



Universidade Federal do Rio de Janeiro

MARCUS FELIPE TAVARES MACHADO

**UM ARCABOUÇO PARA O
DESENVOLVIMENTO DE SISTEMAS
MÓVEIS NO DOMÍNIO DE
GESTÃO DE EMERGÊNCIAS**

DISSERTAÇÃO DE MESTRADO

Rio de Janeiro

2014



Instituto de Matemática



Instituto Tércio Pacitti de Aplicações
e Pesquisas Computacionais

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
INSTITUTO TÉRCIO PACITTI DE APLICAÇÕES E PESQUISAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

MARCUS FELIPE TAVARES MACHADO

UM ARCABOUÇO PARA O DESENVOLVIMENTO DE
SISTEMAS MÓVEIS NO DOMÍNIO DE
GESTÃO DE EMERGÊNCIAS

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.

Orientador: Prof^ª. Adriana Santarosa Vivacqua, D.Sc.

Co-orientador: Prof. Marcos Roberto da Silva Borges, Ph.D.

Rio de Janeiro
2014

M149

Machado, Marcus Felipe Tavares

Um Arcabouço para o Desenvolvimento de Sistemas Móveis no Domínio de Gestão de Emergências / Marcus Felipe Tavares Machado. -- 2014.
194 f.

Dissertação (Mestrado em Informática) - Universidade Federal do Rio de Janeiro, Instituto de Matemática, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Programa de Pós-Graduação em Informática, Rio de Janeiro, 2014.

Orientadora: Adriana Santarosa Vivacqua.

Co-orientador: Marcos Roberto da Silva Borges.

1. Informática. 2. Gestão de Emergências. 3. Reutilização de Software. 4. Dispositivos Móveis -Teses. I. Vivacqua, Adriana Santarosa (Orient). II. Borges, Marcos Roberto da Silva (Co-orient). III. Título.

CDD

MARCUS FELIPE TAVARES MACHADO

UM ARCABOUÇO PARA O DESENVOLVIMENTO DE
SISTEMAS MÓVEIS NO DOMÍNIO DE
GESTÃO DE EMERGÊNCIAS

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 ____/____/_____.

Prof^a. Adriana Santarosa Vivacqua, D.Sc., PPGI/IM/UFRJ

Prof. Marcos Roberto da Silva Borges, Ph.D., PPGI/IM/UFRJ

Prof. Eber Assis Schmitz, Ph.D., DCC/UFRJ

Prof^a. Anamaria Martins Moreira, D.Sc., DCC/UFRJ

Prof. Ricardo Choren Noya, D.Sc., IME

A Deus,
pela graça e pela vida.

Agradecimentos

A Deus, por permitir a concretização deste trabalho.

Aos meus pais, Odílma e Rogério, pelo amor e esforço dedicado à minha formação.

Aos meus familiares e amigos, pela compreensão das ausências ao longo destes anos.

Aos meus orientadores, Adriana Vivacqua e Marcos Borges, pelos conhecimentos transmitidos e pela oportunidade e confiança depositada em mim.

Aos participantes, Bruna Diirr, Bruno Nascimento, Tiago Marino, Daniel Assumpção, João Silva, Lucas Rodrigues e Leonardo Fanzeres, que colaboraram na avaliação dessa pesquisa.

Aos professores e colegas do Grupo de Engenharia do Conhecimento (GRECO/PPGI), pela experiência adquirida e contribuição à minha formação.

Aos professores, Giuliano Prado, Marcos Quinet e Wagner Arbex, pelo incentivo no início desta jornada.

A CAPES, pelo apoio financeiro para a realização deste trabalho.

Resumo

MACHADO, Marcus Felipe Tavares. **Um Arcabouço para o desenvolvimento de sistemas móveis no domínio de gestão de emergências**. 2014. 194 f. Tese (Mestrado em Informática) – Instituto de Matemática, Instituto Tércio Pacciti, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2014.

Nos últimos anos, o domínio de Gestão de Emergências tornou-se alvo de vários esforços em pesquisas científicas. Este domínio é caracterizado por aspectos colaborativos e muitos pesquisadores passaram a abordar o uso de sistemas móveis para complementar as formas tradicionais de compartilhamento de informações em situações de crise. No entanto, pouco se discute sobre a reutilização de software nas soluções propostas, bem como muito poucos projetos têm-se concentrado no desenvolvimento de componentes reutilizáveis. A reutilização de software é desejável para que se aproveitem as capacidades e os pontos fortes dos sistemas existentes, acelerando o desenvolvimento e aumentando a confiabilidade na construção de novos sistemas. Neste trabalho, é apresentado um arcabouço para apoiar a reutilização de software no desenvolvimento de sistemas móveis, com base em especificações do domínio de Gestão de Emergências. Ele fornece orientações para a construção de novos sistemas e simplifica o seu desenvolvimento, conforme necessário para o uso por analistas de sistemas e desenvolvedores.

Palavras-chave: Gestão de Emergências, Reutilização de Software, Dispositivos Móveis.

Abstract

MACHADO, Marcus Felipe Tavares. **Um Arcabouço para o desenvolvimento de sistemas móveis no domínio de gestão de emergências**. 2014. 194 f. Tese (Mestrado em Informática) – Instituto de Matemática, Instituto Tércio Pacciti, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2014.

In recent years, Emergency Management has become the target of multiple research efforts. This domain is characterized by collaborative aspects and several researchers discuss the use of mobile systems to complement traditional forms of information sharing in crisis situation. However, there is little software reuse in proposed solutions, as very few projects have focused on developing reusable components. Reusing components is desirable for it harnesses the strengths of existing systems, speeds up development and increases new systems reliability. In this work, it is present a framework to support software reuse in mobile systems development, based on specifications from the Emergency Management domain. It provides guidance for the construction of new systems and simplifies their development, as required by the systems analysts and developers.

Keywords: Emergency Management, Software Reuse, Mobile Devices.

Lista de Figuras

Figura 1: Quatro fases da Gestão de Emergências	27
Figura 2: Interface do Sistema NOAH	34
Figura 3: Interfaces do Sistema MobileMap	35
Figura 4: Interface do Sistema BUMBONET	36
Figura 5: Interface do Sistema DMT	37
Figura 6: Interfaces do Sistema MIKoBOS.....	38
Figura 7: Interface do Sistema GeoBIPS.....	38
Figura 8: Interfaces do Sistema WORKPAD	39
Figura 9: Interfaces do Sistema SisC2Celular.....	40
Figura 10: Interfaces do Sistema EMERGENCY	41
Figura 11: Interfaces do Sistema de Emergência	42
Figura 12: Interfaces do Sistema ProRad	43
Figura 13: Modelo 3C da Colaboração instanciado para o trabalho em grupo (adaptado de PIMENTEL; FUKS, 2011).....	48
Figura 14: Classificação dos sistemas de informação para GE de acordo com o Modelo 3C da Colaboração	48
Figura 15: Especificação e implementação com arcabouços (adaptado de D’SOUZA; WILLS, 1999).....	60
Figura 16: Etapas da pesquisa	63
Figura 17: Ambiente de resposta à emergência.....	66
Figura 18: Principais funcionalidades estabelecidas para o JEMF.....	69
Figura 19: Principais atores envolvidos em Gestão de Emergências	71
Figura 20: Casos de Uso da funcionalidade Gerenciar Emergência	73
Figura 21: Diagrama de classes do modelo de domínio.....	75
Figura 22: Pacote Núcleo do JEMF destacando as principais classes (abstratas).....	76
Figura 23: Pacote Núcleo do JEMF destacando classes do aspecto de coordenação.....	76
Figura 24: Pacote Núcleo do JEMF destacando as classes para o aspecto de comunicação....	78
Figura 25: Pacote Núcleo do JEMF destacando as classes para o aspecto de cooperação.....	78
Figura 26: Pacote Móvel do JEMF destacando as superclasses e os padrões de projetos (Parte 1).....	80
Figura 27: Pacote Móvel do JEMF destacando as superclasses e os padrões de projetos (Parte 2).....	82

Figura 28: Pacote Móvel do JEMF destacando a classes que compõem a funcionalidade	83
Figura 29: Pacote Móvel do JEMF destacando as classes que compõem as funcionalidades .	84
Figura 30: Diagrama do Modelo Entidade-Relacionamento do JEMF	88
Figura 31: Percentual de distribuição de dispositivos móveis para cada versão da plataforma Android (Android, 2014d).....	90
Figura 32: Mensagens de teste bem-sucedido do JEMF pelo JUnit.....	94
Figura 33: Interfaces do sistema móvel de exemplo do arcabouço	96
Figura 34: Resultados sobre a experiência dos desenvolvedores em OO e Java	102
Figura 35: Resultados sobre a experiência dos desenvolvedores em Dispositivos Móveis e Plataforma.....	102
Figura 36: Resultados sobre a experiência dos desenvolvedores em Gestão de Emergências e sua escolaridade	103
Figura 37: Resultados sobre a execução dos exercícios práticos do JEMF pelos desenvolvedores	104
Figura 38: Resultados sobre a facilidade de uso do JEMF e sua documentação	105
Figura 39: Resultados sobre a facilidade no desenvolvimento de sistema móvel para a gestão de emergências	106
Figura 40: Resultados sobre o apoio da abordagem para o desenvolvimento de sistemas móveis no domínio	108
Figura 41: Resultados sobre a aplicabilidade do JEMF em novos projetos	108
Figura 42: Resultados sobre a geração de benefícios pelo JEMF em novos projetos	109
Figura 43: Resultados sobre a personalização do JEMF em organizações de resposta a emergências	109
Figura 44: Resultados sobre a escolaridade dos especialistas e sua experiência no domínio	111

Lista de Quadros

Quadro 1: Classificação dos sistemas de informação para GE de acordo com as categorias de funcionalidade e outras características	46
Quadro 2: Associação entre as categorias de funcionalidades gerais e do JEMF	68

Lista de Siglas

ADT	<i>Android Developer Tools</i>
API	<i>Application Programming Interface</i>
C2	Comando e Controle
CCCMS	<i>Car Crash Crisis Management System</i>
CIMS	<i>Crisis Information Management Systems</i>
COM	<i>Component Object Model</i>
CORBA	<i>Common Object Request Broker Architecture</i>
DMT	<i>Disaster Management Tool</i>
DUMBONET	<i>Digital Ubiquitous Mobile Broadband OLSR Network</i>
EMERGENCY	<i>Mobile Decision Support in Emergency Situations</i>
FEMA	<i>Federal Emergency Management Agency</i>
FHWA	<i>Federal Highway Administration</i>
GE	Gestão de Emergência
GeoBIPS	<i>Geographic Broadband Integration for Public Services</i>
GPL	<i>General Public License</i>
GPS	<i>Global Positioning System</i>
HHS	<i>Department of Health And Human Services</i>
HLMP	<i>High Level MANET Protocol</i>
HTML	<i>HyperText Markup Language</i>
Java ME	<i>Java Platform Micro Edition</i>
JEMF	<i>Java Emergency Management Framework</i>
MANET	<i>Mobile Ad Hoc Network</i>
MER	Modelo Entidade Relacionamento
MIKoBOS	<i>Mobile Information and Communication System for Public Safety Organizations</i>
MVC	<i>Model-View-Controller</i>
NOAH	<i>Emergency Organisation and Administration Aid</i>
OLSR	<i>Optimized Link State Routing Protocol</i>
ONG	Organização Não Governamental
OO	Orientação a Objeto
PDA	<i>Personal Digital Assistant</i>
ProRad	Sistema de Proteção Radiológica
SDK	<i>Software Development Kit</i>
Sis.Emergência	Sistema de Emergência

SOA	<i>Service-Oriented Architecture</i>
SOAP	<i>Simple Object Access Protocol</i>
TCP	<i>Transmission Control Protocol</i>
UHF	<i>Ultra High Frequency</i>
UML	<i>Unified Modeling Language</i>
XML	<i>Extensible Markup Language</i>

Sumário

Introdução	17
1.1 Motivação.....	18
1.2 Problema.....	19
1.3 Hipótese.....	20
1.4 Enfoque da Solução.....	20
1.5 Estrutura da Dissertação.....	21
2 Domínio de Gestão de Emergências	23
2.1 Emergência e Desastre	23
2.2 Gestão de Emergências	24
2.3 Fases da Gestão de Emergências.....	26
2.3.1 Mitigação.....	27
2.3.2 Preparação	28
2.3.3 Resposta	28
2.3.4 Recuperação	29
2.4 Comando e Controle.....	30
2.5 Desafios de Comunicação na Fase de Resposta.....	31
3 Sistemas de Informação na Gestão de Emergências.....	33
3.1 NOAH	33
3.2 MobileMap.....	34
3.3 CCCMS	35
3.4 DUMBONET	35
3.5 DMT.....	36
3.6 MIKoBOS	37
3.7 GeoBIPS.....	38
3.8 WORKPAD.....	39
3.9 SisC2Celular.....	40
3.10EMERGENCY	41
3.11 Sistema de Emergência	41
3.12ProRad.....	42
3.13 Classificação dos Sistemas de Informação.....	43
3.13.1 Classificação dos sistemas segundo suas funcionalidades	43
3.13.2 Classificação dos sistemas segundo os 3Cs da Colaboração	47
4 Reuso de Software.....	49
4.1 Conceituação	49

