligado, mas não habilitado) ou nunca enviar (desligado).
- Disponibilização do *ranking* de proximidade social dos contatos do dispositivo móvel (envio da listagem de contatos que correspondem à consulta pelo nome requisitada pelo Servidor Comando e Controle): enviar automaticamente quando solicitado (ligado e habilitado), perguntar ao usuário antes de enviar (ligado, mas não habilitado) ou enviar apenas em determinados períodos de tempo (configurado pelo usuário).

O primeiro aspecto comportamental da ferramenta citado anteriormente se refere a um processo realizado através de um serviço executado em *background* que diariamente coleta as informações de interação social do usuário com seus contatos do dispositivo móvel, realiza a compilação dessas informações e as envia ao Servidor Web para armazenamento e futuro

processamento do índice de relevância social desses contatos para o usuário. Para cada novo contato encontrado por esse serviço, suas informações básicas são enviadas para o Servidor Web, juntamente com a compilação das informações de interação social entre esse contato e o usuário.

Portanto, na primeira execução pós-instalação dessa ferramenta (Cliente Móvel), é formada uma lista de contatos atuais do usuário no dispositivo móvel e as informações básicas desses contatos são enviadas para o Servidor Web, juntamente com a compilação das suas informações de interação social com o usuário. Desse momento em diante, cada alteração realizada na lista de contatos do dispositivo móvel do usuário é refletida em nossa ferramenta, desde simples alterações nos dados básicos dos contatos até a adição de novos contatos.

A exclusão de contatos na lista de contato do dispositivo móvel do usuário simplesmente desabilita esses contatos em nossa ferramenta. Esses contatos desabilitados são mantidos apenas por motivos históricos. Nenhum outro tipo de informação sobre esses contatos desabilitados são coletados, pois eles já não estão mais presentes na lista de contatos do dispositivo móvel do usuário. Já para os contatos habilitados, as informações de interação social com o usuário continuam sendo coletadas, compiladas e enviadas para o Servidor Web a cada nova execução desse serviço.

O Servidor Web então armazena os dados coletados do dispositivo móvel do usuário e analisa as informações contextuais da rede social desse usuário através das mídias sociais na Internet. As informações coletadas através das mídias sociais se dividem em dois tipos: básicas e de interação social. As informações básicas dos contatos do usuário são: nome, sexo e data de nascimento. As informações de interação social são: *tweets*, *likes*, compartilhamento de conteúdo, mensagens, comentários, citações, postagens, *tags* em fotos e vídeos, etc.

Conforme mencionado na Seção 5.2.1, a análise contextual da rede social do usuário tem o objetivo de inferir a distância social entre um contato e o usuário. Para isso, pesos diferentes são atribuídos a tipos de informações distintos no cálculo do índice de proximidade social dos contatos com o usuário, de acordo com a importância da forma de interação social utilizada. Na próxima seção, após tratarmos especificamente da utilização de nossa proposta de solução em um cenário de emergência, exemplificaremos o cálculo do fator de relevância social dos contatos com o usuário.

O Servidor Web coleta informações das mídias sociais do usuário periodicamente e pode realizar a análise contextual da rede social do usuário independentemente de qualquer ação externa. Portanto, o Servidor Web mantém sempre atualizadas as informações coletadas do dispositivo móvel do usuário, enviadas pelo Cliente Móvel, e as informações coletadas

diretamente das mídias sociais do usuário. Dessa forma, ele consegue calcular o índice de proximidade social de cada contato com o usuário e mantê-lo o mais atualizado possível.

A Figura 43 ilustra todo o fluxo de trabalho envolvendo o Cliente Móvel e o Servidor Web antes da ocorrência de uma emergência.

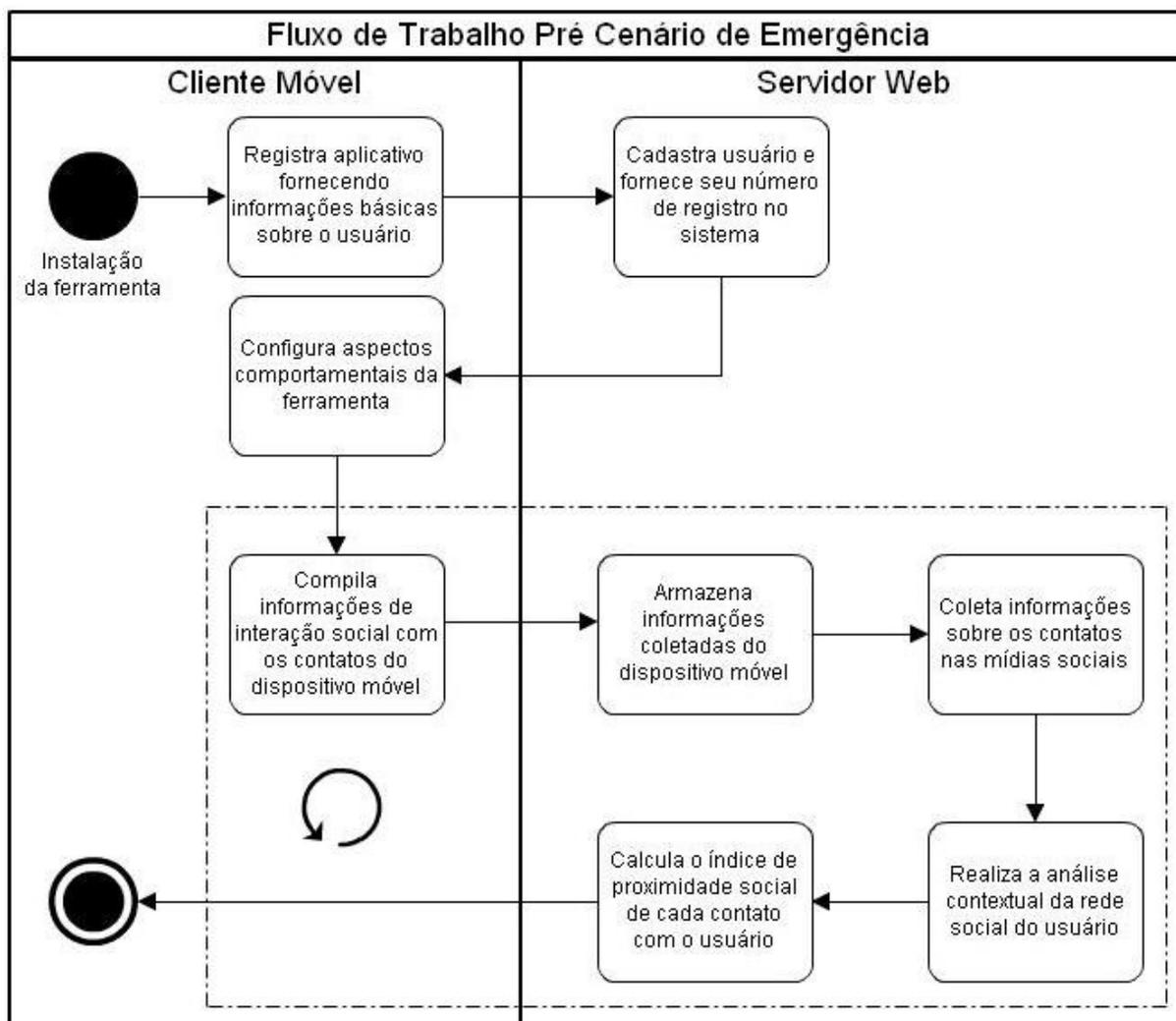


Figura 43. Fluxo de trabalho de nossa proposta de solução antes de um cenário de emergência

Os dois aspectos comportamentais restantes da ferramenta se referem aos processos realizados sob a demanda do Servidor Comando e Controle (equipe de resposta a emergências) durante a ocorrência de um cenário de emergência. O primeiro desses processos é uma requisição feita pela equipe de resposta a emergências para o usuário disponibilizar sua localização posicional (coordenadas GPS), visando o planejamento de rotas para o usuário chegar ao local do acidente, ou ainda uma tentativa de se localizar as vítimas do desastre.

O segundo desses processos é a requisição feita pela equipe de resposta a emergências na busca por informações sobre vítimas de um acidente. Esse processo é realizado durante a fase de resposta a uma emergência. O fluxo de trabalho desse processo se inicia quando a equipe de resposta a emergências identifica uma vítima que está desaparecida ou inconsciente,

seja através de observações realizadas diretamente pela equipe de Operações ou pelas informações passadas pelo Centro de Controle. De posse do nome dessa vítima desaparecida ou inconsciente, a equipe de resposta a emergências acionará o Servidor Comando e Controle, pesquisando pelo nome da vítima em nossa proposta de solução.

O Servidor Comando e Controle então dispara requisições para os Clientes Móveis dentro de um determinado raio de ação consultando a lista de contatos do dispositivo móvel desses usuários em busca de uma pessoa que possua o nome passado pela equipe de resposta. Esses Clientes Móveis então verificam a lista de contatos do usuário, pesquisando pelo nome requisitado pelo Servidor Comando e Controle. Cada Cliente Móvel, encontrando uma lista de contatos que satisfazem a consulta pelo nome solicitado, requisita ao Servidor Web algumas informações desses contatos, incluindo seus índices de proximidade social com o usuário.

O Servidor Web, tendo previamente realizado a análise contextual da rede social do usuário, retorna os índices de proximidade social desses contatos com o usuário, juntamente com as outras informações solicitadas. O Cliente Móvel então calcula o fator de relevância social de cada contato da lista em relação aos outros contatos dessa lista, tendo como base os índices de proximidade social desses contatos retornados pelo Servidor Web. Depois, o Cliente Móvel cria o *ranking* de contatos que possuem o nome pesquisado, ordenando essa lista de contatos em ordem decrescente de fator de relevância social, ou seja, do contato socialmente mais relevante até o contato socialmente menos relevante para o usuário.

Por fim, cada Cliente Móvel enviará para o Servidor Comando e Controle a sua lista de contatos ordenados segundo o *ranking* de fator de relevância social para aquele usuário, contendo as informações básicas de cada contato e algumas informações adicionais para auxiliar a equipe de resposta na identificação da possível vítima dentre aqueles contatos, evitando dessa forma o problema de pessoas homônimas. Assim, o Servidor Comando e Controle exibirá essas listagens de pessoas que satisfazem a pesquisa pelo nome realizada, conforme for recebendo as respostas de cada Cliente Móvel dentro do raio de ação.

Esse fluxo de trabalho de nossa proposta de solução para um cenário de emergência, descrito anteriormente, é ilustrado na Figura 44.

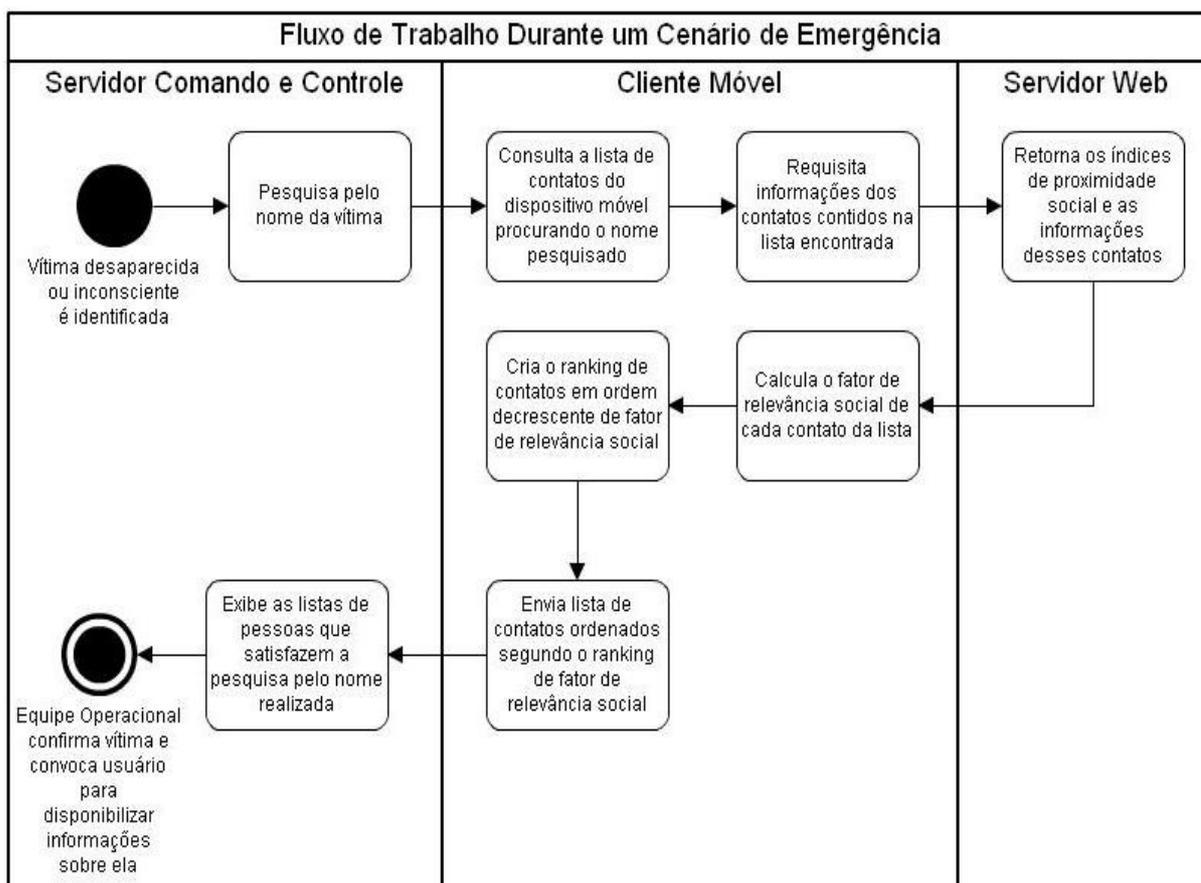


Figura 44. Fluxo de trabalho de nossa proposta de solução durante um cenário de emergência

Caso o Servidor Comando e Controle não receba nenhuma resposta dentro de um limite de tempo especificado pela equipe de resposta a emergências, configurado na própria ferramenta, ele realizará a mesma pesquisa através de uma requisição ao Servidor Web. Dessa vez, o Servidor Comando e Controle consultará o Servidor Web requisitando uma pesquisa pelo nome da vítima desaparecida ou inconsciente nas redes sociais de todos os usuários registrados no sistema.

O Servidor Web, de posse do nome enviado pela requisição do Servidor Comando e Controle, iniciará a pesquisa nas redes sociais de todos os usuários cadastrados em sua base de dados. Para cada usuário registrado, o Servidor Web deverá montar a lista de contatos desses usuários que satisfazem a consulta pelo nome pesquisado. Essa lista de contatos de cada usuário deverá ser ordenada segundo o *ranking* de relevância social para o usuário explicado anteriormente.