4.2 Bibliotecas	49
4.3 Componentes	50
4.4 APIs	52
4.5 Padrões de Projeto	52
4.6 Arcabouços	54
4.6.1 Classificação	56
4.6.2 Métodos Gancho	57
4.6.3 Caixa-Branca e Caixa-Preta	58
4.6.4 Inversão de Controle	58
4.6.5 Processo de Desenvolvimento	59
5 JEMF: Um Arcabouço para o Desenvolvimento de Sistemas Móveis no Domínio de Gestão Emergências	63
5.1 Metodologia	63
5.2 Escopo	65
5.3 Especificação do Arcabouço	66
5.3.1 Modelagem de Análise	66
5.3.2 Modelagem de Projeto	73
5.4 Implementação do Arcabouço	89
5.4.1 Plataforma Móvel	89
5.4.2 Programação-por-diferença	90
5.4.3 Código Aberto	91
5.4.4 Gerência de Configuração	92
5.4.5 Ambiente de Desenvolvimento	92
5.5 Teste do Arcabouço	93
5.5.1 Teste de Unidade e de Integração	93
5.5.2 Documentação	95
5.5.3 Instanciação	96
6 Avaliação do Arcabouço	98
6.1 Avaliação por Desenvolvedores de Software	98
6.1.1 Caracterização do Participante	98
6.1.2 Exercícios de Avaliação do Arcabouço	99
6.1.3 Avaliação de Experiência Geral do Arcabouço	101
6.1.4 Resultados Coletados dos Desenvolvedores	101
6.2 Avaliação por Especialistas do Domínio	107
6.2.1 Avaliação Crítica do Arcabouço	107
6.2.2 Resultados Coletados dos Especialistas	107

6.3 Conclusões da Avaliação do Arcabouço.....	111
7 Trabalhos Relacionados	113
7.1 ESCAPE.....	113
7.2 CIMS.....	114
7.3 HLMP API.....	114
7.4 Sahana.....	115
7.5 Discussão.....	116
8 Conclusão.....	120
8.1 Contribuições.....	121
8.2 Limitações.....	122
8.3 Trabalhos Futuros.....	123
Referências	125
Apêndices	134
APÊNDICE A – QUADRO DE FUNCIONALIDADES DOS SISTEMAS DE INFORMAÇÃO NA GESTÃO DE EMERGÊNCIAS.....	134
APÊNDICE B – DIAGRAMAS DE CASOS DE USO E SUAS DESCRIÇÕES PARA AS FUNCIONALIDADES DO JEMF.....	137
APÊNDICE C – DIAGRAMAS DE SEQUÊNCIA PARA AS FUNCIONALIDADES DO JEMF.....	163
APÊNDICE D – DICIONÁRIO DE ATRIBUTOS DAS ENTIDADES DO MODELO DO BANCO DE DADOS.....	175
APÊNDICE E – QUESTIONÁRIOS PARA A AVALIAÇÃO DO JEMF.....	183

Introdução

Emergências acontecem o tempo todo e muitas vezes resultam em perdas materiais e humanas. Recentemente vivenciamos uma das maiores tragédias na cidade de Santa Maria, Rio Grande do Sul. O incêndio na boate Kiss ocorreu no dia 27 de janeiro de 2013 e deixou pelo menos 240 mortos. Esse foi o segundo incêndio mais mortal e a quinta maior tragédia da história do país (SOUCHAUD; FUSCO, 2012; UOL, 2013).

Da mesma forma, os desastres naturais podem gerar efeitos graves na população mundial, dado que os fenômenos naturais que os desencadeiam, como as tempestades, os terremotos e os vulcões existem em diferentes partes do globo. Considerando somente o ano de 2011, foram apontados 332 desastres naturais, contudo os impactos humanos e econômicos foram enormes, uma vez que mataram 30.773 pessoas e causaram os maiores danos econômicos já registrados, U\$ 366.100 bilhões (GUHA-SAPIR et al., 2012).

As fragilidades sociais agregadas à inexistência ou à insuficiência de processos de gestão de emergências e de capacidade de resposta podem produzir vulnerabilidades aos desastres, tornando indispensável a criação de mecanismos que possam ajudar na redução de danos suscetíveis a sociedade. À vista disso, o Brasil deve continuar na corrida contra o tempo para alcançar uma gerência em emergências a nível nacional para reparar as falhas em equipes com recursos materiais e humanos insuficientes, ou seja, faz-se essencial alinhar essa área do conhecimento com a comunidade científica e profissional, com a busca de novos estudos e tecnologias nacionais para a evolução desta importante área que preserva a vida, o meio ambiente e o patrimônio (SEITO et al., 2008).

Novos sistemas computacionais surgiram para apoiar o tratamento de emergências e gerenciar as atividades das pessoas que estão incluídas nesse domínio. Cada vez mais se torna fundamental apoiar o trabalho dessas pessoas e para isso é necessário fornecer um meio que auxilie analistas e desenvolvedores de software no desenvolvimento de novos projetos. Existem algumas técnicas de Engenharia de Software que permitem a reutilização de código ou de projeto de um sistema. Uma dessas técnicas de desenvolvimento de software é o arcabouço, em inglês *framework*. Quando um arcabouço é desenvolvido, os aspectos genéricos do domínio ao qual ele se refere são capturados, e todos os diferentes sistemas que possam ser desenvolvidos para esse domínio se utilizam dos conceitos presentes no arcabouço, havendo dessa forma, uma economia de esforços na construção desses sistemas

(JOHNSON; FOOTE, 1988). Diante disso, este trabalho abordará o desenvolvimento de um arcabouço que permitirá a construção de sistemas móveis para o domínio de emergências.

1.1 Motivação

Nas últimas décadas, cada vez mais o domínio de Gestão de Emergências se tornou alvo de diferentes pesquisas. Esse domínio se caracteriza por aspectos colaborativos, visto que envolve o trabalho de diferentes pessoas e organizações intervenientes nas situações de crise, como Corpo de Bombeiros, Defesa Civil e Polícia Militar. Seito e outros (2008) defendem a geração de uma cultura de pesquisa e inovação a fim de que ideias possam ser desenvolvidas neste domínio para que a distância no progresso tecnológico entre nós e os outros países diminua através da pesquisa nacional com o uso de sistemas computacionais, automatização, dispositivos, etc.

Com a popularização dos dispositivos móveis, eles se tornaram uma opção frequente para apoiar o tratamento de emergências devido à combinação da mobilidade com os mais variados tipos de recursos e capacidades. Tais características destes dispositivos favoreceu o surgimento de sistemas que possibilitam a consulta e atualização de informações em quase todos os lugares. Dessa forma, estes dispositivos vêm adquirindo estímulos com o desenvolvimento de sistemas móveis ao suportar os diferentes tipos de usuários e organizações que atuam no domínio.

Vários estudos abordam soluções de software e experimentos complexos, onde algumas especificações em comum podem ter sido elaboradas nesses trabalhos. Na medida em que os sistemas de software se tornam cada vez mais complexos, maior importância é dada às técnicas de desenvolvimento de software que buscam facilitar a produção de novos sistemas. Assim, um dos objetivos mais perseguidos pela Engenharia de Software tem sido a reutilização.

Essa perspectiva reforça a necessidade de aprimorar o desenvolvimento de sistemas móveis e promover o reuso de software no domínio de Gestão de Emergências. Dentre as técnicas de reutilização mais atuais, destaca-se a utilização de arcabouços de software. Um arcabouço de software é formado por um conjunto de classes cooperantes que representam os objetos de um domínio em particular. Somente os aspectos específicos dos novos sistemas precisam ser modelados, pois a modelagem de análise e projeto para obter os requisitos genéricos e a sua respectiva implementação, já foram feitos e testados.

1.2 Problema

A necessidade de prover dispositivos tecnológicos para as equipes de emergência que ampliem seu acesso às informações vem aumentando ao longo do tempo. O uso da tecnologia pode melhorar o entendimento comum dos envolvidos em suas atividades, tornando a resposta às emergências mais eficiente.

Novos trabalhos começaram a surgir com o objetivo de apoiar a colaboração e a capacidade de coordenação e cooperação das equipes em gestão de emergências. Dentre eles se destacam a criação de arcabouços conceituais relacionados com o conhecimento contextual no domínio de emergências por Diniz (2006). Turoff e outros (2004) apresentam alguns requisitos para a concepção de sistemas de informação de gestão de emergência e também um arcabouço conceitual. Sharma e outros (2003) e Paelke e outros (2012) evidenciaram estudos com o objetivo de aperfeiçoar a gestão de situações de emergências através de interfaces e interações avançadas. Ferreira (2011) efetuou um estudo sobre as decisões tático-operacionais tomadas por gestores durante um desastre e elaborou uma ferramenta colaborativa para telas sensíveis ao toque com o objetivo de melhorar a percepção situacional. Swiatek (1999) dedicou seu estudo no aprimoramento de medidas de enfrentamento em situações de emergência com uma ferramenta que permite o planejamento e simulação sobre de vários aspectos, por exemplo, efeitos, riscos e prejuízos. Zhang e Li (2008) focaram seu estudo no apoio a comunicação através de uma infraestrutura de suporte em áreas afetadas pela emergência. Vários outros projetos foram elaborados para o contexto móvel e um conjunto deles será apresentado mais adiante neste documento.

Ao analisar estes trabalhos descritos na literatura é possível notar o desenvolvimento de propostas significativas para a resolução de problemas de comunicação e compartilhamento de informações, focando nos aspectos do domínio de emergências. Eles geralmente abordam soluções para apoiar a colaboração entre todos os envolvidos. Entretanto, o processo de desenvolvimento de cada protótipo possivelmente não foi totalmente descrito ou os trabalhos não descrevem se contaram com algum mecanismo de suporte para as implementações das funcionalidades especificadas em seus experimentos. Ademais, as abordagens podem não ter considerado a reutilização de especificações comuns com sistemas existentes no mesmo domínio. Alguns projetos estão muitas vezes dispostos a discutir como superar problemas heterogêneos de rede e desafios particulares, tendo assim menos espaço para discutir semelhanças com outras soluções (JOSHI, 2011).

Araujo (2012) cita que o principal desafio é definir requisitos e funcionalidades que sejam úteis e fáceis de manipular durante o uso de novos dispositivos pelos agentes. Mas, existe outro desafio que precisa ser considerado que é a codificação do software que atenderá todas essas especificações criadas. Entre essas propostas supracitadas, algumas não apresentaram detalhadamente seus artefatos, como modelos, diagramas de casos de uso, diagramas de classes, e os seus requisitos, que são importantes para que outros desenvolvedores possam compreender melhor esses projetos e replicar as funcionalidades em novas propostas.

Portanto, há indícios de que existem pontos que podem ser abordados para apoiar o processo de desenvolvimento de sistemas e o uso dos recursos oferecidos atualmente pelos dispositivos móveis. Dessa forma, o problema que este trabalho visa abordar é a necessidade de um ferramental que viabilize a construção de sistemas móveis no domínio de emergências através da reutilização de software.

1.3 Hipótese

Este trabalho parte da hipótese de que um arcabouço de software que apoie e facilite o desenvolvimento de sistemas móveis com especificações baseadas nos aspectos do domínio de Gestão de Emergências, viabilizará o reuso de software para a construção de soluções tanto para o processo de resposta aos desastres quanto em outros domínios colaborativos similares.

1.4 Enfoque da Solução

Os engenheiros de software constataram com o passar do tempo que muitos sistemas compartilhavam semelhanças, e que o retrabalho do desenvolvimento de software poderia ser diminuído através da reutilização de especificações, modelos e código-fonte. A prática de reutilização de software implica um aumento da produtividade da equipe de desenvolvimento e da confiabilidade do produto, além de contribuir para a diminuição dos custos. Contudo, para que a reutilização seja possível, o desenvolvimento de software precisa passar por uma etapa anterior, onde são analisadas, projetadas e implementadas as características comuns em relação ao domínio em questão.

A proposta desse estudo é o desenvolvimento de um arcabouço para dispositivos móveis, baseado a partir de especificações abordadas em trabalhos prévios no domínio de emergências e no desenvolvimento de novas especificações. Para isso, serão definidos e descritos os requisitos e as análises básicas necessárias para atender os aspectos de desenvolvimento em dispositivos móveis. O arcabouço proposto deverá permitir que

elementos do processo de criação de um sistema (requisitos, modelagens, código, etc.) sejam reaproveitados para o desenvolvimento de outros sistemas móveis.

Será dado um enfoque no uso da *Unified Modeling Language* (UML) e da programação orientada a objetos. Assim, o arcabouço descreverá a arquitetura de um sistema orientado a objeto, seus tipos de objetos e como esses objetos interagem. Essa descrição é feita na forma de um conjunto de classes, uma para cada tipo de objeto do domínio do sistema, sendo os modelos de interação e associação entre os objetos uma parte importante do arcabouço. Um arcabouço de software normalmente utiliza diversos padrões de projeto na sua construção, trazendo um benefício para futuras expansões. As partes invariantes de um domínio são implementadas no arcabouço e reusadas nas instanciações. O arcabouço também permite aos desenvolvedores desprender-se das complexidades técnicas, já que foi definida uma arquitetura parcialmente implementada e encapsulado os detalhes dessa implementação (PIMENTEL; FUKS, 2011).

Dessa forma, a utilização metodológica de arcabouço na modelagem, especificação e implementação de sistemas, tem por finalidade aperfeiçoar o processo de desenvolvimento de software, garantindo menor esforço e maior qualidade do produto final.

1.5 Estrutura da Dissertação

Esta dissertação está organizada em oito capítulos e cinco apêndices que objetivam descrever os aspectos teóricos e práticos relacionados ao desenvolvimento deste trabalho.

O Capítulo 2 apresenta os conceitos, as fases e os desafios encontrados no domínio de Gestão de Emergências.

O Capítulo 3 analisa as características comuns dos sistemas móveis que favorecem a colaboração entre as equipes durante as situações de emergências.

O Capítulo 4 relaciona as principais técnicas de reuso de software para o conhecimento básico sobre os termos utilizados neste estudo.

No Capítulo 5, o arcabouço proposto é detalhado com base na metodologia de desenvolvimento de software que envolverá as principais etapas nesse processo: o levantamento de requisitos, o projeto, a implementação e o teste do arcabouço.

O Capítulo 6 aborda uma avaliação qualitativa por desenvolvedores de software e especialistas do domínio com o intuito de validar a proposta deste trabalho.

O Capítulo 7 discute alguns trabalhos relacionados ao tema.

O Capítulo 8 expõe as considerações finais e os trabalhos futuros.

O Apêndice A é um quadro do levantamento das funcionalidades identificadas dos sistemas de informação de Gestão de Emergências abordados no Capítulo 3.

O Apêndice B apresenta os diagramas de casos de uso e suas descrições para as funcionalidades do arcabouço.

O Apêndice C apresenta os diagramas de sequências para os casos de uso descritos no Apêndice B.

O Apêndice D é um dicionário de atributos das entidades do modelo do banco de dados do arcabouço.

Os Apêndices B, C e D estão relacionados com o Capítulo 5 deste trabalho.

O Apêndice F apresenta os questionários e os exercícios estabelecidos para a avaliação do arcabouço discutido no Capítulo 6.

2 Domínio de Gestão de Emergências

Este capítulo tem por objetivo apresentar os princípios relacionados à Gestão de Emergências. São apresentados os conceitos, a descrição do conjunto de fases, a estrutura de comando nas situações de emergência e os desafios encontrados nesse domínio.

2.1 Emergência e Desastre

Na literatura os termos emergência e desastre muitas vezes são usados de forma intercambiável. A emergência é um evento inesperado que ameaça as pessoas, a propriedade, a continuidade dos negócios, ou a comunidade, exigindo ações imediatas, esforços e recursos locais para minimizar suas consequências adversas ou salvar vidas (PORFIRIEV, 1995; JONES, 2000; TANGEN, 2008).

O desastre é um evento, natural ou produzido pelo homem, súbito ou progressivo, que afeta com gravidade a comunidade, manifestando o rompimento ou destruição, total ou parcial, do sistema social ou dos meios de subsistência. O desastre impõe uma dedicação de recursos além do âmbito normal de uma jurisdição ou ramo do governo para enfrentá-lo através de medidas excepcionais ou para restaurar a estabilidade dessa comunidade (CARTER, 1991; PORFIRIEV, 1995; CARROLL, 2001). A Administração Federal de Rodovias dos Estados Unidos¹ (FHWA) sintetiza a sua definição pelas seguintes características (FHWA, 2006):

- **Repentino:** imprevisível;
- **Escala:** tem grande impacto e envolve uma grande parte da comunidade;
- **Resposta institucional:** além da capacidade normal para lidar com o evento e envolvendo muitas instituições;
- **Prolongada duração do efeito:** impacto não se dissipa rapidamente;
- **Incerteza no comportamento de resposta:** fora da faixa habitual de experiência para o tratamento do evento.

O uso comum dos termos emergência e desastre pode ser confuso. Mas para facilitar o entendimento, pode-se considerá-los como duas extremidades de uma escala, na qual um incidente é comparado pela quantidade de recursos para lidar com ele ou pelo seu tamanho.

¹ Federal Highway Administration: <https://www.fhwa.dot.gov/>.

Numa extremidade desta escala, as emergências são, geralmente, incidentes pequenos que são resolvidos utilizando recursos locais. E na outra extremidade, os desastres tipicamente extrapolam os limites geográficos e políticos de comunidades locais e exigem um nível superior de resposta e recuperação que elas podem proporcionar (FEMA, 2002). “Quando algo ruim e/ou inesperado acontece, ele pode ser chamado uma emergência, um desastre ou uma crise dependendo da magnitude do evento” (CDC, 2002).

Os desastres podem ser classificados em dois tipos: os naturais e os tecnológicos. “Um desastre natural é uma perturbação grave a uma comunidade ou região, causada pelo impacto de um evento natural de início instantâneo que ameaça ou cause morte, lesão ou dano à propriedade ou ao meio ambiente” (DOTARS, 2002). Tal perturbação grave pode ser causada por um, ou a combinação, dos seguintes eventos: furacão, tornado, tempestade, ciclone, inundação, maremoto, tsunami, terremoto, erupção vulcânica, deslizamento de terra, tempestade de neve, secas, incêndio florestal, ou qualquer catástrofe com mudança violenta, repentina e destrutiva no ambiente sem causa da atividade humana (DOTARS, 2002; EEA, 2007).

Os desastres tecnológicos são eventos “artificiais devido a uma avaria súbita ou lenta, falha técnica, erro ou ato humano voluntário ou involuntário que provoca destruição, morte, poluição, e danos ambientais” (GUNN, 1990). Esses desastres podem ser causados por derramamentos de resíduos nucleares, radiológicos, substâncias tóxicas, ou acidentes com materiais perigosos, poluição, epidemias, acidentes em meios de transporte terrestres, marítimos ou aéreos, explosões, incêndios urbanos, perturbação da ordem pública, terrorismo, etc., “que surgem a partir de nosso uso individual e coletivo de tecnologia” (CUTTER, 1993). “Dedutivamente, a humanidade é responsável pelas consequências de suas ações, bem como de suas omissões” (DOMBROWSKY, 1995).

2.2 Gestão de Emergências

A Agência Federal de Gestão de Emergência dos Estados Unidos² (FEMA) define que Gestão de Emergência (GE) é a gerência encarregada de criar a estrutura dentro das quais as comunidades reduzem a vulnerabilidade aos riscos e lidam com desastres. Ela viabiliza a integração coordenada e colaborativa de todos os interessados nas fases de gestão relacionadas com os riscos naturais e tecnológicos. A Gestão de Emergência tem como objetivo tornar as comunidades mais seguras através da organização das atividades

² Federal Emergency Management Agency: <http://www.fema.gov/>.

necessárias para construir, manter e melhorar a capacidade de prevenção, preparação, resposta e recuperação aos desastres (FEMA, 2007a, 2007b).

A GE é “todo o processo de planejamento e intervenção de resgate para reduzir o impacto de emergências, bem como as medidas de resposta e recuperação, para atenuar as consequências sociais, econômicas e ambientais significativas para a comunidade” (NIMPUNO; HILMAN, 1998). Ela envolve planos e acordos institucionais para engajar e orientar os “esforços do governo, organizações não-governamentais, agências voluntárias e privadas de forma abrangente e coordenada para responder a todo o espectro de necessidades de emergência” (UNISDR, 2009).

Drabek e Hoetmer (1991) descrevem que a “GE é a disciplina e profissão de aplicação da ciência, tecnologia, planejamento e gestão para lidar com eventos extremos que podem ferir ou matar um grande número de pessoas e fazer grandes danos à propriedade”, interrompendo a vida da comunidade. Já para Haddow, Bullock e Coppola (2007) e Waugh (1999) é a disciplina que trata de riscos e a prevenção de riscos, de modo que as sociedades possam conviver com riscos ambientais e tecnológicos e lidar com os desastres que eles causam.

O Departamento de Saúde e Serviços Humanos dos Estados Unidos³ (HHS) define como “a ciência da gestão de sistemas complexos e equipes multidisciplinares para lidar com eventos extremos, através de todos os perigos” (HHS, 2004).

Este trabalho segue a definição da GE descrita pela FEMA anteriormente. A FEMA (2007a) estabelece alguns princípios básicos que compõem a GE, destacando que eles precisam ser claramente entendidos e promovidos, pois são universais, ou seja, transcendem fronteiras geopolíticas.

A GE deve ser abrangente, isto é, deve considerar todos os riscos, todas as fases, todos os impactos e todas as partes interessadas relevantes para uma emergência. Todos os riscos devem ser considerados no processo completo de avaliação de risco e priorizados com base na probabilidade de sua ocorrência. Assim como todos os impactos ou consequências previsíveis relativas a esses riscos também devem ser analisados e tratados. A GE bem-sucedida coordena atividades em todas as fases de gerenciamento, desde a prevenção e o planejamento eficaz contra os desastres, até a reação direta a eles e a restauração da comunidade, envolvendo entidades de todos os níveis de setores públicos e privados.

³ Department of Health and Human Services: <http://www.hhs.gov/>.

A GE tem que ser progressiva e não somente reativa, aplicando esforços em medidas preventivas para a criação de uma comunidade resistente e resiliente aos desastres, por exemplo, com ações de gestão ambiental, aplicação de leis para o crescimento urbano de maneira organizada, e no planejamento de estratégias que reduzam as vulnerabilidades.

Os princípios da gestão de risco precisam ser usados na GE. Isso significa numa definição de políticas e na atribuição de prioridades para a identificação de riscos naturais e artificiais que podem ter efeito significativo sobre a comunidade ou organização. A análise da natureza e do potencial impacto da emergência é responsável pela alocação adequada de recursos para lidar com uma maior variedade de riscos.

A GE deve assegurar uma unidade de esforço entre todos os níveis de governo e todos os elementos de uma comunidade. A integração vertical e horizontal permite a sincronização de ações desde níveis locais até federais, construindo parcerias entre vários setores para a proteção da comunidade.

A colaboração deve ser vista como uma atitude ou uma cultura organizacional que caracteriza o grau de unidade que existe dentro da GE. O compromisso com a colaboração possibilita que papéis e funções essenciais de coordenação funcionem nesse ambiente, criando uma atmosfera de equipe e relações sinceras entre os envolvidos. Isso contribui na comunicação, favorece o consenso e estimula a confiança entre eles.

A coordenação é um princípio aplicável em todas as fases da GE. Através da coordenação se obtém o acordo entre as diferentes entidades e autoridades sobre um objetivo comum, certificando que ações independentes cumpram esse objetivo. Ao sincronizar as atividades entre todas as partes interessadas é possível alcançar o sucesso no planejamento e atividades operacionais para uma emergência.

Devido à variedade de pessoas e responsabilidades, a GE requer flexibilidade. A implementação de uma maior variedade de táticas e procedimentos durante a emergência viabiliza a criação de soluções criativas para resolver os problemas e atingir as metas. Isso permite a adaptação rápida em situações de constante mudança e incerteza.

Além disso, a GE demanda o profissionalismo das pessoas envolvidas. A valorização e melhoria contínua da educação, formação e experiência favorece o comprometimento delas com a GE.

2.3 Fases da Gestão de Emergências

As atividades relacionadas à Gestão de Emergência estão agrupadas em quatro fases interconectadas em função do tempo, para todos os tipos de desastres. Além disso, elas estão

associadas entre si, ou seja, formando um ciclo onde cada uma envolve diferentes tipos de habilidades (NGA, 1979). A Figura 1 apresenta esse ciclo com as quatro fases: mitigação, preparação, resposta e recuperação. Uma abordagem para cada fase é realizada nas subseções seguintes.