Por fim, o Servidor Web envia para o Servidor Comando e Controle todas essas listas de contatos por usuário que satisfazem a pesquisa pelo nome realizada. Novamente, essas listas de contatos possuem informações básicas de cada um desses contatos, além de informações que possam auxiliar a equipe de resposta a emergências na identificação da verdadeira vítima.

A partir daí, o Servidor Comando e Controle exibe a listagem de contatos que possuem o nome pesquisado por usuário para que a equipe de resposta a emergências possa identificar a vítima dentre esses contatos dos usuários. Uma vez identificada a verdadeira vítima dentre esses contatos, a equipe de resposta poderá então escolher um dos usuários (provavelmente aquele cujo contato — a vítima identificada — possui um maior fator de relevância social) para convocá-lo a contribuir com o trabalho da equipe de resposta, disponibilizando informações confiáveis sobre a vítima.

Esse fluxo de trabalho alternativo executado quando não há resposta dos Clientes Móveis próximos ao local da emergência é ilustrado na Figura 45.

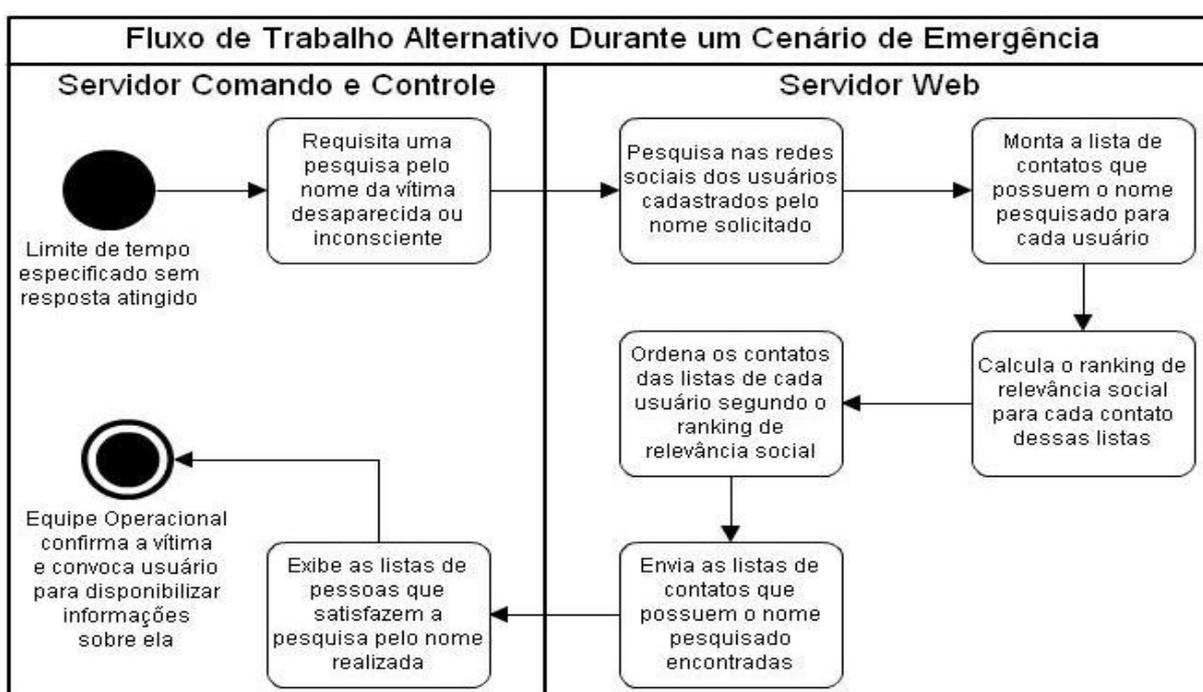


Figura 45. Fluxo de trabalho alternativo de nossa proposta de solução durante um cenário de emergência

6.3 Exemplo Prático Baseado no Protótipo Construído

Nesta seção iremos exemplificar o cálculo do índice de proximidade social de cada contato com o usuário, bem como a ordenação da lista de contatos segundo o *ranking* que considera o fator de relevância social calculado para cada contato, em relação aos demais contatos da lista. Para isso, iremos utilizar como base o protótipo construído durante o desenvolvimento deste trabalho e explicado na Seção 5.5.3.

Digamos que o nome pesquisado pela equipe de resposta a emergências (ou seja, o nome da vítima identificada) tenha sido “Paulo”, conforme é mostrado na Figura 46. O Servidor Comando e Controle então dispara requisições para os Clientes Móveis dentro de um determinado raio de ação, consultando a lista de contatos desses dispositivos móveis em busca de pessoas que contenham o nome pesquisado, no caso desse exemplo, “Paulo”.

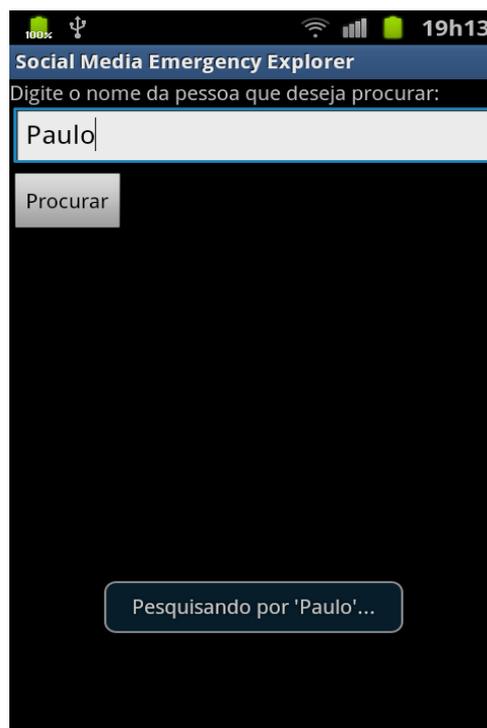


Figura 46. Tela de pesquisa do Servidor Comando e Controle

O Cliente Móvel, ao receber a requisição do Servidor Comando e Controle, começa a procurar na lista de contatos do dispositivo móvel do usuário pelos contatos que possuem aquele nome pesquisado. No nosso exemplo, encontramos os seguintes contatos utilizando-se o protótipo construído para este trabalho: “Paulo Garcia”, “Paulo Heckmaier”, “Paulo Machado”, “Paulo Rodrigo Carneiro” e “Paulo Sérgio Medeiros”. Para cada um desses contatos que correspondem à busca do nome realizada são verificadas as interações sociais realizadas através do dispositivo móvel.

No protótipo construído verificamos as seguintes informações de dispositivo móvel: quantidade de ligações (realizadas e recebidas), data da última ligação (realizada ou recebida), duração total de todas as ligações (realizadas e recebidas) e total de mensagens trocadas (SMS e MMS, enviados e recebidos). Além disso, todas essas informações de dispositivo móvel são coletadas para cada um dos números de telefones registrados do contato e compiladas em um único modelo de informações para o contato.

No caso do contato “Paulo Garcia” são identificadas zero ligações. Portanto o usuário nunca telefonou para esse contato de seu dispositivo móvel, muito menos recebeu um telefonema desse contato. Logo, as informações de data de última ligação e da duração total de todas as ligações também são iguais a zero para esse contato. Em relação à troca de mensagens SMS e MMS, também não foram identificadas nenhuma mensagem desse tipo, seja recebida ou enviada. Portanto, o total de mensagens trocadas via dispositivo móvel também é zero.

Para o contato “Paulo Heckmaier” são identificadas zero ligações. Portanto as informações de data de última ligação e da duração total de todas as ligações também são iguais a zero para esse contato. Em relação à troca de mensagens SMS e MMS, também não foram identificadas nenhuma mensagem desse tipo, seja recebida ou enviada. Portanto, o total de mensagens trocadas via dispositivo móvel também é zero.

Já o contato “Paulo Machado” possui 44 ligações, dentre realizadas e recebidas, totalizando aproximadamente 114,82 minutos de duração total dessas ligações. A data de última ligação para esse contato (ou seja, a ligação mais recente) é 19/07/2012 às 11 horas e 47 minutos. Em relação à troca de mensagens SMS e MMS, foram identificadas cinco mensagens desse tipo, dentre recebidas e enviadas.

O contato “Paulo Rodrigo Carneiro” também não possui ligações identificadas nesse dispositivo móvel. Portanto as informações de data de última ligação e da duração total de todas as ligações também são iguais a zero para esse contato. Em relação à troca de mensagens SMS e MMS, também não foram identificadas nenhuma mensagem desse tipo, seja recebida ou enviada.

Por fim, o contato “Paulo Sérgio Medeiros” não possui ligações identificadas nesse dispositivo móvel. As informações de data de última ligação e da duração total de todas as ligações também são iguais a zero para esse contato. Em relação à troca de mensagens SMS e MMS, também não foram identificadas nenhuma mensagem desse tipo, seja recebida ou enviada. Portanto, o total de mensagens trocadas via dispositivo móvel também é zero.

Em seguida, é hora de identificar e analisar as interações sociais entre o usuário e cada um desses contatos provenientes das mídias sociais. No caso de nosso protótipo, analisamos apenas as interações sociais provenientes do Facebook. Após compilar as informações de interação social via dispositivo móvel, o Cliente Móvel então identifica quais desses contatos possuem um perfil no Facebook. Nesse nosso exemplo, o protótipo identificou que todos esses contatos encontrados possuem um perfil no Facebook, através das informações encontradas no próprio dispositivo móvel do usuário.

Primeiramente, nosso protótipo identifica as informações básicas de cada um desses contatos no Facebook, como o sexo e a idade (através da data de nascimento que cada usuário preenche em seu perfil). No caso do “Paulo Garcia”, identificamos que é uma pessoa do sexo masculino e possui 31 anos de idade. Para o “Paulo Heckmaier”, é identificado o sexo masculino e a idade de 29 anos. Já o “Paulo Machado” também é do sexo masculino e possui 30 anos de idade. O “Paulo Rodrigo Carneiro” é do sexo masculino e possui 29 anos de idade.

E para o “Paulo Sérgio Medeiros” foi identificado o sexo masculino, mas sua idade não pôde ser calculada, pois ele não preencheu sua data de nascimento no perfil do Facebook.

Após a compilação dessas informações básicas dos contatos, retiradas da mídia social (Facebook), nosso protótipo passa então a analisar as informações de interação social entre o usuário e cada um desses contatos no Facebook. As informações analisadas pelo protótipo construído como prova de conceito para este trabalho são: número total de comentários (do contato nas postagens do usuário e vice versa), número total de *likes* (do contato nas postagens do usuário e vice versa), número total de citações em postagens (quantidade de postagens do usuário em que ele cita o contato e vice versa), número total de compartilhamento de conteúdos (quantidade de postagens que o usuário compartilha com o contato e vice versa) e número total de mensagens públicas diretas (quantidade de postagens feitas pelo usuário no perfil do contato e vice versa).

No caso do “Paulo Garcia” foram identificados um comentário, quatro *likes*, nenhuma citação em postagens, nenhum compartilhamento de conteúdo e uma mensagem direta. Para o “Paulo Heckmaier” foram identificados nenhum comentário, oito *likes*, nenhuma citação em postagens, nenhum compartilhamento de conteúdo e nenhuma mensagem direta. Já para o “Paulo Machado” foram identificados um comentário, um *like*, uma citação em postagens, um compartilhamento de conteúdo e uma mensagem direta. E para o “Paulo Rodrigo Carneiro” foram identificados dois comentários, 15 *likes*, nenhuma citação em postagens, nenhum compartilhamento de conteúdo e nenhuma mensagem direta. Finalmente, para o “Paulo Sérgio Medeiros” foram identificados nenhum comentário, nenhum *like*, nenhuma citação em postagens, nenhum compartilhamento de conteúdo e nenhuma mensagem direta.

O Quadro 6 mostra o comparativo dessas informações de interação social entre o usuário e cada um dos contatos encontrados.

Contato	Comentários	Likes	Citações	Compartilhamentos	Mensagens
Paulo Garcia	1	4	0	0	1
Paulo Heckmaier	0	8	0	0	0
Paulo Machado	1	1	1	1	1
Paulo Rodrigo Carneiro	2	15	0	0	0
Paulo Sérgio Medeiros	0	0	0	0	0

Quadro 6. Interação social do usuário com os contatos

Agora, com todas essas informações de interação social através do dispositivo móvel e do Facebook, nosso protótipo está pronto para calcular o índice de proximidade social dos contatos com o usuário. Esse índice é calculado através de uma média aritmética ponderada de cada um desses tipos de interações sociais identificados. Os pesos utilizados para cada tipo de interação social nessa média são: citações e *likes* (peso um); comentários e compartilhamento de conteúdo (peso dois); mensagens diretas (peso três); mensagens SMS/MMS (peso quatro) e total de ligações (peso cinco).

Depois de calculada essa média aritmética ponderada, é levada em consideração também no cálculo desse índice de proximidade social dos contatos com o usuário a duração total das ligações efetuadas entre eles (em minutos). Vamos então agora demonstrar o cálculo realizado pelo nosso protótipo para cada um desses contatos encontrados. No caso do “Paulo Garcia”, nosso cálculo fica:

$$I_{PG} = \left(1 * L + 2 * Com + 1 * Cit + 2 * Cont + 3 * M + 4 * Sms + 5 * T / 18\right) + D \quad (3)$$

$$I_{PG} = \left(1 * 4 + 2 * 1 + 1 * 0 + 2 * 0 + 3 * 1 + 4 * 0 + 5 * 0 / 18\right) + 0 \quad (3.1)$$

$$I_{PG} = 0,5 \quad (3.2)$$

Para o “Paulo Heckmaier” o cálculo fica:

$$I_{PH} = \left(1 * L + 2 * Com + 1 * Cit + 2 * Cont + 3 * M + 4 * Sms + 5 * T / 18\right) + D \quad (4)$$

$$I_{PH} = \left(1 * 8 + 2 * 0 + 1 * 0 + 2 * 0 + 3 * 0 + 4 * 0 + 5 * 0 / 18\right) + 0 \quad (4.1)$$

$$I_{PH} \cong 0,44 \quad (4.2)$$

Já para o “Paulo Machado” temos o seguinte cálculo:

$$I_{PM} = \left(1 * L + 2 * Com + 1 * Cit + 2 * Cont + 3 * M + 4 * Sms + 5 * T / 18\right) + D \quad (5)$$

$$I_{PM} = \left(1 * 1 + 2 * 1 + 1 * 1 + 2 * 1 + 3 * 1 + 4 * 5 + 5 * 44 / 18\right) + 114,82 \quad (5.1)$$

$$I_{PM} = 128,65 \quad (5.2)$$

E para o “Paulo Rodrigo Carneiro” o cálculo ficaria assim:

$$I_{PRC} = \left(1 * L + 2 * Com + 1 * Cit + 2 * Cont + 3 * M + 4 * Sms + 5 * T / 18\right) + D \quad (6)$$

$$I_{PRC} = \left(1 * 15 + 2 * 2 + 1 * 0 + 2 * 0 + 3 * 0 + 4 * 0 + 5 * 0 / 18\right) + 0 \quad (6.1)$$

$$I_{PRC} \cong 1,06 \quad (6.2)$$

Finalmente, para o “Paulo Sérgio Medeiros” temos como cálculo:

$$I_{PSM} = \left(1 * L + 2 * Com + 1 * Cit + 2 * Cont + 3 * M + 4 * Sms + 5 * T / 18\right) + D \quad (7)$$

$$I_{PSM} = \left(1*0 + 2*0 + 1*0 + 2*0 + 3*0 + 4*0 + 5*0 / 18\right) + 0 \quad (7.1)$$

$$I_{PSM} = 0 \quad (7.2)$$

A partir desse índice de proximidade social dos contatos com o usuário é calculado o fator de relevância social desses contatos para o usuário. Esse fator de relevância social é calculado para cada contato em relação a todos os contatos dessa lista de contatos que possuem o nome pesquisado anteriormente (“Paulo”). Então, para obtermos esse fator, precisamos primeiramente calcular o somatório de todos esses índices de proximidade social dos contatos com o usuário. No caso de nosso exemplo, o cálculo ficaria:

$$TI = I_{PG} + I_{PH} + I_{PM} + I_{PRC} + I_{PSM} \quad (8)$$

$$TI = 0,5 + 0,44 + 128,25 + 1,06 + 0 \quad (8.1)$$

$$TI = 130,25 \quad (8.2)$$

Agora, podemos calcular esse fator de relevância social dos contatos para o usuário. No caso do “Paulo Garcia”, nosso cálculo fica:

$$F_{PG} = I_{PG} / TI \quad (9)$$

$$F_{PG} = 0,5 / 130,25 \quad (9.1)$$

$$F_{PG} \cong 0,0038 \quad (9.2)$$

Para o “Paulo Heckmaier” o cálculo fica:

$$F_{PH} = I_{PH} / TI \quad (9)$$

$$F_{PH} = 0,44 / 130,25 \quad (9.1)$$

$$F_{PH} \cong 0,0034 \quad (9.2)$$

Já para o “Paulo Machado” temos o seguinte cálculo:

$$F_{PM} = I_{PM} / TI \quad (10)$$

$$F_{PM} = 128,25 / 130,25 \quad (10.1)$$

$$F_{PM} \cong 0,9847 \quad (10.2)$$

E para o “Paulo Rodrigo Carneiro” o cálculo ficaria assim:

$$F_{PRC} = I_{PRC} / TI \quad (11)$$

$$F_{PRC} = 1,06 / 130,25 \quad (11.1)$$

$$F_{PRC} \cong 0,0081 \quad (11.2)$$

Finalmente, para o “Paulo Sérgio Medeiros” temos como cálculo:

$$F_{PSM} = I_{PSM} / TI \quad (12)$$

$$F_{PSM} = 0 / 130,25 \quad (12.1)$$

$$F_{PSM} = 0 \quad (12.2)$$

Portanto, o fator de relevância social é um número no intervalo [0, 1] que indica a importância daquele contato para o usuário, em relação aos demais contatos da lista. Quanto mais próximo de um o fator é, mais importante para o usuário aquele contato se mostra. Com todos esses dados calculados, podemos finalmente ordenar essa lista de contatos para criar o *ranking* de contatos socialmente relevantes para o usuário. Esse *ranking* é criado ordenando-se a lista de contatos em ordem decrescente do fator de relevância social.

Dessa forma, tudo o que resta agora para o Cliente Móvel é retornar para o Servidor Comando e Controle essa listagem de contatos ordenados de acordo com o *ranking* de relevância social. Essa listagem exibe informações sobre os contatos a fim de auxiliar a equipe de resposta a emergências na confirmação da vítima, procurando fornecer subsídios à equipe de Operações para evitar o problema de pessoas homônimas. Em nosso protótipo exibimos essa listagem de contatos, conforme pode ser visto na Figura 47.



Figura 47. Exemplo de resposta de um Cliente Móvel

Com o desenvolvimento computacional dessa heurística e das demais funcionalidades anteriormente expostas, a ferramenta aqui proposta pode ser considerada suficientemente pronta para a execução de avaliações mais profundas da proposta de solução deste trabalho. Entretanto, tal experimentação deve ser planejada cautelosamente, a fim de se evitar colocar em risco operações reais ou, no outro extremo, torná-la superficial.

O simples caso de uso da ferramenta abordado neste capítulo pode ser analisado sob diversos ângulos para que indícios e conclusões mais sólidos possam ser inferidos. Porém, a mera reflexão sobre tal ilustração não é garantia para a obtenção de interpretações fundamentadas para definirem a continuidade do trabalho. Por esta razão, é imprescindível que estudos exploratórios sejam realizados considerando-se o contexto de realização desta pesquisa, o que pode apontar, inclusive, mudanças na abordagem adotada.

Na próxima seção detalharemos como deverão ser realizadas as avaliações da heurística de recomendação. Também mostraremos o planejamento dos testes futuros a serem realizados visando à obtenção de resultados mais sólidos, capazes de indicar a comprovação da hipótese deste trabalho e realizar uma análise geral da solução proposta.

6.4 Avaliação das Recomendações

Durante a criação do protótipo, diversos testes unitários foram realizados em alguns dispositivos móveis para ajustar nossa heurística de recomendação, visando melhorar todo o processo descrito no presente trabalho. Esses testes unitários, apesar de não poderem ser considerados testes conclusivos para a hipótese deste trabalho, apresentaram bons resultados na recomendação, conforme era o esperado para os cenários testados. Eles também serviram para guiar a formalização de um processo de teste mais rigoroso, visando à experimentação de nosso protótipo a fim de podermos testar a hipótese apresentada neste trabalho.

Tais testes unitários foram realizados da seguinte maneira: ao usuário do dispositivo móvel onde nosso protótipo estava instalado era solicitada uma listagem de nomes comuns (por exemplo, Carlos, Eduardo, Luís, Paulo ou Rodrigo) de pessoas que ele conhecesse. Para cada pessoa sugerida pelo próprio usuário, o nome era pesquisado em nosso protótipo. Após o processamento dos contatos do usuário pela heurística apresentada neste trabalho, o protótipo apresentava a listagem de contatos do dispositivo móvel, ordenados segundo o *ranking* de relevância social. O usuário era então confrontado com essa listagem e deveria informar se tal *ranking* condizia com a realidade, ou seja, se a ordem das pessoas homônimas apresentadas pelo protótipo expressava realmente quais eram socialmente mais próximas a ele.

Na maioria dos casos testados, os usuários reportaram que o *ranking* apresentado pelo protótipo condizia com a realidade experimentada por ele. A pessoa em primeiro lugar da listagem apresentada pelo protótipo era, dentre as demais, aquela que mais possuía intimidade com o usuário e, no caso de alguma emergência, poderia prestar informações sobre ele ou, pelo menos, indicar pessoas mais próximas ao usuário (como pais, irmãos ou familiares) que poderiam ajudar em algum caso onde fossem necessárias informações mais precisas.

A única diferenciação observada nesses casos foi que nem sempre a listagem era 100% correta. Apesar do primeiro lugar da lista ser a pessoa que, sem dúvidas, possuía uma maior intimidade com o usuário que as demais, nas colocações seguintes foram encontradas alguns disparates em relação à realidade do usuário. Por exemplo, uma pessoa que aparecia na segunda colocação do *ranking* para um usuário, na realidade não era para ele assim tão mais íntima que o terceiro ou o quarto colocado. Essas inversões nas outras colocações do *ranking* não puderam ser investigadas mais a fundo, porém acreditamos que com algumas medidas extras na heurística apresentada neste trabalho essas inexatidões possam ser consertadas.

Com a execução desses testes unitários já foi possível identificar alguns pontos fortes e fracos da ferramenta construída. Os ajustes na heurística identificados durante essa fase de testes unitários, bem como as melhorias gerais no processo e na arquitetura do *framework* apresentado neste trabalho, serão discutidos no Capítulo 7. Na próxima seção apresentaremos um guia de testes a serem realizados futuramente.

6.5 Experimentações Futuras

A realização de experimentos científicos que envolvam tomadas de decisão rápidas e sob pressão não é trivial. Em qualquer domínio de aplicação que seja, os riscos oriundos da utilização de um novo método e/ou ferramenta em situações reais são considerados muito altos pelos próprios participantes. Se tratando de casos que envolvam vidas humanas, como é o cenário de emergências, a experimentação em situações reais é praticamente impossível.

Por este motivo, o mais recomendado para o domínio de aplicação deste trabalho é que se inicie a validação das teorias aqui apresentadas por meio de eventos controlados em laboratório. Essas simulações podem não ser capazes de gerar a carga emocional experimentada na realidade, mas, por outro lado, permitem uma observação mais minuciosa das variáveis estudadas.

Além disso, para se avaliar métodos de recomendação de pessoas é importante que os indivíduos envolvidos acompanhem em seu raciocínio todas as etapas dos cálculos e os resultados obtidos. Somente assim eles poderão fazer uma avaliação mais criteriosa e fornecer

valiosos pareceres para se realizarem ajustes ou melhorias nesse processo. Por essa razão, a redução da pressão e a disponibilidade de maior tempo na simulação em laboratório podem ser fundamentais para refinar a heurística proposta nesta dissertação antes de utilizá-la em experimentos de campo.

Dessa forma, a estratégia de validação traçada para este trabalho consiste na realização de experimentos diversos direcionados aos dois perfis de usuários de nossa ferramenta:

- Usuários Comuns (cidadãos): os experimentos envolvendo esse grupo de usuários servirão para validar e aprimorar a heurística de recomendação de pessoas socialmente próximas a eles;
- Usuários Avançados (equipe operacional de emergência): os experimentos envolvendo esse grupo de usuários servirão para validar e aprimorar a utilidade da ferramenta proposta para as organizações de gestão de emergências.

Para o primeiro grupo de usuários (cidadãos) devem-se utilizar experimentos simulatórios baseados na identificação das pessoas socialmente mais próximas a cada um deles. Após a conclusão dessa primeira etapa de validações, o segundo grupo de usuários (equipe operacional de emergência) seria acionado para um caso de estudo exploratório com a execução de uma simulação baseada em uma ocorrência real, sem interferência do pesquisador durante seu desenrolar e com coleta de dados qualitativa por meio de questionários. A seguir descreveremos alguns cenários de testes para cada um desses grupos, identificando também as situações limites de recomendação.

6.6 Plano de Testes

Nosso plano de testes é dividido em alguns cenários de testes específicos para cada grupo de usuários de nossa ferramenta: os cidadãos comuns e a equipe operacional de emergência. O objetivo principal desses estudos de casos é realizar uma avaliação da heurística de recomendação de pessoas apresentada neste trabalho, com o intuito de se obterem indícios sobre sua exatidão e as perspectivas de contribuição das pessoas recomendadas para a disponibilização de informações sobre o usuário que possam auxiliar o processo de tomada de decisão pela equipe operacional em situações de emergências. Os cenários que visam tal objetivo são aqueles orientados aos usuários comuns (cidadãos).

De forma secundária, pretende-se também obter orientações sobre o emprego da tecnologia proposta e da ferramenta desenvolvida para que se possam realizar os ajustes necessários para alcançar uma melhor aderência ao trabalho real desempenhado pelas equipes de operação em emergências. Desse modo, pretendemos avaliar a utilidade de nossa

ferramenta em cenários simulados de situações de emergência, com a presença de oficiais e membros de organizações de gestão de emergências. Portanto, esses cenários que verificarão a utilidade da ferramenta construída são orientados aos usuários avançados (equipe operacional de emergência).

Para tanto, é preciso identificar previamente as variáveis a serem observadas e aquelas a serem manipuladas durante as experimentações de cada grupo. As primeiras são referentes ao funcionamento da heurística, sua correção e precisão, e à perspectiva de capacidade dos indicados para disponibilizar informações de qualidade sobre o usuário. Esse grupo, que visa ao cumprimento do objetivo principal desses estudos de casos, apresenta variáveis qualitativas e nominais que permitem extrair os pontos fortes e as fraquezas sob a perspectiva dos usuários comuns. O Quadro 7 apresenta tais variáveis, explicando o significado adotado para elas.

Variável	Significado
Correção	A ordenação do <i>ranking</i> de contatos recomendados, em relação à proximidade social com o usuário, é a mesma percebida por ele na vida real?
Precisão	O fator de relevância social dos contatos recomendados é o mesmo percebido pelo usuário na vida real?
Contribuição	As pessoas recomendadas são capazes de disponibilizarem informações sobre o usuário?

Quadro 7. Variáveis relativas à heurística de recomendação de pessoas

Para se avaliar a variável de correção da heurística, os usuários devem levar em consideração somente a ordenação do *ranking* de contatos que é exibido pela ferramenta. Os usuários devem identificar se a ordem em que os contatos aparecem nessa listagem realmente traduz uma possível ordenação daquelas pessoas de acordo com seus níveis de importância ou proximidade social para ele. Por exemplo, dado que foi pesquisado na ferramenta pelo nome “Carlos” e obtido como resposta uma listagem com os contatos “Carlos Eduardo”, “Carlos Alberto” e “Carlos Lima”, nessa mesma ordem, o usuário deverá indicar se, realmente, o “Carlos Eduardo” é mais íntimo para ele do que o “Carlos Alberto”, que por sua vez é socialmente mais próximo a ele do que o “Carlos Lima”.

Já ao avaliar a variável de precisão da heurística, os usuários devem levar em consideração puramente os fatores de relevância social que são mostrados para cada contato

recomendado pela ferramenta. Então, eles deverão analisar se o índice de relevância social do contato, que é medido em percentual referente aos outros contatos da mesma lista, é de fato próximo ao que é percebido pelo usuário na vida real. A medição dessa variável se faz necessária para avaliarmos se a proximidade social daqueles contatos em relação ao usuário é confiável o suficiente para que o *ranking* exibido pela ferramenta seja o mais preciso possível.

Note que o fato de o *ranking* exibido pela ferramenta estar de acordo com o percebido pelo usuário na vida real (correção) não necessariamente implica em uma precisão nos fatores de relevância social dos contatos. Aproveitando o exemplo anteriormente dado, levemos em consideração os seguintes índices de relevância social dos contatos recomendados pela ferramenta: “Carlos Eduardo – 57%”, “Carlos Alberto – 29%” e “Carlos Lima – 14%”.

Ao avaliar esses números, um usuário poderia informar que a diferença do fator de relevância social entre os dois primeiros contatos, conforme exibido pela ferramenta, não é tão grande assim conforme o percebido por ele na vida real. Ou seja, a diferença de importância entre o “Carlos Eduardo” e o “Carlos Alberto” para o usuário na vida real pode ser menor, digamos 48% e 43%, respectivamente, enquanto que, conseqüentemente, a importância do “Carlos Lima” para o usuário cairia para 9%.