Figura 1: Quatro fases da Gestão de Emergências

2.3.1 Mitigação

A fase de mitigação ou prevenção inclui qualquer atividade que, de fato, elimine ou minimize a probabilidade de ocorrência de uma emergência. Inclui também atividades de longo prazo destinadas a reduzir os potenciais efeitos adversos de uma emergência, como a criação de programas e/ou legislações para gestão da segurança pública, para o uso adequado do solo, o monitoramento ambiental, inspeções e operações de vigilância ou segurança, construção de abrigos, dentre outros. A mitigação é o método mais proativo e bem-sucedido para a contenção de prejuízos físicos e financeiros causados pelas emergências, pois possibilita antecipadamente o controle dos riscos e seus efeitos (NGA, 1979; HSC, 2006).

As atividades de mitigação são muitas vezes desenvolvidas de acordo com as lições aprendidas a partir de emergências anteriores, demonstrando-se mais eficazes quando baseada num plano abrangente e detalhado que deve ser elaborado antes que ocorram novas emergências e ao fornecer valores e informações para o público para a criação de comunidades mais seguras (FEMA, 2008).

2.3.2 Preparação

As atividades de preparação são necessárias na medida em que as atividades de mitigação não podem prevenir uma emergência. Na fase de preparação, os governos, as organizações e a comunidade desenvolvem uma combinação de atividades de planejamento, organização, treinamento, exercícios e avaliação para salvar vidas em eventuais operações de resposta à emergência (NGA, 1979; DHS, 2004a).

O objetivo principal é a manutenção de planos, procedimentos, programas e sistemas para melhorar a disponibilidade e garantir uma coordenação eficaz em uma situação de crise. Conseqüentemente, medidas e esforços tomados previamente podem assegurar uma resposta adequada aos impactos dos desastres, estabelecendo atividades em quantidades suficientes para a execução de tarefas e missões (DHS, 2007).

Dentre as principais atividades estão: (1) a pré-criação de postos de comando, áreas de preparação e outras instalações; (2) o estabelecimento de diretrizes e protocolos necessários para promover a interoperabilidade entre agências a nível federal, estadual e local; (3) a manutenção dos recursos físicos e humanos; (4) o planejamento, o treinamento e a realização de exercícios de resposta; (5) a elaboração de normas de qualificação e certificação de pessoal e equipes; (6) a aquisição de equipamentos; (7) o armazenamento de alimentos e suprimentos médicos; (8) a instalação de sistemas de alerta e/ou a emissão antecipado de alertas; e (9) a evacuação temporária de pessoas e bens de um local ameaçado.

2.3.3 Resposta

As atividades de resposta ocorrem imediatamente após o início de uma emergência, isto é, todas as ações são imediatas e de curto prazo. O objetivo principal é fornecer assistência emergencial para as vítimas, atender as necessidades básicas humanas, preservar a propriedade, o ambiente, e a estrutura social, econômica e política da comunidade. As atividades são efetuadas para o reconhecimento, a avaliação, e a declaração da emergência, assim como na execução dos planos de preparação e na gestão de recursos. Elas também buscam reduzir a probabilidade de danos secundários e acelerar as operações de recuperação com a notificação desses prejuízos gerados pela emergência para as autoridades competentes (NGA, 1979; TANGEN, 2008; DHS, 2004b; DHS, 2007).

Conforme a situação do desastre evolui, os serviços de resposta podem incluir a aplicação de inteligência e outras informações destinadas a limitar os efeitos, as conseqüências ou outros resultados desfavoráveis (DHS, 2009). Dentre as principais atividades estão: (1) as investigações da natureza e da origem da ameaça; (2) a manutenção da

consciência situacional entre equipes de resposta; (3) as operações de busca e salvamento; (4) os serviços de emergência médica e funerário; (5) as operações de segurança e vigilância que visam interromper atividades ilegais e apreensão de criminosos; (6) a proteção da cena da emergência para investigar e recolher provas; (7) os serviços de transporte e logística; (8) a mobilização, o monitoramento e a desmobilização dos recursos físicos e humanos; (9) a remoção de ameaças ao meio ambiente; (10) a oferta de abrigos, habitação e alimentação em massa; (11) a descontaminação, a imunização ou o isolamento após um ataque químico, biológico ou radiológico; (12) a restauração dos serviços essenciais (energia elétrica, água, esgoto, telefone); e (13) o fornecimento de bens e serviços necessários através de contratos ou doações pelo setor privado.

2.3.4 Recuperação

As atividades de recuperação são desempenhadas até que todo o sistema social retorne ao normal ou aos padrões mínimos de funcionamento. A recuperação compreende a reconstrução da infraestrutura física da área afetada e na restauração econômica, política, e do bem-estar emocional da comunidade, além do meio ambiente. Essas atividades também incluem o desenvolvimento, a coordenação e a execução de planos estratégicos e serviços para reconstituir as operações do governo e do setor privado, apoiando na restauração completa e na melhoria desse sistema social. Tudo isso apoia o crescimento de uma população de forma sustentável e na aquisição de sua confiança (NGA, 1979; DOTARS, 2002; HHS, 2004; DHS, 2007).

As atividades de recuperação podem ser de curto prazo, mas muitas vezes elas podem se estender muito tempo após o desastre (DHS, 2004b). As atividades típicas podem incluir: (1) a limpeza e remoção de detritos; (2) a reparação e substituição de instalações públicas (estradas, pontes, edifícios públicos, escolas, hospitais, organizações sem fins lucrativos, etc.); (3) a restauração de serviços de energia elétrica, água, esgoto, telefone; (4) a avaliação do desastre para identificar lições aprendidas; (5) a comunicação e a divulgação de relatórios e documentos pós-incidente; (6) a habitação temporária para as vítimas; (7) a terapia, saúde mental e outras formas de assistência para as vítimas; (8) a assistência jurídica e replanejamento da comunidade; (9) o empréstimo a juros baixos para ajudar indivíduos e empresas com reconstrução em longo prazo; e (10) os programas de estabilização econômica de longo prazo.

2.4 Comando e Controle

Diante da diversidade de atividades na fase de resposta, fica evidente a existência de um desafio presente nesse contexto: a integração dos diferentes tipos de sistemas de gestão e abordagens utilizadas por todos os níveis de governo em uma resposta abrangente e unificada para atender às necessidades exclusivas e exigências de cada emergência. Desta forma, a gestão das ações federais, estaduais e municipais de resposta requer uma flexibilidade para abordar toda a gama de capacidades e recursos de resposta (DOD, 2001).

Comando e Controle (C2) é “o exercício da autoridade e direção devidamente designada por um comandante sobre as forças atribuídas e anexadas no cumprimento da missão” (DOD, 2010) que tem como objetivo realizar esta integração através de uma estrutura unificada para a elaboração, organização e execução de planos e ordens de acordo com a complexidade da emergência (DOD, 2001). Logo, esse comandante é responsável pelo controle de forças e operações, coordenação de pessoal, equipamentos, comunicações, instalações e procedimentos empregados na fase de resposta de forma centralizada. O comandante é o oficial mais graduado no cenário do incidente e delega tarefas aos demais membros de sua equipe.

Em geral, a função de comando pode ser realizada de duas formas (DHS, 2004a):

- **Comando Simples:** Quando um incidente ocorre numa única jurisdição e não há nenhuma sobreposição funcional de agências. Assim, um único comandante deve ser designado com a responsabilidade global de gerenciamento de emergências pela autoridade adequada;
- **Comando Unificado:** Unidade de comando significa que cada indivíduo tem um supervisor designado a quem eles se reportam no local da emergência. Este princípio assegura relacionamentos hierárquicos e eliminam equívocos causados por diretivas múltiplas e/ou conflitantes.

A necessidade de identificação de uma pessoa responsável pelo comando surgiu nos anos 70 durante o atendimento emergencial a uma sequência de incêndios florestais na Califórnia, Estados Unidos. Por tal motivo, a FEMA percebeu que o método mostrou-se apropriado para todos os tipos de emergências e passou a adotá-lo em qualquer emergência (SEITO et al., 2008).

Segundo Seito e outros (2008), não há uma legislação que defina claramente o comando nos locais de emergência no Brasil. Como regra geral, o comando das operações deve ser passado para a autoridade pública presente no local, estabelecendo um protocolo

entre as organizações de resposta para manter uma visão completa da situação de emergência. Uma análise detalhada sobre alguns dos principais modelos de C2 para o domínio de emergências é apresentado no trabalho de Padilha (2011).

2.5 Desafios de Comunicação na Fase de Resposta

A fase de resposta envolve e requer a colaboração de muitas pessoas. Para executar e coordenar suas atividades, elas devem contar com informações contextuais detalhadas e precisas sobre a situação atual da emergência para assumir o controle dos problemas encontrados no local, como por exemplo, a localização exata da emergência, a localização de hidrantes, as áreas perigosas, os hospitais, as delegacias, as redes de eletricidade e gás, os projetos de construção, os mapas, etc.

Algumas formas de comunicação podem ser utilizadas pelas organizações para transmitir estas atividades. Quando a área de atuação dos envolvidos é pequena, a voz se torna o meio mais empregado. Caso o ambiente apresente muito ruído ou tumulto é possível que a informação não seja compreendida. A corneta é uma opção que pode anunciar ações com toques padronizados e antecipadamente estabelecidos entre os agentes para áreas de trabalho maiores, porém é necessário que cada equipe possua um agente corneteiro, o que nem sempre é possível (ARAUJO, 2012).

No país, as organizações de defesa social, como Corpo de Bombeiros e Polícia Militar, utilizam o sistema de rádio UHF como o principal equipamento de comunicação nas emergências. Segundo Ibarra e outros (2012), esse sistema de rádio de um centro de comando e controle geralmente é capaz de enviar ou receber mensagens de voz dentro de uma área com um raio de trinta quilômetros. Para os caminhões de bombeiros e os dispositivos de rádio portáteis, a área de comunicação é entre dois e quatro quilômetros ao redor deles, dependendo da potência de transmissão do equipamento.

Existe ainda outra forma de comunicação que é por meio de satélites, permitindo abranger áreas muito superiores ao sistema de rádio. Essa opção pode ser encontrada em alguns países desenvolvidos localizados na Europa, por exemplo, na Alemanha. Entretanto o Brasil ainda não possui a capacidade tecnológica para uso de tal sistema de comunicação.

Durante um desastre, existe uma constante troca de informação entre a equipe de comando, responsável pela gestão das atividades, e a equipe de operação, responsável pela execução das atividades planejadas. Todavia, o canal de radiocomunicação na operação de resgate pode ficar congestionado, pois normalmente são apenas dois canais disponíveis para

suportar todas as atividades. Essa limitação se agrava quando a equipe de comando precisa coordenar mais de duas emergências simultaneamente (IBARRA et al., 2012).

Além disso, as mensagens podem ser transmitidas e recebidas fora de ordem lógica, misturando diversos assuntos. Assim, o volume de informações se torna um desafio para o processamento humano. A pressão por tempo, o estresse e a divisão da atenção dos indivíduos envolvidos em muitas atividades simultâneas também contribuem para ocasionar falhas de compreensão das mensagens (FERREIRA, 2011; BARR et al., 2011).

Mensagens textuais, mapas, fotos, vídeos são informações digitais relevantes para organizar e coordenar o processo de resposta a uma emergência, no entanto, este tipo de informação é necessário por diversas vezes, e não há possibilidade de serem entregues ou compartilhados através de um sistema de rádio (IBARRA et al., 2012).

Esses desafios de comunicação motivaram a pesquisa de novas alternativas para apoiar estas equipes de resposta no compartilhamento de dados através de sistemas computacionais de maneira mais adequada.

3 Sistemas de Informação na Gestão de Emergências

Nas últimas décadas os computadores e a tecnologia móvel têm atraído o interesse da sociedade com a sua crescente adoção no cotidiano das pessoas. Com a redução do tamanho e do preço, surgiram novas possibilidades para o desenvolvimento de soluções em variados cenários de aplicação, inclusive para apoiar as atividades de Gestão de Emergência.

As características citadas na seção 2.5 evidenciam a complexidade na comunicação e colaboração em situações críticas e a manutenção das informações pelas equipes envolvidas nesse contexto. Com o avanço da computação, eleva-se a possibilidade de que sistemas computacionais possam ser concebidos para beneficiar tanto os agentes em campo quanto a equipe de comando, algo que antes não poderia ser construído, tendo em vista as limitações dos computadores.

O uso da tecnologia pode melhorar o entendimento comum das equipes de emergência em suas atividades através do acesso às informações, tornando a resposta às emergências mais eficiente. Desta forma, os sistemas para apoiar a colaboração nesse domínio têm evoluído motivados principalmente pelo desenvolvimento e integração de dispositivos não convencionais, como sensores, redes sem fio e dispositivos móveis.

Recentemente, vários projetos surgiram com o objetivo de apoiar a colaboração e a tomada de decisão em gestão de emergências. Dentre os trabalhos abordados na literatura, alguns se destacam com a utilização da computação móvel, como é o caso dos projetos descritos a seguir.

3.1 NOAH

Emergency Organisation and Administration Aid (NOAH) é um sistema que apoia os médicos na documentação inicial dos dados do paciente no local da emergência (SCHÄCHINGER et al., 2000). A coleta dos sinais vitais e outros dados clínicos é realizada em ordem cronológica e registrada com a devida validação e integridade num banco de dados. A documentação eletrônica permite o registro completo e imediato sobre os tipos de ferimentos ocorridos e a terapia apropriada a ser realizada para cada ferimento (Figura 2). O sistema faz a conexão da ambulância e os médicos com o centro de comando e o hospital. Dessa forma, um registro eletrônico do paciente é processado e transmitido eletronicamente do local de emergência para o hospital. Isso permite diminuir a ocorrência de erros na

transferência de vítimas e reduz a necessidade de comunicação por voz. As informações dos pacientes de uma emergência crítica podem estar disponíveis cerca de 20 minutos mais cedo para o hospital, permitindo que medidas imediatas para o atendimento ao paciente sejam realizadas antes da sua chegada e melhorando o prognóstico. Além disso, o hospital pode ser escolhido baseado na sua disponibilidade de leitos e equipamentos. O sistema também provê uma ferramenta de análise para a identificação do tratamento no local, caso o ferimento não seja de fácil reconhecimento, como um envenenamento.

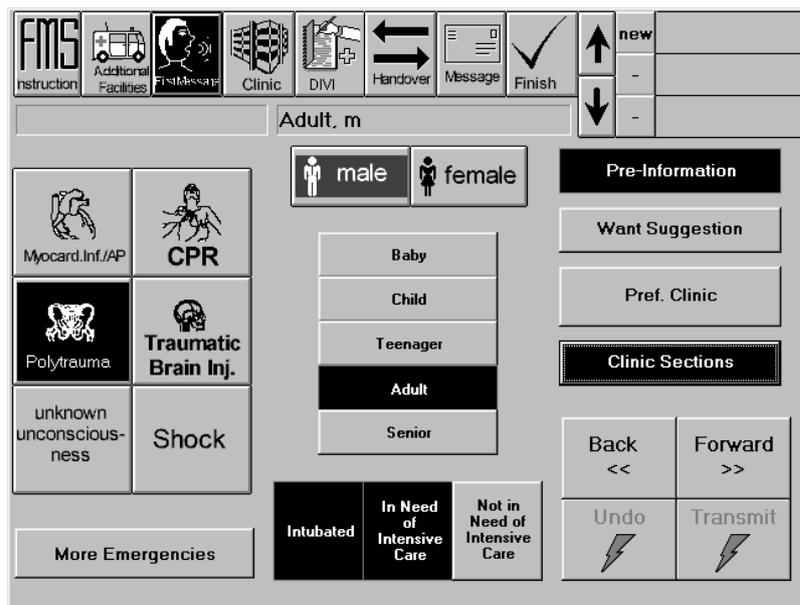


Figura 2: Interface do Sistema NOAH

3.2 MobileMap

MobileMap é um sistema móvel colaborativo que usa dados do GPS para localizar o local de uma emergência e recursos materiais e humanos disponíveis (IBARRA et al., 2012). O sistema proporciona a interação entre um bombeiro e centro de comando, a fim de recuperar e reportar informações sobre a emergência. Ele fornece comunicação e compartilhamento de dados sobre a ameaça e a área afetada, na organização do combate aos incêndios e na gravação e coleta de informações de campo, bem como ajuda o comandante do incidente, que está no comando do processo de resposta, a fazer e comunicar suas decisões. Essa ferramenta também indica a localização dos principais recursos que podem ser usados para apoiar o processo de resposta, tais como bombeiros, hidrantes, hospitais, delegacias, caminhões de bombeiros, etc (Figura 3). MobileMap possibilita aperfeiçoar o processo de aprendizado dos bombeiros com o resultados obtidos em eventos anteriores, assim eles realizam tomada de decisões melhores e se sentem mais confiantes. Além disso, reduz a necessidade de comunicação via rádio.

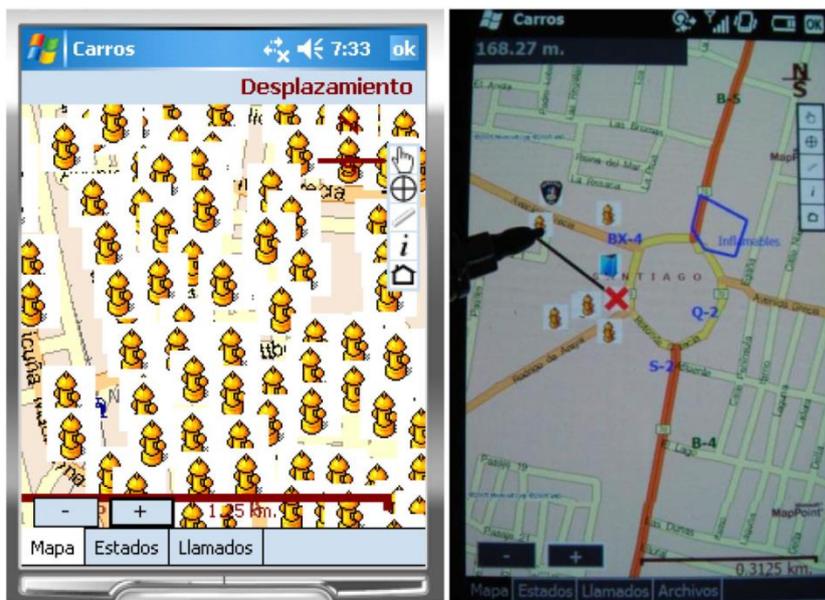


Figura 3: Interfaces do Sistema MobileMap

3.3 CCCMS

Car Crash Crisis Management System (CCCMS) é um sistema responsável por uma resposta coordenada às situações de emergência após um acidente de carro (KIENZLE; GUELFY; MUSTAFIZ, 2010). Esta é uma aplicação que auxilia as autoridades que lidam com acidentes de carro de forma mais eficiente, centralizando todas as informações relacionadas ao acidente, facilitando a comunicação entre os agentes, e propondo uma estratégia de resolução da crise de forma adequada. Através dele é possível despachar os recursos humanos para o local do acidente, e reavaliar a estratégia de resposta quando novas informações sobre o incidente são capturadas. Portanto, o CCCMS apoia o processo de resposta por orquestrar a comunicação e coordenação entre todas as partes envolvidas que devem lidar com a crise, aumentando fluxo de informações compartilhadas e facilitando a gestão das atividades. A alocação apropriada dos recursos humanos permite o resgate de vítimas e o seu transporte mais rápido para os hospitais. Além disso, ajuda na recuperação do cenário da emergência com a remoção dos obstáculos e veículos danificados do local.

3.4 DUMBONET

Digital Ubiquitous Mobile Broadband OLSR Network (DUMBONET) combina tecnologias de rede via satélite, Ad hoc e internet convencional para fornecer ferramentas aos seus usuários para a coordenação, comunicação e gestão da informação (KANCHANASUT et al., 2007). O sistema propõe a troca de informações multimídia em tempo real nas operações de busca e resgate de vítimas em áreas afetadas por desastres (Figura 4). Ele possibilita a

conexão de várias áreas afetadas e o centro de comando através de *links* via satélite. Cada uma dessas áreas tem a sua própria rede móvel Ad hoc para a comunicação entre os socorristas e cada unidade móvel pode ser um computador portátil ou um Assistente Pessoal Digital (PDA).

DUMBONET se concentra em apoiar os comandantes que trabalham em salas de comando e controle que precisam para tomar decisões e fornecer coordenação de alto nível das equipes de socorristas que trabalham nessas áreas afetadas. Deste modo, o sistema interliga os envolvidos para se comunicarem e coordenarem suas ações por meio de mensagens textuais, áudio e vídeo. Logo, ele foi projetado para operações colaborativas e simultâneas de resposta, alcançando uma disponibilidade confiável da rede de comunicação que é necessária em cenários previstos de emergência onde não há uma infraestrutura adequada.



Figura 4: Interface do Sistema BUMBONET

3.5 DMT

Disaster Management Tool (DMT) é um sistema capaz de oferecer auxílio no atendimento de missões de reconhecimento na fase de resposta e ele agrega um conjunto de funcionalidades para acompanhamento das tarefas nas etapas subsequentes, como a coordenação de várias equipes de socorro no local do desastre (ANGERMANN et al., 2009). O DMT é composto por três partes que se complementam. A primeira parte é a de simulação, que estabelece funções como a estimativa rápida de prejuízos e possíveis vítimas, a simulação de progressão futura do desastre, como propagação do fogo, e as consequências de decisões tomadas durante exercícios praticados anteriormente. A segunda parte envolve a análise de

danos obtidos do local do desastre e de dados do providos pelo centro de comando para a tomada de decisão, bem como para a atribuição de tarefas e recursos de forma eficiente durante a fase de resposta à emergência. A terceira parte inclui as funções de comunicação para a integração dos agentes na realização das tarefas, administrando a agregação, seleção e distribuição de informações relevantes entre eles. O sistema também conta com uma interface gráfica para a visualização de informações geográficas (Figura 5).

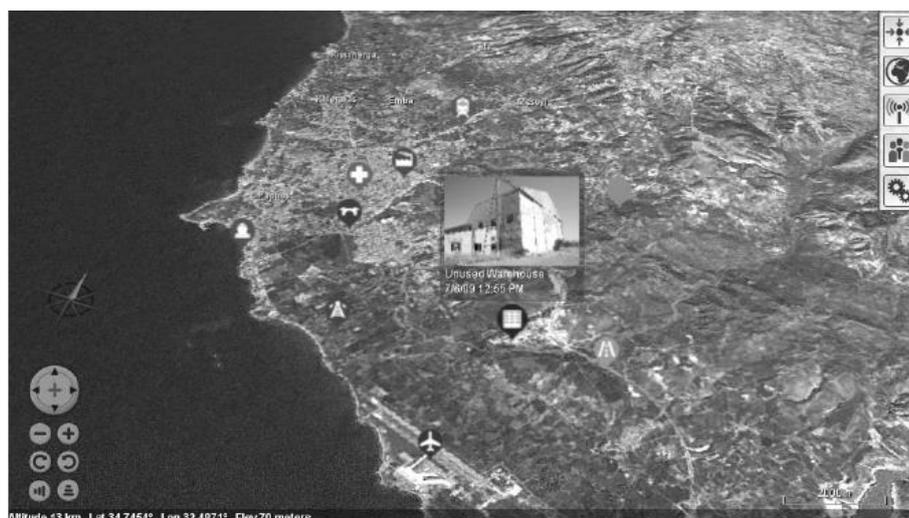


Figura 5: Interface do Sistema DMT

3.6 MIKOBOS

Mobile Information and Communication System for Public Safety Organizations (MIKOBOS) é um sistema de resposta de emergência para estender a comunicação do corpo de bombeiros para o domínio móvel (MEISSNER et al., 2006). Ele permite ao pessoal em campo receber ordens de comandantes e acessar informações que só poderiam estar disponíveis para eles por meio de troca de mensagens de voz em transmissões de rádio. Dessa forma, esses agentes que se encontram encarregados nas ações de resposta, podem interagir eletronicamente com colegas, chefes ou peritos para o compartilhamento de dados de sensores, relatórios de situação com imagens relacionadas ao incidente, instruções de manuseio de materiais perigosos, ou os recursos disponíveis para ajudar em suas atividades (Figura 6). Os membros das equipes de resposta podem estar espalhados em diferentes locais, porém o sistema fornece informações de sua localização com a utilização de navegação via GPS, ajudando um socorrista a encontrar rotas para o local da emergência. Além disso, durante o transporte das vítimas, as informações coletadas podem ser transferidas para os hospitais através da arquitetura de rede implantada para o MIKOBOS.



Figura 6: Interfaces do Sistema MIKoBOS

3.7 GeoBIPS

O sistema móvel de gestão de crises Geographic Broadband Integration for Public Services (GeoBIPS) permite que os serviços de emergência possam coletar e combinar dados dinâmicos do local do incidente com dados estáticos armazenados em servidores locais ou distantes, dispondo aos agentes de resposta o acesso rápido e fácil de informações atualizadas em situações de emergência (LUYTEN et al., 2006).