Dessa forma, teríamos que na vida real o “Carlos Eduardo” e o “Carlos Alberto” apresentariam níveis de distância social com o usuário bem próximos, enquanto que o “Carlos Lima” teria uma distância social pouco maior que a apurada pela ferramenta. Devemos destacar também que, em casos limite, a precisão pode afetar a correção da heurística. Por exemplo, se a ferramenta auferir fatores de relevância social bem próximos para dois contatos, como “João Carlos – 7%” e “José Carlos – 5%”. Em uma avaliação do usuário poderia constatar-se que, na realidade, o “João Carlos” apresenta um nível de importância para o usuário de 6%, enquanto que o “José Carlos” apresenta um nível de importância para o usuário de 8%, invertendo-se assim suas posições no *ranking* apresentado pela ferramenta.

Por fim, para se avaliar a variável de contribuição das pessoas recomendadas pela ferramenta, o usuário deverá ser capaz de aferir se os contatos que aparecem no *ranking* são realmente capazes de disponibilizar informações sobre ele que possam auxiliar a equipe operacional de emergência em um cenário onde ele esteja envolvido. Principalmente, deve ser indicado se o contato socialmente mais próximo ao usuário, conforme averiguado pela ferramenta, é capaz de fornecer tais informações.

A avaliação mínima desejável por este estudo nesse quesito é que a maioria dos contatos indicados pela ferramenta possa ao menos indicar outra pessoa mais próxima ao usuário (tais como cônjuge, pais ou irmãos) para fornecer informações de qualidade que

possam auxiliar a equipe de emergência em uma possível situação de resgate envolvendo o usuário. Dessa forma, esperamos que a solução proposta sirva como, no mínimo, um bom ponto de partida para a obtenção de informações sobre o usuário.

Para o cumprimento do objetivo secundário deste trabalho, faz-se necessário obter indícios sobre o balanço entre a dificuldade de aprendizado e de uso da ferramenta e os benefícios proporcionados pelo método e pela tecnologia propostos, considerando-se sua adoção no trabalho cotidiano. Para tanto, devem ser utilizadas as variáveis descritas no Quadro 8.

Variável	Significado
Formato de Saída	Divisão da informação em listas aninhadas.
Volume de Saída	Quantidade de informação exibida.
Facilidade de Entendimento	Esforço necessário para entender as informações disponibilizadas.
Desempenho	Velocidade com a qual as informações são exibidas.
Utilidade	Balanço entre o custo de utilização da ferramenta e a utilidade considerada para conclusão da atividade.
Relevância	Coerência entre a real necessidade do usuário e o auxílio providenciado pela ferramenta.
Facilidade de Uso	Esforço necessário para se utilizar a ferramenta.
Facilidade de Aprendizado	Esforço necessário para aprender a manipular a ferramenta.

Quadro 8. Variáveis de aderência da solução proposta ao trabalho real

Deve-se ter em mente também a influência de algumas outras variáveis que podem, inclusive, serem manipuladas com o intuito de se obterem novos indícios. Neste trabalho, essas correspondem à duração e velocidade de execução da simulação, à possíveis cenários adversos encontrados e à qualidade das informações adicionais, tanto as providas pela ferramenta quanto aquelas que serão disponibilizadas pelas pessoas recomendadas, que podem interferir na utilização da solução proposta.

O próprio evento simulado pode ser planejado de forma a destacar algumas características do processamento da heurística elaborada. Assim, por meio da introdução de condições como acessos a determinadas ruas fechadas, condições climáticas adversas, malha

de transportes alternativos, e outras, pode-se facilitar a percepção de determinados aspectos do método como um todo, conforme apresentado no Capítulo 5.

Existe ainda outro conjunto de variáveis extrínsecas à pesquisa, mas que mesmo assim podem ser consideradas na avaliação dos resultados. Essas estão associadas à experiência dos oficiais participantes, sua familiaridade com tecnologias de ponta, motivação e expectativas para a participação no experimento. O plano de realização dos experimentos é elaborado de forma a se ter o controle sobre a maior parte dessas variáveis, manipuláveis ou extrínsecas.

6.6.1 Cenários de Teste para o Grupo dos Usuários Comuns

O objetivo principal dos cenários de teste desenvolvidos para esse grupo de usuários é avaliar o método de recomendação de pessoas, baseado em informações contextuais presentes nos dispositivos móveis e nas redes sociais, exposto no presente trabalho. Ao mesmo tempo, faz-se necessário coletar e analisar indícios que comprovem a hipótese apresentada por este estudo. Portanto, tal grupo de usuários deverá também fornecer o *feedback* necessário para avaliar a hipótese deste trabalho, durante a execução desses cenários de teste.

O primeiro cenário imaginado visando alcançar tais objetivos seria a execução de testes individualizados utilizando a ferramenta descrita neste trabalho, instalada nos dispositivos móveis desse grupo de usuários. Para a execução dos testes, um grupo de usuários poderia ser selecionado completamente ao acaso, já que nesse primeiro momento estamos preocupados somente com a avaliação individual dos usuários sobre o método de recomendação de pessoas e o *feedback* deles sobre a capacidade das pessoas recomendadas de disponibilizarem informações a respeito do usuário.

Dessa forma, primeiramente, cada usuário selecionado para esse cenário de teste seria arguido a citar três nomes comuns à sua rede social que ele considerasse relevante (somente o primeiro nome, por exemplo, “José”). De posse desses nomes, o usuário seria então instruído de elencar esses contatos homônimos de acordo com a ordem de proximidade social desses contatos com ele, dos contatos mais próximos aos socialmente mais distantes. A ferramenta então seria instalada no dispositivo móvel do usuário, para realizar a análise contextual das informações dos contatos.

Terminada a preparação para esse cenário de teste, os nomes citados pelo usuário seriam pesquisados na ferramenta, um de cada vez. Cada resultado de recomendação oferecido pela ferramenta seria então confrontado com a listagem original feita pelo usuário. Após analisarem-se ambas as listagens, o usuário deverá indicar se a recomendação fornecida pela ferramenta procede, de acordo com o que é percebido por ele na vida real. Tanto a

ordenação da lista de recomendações quanto o fator de relevância social dos contatos auferido pela ferramenta deverão ser analisados.

Por fim, o usuário também deverá ponderar sobre a capacidade dos contatos recomendados pela ferramenta em disponibilizar informações sobre ele, capazes de auxiliar a equipe de operação em alguma situação de emergência hipotética que o usuário esteja envolvido. Principalmente, deverá ser analisado se a primeira pessoa recomendada é capaz de fornecer informações de qualidade sobre o usuário. Em segundo lugar, pretende-se avaliar se as pessoas recomendadas são capazes de, no mínimo, indicarem outras pessoas mais capacitadas para disponibilizarem informações sobre o usuário.

Já o segundo cenário de teste para este grupo de usuários visa avaliar a heurística de recomendação de pessoas e a hipótese apresentadas neste estudo de uma maneira geral. Para tanto, faz-se necessária a participação de um grupo de pessoas que possuam uma rede social em comum. Não necessariamente todos os participantes desse experimento deverão ser de uma comunidade específica, porém é recomendado que possam ser identificadas redes sociais em comum em determinados grupos de pessoas. Por exemplo, os participantes desse experimento poderão ser selecionados dentre os alunos de uma Faculdade. Mas dentre eles, grupos de alunos de diferentes períodos seriam escolhidos, configurando-se assim distintas redes sociais dentre os participantes.

Uma vez formado esse grupo de pessoas que participarão do experimento, a ferramenta proposta neste trabalho seria instalada nos dispositivos móveis dos participantes. Para este experimento é necessário que todos os participantes estejam presentes, pois a intenção agora é avaliar se a ferramenta proposta consegue identificar, dentre um grupo de pessoas, qual delas possui uma maior proximidade social com alguém em específico (no caso de nossa hipótese, a vítima da emergência). Portanto, a cada rodada de teste, um dos participantes deverá ser escolhido para representar o papel da vítima de uma situação de emergência hipotética.

Após ser escolhido quem representará o papel da vítima da emergência, o nome da vítima é então pesquisado nos dispositivos móveis dos demais participantes do experimento. Os resultados obtidos em cada dispositivo móvel deverão ser analisados comparativamente, visando descobrir qual dos participantes a ferramenta indica como sendo aquele socialmente mais próximo à vítima. Uma vez obtido o resultado final, que pode ser uma listagem dos participantes ordenada de acordo com a indicação da ferramenta de suas proximidades sociais com a vítima, o participante escolhido para representar o papel da vítima deverá averiguar se esse resultado condiz com a realidade.

Portanto, ao analisar o resultado final do teste, o participante deverá indicar se as pessoas recomendadas pela ferramenta são realmente capazes de disponibilizarem informações sobre ele. Por fim, vale ressaltar que ambos os cenários de teste descritos nesta seção foram desenvolvidos visando obter-se uma avaliação qualitativa imparcial sobre a heurística de recomendação e a hipótese apresentadas neste trabalho. Para a obtenção dos resultados das avaliações desse grupo de usuários poderão ser utilizados questionários que procurem extrair dos usuários o máximo possível de informações referentes às análises da heurística de recomendação e da hipótese deste estudo.

6.6.2 Cenário de Teste para o Grupo dos Usuários Avançados

O objetivo do cenário de teste desenvolvido para esse grupo de usuários é avaliar a utilidade da ferramenta apresentada neste trabalho em um cenário de emergência. Para tanto, faz-se necessária a presença de oficiais e comandantes de organizações de gestão de emergências. Como a experimentação de novas tecnologias e ferramentas em situações de emergência é praticamente inviável, devido aos riscos envolvidos, inclusive pelo fato de as pessoas envolvidas no processo estarem expondo suas vidas ao perigo, optou-se por uma simulação em laboratório baseada em cenários reais.

Os participantes da simulação poderão ser escolhidos aleatoriamente, uma vez que todos deverão ser treinados na manipulação da ferramenta. Porém, pode ser desejável que alguns dos participantes, principalmente os membros da equipe de operação em emergências, possuam certa familiaridade com a tecnologia de dispositivos móveis a ser empregada durante o curso do experimento. Após a escolha dos participantes, todos deverão passar por um processo de ambientação com a tecnologia e a ferramenta proposta no presente trabalho.

Uma vez finalizado esse período de ambientação, onde os participantes serão treinados na tecnologia e na ferramenta propostas por este estudo, o cenário da simulação deverá ser explicado. Ainda, os objetivos deste experimento deverão ser explicitados, bem como a forma de avaliação da ferramenta pelos próprios usuários. Durante a simulação do cenário escolhido, os participantes deverão seguir livremente o curso normal de seus procedimentos durante a fase de resposta a uma emergência. O mínimo de intervenção possível por parte dos pesquisadores é sugerido.

É importante também levar em consideração durante a simulação do cenário de emergência determinados aspectos que poderiam configurar situações limítrofes à utilização da ferramenta em um cenário de emergência real. Para avaliar a ferramenta em seu aspecto mais geral, englobando as funcionalidades de auxílio à logística da equipe de operação em

emergências, podem-se adicionar desafios adicionais ao cenário de emergência, como, por exemplo, o bloqueio de determinada rua de acesso ao local de emergência, ou o fato de não haver pessoa próxima ao local de emergência que conheça a vítima encontrada.

Ao final da simulação do cenário, os participantes deverão analisar o uso da ferramenta e deixarem suas impressões sobre os pontos fracos e os pontos fortes exibidos pela ferramenta. O *feedback* desse grupo de usuários será importante na avaliação geral da ferramenta e também poderá fornecer novos parâmetros para guiar o desenvolvimento de melhorias a serem aplicadas no avançar desta pesquisa. A avaliação dos usuários deverá ser realizada através de questionários que visem extrair o máximo de informações possíveis sobre a utilização da ferramenta e seu comportamento.

Apesar de até o momento da conclusão deste trabalho não termos realizado nenhum experimento conclusivo sobre o processo de recomendação e a ferramenta apresentados, alguns pontos fracos e fortes já puderam ser identificados no decorrer de nossa pesquisa. No capítulo seguinte iremos enumerar as principais fraquezas encontradas, já sugerindo melhorias na abordagem adotada, além de clarificar as maiores virtudes deste trabalho.

Capítulo 7 - Conclusões e Trabalhos Futuros

Para que uma pesquisa científica obtenha sucesso, não basta apenas encontrar um bom problema e seguir um rigoroso método durante sua realização, mas também se devem realizar análises críticas periodicamente a fim de corrigir seu direcionamento e garantir sua continuidade. Justamente por esses motivos, este capítulo visa realizar uma análise retrospectiva de todo o fluxo de trabalho da presente pesquisa, identificando seus erros e acertos, seus pontos fortes e fraquezas, os problemas encontrados e as limitações descobertas para, finalmente, determinar os ajustes que forem julgados necessários no método de execução dessa pesquisa e de seus próximos passos.

O presente trabalho teve como objetivo inicial investigar as informações contextuais presentes nos dispositivos móveis tão amplamente difundidos em nossa sociedade atual e também nas mídias sociais, que hoje em dia usufruem de alta popularidade entre a população mundial, como ferramentas capazes de melhorar o processo de recomendação de pessoas através do cálculo da distância social e física entre elas. As informações avaliadas são aquelas diretamente ligadas ao contexto do problema. Por exemplo, informações sobre as interações sociais entre os indivíduos, onde as pessoas se encontram, etc. Nesse sentido, os dispositivos móveis possuem informações que carregamos conosco o tempo inteiro e que podem auxiliar na representação do contexto onde estamos inseridos: localização posicional (coordenadas GPS), mensagens (SMS ou MMS), últimas ligações (realizadas ou recebidas), e-mails, etc.

Os dispositivos móveis podem auxiliar na colaboração e na troca de dados, assistindo na captura e no provimento de informações contextuais e servindo como forma de acesso à rede social do indivíduo. Essa rede social encontrada nos dispositivos móveis, aliada às informações contextuais também presentes neles, são valiosos construtos para a melhoria no processo de recomendação de pessoas. A vantagem de utilizar um sistema móvel está na transposição das dificuldades da coleta e uso dessas informações contextuais em uma situação onde o tempo é um fator crucial para o sucesso de uma operação: os celulares possuem tecnologias como o Bluetooth e 3G, que são capazes de obter e transmitir facilmente informações em questões de segundos, no máximo minutos.

O domínio de aplicação de nossa proposta de solução foi a área de gestão de emergências, visando atacar a questão crítica que é a ausência de informações confiáveis e íntegras sobre as vítimas durante a fase de resposta de uma emergência. Tais informações poderiam partir de pessoas que possuíssem algum convívio com a vítima e, portanto, para a

obtenção dessas informações precisamos abordar o problema de detecção e recomendação de pessoas baseados em seus perfis sociais, para colaborar na resolução de um determinado cenário-problema. O problema de detecção e recomendação de pessoas nesse caso não dependeu somente de perfis sociais, que se enquadravam como objetos mais estáticos nessa análise, mas também do contexto onde essa rede social estava inserida.