Figura 7: Interface do Sistema GeoBIPS

Desta forma, as decisões podem ser realizadas para poupar vidas durante os momentos críticos pela intervenção desses agentes. O sistema é usado pelos membros da equipe de controle para compartilhamento de dados e enviá-los através de redes sem fio para o oficial de comando. O GeoBIPS permite ao oficial de comando adquirir informações atualizadas dos agentes em campo, por exemplo, através de transmissão de vídeo digital (Figura 7). Ao mesmo tempo, o oficial de comando pode enviar dados úteis e instruções aos membros da equipe de resposta, por exemplo, uma ficha sobre os materiais perigosos que serão

encontrados no cenário do incidente. Por outro lado, o sistema possibilita a transmissão de informações para o centro de crise para a tomada de decisão e receber desse centro outras informações, como planos de intervenção.

3.8 WORKPAD

O projeto WORKPAD provê uma infraestrutura de software e comunicação para apoiar os operadores em situações de emergência, concentrando-se nas fases mais críticas, a resposta e a recuperação de curto prazo (HUMAYOUN et al., 2009). O sistema tem um papel fundamental na realização dos objetivos das equipes de trabalho, com foco no gerenciamento de tarefas, suporte a tomada de decisão e planejamento, e uma interface de usuário de fácil entendimento para dispositivos móveis (Figura 8).

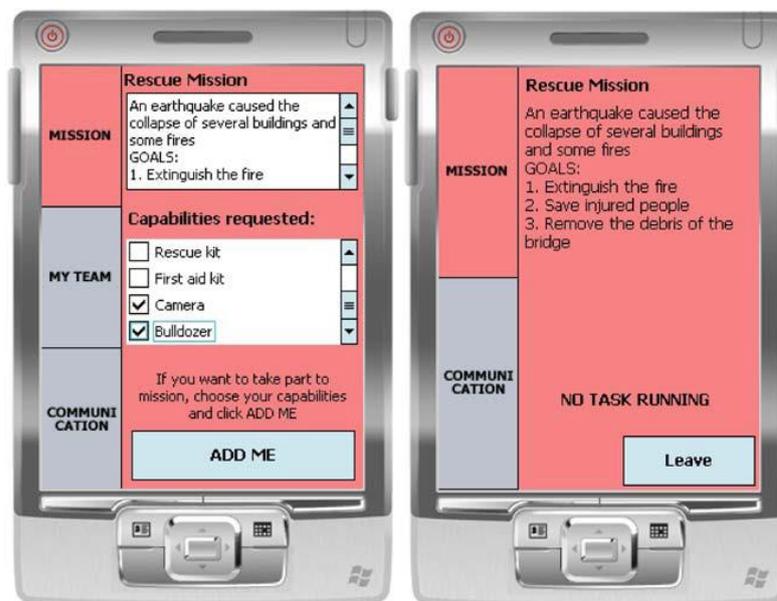


Figura 8: Interfaces do Sistema WORKPAD

Outras funções também foram desenvolvidas para facilitar a execução das atividades dos operadores, como o registro de informações contextuais, o registro de fotos e recuperação de informações relevantes. Ademais, o WORKPAD permite aos usuários ter uma visão geral da área afetada e a visualização de pontos de interesse exibidos no mapa. Os agentes de resposta têm uma percepção imediata das tarefas e suas posições no mapa, facilitando a execução das tarefas planejadas e o monitoramento eficiente dessas tarefas pelos gestores da crise. Quando as tarefas são atribuídas, metáforas visuais permitem ao operador detectá-las mais facilmente, por exemplo, as tarefas próximas da posição do operador ganham destaque ao piscar na tela.

3.9 SisC2Celular

SisC2Celular é um sistema de informação criado para gerenciar o processo de resposta a emergência baseado em Sistemas de Comando e Controle (PADILHA, 2010). O telefone celular é utilizado como plataforma principal para capturar, transmitir e receber informações entre as equipes de comando e controle para alcançar um entendimento comum mais rápido e uma melhor consciência situacional, aumentando a agilidade na tomada de decisão. As decisões tomadas pelo comando são enviadas para as equipes de resposta de campo através de tarefas operacionais. Essas equipes recebem instruções sobre a situação de emergência e a cada novo fato encontrado os participantes decidem quais as ações que precisam ser desempenhadas (Figura 9).



Figura 9: Interfaces do Sistema SisC2Celular

Como a situação é dinâmica, algumas instruções podem ser alteradas para se adaptar a uma nova realidade. Decididas às linhas de ação, os participantes encaminham solicitações e relatórios para o centro de comando e o resultado dessas ações pode modificar o contexto da situação continuamente. A troca de mensagens pré-definidas entre as equipes facilita o processo de compartilhamento da informação e uma designação mais rápida dos recursos disponíveis. A capacidade de registrar as informações do incidente permite realizar análises das atividades que foram realizadas, possibilitando validar a eficácia das ações e o melhoramento do processo de resposta.

3.10 EMERGENCY

Mobile Decision Support in Emergency Situations (EMERGENCY) tem como objetivo explorar a alocação de recursos através de uma interface baseada em mapa para apoiar as ações do comandante de incidente na Polícia com dispositivos móveis (JOSHI, 2011). As tarefas, os policiais e o contexto desses policiais, ou seja, a gestão de recursos durante situações de emergência podem ser interpretadas de forma mais intuitiva. O sistema permite que o comandante mantenha facilmente uma visão geral da resposta através da apresentação baseada em mapa, contendo símbolos e metáforas em vez de apenas texto. Deste modo, a manipulação de recursos no local do incidente pode se tornar mais fácil e as informações são compreendidas rapidamente, exigindo menos concentração do comandante. Então, este projeto explora as alternativas encontradas em funções desenvolvidas para a visualização de informações em mapas em conjunto com as ações de alocação, realocação, controle e requisição de novos recursos para fornecer uma melhor gestão da resposta e o apoio à decisão em situações de crise (Figura 10).



Figura 10: Interfaces do Sistema EMERGENCY

3.11 Sistema de Emergência

O Sistema de Emergência (Sis.Emergência) é um sistema para a gestão de informações de crise que visa apoiar a colaboração e tomada de decisão entre a equipe de operação e o comandante de incidente (ARAUJO, 2012). Esse sistema tem como objetivo registrar as principais informações relacionadas aos atores envolvidos no procedimento de resposta, como dados relevantes sobre os socorristas, por exemplo, sua especialidade e suas

tarefas, e também os dados das vítimas encontradas no local do incidente, como o seu estado de saúde. Além disso, ele gerencia os dados da emergência, como o seu tipo, a origem, a intensidade, as condições climáticas e os dados capturados sobre as infraestruturas afetadas possibilita uma avaliação mais detalhada pelos gestores do incidente (Figura 11). O Sistema de Emergência foi construído para ser intuitivo e de fácil entendimento, visto que nem todos os bombeiros possuem uma vasta experiência no uso de smartphones ou tablets modernos. Por fim, é possível trocar mensagens entre os bombeiros através de um chat, tirar fotos e visualizar as principais informações coletadas em mapas digitais, conduzindo de forma mais eficiente o processo de resposta a emergências.

The image displays two side-by-side screenshots of a mobile application interface titled 'Sistema de emergência'. The left screenshot shows the 'Nova Emergência' (New Emergency) form, and the right screenshot shows the 'Nova Vítima' (New Victim) form. Both forms are designed for data entry on a mobile device, featuring text input fields, dropdown menus, and buttons for navigation and saving.

Nova Emergência (Left Screenshot):

- Nome: Emergência 1
- Data: 31/07/2012
- Tipo: Incêndio
- Origem: Escapamento de gas
- Intensidade: Moderada
- Evolução:
- Condições Climáticas: Chuva
- Observações:

Nova Vítima (Right Screenshot):

- Número de identificação: 001
- Nome: Ana
- Data e hora: 01/08/2012 12:06
- Telefone: 9500-0000
- Idade: 17
- Estado de saúde: Ferimentos leves
- Observações: Celular da mãe

Figura 11: Interfaces do Sistema de Emergência

3.12 ProRad

O Sistema de Proteção Radiológica (ProRad) consiste em apoiar o agente de resposta através da utilização de interface humano-computador amigável e bases de dados específicas embarcadas localmente no dispositivo móvel, disponibilizando informações, calculando e classificando categorias de riscos (SILVA, 2012). Além disso, ele permite fazer registros de informações e consultas a Web, possibilitando em curto intervalo de tempo obter um panorama geral da situação de uma emergência de natureza radiológica. O sistema tem como objetivo guiar o agente de resposta em suas principais ações, por exemplo, a verificação do nível de radiação no local do acidente e a classificação da categoria da fonte de radiação, para dimensionar uma área de isolamento adequada de acordo com a gravidade do desastre (Figura 12). Ele também auxilia no cadastro das vítimas e no levantamento de riscos eminentes para as pessoas e o meio ambiente, em função dos dados climáticos e geográficos providos pelo

sistema. Assim, o ProRad apoia as atividades desses agentes de resposta radiológica ao reduzir a quantidade de carga cognitiva, fornecer acesso à dados locais para suporte de suas ações, facilitar a investigação do cenário atual do desastre e, conseqüentemente, reduzir sua exposição à radiação.

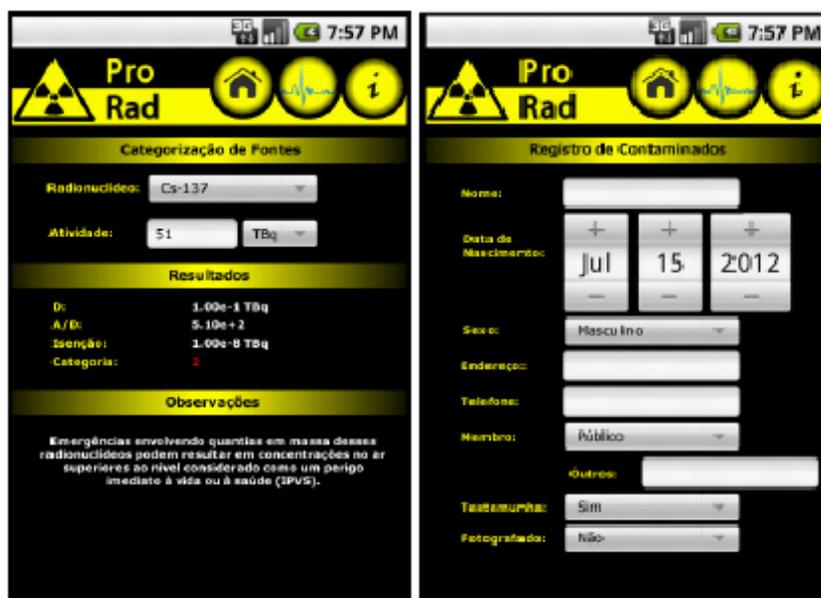


Figura 12: Interfaces do Sistema ProRad

3.13 Classificação dos Sistemas de Informação

Nota-se que vários trabalhos desenvolveram propostas significativas para a resolução de problemas de comunicação e compartilhamento de informações no domínio de emergências. Nas subseções a seguir são apresentadas classificações baseadas nesse conjunto de trabalhos.

3.13.1 Classificação dos sistemas segundo suas funcionalidades

Nilsson e Stølen (2011) conduziram uma investigação com o propósito de identificar as necessidades comuns dos agentes envolvidos na resposta de emergências e classificá-las em categorias de funcionalidades de sistemas de informação que suportam as atividades desses agentes.

Estudos empíricos foram realizados para encontrar as possíveis categorias de funcionalidade que são compartilhadas por pelo menos dois atores de resposta. Esses estudos envolveram um ou mais atores de cada organização, como Polícia Militar, Corpo de Bombeiros, Ambulâncias de Resgate, Cruz Vermelha e outros, através das seguintes atividades:

- Exercício de treinamento de operação de resgate em avalanches;

- Exercício de treinamento de operação de resgate envolvendo incêndio em um navio de armazenamento de gás;
- Exercício de treinamento organizado pelo serviço de ambulância para um acidente que envolve desafios especiais no acesso e transporte de vítimas;
- Entrevistas com líderes experientes e agentes de serviços da polícia, bombeiros e ambulância.

As informações que descrevem as necessidades foram coletadas por entrevistas e observações. As categorias surgiram como resultado da análise das atividades executadas pelos líderes locais e as informações envolvidas na realização dessas tarefas. Dessa forma, segundo os autores, o somatório das categorias de funcionalidade devem cobrir as necessidades em operações de médio porte na fase de resposta de emergências.

Por operações de tamanho médio, foram excluídas as operações “cotidianas”, que envolve um número pequeno de agentes de resposta (tipicamente um pequeno acidente), bem como operações grandes, como ataques terroristas, terremotos, tsunamis, etc., que são duradouras e raramente ocorrem. Portanto, foi definido um total de 11 categorias de funcionalidades que estão descritas abaixo:

1. **Visão Operacional:** apoia a definição de uma área operacional, ou seja, a principal área em que uma operação de resposta ocorre, bem como no apoio à construção de uma visão operacional comum entre os atores. A visão operacional inclui a necessidade de manter informações detalhadas sobre a emergência de modo geral e a visualização da situação de resposta.
2. **Gestão de Recursos:** apoia à gestão de recursos, ou seja, o pessoal e os equipamentos disponíveis na resposta a emergências. Manter o controle dos recursos é considerado essencial, bem como a forma que os recursos são alocados para diferentes tarefas. A gestão de recursos também pode incluir informações sobre a especialidade de cada ator.
3. **Planos e Ações:** apoia o planejamento e realização das tarefas e na execução das ações pelo pessoal de resposta a emergências. Nos casos em que os planos e as tarefas são gerenciados por um oficial de comando, ter acesso a estas informações é importante, incluindo o conhecimento do progresso e cumprimento das tarefas.