A questão contextual dependeu de objetos mais dinâmicos que estavam inseridos nessa análise, tais como onde estão localizadas as pessoas participantes dessa rede social, quando ocorreu a última interação entre essas pessoas e qual a intensidade dessa interação, além de ingredientes contextuais específicos, assumidos durante determinados instantes, e que representam atividades usuais ou informações sobre o ambiente em torno do qual está inserida a questão, como, por exemplo, qual seria a melhor trajetória para uma pessoa que prestará as informações chegar até o local da emergência (e nesse caso devemos analisar o trânsito e, por conseguinte, a condição climática dos locais de origem e destino, pois esta afeta diretamente o tráfego entre esses pontos).

A partir desses entendimentos foram elaboradas as premissas e a hipótese do trabalho, que objetivaram a criação de um processo de recomendação de pessoas que pudessem disponibilizar informações de qualidade sobre vítimas de uma emergência que estão desaparecidas ou inconscientes, auxiliando dessa maneira a equipe de resposta a emergências no procedimento de resgate dessas vítimas. É importante frisar que, no pior dos casos, espera-se que a pessoa recomendada seja, no mínimo, um bom ponto de partida para a recuperação de informações sobre a vítima, servindo como um meio de acesso às informações necessárias.

Para isso, esse processo de recomendação seria impulsionado pela análise contextual da rede social da vítima, identificando assim a distância social entre a vítima e uma pessoa a partir de suas interações nas mídias sociais e dispositivos móveis. Além disso, utilizaríamos também as informações contextuais sobre a emergência, a vítima e sua rede social, a fim de melhorar a recomendação de pessoas para auxiliar em um cenário de emergência.

A gestão de emergências foi escolhida como domínio de aplicação para esta pesquisa por se tratar de uma das áreas mais representativas do objeto de estudo. A alta necessidade de obtenção de informações confiáveis e íntegras sobre as vítimas de uma emergência que estão desaparecidas ou inconscientes foi o fator principal que nos levou a pensar em um processo de recomendação de pessoas impulsionado pelas informações contextuais e as interações sociais nas mídias sociais e dispositivos móveis. As dificuldades comumente encontradas pela equipe de resposta a emergências em um cenário de emergência também foram levadas em

consideração durante a escolha do domínio de aplicação e o conseqüente desenvolvimento do presente trabalho ao analisar o objeto de estudo.

Dado o pouco tempo para a tomada de decisão e a quantidade de informações a serem coletadas e analisadas pelos integrantes da equipe de resposta a emergências, tal mecanismo implícito de busca e seleção assume grande importância no procedimento de resgate a essas vítimas. Esta pesquisa objetivou, então, determinar uma forma analítica para realizar tal filtro de pessoas de forma que somente aquelas mais aptas a disponibilizar informações úteis sobre a vítima fossem recomendadas. E foi então que vinculamos a ideia da análise contextual da rede social da vítima, utilizando também elementos contextuais característicos de localização posicional, à melhoria no processo de recomendação de pessoas para disponibilizar informações úteis e de qualidade sobre as vítimas de uma emergência.

A heurística de utilidade elaborada foi então embutida na ferramenta móvel construída como prova de conceito, desenvolvida para ser utilizada em dispositivos móveis que possuem a plataforma Android. Essa ferramenta então analisa as informações contextuais contidas nos dispositivos móveis dos usuários e nas mídias sociais, verificando a interação social desses usuários com a vítima, a fim de recomendar os usuários mais socialmente próximos a ela.

Por último, realizamos uma ilustração de uso de nossa ferramenta, simulando as atividades da equipe de resposta a emergências durante um cenário de emergência. O caso de uso escolhido foi inspirado em um desastre real, para se demonstrar o uso de nossa proposta de solução e a contribuição da ferramenta elaborada neste trabalho. Essa escolha de uma ilustração de uso de nossa ferramenta foi devido às limitações impostas por um experimento mais real, devido ao grau de inovação da proposta e do prazo para a conclusão deste trabalho. Ainda assim foi possível identificar as fragilidades do modelo, algumas expectativas sobre suas funcionalidades e as diretrizes para o aprimoramento dessa pesquisa.

Apesar de não conseguirmos realizar nenhum experimento conclusivo com a ferramenta construída até o encerramento deste trabalho, já deixamos aqui apresentado no Capítulo 6 o plano de testes a serem efetuados posteriormente, com o avanço desta pesquisa. Tal plano encontra-se estruturado de maneira a guiar os experimentos planejados durante o desenvolvimento de nossa proposta de solução, detalhando os cenários de teste desenvolvidos, mostrando seus objetivos e explicando quais variáveis deverão ser analisadas no decorrer dos experimentos. Também são sugeridos cenários limites para avaliar melhor as restrições e limitações de nossa proposta de solução.

7.1 Contribuições

A principal contribuição do presente trabalho é a proposta de recomendação de pessoas por meio de critérios analíticos das interações sociais com o intuito de se fornecer informações íntegras e confiáveis para uma equipe de resposta a emergências no momento necessário ao resgate de uma vítima desaparecida ou inconsciente. Para isso, analisamos as informações contextuais contidas nos dispositivos móveis e nas mídias sociais para impulsionar o processo de recomendação de pessoas, a fim de indicar as pessoas mais aptas a disponibilizar informações sobre essa vítima, ou seja, aquelas mais socialmente próximas à vítima e que tenham condições de chegar rapidamente ao local da emergência.

Em relação à utilização de informações contextuais para melhorar processos de recomendações mais gerais, encontramos no trabalho de Yap, Tan e Pang (2005) fortes críticas às abordagens tradicionais de sistemas de recomendação, que ao não levarem em conta as informações situacionais, limitam seriamente a relevância dos resultados recomendados. Eles defendem a percepção contextual como uma abordagem promissora para melhorar o desempenho de sistemas recomendadores e introduzem um mecanismo para realizar esse tipo de abordagem.

Muito se fala na literatura atual sobre a utilização de informações contextuais em processos de recomendação. Dessa forma, baseado em pesquisas nessa área, decidimos desenvolver esse nosso modelo proposto voltado ao cenário de emergência, e acreditamos que a utilização de informações contextuais podem realmente melhorar o processo de recomendação. Nosso modelo de recomendação proposto utiliza as informações contextuais coletadas diretamente dos dispositivos móveis e das mídias sociais, realizando assim uma análise contextual da rede social dos usuários.

Portanto, podemos considerar como contribuição desta dissertação a estruturação da etapa inicial da resolução de problemas de falta de informações confiáveis e íntegras sobre vítimas de uma emergência que estejam desaparecidas ou inconscientes. Consequentemente, temos o aprimoramento desse processo de recomendação de pessoas utilizando-se informações contextuais por meio do processamento computacional possibilitado com nossa proposta de solução.

Isso significa que a atividade de busca e captura de informações sobre vítimas de acidentes, atualmente dependente em grande parte da tarefa realizada manualmente pelos indivíduos da equipe de resposta a emergências, pode ser auxiliada com artefatos tecnológicos

altamente difundidos em nossa sociedade que facilitam a indicação de pessoas capazes de fornecer tais informações, permitindo assim uma tomada de decisões mais acertada e rápida.

Assim sendo, implementamos em um protótipo nosso modelo de recomendação de pessoas utilizando as informações contextuais encontradas em dispositivos móveis e mídias sociais. Essa ferramenta construída foi capaz de analisar as informações contextuais encontradas nos dispositivos móveis dos usuários e, juntamente com as informações contextuais provenientes das mídias sociais, conseguiu criar um *ranking* de relevância social entre aquelas pessoas e os usuários.

Apesar de não termos conseguido avaliar essa ferramenta formalmente, através de experimentos controlados ou reais, algumas simulações foram realizadas no protótipo construído para verificar o processo de recomendação construído. Embora não tenha sido possível confirmar a utilidade dessa ferramenta em um cenário de emergência real, seu comportamento durante as simulações ficou dentro do objetivo esperado como resultado desse modelo de recomendação proposto.

Por fim, nossa proposta de um modelo genérico baseado em contexto reforça o processo de recomendação existente orientado a uma tarefa específica, a saber, a coleta de informações sobre as vítimas de uma emergência que estão desaparecidas ou inconscientes. Esse modelo também leva em consideração o fato de o estado de certos contextos poderem afetar a importância de outros contextos durante um processo de tomada de decisão.

7.2 Problemas Encontrados e Limitações

A maior parte dos problemas encontrados e possíveis limitações da ferramenta dizem respeito a condições técnicas das plataformas utilizadas na construção do protótipo. No caso do Android SDK, para se coletar as interações sociais realizadas via troca de mensagens SMS e MMS tivemos que acessar diretamente um provedor de conteúdo não documentado. O problema com esse tipo de acesso é que ele é característico de algumas versões da plataforma Android, e não há garantias da manutenção dessa forma de acesso em futuras versões.

Já para coletarmos as interações sociais via e-mails, deveríamos consultar provedores de conteúdos privados, dependentes da cada aplicativo de e-mail existente para a plataforma Android. A estrutura altamente versátil da plataforma Android conta em nosso favor nesse caso, pois ela permite interconectar os mais diversos tipos de aplicativos no sistema Android. O mesmo também é válido para a análise das informações contextuais provenientes de outras fontes ou tipos de aplicativos como *chats*, mensageiros eletrônicos, ligações VoIP, etc.

Para cada aplicativo cujos dados devem ser analisados, será necessário interagir com seu provedor de conteúdo próprio, caso seja disponibilizado o acesso a esses dados via um provedor de conteúdo público. A plataforma Android apenas disponibiliza uma forma de as aplicações trocarem informações entre si, mas não pode forçar nenhuma delas a fazer isso, ou prover acesso não permitido a essas informações.

Ao longo dessa dissertação, também citamos os problemas e limitações encontrados com o Facebook SDK. Alguns deles fogem ao nosso alcance, como a observada incapacidade de se recuperar algumas postagens realizadas ou a limitação dos recursos de sua API, que por vezes durante algumas simulações acusaram erro no processamento da requisição. Mas até mesmo para essa última é possível contornarmos o problema com um projeto de arquitetura e execução dos serviços adequados. O problema com a paginação nativa do Facebook API citado anteriormente também pode ser contornado utilizando-se certos padrões de projetos e com a arquitetura da solução proposta.

Considerando a construção do protótipo, certos problemas encontrados devem-se meramente a limitações impostas pelo desempenho desejado na ferramenta proposta como solução final. Ao combinarmos a limitação nativa do Facebook SDK utilizando a ferramenta FQL para consultar as informações do usuário no Facebook e o fato de o protótipo buscar essas informações em tempo real, constatou-se que a coleta de determinadas informações como *likes* ou citações em comentários seria praticamente inviável de ser implementada por conta do desempenho desejado para o protótipo.

Porém, tecnicamente nada impede que essas ou outras informações sejam coletadas e analisadas seguindo os padrões de projeto e arquitetura definidos nesta proposta de solução. A proposta descrita por este trabalho já inclui o tratamento desses problemas e limitações identificados durante a construção do protótipo como prova de conceito de nossa pesquisa.

Algumas limitações que não poderiam ficar de fora da análise desse capítulo são aquelas provenientes das questões de privacidade, segurança e dependência da infraestrutura de comunicação. Em relação às questões de privacidade e segurança, Wagner et al. (2010) mostra que nas redes sociais os participantes são muito mais diligentes e cuidadosos sobre o compartilhamento de informações sobre outras pessoas.

Os usuários novatos podem ser insensíveis à privacidade devido à falta de compreensão de como a informação é revelada. No entanto, posteriormente eles reconhecem a importância de controlar a disponibilidade dessas informações através de mecanismos como desabilitação do serviço. A utilidade de serviços de compartilhamento de informações foi

reconhecida em situações mais estressantes que envolvem cenários de crise e de segurança em geral. Em tais situações, a utilidade de informações supera a preocupação com a privacidade.

Nós estamos conscientes das implicações dessa pesquisa em relação à privacidade dos usuários. Sabemos que pelo fato da nossa solução precisar coletar informações sobre os contatos do usuário, suas chamadas, as mensagens trocadas, dentre outras, é importante abordarmos esta questão de segurança e privacidade. Inicialmente, o contrato para utilizar a solução conterà um aviso dizendo que usaremos os dados coletados apenas em situações de emergência, e que somente a equipe de resposta a emergências terá acesso a esses dados. Além disso, o sistema inclui opções para I) fornecer os dados automaticamente (sempre fornecer os dados), II) fornecer os dados apenas em determinados períodos de tempo (configurado pelo usuário) e III) fornecer os dados manualmente (solicitando permissão para o usuário antes de fornecer esses dados).

Quanto à dependência da infraestrutura de comunicação, a arquitetura e o padrão de projeto para a construção dessa solução proposta preveem as possíveis falhas durante a comunicação com os serviços externos. Justamente por causa dessa dependência de serviços externos para a coleta e utilização de informações em nossa proposta de solução foi que decidimos executar a coleta e análise dessas informações em uma atividade executada periodicamente em segundo plano, inclusive criando-se um *cache* de informações para quando esses serviços externos se mostrarem indisponíveis.

Em relação à utilização de fontes de dados externas para o contexto do usuário, nós tratamos apenas da aquisição dessas informações de contextualização. No tocante às questões da confiabilidade, segurança e privacidade associadas a essas informações, apesar de serem assuntos altamente interessantes, estão fora do escopo deste trabalho e devem ser tratadas internamente pelas próprias fontes de dados utilizadas.

Por fim, uma importante limitação deste trabalho foi não podermos realizar uma avaliação de nosso protótipo em uma simulação envolvendo os principais agentes de uma organização de resposta a emergências. Essa limitação de uma avaliação mais profunda de nosso protótipo deveu-se a certa complexidade de preparação de uma simulação deste tipo, aliada ao prazo de conclusão do presente trabalho. Simulações mais simples, realizadas em um ambiente totalmente controlável poderiam ter sido realizadas. Porém, seus resultados não garantiriam a validade de nossa proposta de solução, uma vez que num ambiente desse tipo fatalmente haveria uma rede social muito comum entre os indivíduos envolvidos.

7.3 Trabalhos Futuros

O aspecto mais importante dessa pesquisa é que, portanto, apresenta um grande potencial de evolução é o detalhamento da heurística que calcula a distância social entre as pessoas de uma rede social — aqui chamado de índice de proximidade social. A versão apresentada neste trabalho é uma equação simples, que leva em consideração diferentes tipos de interações sociais com graus de importância distintos (média aritmética ponderada). Para o protótipo construído, apenas algumas dessas interações sociais identificadas e planejadas durante a pesquisa foram implementadas.

Portanto, a continuidade dessa pesquisa indica que devemos desenvolver uma versão mais completa da solução proposta, conforme o planejamento e as diretrizes iniciais dados por este trabalho, de forma que as mais distintas maneiras de interação social consigam ser coletadas e computadas nessa heurística, inclusive usando as diversas fontes de informações citadas no decorrer desta dissertação (nas mídias sociais e nos dispositivos móveis). Também nessa nova versão da solução proposta poderíamos implementar a avaliação das informações contextuais relativas à distância física, para adicionarmos as funcionalidades de localização de pessoas e planejamento de rotas de acesso dessas pessoas ao local de emergência.

Outra possível extensão de nossa proposta de solução seria avaliar informações que permitissem identificar profissionais especialistas em determinadas áreas, que estivessem próximos ao local do acidente, para auxiliar a equipe de resposta a emergências durante suas atividades no cenário de emergência. Ainda, utilizando o histórico de localizações do usuário através do Google Latitude juntamente com as informações do dia e hora do acidente, seria possível então traçar a rotina de uma pessoa para sabermos a probabilidade de ela estar no local da emergência.

Visando melhorar o tempo de resposta de nossa solução, seria interessante criar no Cliente Móvel um *cache* dos índices de proximidade social dos contatos calculados pelo Servidor Web. Dessa forma, o problema citado de dependência da infraestrutura de comunicação seria contornado, pois caso durante um cenário de emergência não seja possível estabelecer comunicação entre o Cliente Móvel e o Servidor Web, os índices de proximidade social poderiam ser coletados diretamente desse *cache*.

Ainda em relação à heurística de cálculo da distância social inicialmente proposta neste trabalho, cabe ressaltar que a janela de tempo das informações coletadas e utilizadas na análise contextual da rede social dos usuários é desde o marco zero. Isso significa que utilizamos todas as informações contextuais contidas nos dispositivos móveis e mídias

sociais, desde o momento em que o usuário começou a usá-los até o presente, sem fazer distinção em relação ao período em que essas informações foram coletadas.

Isso pode não refletir precisamente na qualidade das informações disponibilizadas pela pessoa indicada por nossa proposta de solução. Por exemplo, um usuário pode ter um longo histórico de contato com a pessoa procurada pela equipe de resposta a emergências, mas em algum momento de sua vida ele perdeu contato com essa pessoa, não sabendo mais onde ela mora, se está casada, etc. Já outro usuário, que não possui um histórico de contato com essa pessoa tão extenso quanto o usuário anterior, possui contato frequente e mais atual com ela.

Na heurística atual de cálculo da distância social, fatalmente o primeiro usuário seria o mais indicado pela nossa proposta de solução, por possuir um longo histórico de contato com a pessoa procurada pela equipe de resposta a emergências, apesar de suas informações sobre a vítima não serem as mais atuais possíveis. Para se corrigir esse problema, será necessário atribuir pesos distintos às informações contextuais coletadas de acordo com o período em que elas foram coletadas, sendo que informações mais antigas receberiam pesos menores, enquanto as informações mais recentes receberiam pesos maiores.

Além dessa diferenciação de pesos em relação à antiguidade da informação coletada, também poderia ser criado um mecanismo que altera os pesos das variáveis relativas às formas de comunicação utilizadas nessa heurística de acordo com a preferência de comunicação do usuário. Por exemplo, esse mecanismo poderia identificar que um usuário utiliza mais as redes sociais como forma de comunicação do que o e-mail. Portanto, para esse usuário, a heurística de cálculo da distância social seria ajustada para se dar um maior peso às mensagens trocadas via rede social do que àquelas trocadas via e-mail.

Por fim, ainda há espaço também para se explorar o problema dos homônimos que apenas introduzimos neste trabalho, sem considerá-lo o foco do objeto de estudo. Ainda, após a evolução do protótipo da ferramenta de solução inicialmente construído, pode-se eventualmente realizar avaliações mais criteriosas deste trabalho, inclusive conduzindo experimentos controlados em laboratório visando à melhoria na heurística de análise contextual da rede social dos usuários, ou ainda simulações de cenários de emergência envolvendo os principais agentes de organizações de resposta a emergências, para aferir a viabilidade de uso da ferramenta proposta por este trabalho em cenários de emergência reais.

Os futuros passos desta pesquisa não se limitam apenas às sugestões aqui comentadas. Elas podem ser consideradas como as mais imediatas para o aperfeiçoamento da solução, da heurística e da ferramenta. Os problemas discutidos na seção anterior também apontam melhorias a serem perseguidas no próprio método de trabalho e que podem ser estudadas a

parte, pois constituem problemas comuns a outros âmbitos. As características do problema estudado e do domínio de aplicação demonstram que são duas áreas repletas de oportunidades para a pesquisa científica, principalmente para a área de tecnologia.

Referências

ADLER, C. et al. IT-Supported management of mass casualty incidents: the e-triage project. In: ISCRAM 2011 – INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS FOR CRISIS RESPONSE AND MANAGEMENT , 8. 2011, Lisbon. **Proceedings ...** Lisbon: LNEC/SCRAM, 2011.

ADOMAVICIUS, G. ; TUZHILIN, A. Extending recommender systems: a multidimensional approach. In: IJCAI-01 - THE INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 2001, Seattle, ITWP 2001 - WORKSHOP ON INTELLIGENT TECHNIQUES FOR WEB PERSONALIZATION, 2001. Seattle. **Proceedings...** Menlo Park, CA: AAAI, 2001. p. 4-6.

_____. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. **IEEE Transactions on Knowledge And Data Engineering**, New York, v. 17, n. 6, p. 734-749. June, 2005.

ANDROID. **Android**. 2008. Disponível em: <http://www.android.com/>. Acesso em: out. 2012.

_____. **API Guides**. 2012. Disponível em: <http://developer.android.com/guide/components/index.html>. Acesso em: jul. 2012.

_____. **Exploring the SDK**. 2012. Disponível em: <http://developer.android.com/sdk/index.html>. Acesso em: jul. 2012.

ANKOLEKAR, A. et al. Friendlee: a mobile application for your social life. In: MobileHCI '09 - INTERNATIONAL CONFERENCE ON HUMAN-COMPUTER INTERACTION WITH MOBILE DEVICES AND SERVICES, 11., 2009, Bonn. **Proceedings...** New York, ACM, 2009.

ARNO, C. Global social media. **SEJ. Search Engine Journal**. 2012. Disponível em: <http://www.searchenginejournal.com/global-social-media/46506/>. Acesso em: agos. 2012.

BALABANOVIC, M. ; SHOHAM, Y. Fab: content-based, collaborative recommendation. **Communications of the ACM**, New York, v. 40, n. 3, p. 66-72, Mar. 1997.

BALDAUF, M. ; DUSTDAR, S. ; ROSENBERG, F. A Survey on context-aware systems. **International Journal of Ad Hoc and Ubiquitous Computing**, [S.l.],v. 2, n. 4, p. 263-277, Jun. 2007.

BARKHUUS, L. Privacy in location-based services, concern vs. coolness. In: Mobile HCI 2004 WORKSHOP: LOCATION SYSTEM PRIVACY AND CONTROL, 2004, Glasgow, UK, **Proceedings...** Glasgow, UK: University of Strathclyde, 2004.

BELL, R. ; KOREN, Y. ; VOLINSKY, C. Matrix factorization techniques for recommender systems. **IEEE Computer**, New York, v. 42, n. 8, p. 30-37, Aug. 2009.

BELLO, P. G. et al. Improving communication for mobile devices in disaster response. In: LÖFFLER, J. ; KLANN, M. (Ed.). **Mobile response**. First International Workshop on Mobile Information Technology for Emergency Response , response Mobile 2007. - Sankt Augustin, Germany. Berlin: Springer, 2007. p. 126-134. (Lecture Notes in Computer Science, v. 4458). ISBN 978-3-540-75667-5.

BILLSUS, D. ; PAZZANI, M. J. Learning collaborative information filters. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 15., 1998. Madison, Wisconsin. **Proceedings ...** San Francisco: Morgan Kaufmann, CA, 1998. p. 46-54.

BRASIL. Secretaria Nacional de Defesa Civil. 2012. Disponível em: <http://www.defesacivil.gov.br/>. Acesso em: out. 2012.

_____. Secretaria Nacional de Defesa Civil. 2012. **Histórico da Defesa Civil**. Disponível em <http://www.integracao.gov.br/historico-sedec>. Acesso em dez.2012.

_____. **Glossário de defesa civil, estudos de riscos e medicina de desastres**. 5ª Ed. Brasília: Secretaria Nacional de Defesa Civil / Ministério da Integração Nacional, 2008. 283 p. volume único. Disponível em: <http://www.defesacivil.gov.br/publicacoes/publicacoes/glossario.asp>. Acesso em: fev. 2012.

_____. **Manual de planejamento em defesa civil**, v. 1, 1999. Disponível em: <http://www.defesacivil.gov.br/publicacoes/publicacoes/planejamento.asp>. Acesso em: mar. 2012.

_____. **Política nacional de defesa civil**, 2007. Disponível em: <http://www.defesacivil.gov.br/publicacoes/publicacoes/pndc.asp>. Acesso em: mar. 2012.

BREESE, J. S. ; HECKERMAN, D. ; KADIE, C. Empirical analysis of predictive algorithms for collaborative filtering. In: CONFERENCE ON UNCERTAINTY IN ARTIFICIAL INTELLIGENCE, 14., 1998, Madison, Wisconsin. **Proceedings ...** San Francisco: Morgan Kaufmann, CA, 1998.

BRÉZILLON, P. Context in problem solving: a survey. **The Knowledge Engineering Review**, Cambridge, Eng.: v. 14, n. 1, p. 47-80, May 1999.

_____. Using context for supporting users efficiently. In: HICSS'03 ANNUAL HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, 36., 2003, Waikoloa, Hawaii. **Proceedings ...** Washington DC: IEEE, 2003. v.5, Track 5, p. 127.3.

BRIDGE, D. et al. Case-based recommender systems. **The Knowledge Engineering Review**, Cambridge, Eng.: v. 20, n. 3, p. 315-320, Sept. 2005.

BULLAS, J. 48 Significant social media facts, figures and statistics plus 7 infographics. **jeffbullas.com**. 2012. Disponível em: <http://www.jeffbullas.com/2012/04/23/48-significant-social-media-facts-figures-and-statistics-plus-7-infographics/>. Acesso em: agos. 2012.

BURKE, R. et al. Segment-based injection attacks against collaborative filtering recommender systems. In: ICDM '05 - IEEE INTERNATIONAL CONFERENCE ON

DATA MINING, 5., 2005. Houston, TX. **Proceedings ...**, Washington, DC: IEEE, 2005. p. 577-580.

BURKE, R. Hibrid web recommender systems In: BRUSILOVSKI, P. ; KOBASA, A. ; NEJDL, W. (Eds). **The Adaptive web**. methods and strategies of web personalization. Berlin: Springer-Verlag, 2007. p. 377-408, (Lecture Notes in Computer State-of-the Art Surveys, .v. 4321).

CAMPBELL, B. D. et al. Emergency response planning and training through interactive simulation and visualization with decision support. In: IEEE 2008 INTERNATIONAL CONFERENCE ON TECHNOLOGIES FOR HOMELAND SECURITY, 2008. Waltham, MA, **Proceedings ...** Waltham, MA: IEEE, 2008. p. 176-180.

CBMERJ. Corpo de Bombeiros Militar do Estado do Rio de Janeiro. Museu Histórico do CBMERJ. Disponível em: <http://www.museu.cbmerj.rj.gov.br/>. Acesso em dez. 2012.

CHADE, J. Governo brasileiro admite à ONU despreparo em tragédias. **O Estado de São Paulo**. 2011. Disponível em: <http://www.estadao.com.br/noticias/cidades,governo-brasileiro-admite-a-onu-despreparo-em-tragedias,666689,0.htm>. Acesso em: ago. 2012.

CHEVERST, K. et al. Developing a context-aware electronic tourist guide: some issues and experiences. In: SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 200, The Hague, Netherlands: **Proceedings ...** New York: ACM, 2000. p. 17-24.

CHEVERST, K. et al. Using context as a crystal ball: rewards and pitfalls. **Personal and Ubiquitous Computing**, London, v. 5, n. 1, p. 8-11, Feb. 2001.

CHOUDHURY, T. ; PENTLAND, A. The Sociometer: a wearable device for understanding human networks. In CSCW '02 WORKSHOP: AD HOC COMMUNICATIONS AND COLLABORATION IN UBIQUITOUS COMPUTING ENVIRONMENTS, 2002, New Orleans. **Proceedings ...** New York: ACM, 2002.

CLAYPOOL, M. ; GOKHALE, A. ; MIRANDA, T. Combining content-based and collaborative filters in an online newspaper. In SIGIR-99 WORKSHOP ON RECOMMENDER SYSTEMS: ALGORITHMS AND EVALUATION, 1999, Berkeley, CA. **Proceedings ...** New York: ACM, 1999.

CLIMATEMPO. **Climatempo**. 1997. Disponível em: <http://www.climatempo.com.br/>. Acesso em: jun. 2012.

_____. **Climatempo API**. 2009. Disponível em: <http://www.climatempo.com.br/tempo-no-seu-site>. Acesso em: jun. 2012.

COTTER, P. ; SMYTH, B. PTV: intelligent personalized TV guides. In: CONFERENCE ON INNOVATIVE APPLICATIONS OF ARTIFICIAL INTELLIGENCE, 12., 2000, Austin, TX. **Proceedings...** Menlo Park, CA: AAAI, 2000. p. 957-964.

CRISIS. **Crisis' search**. 2005. Disponível em: <http://www.crisissearch.com/>. Acesso em: nov. 2011.

DE PESSEMIER, T. ; DERYCKERE, T. ; MARTENS, L. Context aware recommendations for user-generated content on a social network site. In: EUROPEAN CONFERENCE ON EUROPEAN INTERACTIVE TELEVISION CONFERENCE, 7., 2009, Leuven, Belgium. **Proceedings ...** New York: ACM, 2009. p. 133-136.

DE VOS, H. ; HOFTE, G. H. ; DE POOT, H. IM [@Work]: adoption of instant messaging in a knowledge worker organisation. In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, 37., 2004. Big Island, **Proceedings ...** Los Alamitos, CA: IEEE, 2004. v. 1. p. 10019, 2004.

DEVELOPERS. **Android developers**. 2012. Android platform versions. Disponível em: <http://developer.android.com/about/dashboards/index.html>. Acesso em: jul. 2012.

DEY, A. K. Understanding and using context. **Personal and Ubiquitous Computing**, London, v. 5, n. 1, p. 4-7, Feb. 2001.

DEY, A. K. ; ABOWD, G. D. ; SALBER, D. A Conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. **Human-Computer Interaction**, Hillsdale, NJ, v. 16, n. 2, p. 97-166, Dec. 2001.

DHA/ONU. **Internationally agreed glossary of basic terms related to disaster management**. Geneva Switzerland: Department of Humanitarian Affairs / United Nations, 1992. Disponível em <http://reliefweb.int/node/21195>. Acesso em: fev. 2012.

DI MAIO, P. An open ontology for open source emergency response system. In: **Open Source Research Community**, 2007.

DRABEK, T. E. **The Social dimensions of disaster** -FEMA Emergency Management Higher Education Project College Course Instructor Guide. Emmitsburg, MD: Emergency Management Institute, 1996.

EAGLE, N. ; PENTLAND, A. Social network computing. In: DEY, A. K. ; SCHIMIDT, A. ; CCARTHY, J. F. (Eds). **UbiComp 2003: Ubiquitous computing**. 5th International Conference. Seattle, WA, Oct. 2003. Berlin: Springer-Verlag, 2003. p. 289-296. (Lecture Notes in Computer Science, v. 2864).