4. **Monitoramento:** apoia o monitoramento do estado de saúde dos atores de resposta ou de vítimas de um incidente. O monitoramento também pode envolver o acompanhamento em situações de risco.
5. **Gestão da Comunicação:** apoia os usuários na utilização de mecanismos de comunicação existentes, como os dispositivos móveis. Eles são utilizados extensivamente para a comunicação entre pessoas ou entre grupos, principalmente onde a comunicação é considerada inviável através os canais de rádio comuns.
6. **Serviço de Informação:** apoia o acesso a serviços de fornecimento de informações úteis durante uma operação de resposta, como relatórios, manuais, dentre outros documentos. Serviço de informação também pode incluir a visualização de informações de diferentes fontes em conjunto.
7. **Transmissão:** apoia transmissão de imagens entre os agentes no local de um incidente ou entre estes agentes e um centro de comando e controle. A transmissão também pode incluir o envio de imagens capturadas por veículos de resposta para agentes em diferentes níveis na operação de emergência.
8. **Mecanismo Especial de Interação:** apoia com interações especiais através de interfaces humano-computador diferenciadas, podendo incluir a utilização de interfaces multimodais, bem como técnicas de realidade aumentada, dentre outras formas de interação.
9. **Detalhes do Incidente:** apoia a manutenção de informações sobre objetos e pessoas envolvidas em um incidente, como vítimas, pessoas desaparecidas, infraestruturas afetadas, etc.
10. **Análise Automatizada:** apoia na obtenção automática de informações de controle, visualização e análises da área de cobertura das tarefas relacionadas com uma operação de resposta, como pontos de interesse em mapas digitais.
11. **Registro:** apoia o registro de ações executadas pelos agentes ou de eventos ocorridos durante a operação de resposta.

Quadro 1: Classificação dos sistemas de informação para GE de acordo com as categorias de funcionalidade e outras características

<i>Funcionalidade</i>	<i>Projeto</i>											
	NOAH	MIKO-BOS	Geo-BIPS	DUM-BONET	DMT	WORK-PAD	SisC2-Celular	CCCMS	Mobile-Map	EMERGENCY	Sis.Emergência	ProRad
Visão Operacional	Sim	Sim	Sim		Sim		Sim		Sim		Sim	Sim
Gestão de Recursos			Sim			Sim	Sim	Sim	Sim	Sim	Sim	
Planos e Ações			Sim			Sim	Sim	Sim				Sim
Monitoramento			Sim						Sim			Sim
Gestão da Comunicação	Sim	Sim	Sim	Sim		Sim	Sim		Sim		Sim	
Serviço de Informação		Sim			Sim	Sim			Sim			Sim
Transmissão	Sim	Sim	Sim	Sim					Sim		Sim	
Mecanismo Especial de Interação												
Detalhes do Incidente	Sim	Sim		Sim				Sim			Sim	Sim
Análise Automatizada			Sim		Sim	Sim	Sim		Sim	Sim	Sim	Sim
Registro									Sim			
<i>Características</i>												
Ano	2000	2006	2006	2007	2009	2009	2010	2010	2011	2011	2012	2012
País	Alemanha	Alemanha	Bélgica	Tailândia	Alemanha	Itália	Brasil	Canada	Chile	Noruega	Brasil	Brasil
Dispositivo	Computador Portátil, PDA	PDA	Computador Portátil, PDA	Computador Portátil, PDA	Computador Portátil	PDA	Celular	PDA	PDA	Smart-Phone, Tablet	Smart-Phone, Tablet	Smart-Phone, Tablet
Plataforma	Windows	Windows	Windows	Windows	Windows	Windows	Java ME	Windows	Windows	Android	Android	Android

Uma classificação dos projetos já apresentados na seção anterior foi realizada baseada nessas 11 categorias com objetivo de analisar as principais funções providas por cada projeto (Quadro 1). Além desse panorama, nota-se que as tecnologias empregadas evoluíram com o passar do tempo. Entre essas propostas supracitadas, algumas avançaram com o uso dos recursos providos pelos dispositivos móveis. Mas, devido às características particulares de cada proposta, ou talvez pela inexistência desses recursos em dispositivos antigos, os trabalhos podem não ter abordado o desenvolvimento de aplicações de forma mais completa e atender novas especificações que podem surgir dentro do domínio. No Apêndice A é possível observar com maior detalhamento as funcionalidades de cada sistema retratado neste capítulo.

3.13.2 Classificação dos sistemas segundo os 3Cs da Colaboração

Além da classificação anterior, complementamos essa análise com base no Modelo 3C da colaboração. Originalmente proposto por Ellis, Gibbs e Rein (1991), o Modelo 3C é um modelo que referencia o sistema colaborativo, sendo utilizado em diversos domínios para representar a colaboração. O modelo divide a colaboração em três aspectos: Comunicação, Coordenação e Cooperação (Figura 13).

Os Cs se inter-relacionam para que a colaboração ocorra, uma vez que são intimamente dependentes e que as pessoas precisam Comunicar, Coordenar e Cooperar para que exista a colaboração. No trabalho em grupo, as pessoas se comunicam por mensagens para negociar e tomar decisões. Durante a coordenação, os membros do grupo lidam com conflitos e organizam as atividades e recursos, evitando desperdícios de comunicação e esforços de cooperação. Enquanto se cooperam em ação conjunta num espaço compartilhado para alcançar um objetivo comum, a necessidade de renegociar sobre situações imprevistas demanda a comunicação, e conseqüentemente, demanda coordenação para reorganizar as tarefas (PIMENTEL; FUKS, 2011).

O domínio de emergências é caracterizado pelo trabalho colaborativo e cada sistema apresentado anteriormente enfatiza a combinação desses aspectos de forma mais relevante que o outro. Dessa forma, por meio de um diagrama de Venn, uma classificação dos sistemas é apresentada de acordo com os aspectos do Modelo 3C que cada um mais apoia pelo seu conjunto de funcionalidades (Figura 14), por exemplo, troca de mensagens está relacionada com a Comunicação, gestão de recursos humanos está relacionada com a Coordenação e a aquisição de informações no local do incidente está relacionada com a Cooperação.

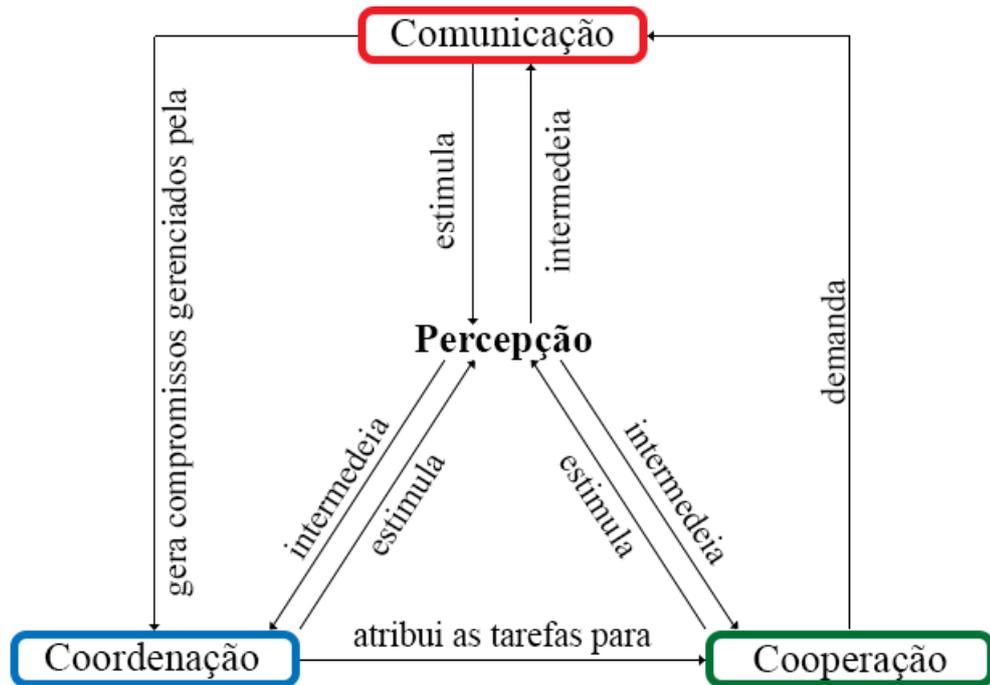


Figura 13: Modelo 3C da Colaboração instanciado para o trabalho em grupo (adaptado de PIMENTEL; FUKS, 2011)

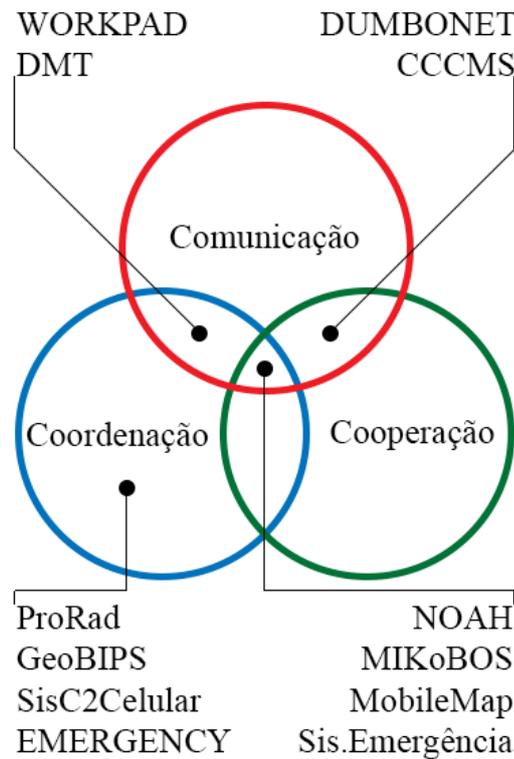


Figura 14: Classificação dos sistemas de informação para GE de acordo com o Modelo 3C da Colaboração

4 Reuso de Software

Este capítulo tem por objetivo apresentar algumas das principais técnicas para o reuso de software, abordando suas principais características e fornecendo um conhecimento básico sobre alguns dos termos empregados neste trabalho.

4.1 Conceituação

Nas últimas de décadas, um dos objetivos mais perseguidos pela Engenharia de Software tem sido a reutilização, uma vez que os sistemas de software se tornam cada vez mais complexos, maior importância é dada às técnicas de desenvolvimento de software. Nesse contexto, várias são as técnicas que permitem a reutilização de software que buscam facilitar a produção de novos sistemas.

Desde o aparecimento da programação modular nos anos 70, passando pelo surgimento das linguagens orientadas a objetos nos anos 80, até os dias de hoje com o modelo de desenvolvimento baseado em componentes, as técnicas de reuso de software se tornaram de vital interesse tanto para a comunidade científica como para a indústria. Os principais benefícios, como aumento da qualidade, redução do custo e do tempo de desenvolvimento, evidenciaram a constante evolução dessa área de pesquisa (BOSCH et al., 1997).

O reuso de software na construção de sistemas de informação pode ser separado essencialmente entre os seguintes tipos: de código ou de projeto. As primeiras técnicas de reuso de software eram baseadas em reutilização de código, como por exemplo, as bibliotecas de funções. Com o passar do tempo, percebeu-se que era possível também empregar a reutilização de projeto, dando origem às técnicas mais modernas, como os padrões de projetos. Já um arcabouço está entre esses dois tipos, visto que é possível abordar tanto reutilização de código quanto de projeto. Szyperski, Gruntz e Murer (2002) apresentam uma revisão das abordagens que propõem a reutilização em nível de projeto. Nas subseções a seguir serão discutidas algumas das técnicas mais importantes na Engenharia de Software.

4.2 Bibliotecas

As bibliotecas surgiram devido à necessidade de codificação de funções que eram reimplementadas a cada novo sistema de informação desenvolvido. As funções genéricas, tais como operações comuns, ordenações ou de manipulação de cadeias de caracteres, foram então reunidas em porções modulares de software na tentativa de proporcionar às primeiras

linguagens de programação funcionalidades que viessem a ser compartilhadas por diferentes sistemas por meio de fragmentos de solução (BACHMAN et al., 2000).