_____. **Social serendipity: proximity sensing and cueing**. Cambridge, MA: MIT, 2004. (MIT Media Laboratory Technical Note 580, Oct. 2004report. 2004).

EIIF. Emergency Information Interoperability Frameworks Incubator Group (EIIF XG). **Emergency information interoperability frameworks**. 2009. Disponível em: <http://www.w3.org/2005/Incubator/eiif/XGR-Framework-20090806/>. Acesso em: dez. 2011.

FACEBOOK. **Facebook**. 2004. Disponível em: <http://www.facebook.com/>. Acesso em: nov. 2011.

FACEBOOK DEVELOPERS. 2012. Disponível em: <https://developers.facebook.com/>. Acesso em: out. 2012.

FACEBOOK DEVELOPERS. **Facebook developers docs**. 2012. Disponível em: <https://developers.facebook.com/docs/>. Acesso em: jul. 2012.

_____. Advanced topics - **Facebook query language (FQL)**. 2012. Disponível em: <http://developers.facebook.com/docs/reference/fql/>. Acesso em: jul. 2012.

_____. Blog do desenvolvedor. **How-to: paging with the graph API and FQL**. 2011. Disponível em: <http://developers.facebook.com/blog/post/478/>. Acesso em: jul. 2012.

_____. **Open graph concepts**. 2012. Disponível em: <https://developers.facebook.com/docs/opengraph/>. Acesso em: jul. 2012.

FACEBOOK NEWSROOM. **Key facts**. 2012. Disponível em: <http://newsroom.fb.com/content/default.aspx?NewsAreaId=22>. Acesso em: out. 2012.

FARAZMAND, A. Introduction - crisis and emergency management. In: _____. (Ed.). **Handbook of crisis and emergency management**. New York: CRC Press, 2001.

FEMA. **Federal Emergency Management Agency**. 2012. Disponível em: <http://www.fema.gov/>. Acesso em: out. 2012.

_____. **Emergency management guide for business and industry**. 1993. Disponível em: <http://www.fema.gov/business/guide/index.shtm>. Acesso em: mar. 2012.

_____. **Glossary/Acronyms**. 2012. Disponível em: <http://www.fema.gov/emergency/nrf/glossary.htm>. Acesso em: mar. 2012.

_____. **IS-1 Emergency manager: an orientation to the position**. 2007. Disponível em: <http://training.fema.gov/EMIWeb/is/is1.asp>. Acesso em: out. 2012.

FERREIRA, A. F. E. **Um modelo de apoio à percepção situacional na resposta a emergências**. 2011. 202 f. Dissertação (Mestrado em Informática) – Instituto de Matemática, Instituto Tércio Pacitti, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2011.

FISHER, D. ; DOURISH, P. Social and temporal structures in everyday collaboration. In: SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2004. Vienna. **Proceedings ...** New York: ACM, 2004. p. 551-558.

FOURSQUARE. **About foursquare**. 2009. Disponível em: <https://foursquare.com/about/>. Acesso em: jul. 2012.

G1. Globo.com. 2010. Disponível em: <http://g1.globo.com/Noticias/Rio/0,,MUL1561566-5606,00-HA+AO+MENOS+SOTERRADOS+NO+MORRO+DO+BUMBA+DIZ+SUBSECRETARI+O+DE+DEFESA+CIVIL.html>. Acesso em: dez. 2011.

GARTNER. **Gartner says worldwide smartphone sales soared in fourth quarter of 2011 with 47 percent growth**. Feb. 2012. Disponível em: <http://www.gartner.com/it/page.jsp?id=1924314>. Acesso em: jun. 2012.

GARTON, L. ; HAYTHORNTHWAITE, C. ; WELLMAN, B. Studying online social networks. **Journal of Computer Mediated Communication**, [S.l.], v. 3, n. 1, Jun. 1997.

GOOGLE. **Google**. 1998. Disponível em: <https://www.google.com/intl/pt-BR/about/>. Acesso em: out. 2012.

_____. **Google latitude**. 2009. Disponível em: <https://www.google.com/latitude/>. Acesso em: jun. 2012.

_____. **Google latitude API**. 2010. Disponível em: <https://developers.google.com/latitude/>. Acesso em: jun. 2012.

_____. **Google maps**. 2005. Disponível em: <https://maps.google.com/>. Acesso em: jun. 2012.

_____. **Google maps API**. 2005. Disponível em: <https://developers.google.com/maps/>. Acesso em: jun. 2012.

_____. **Google person finder**. 2010. Disponível em: <http://person-finder.appspot.com/> (aplicativos desativados) ou <http://googlepersonfinder.appspot.com/> (aplicativos atuais). Acesso em: nov. 2011.

_____. _____. 2010. Disponível em: <http://www.google.org/personfinder/global/howitworks>. Acesso em: nov. 2011.

_____. **Google search**. 1997. Disponível em: <http://www.google.com.br/>. Acesso em: nov. 2011.

_____. **Google weather**. 2008. Disponível em: http://www.google.com/ig/directory?type=gadgets&url=www.google.com/ig/modules/built_in_weather.xml. Acesso em: jun. 2012.

GOLBECK, J. Generating predictive movie recommendations from trust in social networks. In: STOLEN, K. et al. **Trust management**. Proceedings of the 4th International Conference on iTrust 2006, Pisa, Italy. Berlin: Springer-Verlag, 2006, p. 93-104. (Lecture Notes in Computer Science, v. 3986).

GOLDBERG, D. et al. Using collaborative filtering to weave an information tapestry. **Communications of the ACM**, New York, v. 35, n. 12, p. 61-70, Dec. 1992.

GRANOVETTER, M. S. The strength of weak ties. **The American Journal of Sociology**, Chicago, v. 78, n. 6, p. 1360-1380, May 1973.

GROH, G. ; EHMIG, C. Recommendations in taste related domains: collaborative filtering vs. social filtering. In: INTERNATIONAL ACM CONFERENCE ON SUPPORTING GROUP WORK, 2007, Sanibel Island, **Proceedings ...** New York: ACM, 2007. p. 127-136.

GUY, I. ; CARMEL, D. Social recommender systems. In: INTERNATIONAL CONFERENCE COMPANION ON WORLD WIDE WEB, 20., 2011, Hiderabad, India. **Proceedings ...** New York: ACM, 2011. p. 283-284.

GUY, I. ; RONEN, I. ; WILCOX, E. Do you know? recommending people to invite into your social network. In: INTERNATIONAL CONFERENCE ON INTELLIGENT USER INTERFACES, 14., 2009, Sanibel Island. **Proceedings ...** New York: ACM, 2009. p. 77-86.

GUY, I. et al. Personalized recommendation of social software items based on social relations. In: ACM CONFERENCE ON RECOMMENDER SYSTEMS, 3., 2009. New York. **Proceedings ...** New York: ACM, 2009. p. 53-60.

GUY, I. et al. Social media recommendation based on people and tags. In: INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 33., 2010, Geneva. **Proceedings ...** New York: ACM, 2010. p. 194-201.

HÄKKILÄ, J. A. S. et al. Context-aware mobile media and social networks. In: MobileHCI '09 - INTERNATIONAL CONFERENCE ON HUMAN-COMPUTER INTERACTION WITH MOBILE DEVICES AND SERVICES, 11., 2009, Bonn. **Proceedings ...** New York: ACM- MobileHCI '09, p. 1, 2009.

HARPALE, A. S. ; YANG, Y. Personalized active learning for collaborative filtering. In: SIGIR '08 - ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 31., 2008, Singapore. **Proceedings ...** New York: ACM, 2008. p. 91-98.

HIGHTOWER, J. ; BORRIELLO, G. Location systems for ubiquitous computing. **Computer**, Piscataway, NJ, v. 34, n. 8, p. 57-66, Aug. 2001.

HOFTE, G. H. ; MULDER, I. Dynamic personal social networks: a new perspective for CSCW research and design. In: **SIGGROUP Bulletin**, New York, v. 24, n. 3, p. 139-142, 2003.

IACHELLO, G. et al. Developing privacy guidelines for social location disclosure applications and services. In: SOUPS 2005 - SYMPOSIUM ON USABLE PRIVACY AND SECURITY, 1., 2005. Pittsburgh, PA. **Proceedings ...** New York: ACM, 2005. p. 65-76. (ACM International Conference Proceedings Series, 93.)

IAEM. International Association of Emergency Managers. **Principles of emergency management**. 2007. Disponível em: <http://www.iaem.com/EMPrinciples/index.htm>. Acesso em: mar. 2012.

IFRC. **International Federation of Red Cross and Red Crescent Societies**. 2012. Disponível em: <http://www.ifrc.org/>. Acesso: em out. 2012.

ITO, M. A New set of social rules for a newly wireless society. **Japan Media Review**, Los Angeles, 17/07/2004. score 0.8169. Disponível em: <http://www.ojr.org/japan/wireless/1043770650.php>. Acesso em: dez. 2010.

JAVED, Y. ; NORRIS, T. ; JOHNSTON, D. Evaluating SAVER: measuring shared & team situation awareness of emergency decision makers. In: INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS FOR CRISIS RESPONSE AND MANAGEMENT, 9.,

2012. Vancouver, Canada. **Proceedings ...** Vancouver, Canada: Simon Fraser University, 2012. Track: Human Experiences in the Design of Crisis Response and Management.

JIN, R. ; SI, L. A Bayesian approach toward active learning for collaborative filtering. In: UAI '04 - CONFERENCE ON UNCERTAINTY IN ARTIFICIAL INTELLIGENCE, 20., 2004, Banff, Canada. **Proceedings ...** Arlington: AUAI Press, 2004. p. 278-285.

JOHANSSON, B. ; TRNKA, J. ; GRANLUND, R. The Effect of geographical information systems on a collaborative command and control task. In: ISCRAM 2007 - INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS FOR CRISIS RESPONSE AND MANAGEMENT, Delft, Netherlands, **Proceedings ...** Darmstadt: Technisch Universitat Darmstadt, 2007. p. 191-200.

KAPFERER, B. Norms and the manipulation of relationships in a work context. In: MITCHELL, J.C. (Ed). **Social networks in urban situations**. Manchester: Manchester University Press, 1969.

KAPLAN, A. M. ; HAENLEIN, M. Users of the world, unite! The challenges and opportunities of social media. **Business Horizons**, Bloomington, Ind., v. 53, n. 1, p. 59-68, 2010.

KATRINA. **Hurricane Katrina**. 2012. Disponível em: <http://www.katrina.com/>. Acesso em: ago. 2012.

KAUTZ, H. A. ; SELMAN, B. ; SHAH, M. Referral web: combining social networks and collaborative filtering. **Communications of the ACM**, New York, v. 40, n. 3, p. 63-65, Mar. 1997.

KONDAVETI, R.; GANZ, A. Decision support system for resource allocation in disaster management. In: ANNUAL INTERNATIONAL CONFERENCE OF THE IEEE ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY, 31., 2009. Minneapolis, MN. **Proceedings ...** Piscataway, NJ: IEEE, 2009. p. 3425-3428.

KONSTAN, J. A. Introduction to recommender systems: algorithms and evaluation. **ACM Transactions on Information Systems**, New York, v. 22, n. 1, p. 1-4, Jan. 2004.

KRUCHTEN, P. et al. A Conceptual model of disasters encompassing multiple stakeholder domains. **International Journal of Emergency Management**, Olney, UK, v. 5, n. 1, p. 25-56, 2008.

KRÜGER, U. ; WUCHOLT, F. ; BECKSTEIN, C. Electronic checklist support for disaster response. In: INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS FOR CRISIS RESPONSE AND MANAGEMENT, 9., 2012. Vancouver, Canada. **Proceedings ...** Vancouver, Canada: Simon Fraser University, 2012. Track: Human Experiences in the Design of Crisis Response and Management.

LAM, S. K. ; RIEDL, J. Shilling recommender systems for fun and profit. In: WWW '04 - INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 13., 2004, New York. **Proceedings ...** , New York: ACM, 2004. p. 393-402.

LANG, X. ; OREGLIA, E. ; THOMAS, S. Social practices and mobile phone use of young migrant workers. In: MobileHCI '10 - INTERNATIONAL CONFERENCE ON HUMAN COMPUTER INTERACTION WITH MOBILE DEVICES AND SERVICES, 12., 2010, Lisbon. **Proceedings ...** New York: ACM, 2010. p. 59-62.

LANGHEINRICH, M. Privacy by design: principles of privacy-aware ubiquitous systems. In: Unbicomp 2001. - INTERNATIONAL CONFERENCE ON UBIQUITOUS COMPUTING, 3., 2001. Atlanta, GA. **Proceedings ...** London, Springer, 2001. p. 273-291, 2001.

LEDERER, S. ; MANKOFF, J. ; DEY, A. K. Who wants to know what when? Privacy preference determinants in ubiquitous computing. In: CHI EA '03. EXTENDED ABSTRACTS ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2003, Fort lauderdale, FL. **Proceedings ...** New York: ACM Press, 2003. p. 724-725.

LI, X. et al. Building a practical ontology for emergency response systems. In: CSSE - INTERNATIONAL CONFERENCE ON COMPUTER SCIENCE AND SOFTWARE ENGINEERING, 2008, Hubei, China. **Proceedings ...** Los Alamitos, CA: IEEE, 2008. v. 4. p. 222-225.

LINUX. **Linux kernel**. 1991. Disponível em: <http://www.kernel.org/>. Acesso em: out. 2012.

LUBERG, A. et al. Context-aware and multilingual information extraction for a tourist recommender system. In: INTERNATIONAL CONFERENCE ON KNOWLEDGE MANAGEMENT AND KNOWLEDGE TECHNOLOGIES, 11., 2011, Graz, Austria. **Proceedings ...** New York, ACM, 2011. p. 13:1-13:8.

MAEDCHE, A. ; STAAB, S. Semi-automatic engineering of ontologies from text. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING, 12., 2000, Chicago. **Proceedings ...** Skokie, IL: Knowledge Systems Institute, 2000. p. 231-239.

MARTERER, R. ; MOI, M. ; KOCH, R. An Architecture for distributed, event-driven systems to collect and analyze data in emergency operations and training exercises. In: INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS FOR CRISIS RESPONSE AND MANAGEMENT, 9., 2012. Vancouver, Canada. **Proceedings ...** Vancouver, Canada: Simon Fraser University, 2012. Track: Human Experiences in the Design of Crisis Response and Management.

MBA online. **A Case study in social media demographics**. 2012. Disponível em: <http://www.onlinemba.com/blog/social-media-demographics/>. Acesso em: ago. 2012.

MCDONALD, D. W. Recommending collaboration with social networks. In: CHI'03 - SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2003, Ft. Lauderdale, FL. **Proceedings ...** New York: ACM, 2003. p. 593-600.

MCDONALD, D. W. ; ACKERMAN, M. S. Expertise recommender: a flexible recommendation system and architecture. In: CSCW 00 - 2000 ACM CONFERENCE ON COMPUTER SUPPORTED COOPERATIVE WORK, 2000, Philadelphia, PA. **Proceedings ...** New York: ACM, 2000. p. 231-240.

MECHLER, R. **Natural disaster risk management and financing disaster losses in developing countries**. [S.l.]: Verlag Versicherungswirtschaft (VWV Karlsruhe), 2004. 312 p.

MELVILLE, P. ; MOONEY, R. J. ; NAGARAJAN, R. Content-boosted collaborative filtering for improved recommendations. In: AAAI-02 - NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, 18., 2002, Edmonton, Alberta. **Proceedings ...** Palo Alto, CA: AAAI, 2002. p. 187-192.

MELVILLE, P. ; SINDHWANI, V. Recommender systems. In: SAMMUT, C. ; WEBB, G. I. (Eds.). **Encyclopedia of Machine Learning**. New York: Springer Science+Business Media, 2010.

MILGRAM, S. The Small world problem. **Psychology Today**, New York, v. 1, n. 1, p. 61-67, May 1967. Disponível em: http://measure.igpp.ucla.edu/GK12-SEE-LA/Lesson_Files_09/Tina_Wey/TW_social_networks_Milgram_1967_small_world_problem.pdf. Acesso em: dez. 2011.

MIRZA, M. M. Q. Climate change and extreme weather events: can developing countries adapt?. **Climate Policy**, [S.l.], v. 3, n. 3, p. 233-248, Sept. 2003.

MOBI thinking. **Global Mobile Statistics 2012**. 2012. Disponível em: <http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats>. Acesso em: ago. 2012.

MOBILE internet report. **Stanley Morgan**. 2009. Disponível em: http://www.morganstanley.com/institutional/techresearch/pdfs/2SETUP_12142009_RI.pdf (Resumo e reportagem sobre o relatório Disponível em: <http://mobilebeyond.net/mobile-internet-research-report/>). Acesso em: nov. 2011.

MONARES, A. et al. MobileMap: a collaborative application to support emergency situations in urban areas. In: INTERNATIONAL CONFERENCE ON COMPUTER SUPPORTED COOPERATIVE WORK IN DESIGN, 13., 2009, Santiago, Chile. **Proceedings ...** Santiago, Chile: Universidad de Chile/IEEE, 2009.p. 432-437.

MONCLAR, R. et al. MEK: using spatial-temporal information to improve social networks and knowledge dissemination. **Information Sciences**, New York, v. 179, n. 15, p. 2524-2537, July 2009.

MOONEY, R. J. ; ROY, L. Content-based book recommending using learning for text categorization. In: ACM CONFERENCE ON DIGITAL LIBRARIES, 5., 2000, San Antonio, TX. **Proceedings ...** New York: ACM, 2000. p. 195-204, June 2000.

MURPHY, T. ; JENNEX, M. E. Knowledge management systems for hurricane Katrina response. In: ISCRAM 2006 -INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS FOR CRISIS RESPONSE AND MANAGEMENT, 3., 2006, Newark, NJ. **Proceedings ...** Darmstadt: Technisch Universitat Darmstadt, 2006. p. 615.

NEYEM, A. ; OCHOA, S. F. ; PINO, J. A. Integrating service-oriented mobile units to support collaboration in ad-hoc scenarios. **Journal of Universal Computer Science**, [S.l.], v. 14, n. 1, p. 88-122, 2008.

NIMS. **National Incident Management System**. 2008. Disponível em: <http://www.fema.gov/emergency/nims/>. Acesso em: dez. 2012.

OGATA, H. et al. Computer supported social networking for augmenting cooperation. **Computer Supported Cooperative Work: The Journal of Collaborative Computing**, Dordrecht, v. 10, n. 2, p. 189-209, Jan. 2001.

OLIVEIRA, J. N. ; AMARAL, L. **O papel da qualidade da informação nos sistemas de informação**. 1999. Disponível em: <http://repositorium.sdum.uminho.pt/handle/1822/2183>. Acesso em: nov. 2011.

PAGOTTO, J. ; O'DONNELL, D. Canada's multi-agency situational awareness system – keeping it simple. In: INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS FOR CRISIS RESPONSE AND MANAGEMENT, 9., 2012. Vancouver, Canada. **Proceedings ...** Vancouver, Canada: Simon Fraser University, 2012. Track: Human Experiences in the Design of Crisis Response and Management Service and Systems.

PAZZANI, M. J. ; BILLSUS, D. Learning and revising user profiles: the identification of interesting web sites. **Machine Learning**, Noywell, MA, v. 27, n. 3, p. 313-331, Jun. 1997.

PEDDEMORS, A. J. H. ; LANKHORST, M. M. ; DE HEER, J. Presence, location, and instant messaging in a context-aware application framework. In: INTERNATIONAL CONFERENCE ON MOBILE DATA MANAGEMENT, 4., 2003, Melbourne. **Proceedings ...** New York: ACM, 2003. p. 325-330, 2003.

PFIF. **People Finder Interchange Format**. 2005. Disponível em: <http://zesty.ca/pfif/>. Acesso em: nov. 2011.

R7. **R7 Notícias**. 2011. Disponível em: <http://noticias.r7.com/rio-de-janeiro/noticias/remocao-de-desabrigados-do-morro-do-bumba-e-impedida-por-acao-do-ministerio-publico-20110104.html>. Acesso em: mar. 2012.

RESNICK, P. et al. Grouplens: an open architecture for collaborative filtering of netnews. In: CSCW 94 – CONFERENCE ON COMPUTER SUPPORTED COOPERATIVE WORK, 1994, Chapel Hill, NC. **Proceedings ...** New York: ACM, 1994. p. 175-186.

RESNICK, P. ; VARIAN, H. R. Recommender systems. **Communications of the ACM**, New York, v. 40, n. 3, p. 56-58, Mar. 1997.

RICCI, F. ; NGUYEN, Q. N. Acquiring and revising preferences in a critique-based mobile recommender system. **IEEE Intelligent Systems**, New York, v. 22, n. 3, p. 22-29, Jun. 2007.

RICCI, F. ; ROKACH, L. ; SHAPIRA, B. Introduction to recommender systems handbook. In: RICCI, F.; ROKACH, L.; SHAPIRA, B.; KANTOR, P.B. (Eds.). **Recommender Systems Handbook**. New York: Springer, 2011 p. 1-35.

SANTOS, R. D. O. et al. JoinUs: management of mobile social networks for pervasive collaboration. In: SBSC 2008 - SIMPÓSIO BRASILEIRO DE SISTEMAS COLABORATIVOS, 5., 2008, Vila Velha, ES. **Anais ...** Los Alamitos, CA: SBC/IEEE, 2008. p. 224-234.

SANTOS, R. S. **Um Modelo de referência para avaliação da capacidade de resposta das organizações de emergência**. 2007. Dissertação (Mestrado em Informática), Instituto de Matemática, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2007.

SCHEIN, A. I. et al. Methods and metrics for cold-start recommendations. In SIGIR '02 - ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 25., 2002, Tampere Finland. **Proceedings ...** New York: ACM, 2002. p. 253-260.

SILVA, S. T. F. ; OLIVEIRA, J. ; BORGES, M. R. S. Contextual analysis of the victims social network for people recommendation on the emergency scenario. In: HERSKOVIC, V. et al. (Eds.). **Collaboration and technology**, Berlin: Springer, 2012, p. 200-207.

SINHA, R. ; SWEARINGEN, K. Comparing recommendations made by online systems and friends. In: DELOS-NETWORK OF EXCELLENT WORKSHOP ON PERSONALIZATION AND RECOMMENDER SYSTEMS IN DIGITAL LIBRARIES, 2., 2001, Dublin. **Proceedings ...** Arlington, VA: US National Science Foundation , 2001.

SMITH, I. Social-mobile applications. **IEEE Computer Magazine**, New York, v. 38, n. 4, p. 84-85, Apr. 2005.

SOUZA, D. S. et al. MEK: uma abordagem oportunística para disseminação colaborativa do conhecimento. In: SIMPÓSIO BRASILEIRO DE SISTEMAS COLABORATIVOS, 8., 2011. Paraty. **Anais ...** Porto Alegre: SBC, 2011. v. 1. p. 145-151.

STARBIRD, K. et al. Chatter on the red: what hazards threat reveals about the social life of microblogged information. In: 2010 ACM CONFERENCE ON COMPUTER SUPPORTED COOPERATIVE WORK, 2010, Savannah, Georgia. **Proceedings ...** New York, ACM, 2010. p. 241-250.

STRANG, T. ; LINNHOF-POPIEN, C. A Context modeling survey. In: UbiComp 2004 - INTERNATIONAL CONFERENCE ON UBIQUITOUS COMPUTING, 6., 2004, Nottingham, Eng. WORKSHOP ON ADVANCED CONTEXT MODELLING, REASONING AND MANAGEMENT, 2004, Nottingham, Eng. **Proceedings ...** New York: ACM, 2004.

SU, X. et al. Hybrid collaborative filtering algorithms using a mixture of experts. In: IEEE/WIC/ACM INTERNATIONAL CONFERENCE ON WEB INTELLIGENCE, 2007, Silicon Valley, CA. **Proceedings ...** Los Alamitos, CA: IEEE, 2007. p. 645-649.

_____. Imputation-boosted collaborative filtering using machine learning classifiers. In: SAC '08 - ANNUAL ACM SYMPOSIUM ON APPLIED COMPUTING, 23. 2008, Fortaleza. **Proceedings ...** New York: ACM, 2008. p. 949-950.

SUTTON, J. Twittering Tennessee: distributed networks and collaboration following a technological disaster. In: ISCRAM 2010 -INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS FOR CRISIS RESPONSE AND MANGEMENT, 7., 2010, Seattle. **Proceedings ...** Darmstadt: Technisch Universitat Darmstadt, 2010.

SYMBIAN Foundation. **Symbian OS**. 1997. Disponível em: <http://licensing.symbian.org/>. Acesso em: out. 2012.

TELECO - inteligência em telecomunicações. **Estatísticas de celulares no Brasil**. 2012. Disponível em: <http://www.teleco.com.br/ncel.asp>. Acesso em: ago. 2012.

TERVEEN, L. ; MCDONALD, D. W. Social matching: a framework and research agenda. **ACM Transactions on Computer-Human Interaction**, New York, v. 12, n. 3, p. 401-434, Sept. 2005.

TRAGÉDIA na região serrana do RJ. **Folha de São Paulo**. 2011. Disponível em: <http://www1.folha.uol.com.br/especial/2011/tragedianaregiaooserranadorj/>. Acesso em: ago. 2012.

TRUPTIL, S. ; BÉNABEN, F. ; PINGAUD, H. Collaborative process design for mediation information system engineering. In: ISCRAM'09 - INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS FOR CRISIS RESPONSE AND MANAGEMENT, 2009, Göteborg, Sweden. **Proceedings ...** Darmstadt: Technisch Universitat Darmstadt, 2009.

TUNG, H. W. ; SOO, V. W. A Personalized restaurant recommender agent for mobile e-service. In: EEE 04 – IEEE INTERNATIONAL CONFERENCE ON E-TECHNOLOGY, E-COMMERCE AND E-SERVICES, 2004. Taipei, Taiwan **Proceedings ...** Los Alamitos, CA, IEEE, 2004. p. 259-262.

TWIDEV. **Twitter developers**. 2012. Disponível em: <https://dev.twitter.com/>. Acesso em: out. 2012.

TWITTER. 2006. Disponível em: <http://twitter.com/>. Acesso em: 11-2011.

UOL. **UOL Notícias**. 2010. Disponível em: <http://noticias.uol.com.br/cotidiano/ultimas-noticias/2010/04/09/niteroi-rj-nao-implementa-plano-contra-desastres-financiado-pelo-ministerio-das-cidades.htm>. Acesso em: mar. 2012.

VERBRUGGE, L. M. Multiplexity in adult friendships. **Social Forces**, v. 57, n. 4, p. 1286-1309, Jun. 1979.

VIEWEG, S. et al. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In: CHI 2010 - INTERNATIONAL CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 28., 2010. Atlanta, GA. **Proceedings ...** New York: ACM, 2010. p. 1079-1088.

XU, W. ; ZLATANOVA, S. Ontologies for disaster management response. In: LI, J.; ZLATANOVA, S. ; FABBRI, A. (Eds.) **Geomatics solutions for disaster management**. Heidelberg: Springer, 2007. p. 185-200, 2007. (Lecture Notes in Geoinformation and Cartography).

YAP, G. E. ; TAN, A. H. ; PANG, H. H. Dynamically-optimized context in recommender systems. In: INTERNATIONAL CONFERENCE ON MOBILE DATA MANAGEMENT, 6., 2005, Ayia Napa, Cyprus. **Proceedings ...** New York: ACM, 2005. p. 265-272.

YU, B. ; CAI, G. Coordination of emergency response operations via event-based awareness mechanism. In: INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS FOR CRISIS RESPONSE AND MANAGEMENT, 9., 2012. Vancouver, Canada. **Proceedings ...** Vancouver, Canada: Simon Fraser University, 2012. Track: Human Experiences in the Design of Crisis Response and Management Service and Systems.

WAGNER, D. et al. Hide and seek. In: INTERNATIONAL CONFERENCE ON HUMAN COMPUTER INTERACTION WITH MOBILE DEVICES AND SERVICES, 12., 2010, Lisbon. **Proceedings ...** New York: ACM, 2010. p. 55.

WALLS, J. G. ; WIDMEYER, G. R. ; EL SAWY, O. A. Assessing information system design Theory in perspective: how useful was our 1992 initial rendition?. **Journal of Information Technology Theory and Application**, Atlanta, v. 6, n. 2, p. 43-58, Jul. 2004.

_____. Building an information system design theory for vigilant EIS. **Information Systems Research**, Providence, v. 3, n. 1, p. 36-59, Jan. 1992.

WOERNDL, W. ; SCHUELLER, C. ; WOJTECH, R. A Hybrid recommender system for context-aware recommendations of mobile applications. In: 2007 IEEE INTERNATIONAL CONFERENCE ON DATA ENGINEERING WORKSHOP, 23., 2007, Istanbul. **Proceedings ...** Istanbul, IEEE, 2007. p. 871-878.

ZHENG, R. ; PROVOST, F. ; GHOSE, A. Social network collaborative filtering: preliminary results. In: WEB - WORKSHOP ON E-BUSINESS, 6., 2007, Montreal, Canada. **Proceedings ...** Atlanta, GA: AIS SIFeBIZ, 2007.

ZIESEMER, A. C. A.; OLIVEIRA, J. B. S. How to know what do you want? a survey of recommender systems and the next generation. In: SIMPÓSIO BRASILEIRO DE SISTEMAS COLABORATIVOS, 8., 2011. Paraty. **Anais ...** Porto Alegre: SBC, 2011. v. 1. p. 104-111.