Em vista disso, os programadores poderiam empenhar mais tempo na construção do sistema em si, pois a implementação de rotinas auxiliares já não era necessária em razão delas serem disponibilizadas pela biblioteca, havendo assim um aumento de produtividade no desenvolvimento de software.

Novos paradigmas de programação que evoluíram devido à expansão da orientação a objetos, se basearam na modelagem de características essenciais de um problema em classes de objetos. Isso permitiu o aparecimento das bibliotecas de classes com o objetivo de oferecer funcionalidades com um uso genérico nesse contexto. Assim, uma biblioteca de classes é um conjunto de classes que podem ser agregadas para produzir vários tipos de sistemas (BACHMAN et al., 2000).

As bibliotecas de classes providenciam as funcionalidades que podem ajudar um sistema na execução de uma determinada tarefa, tal como a manipulação de coleções de objetos através de conjuntos, filas, listas e pilhas. Essas funcionalidades não instituem um modelo de arquitetura particular para o sistema, sendo de responsabilidade do desenvolvedor a interconexão do mesmo com a biblioteca. Com a popularização das plataformas, principalmente a plataforma Java, as bibliotecas de classes se tornaram essenciais na implementação de sistemas, oferecendo recursos para extensão das funcionalidades básicas e no reuso de software (Java, 2014a).

Atualmente, as bibliotecas podem permanecer separadas das plataformas de programação, permitindo que suas funcionalidades específicas não sofram alterações ao longo do tempo pelas suas características de abstração e de execução independente. Logo, se faz essencial manter o desenvolvimento de bibliotecas autônomas, mas que possuam uma integração robusta com plataformas ou sistemas.

4.3 Componentes

Um componente pode ser definido como uma unidade independente de distribuição de software que oferece o compartilhamento de determinada funcionalidade. A construção de um componente pode ser baseada em objetos ou em bibliotecas de funções, e podem satisfazer um conjunto de regras de comportamento e suas dependências de contexto necessitam ser explícitas. Ele implementa uma interface, contratualmente especificada, que permite que ele seja composto com outros componentes. A partir da composição de vários componentes é possível desenvolver softwares funcionais (SZYPERSKI; GRUNTZ; MURER, 2002).

Em Pressman (2001) é discutida uma abordagem mais elaborada de reuso sobre essa técnica, chamada de Engenharia de Software Baseada em Componentes, em inglês *Component-Based Software Engineering*. O objetivo dessa abordagem é simplificar o processo de desenvolvimento de software com o uso de componentes padronizados e pré-construídos em vários sistemas. Ao utilizar esses componentes com interfaces padronizadas, reutilizáveis e independentes, os sistemas podem ser produzidos em menos tempo, com maior qualidade e estabilidade.

Esta forma de modularizar o desenvolvimento de software pode trazer outro benefício, a maior adaptabilidade dos sistemas às novas exigências e, conseqüentemente, pode também reduzir os custos de desenvolvimento, o que é fundamental para a estratégia das empresas. Além disso, esta abordagem permite aos desenvolvedores a elaboração de um conjunto de componentes específicos para domínios de negócio, podendo assim facilitar a implementação e manutenção de sistemas complexos ao particionar um problema (SZYPERSKI; GRUNTZ; MURER, 2002).

Os componentes podem ser projetados para serem usados em várias configurações diferentes, permitindo que programadores construam múltiplas variações na sua estrutura e conteúdo. Isso ajuda a esconder algumas complexidades inerentes no tratamento dos aspectos não funcionais, tal como tolerância a erros. Ao delinear totalmente as suas fronteiras, eles fornecem um escopo natural para ações de reconfiguração, supervisão e instrumentação de um sistema.

Ao longo do tempo os componentes passaram a ser comercializados e oferecidos prontos por terceiros para o desenvolvimento de sistemas. O principal benefício é a composição de sistemas com componentes independentes e de fornecedores distintos, na qual não é obrigatório o conhecimento do seu funcionamento interno. Na composição estática, um programador pode combinar vários componentes em tempo de compilação para produzir um sistema. Na composição dinâmica, o programador pode adicionar, remover ou reconfigurar os componentes dentro de um sistema em tempo de execução. Exemplos clássicos de componentes para ambiente desktop são os botões e caixas de texto, e os modelos mais conhecidos são COM/ActiveX, Enterprise JavaBeans, CORBA e Microsoft .Net. Os componentes são uma das principais maneiras de reutilização de código (BACHMAN et al., 2000; SZYPERSKI; GRUNTZ; MURER, 2002).

4.4 APIs

Segundo Bachman (2000), uma Interface de Programação de Aplicação, em inglês *Application Programming Interface* (API), é essencialmente uma especificação em linguagem de programação das propriedades de um módulo e que outros módulos clientes podem ou não depender, por exemplo, as classes ou os métodos de comunicação e como eles devem ser usados.

Em outras palavras, uma API define um ponto de acesso para que um componente de software possa ser usado programaticamente por outros softwares. Portanto, ela possibilita a execução de um conjunto de diferentes funcionalidades pré-estabelecidas desses componentes, exposto para o uso por programadores, tais como requisições ou publicações de dados, eventos diversos ou a realização de tarefas.

O conceito de interface baseia-se na habilidade de reutilizar módulos de software e montar aplicações a partir de componentes independentes, concedendo mecanismos para administrar as dependências que ocorrem entre os diversos módulos de um sistema de informação. Uma vez que o provedor da interface e seus clientes combinam um contrato de associação, é formada uma base mínima para a interação entre as partes e as operações agrupadas em uma interface ocorrem corretamente. Entretanto, uma interface e seus contratos não dizem nada sobre as arquiteturas internas de cada sistema (SZYPERSKI; GRUNTZ; MURER, 2002).

As APIs se tornaram importantes para o crescimento de empresas de desenvolvimento de software, pois quanto mais aplicações criadas para uma determinada plataforma, maior era a sua importância e o seu valor. A Microsoft foi uma das empresas mais bem-sucedidas na exploração de suas APIs para vantagem de negócio. Ela fez investimentos massivos para atrair o maior número de desenvolvedores para escrever sistemas de informação com a API do Windows. Com o surgimento da Internet, houve uma democratização nesse panorama devido à evolução das APIs Web, onde qualquer empresa pode definir e criar sua API para compartilhar dados ou serviços. Logo, uma API aumenta o valor dos ativos digitais da empresa e expande o acesso de clientes aos seus conteúdos através de páginas Web, aplicativos móveis, parceiros e muito mais (CHESHIRE, 2011).

4.5 Padrões de Projeto

Os padrões de projeto representam um conjunto de soluções para problemas de projeto de software dentro de determinados contextos (FAYAD; SCHMIDT, 1997). Cada padrão de

projeto descreve um problema que pode ocorrer frequentemente em diferentes desenvolvimentos de software. Ele descreve também o núcleo da solução do problema, detalhando as classes participantes e suas interações em relação a uma pequena parte de um sistema orientado a objeto, demonstrando como se pode utilizar uma determinada solução por várias vezes, mas obtendo produtos finais diferentes (GAMMA et al., 1995).

A partir do momento em que o problema tenha sido identificado e isolado, deve-se escolher um padrão de projeto adequado e considerar a sua adaptação às circunstâncias específicas. Cada padrão permite mudar aspectos da estrutura do sistema independentemente de outros aspectos para facilitar futuras alterações, bem como no uso de abstrações para reduzir das dependências entre as classes. Assim, os padrões podem ser usados na arquitetura de software e, se aplicados corretamente, eles aumentam a flexibilidade e a capacidade de reutilização de um sistema. Um catálogo com 23 padrões de projetos baseados na experiência de desenvolvedores de sistemas orientados a objeto ao longo do tempo é apresentado por Gamma e outros (1995). Em geral, um padrão tem os seguintes elementos essenciais: o nome do padrão, o problema, a solução, e finalmente, as suas consequências.

Os padrões de projeto, além de permitirem a reutilização de software, proporcionam aos programadores menos experientes o reuso de conhecimento por causa da literatura criada por arquitetos de software e especialistas para documentá-los. Para programadores mais experientes, eles são fundamentais para obter uma visão melhor do sistema sem ter que conhecer o código-fonte em detalhe. Isso tudo favorece as relações entre os desenvolvedores com um vocabulário para dialogar e trabalhar em um problema em um nível geral. Portanto, os padrões de projeto podem colaborar com o aumento de qualidade do projeto em razão de resolver problemas conhecidos com soluções comprovadamente eficazes.

O padrão arquitetural Model-View-Controller (MVC) é um exemplo bem-sucedido na construção de interfaces para interação com o usuário (FROEHLICH et al., 1998). Um problema comum na construção de interfaces dos sistemas é a constante mudança de métodos e funcionalidades, como por exemplo, substituição de botões e menus, além de formas distintas de visualização, através de janelas de sistemas desktop ou páginas Web. Por esse motivo, o MVC visa separar a interface do sistema de suas funcionalidades através de três camadas: o modelo, a visão e o controlador. A camada de modelo abrange as funcionalidades do sistema e os seus dados. A camada visão exibe de forma adequada os dados trabalhados pela camada de modelo. E a camada de controle, após identificar as ações realizadas pelo usuário, aciona os respectivos métodos da camada de modelo, operando uma determinada

tarefa do sistema. Dessa forma, o MVC pode ser decomposto em três principais padrões de projeto, e vários outros menos importantes. Ele usa o padrão *Observer* para garantir que a camada de visão está atualizada com a camada modelo. O padrão *Composite* é aplicado para aninhar visões. E o padrão *Strategy* é utilizado para que as visões deleguem a responsabilidade de lidar com eventos do usuário ao seu respectivo controlador.

Por fim, a detecção de padrões de projetos contribui na construção de sistemas de software bem estruturados, de fácil manutenção e reutilizáveis. Eles são reconhecidos há muito tempo como soluções elegantes e simples para problemas específicos pela captura de estratégias bem-sucedidas de desenvolvimento de projetos abstratos, modelos conceituais e arquiteturas de software independentes. Eles também podem estar diretamente relacionados com outras técnicas de reuso de software, tais como componentes e arcabouços.

4.6 Arcabouços

Um arcabouço, em inglês *framework*, orientado a objeto é um projeto reutilizável de toda (ou parte de) uma arquitetura que é representada por um conjunto de classes. Além disso, ele descreve a maneira que as instâncias (objetos) destas classes se interagem e os seus tipos (FAYAD; SCHMIDT, 1997; GAMMA et al., 1995; JOHNSON, 1997a). Um arcabouço é um projeto reutilizável parcialmente pronto que pode ser especializado para produzir sistemas personalizados (JOHNSON; FOOTE, 1988). Ele fornece a implementação de um sistema (ou subsistema) para problemas de um domínio em particular, fornecendo todo o conhecimento genérico exigido para este domínio (FROEHLICH et al., 1998).

O arcabouço tira proveito de três características distintas da orientação a objeto. A primeira característica é a abstração de classes, representando uma interface na qual a implementação pode mudar. A segunda característica é a capacidade de uma única variável ou procedimento assumir valores de vários tipos, mais conhecido como polimorfismo. A última característica é a herança, facilitando a criação de novos componentes (FAYAD; SCHMIDT, 1997; JOHNSON, 1997a).

Um arcabouço, diferentemente das técnicas já apresentadas, é uma técnica que oferece tanto reutilização de código quanto de projeto (JOHNSON; FOOTE, 1988; JOHNSON, 1997a). Tecnicamente ele é composto de classes abstratas e concretas, descrevendo os tipos de objetos que são importantes e fornecendo um vocabulário comum para trabalhar com um problema específico. No entanto, arcabouços são mais do que apenas ideias, eles também são código.

Arcabouços também podem ser formados por diversos padrões de projetos. A principal diferença entre eles é que os arcabouços focam no reuso de projetos concretos, algoritmos e componentes codificados em uma linguagem de programação particular. Já os padrões de projeto focam no reuso de projetos abstratos, modelos conceituais e microarquiteturas de software independentes de implementações específicas (JOHNSON; FOOTE, 1988; GAMMA et al., 1995).

Um especialista em um determinado arcabouço vê o mundo em termos do domínio, e, naturalmente, vai dividi-lo em uma biblioteca de componentes. O arcabouço permite que um novo componente possa herdar a maior parte de sua implementação de uma superclasse abstrata. Apesar de uma boa biblioteca de componentes ser crucial, a essência de um arcabouço é o modelo de interação e fluxo de controle entre seus objetos. Esses componentes podem ser facilmente combinados uns com os outros porque todos usam as interfaces do arcabouço, facilitando a elaboração uma grande variedade de sistemas com um pequeno número de componentes na biblioteca existente. Ou seja, uma família de sistemas semelhantes, embora não idênticos, pode ser construída a partir de um único arcabouço. Porém a reutilização através da herança requer uma compreensão mais profunda das classes e componentes (JOHNSON; FOOTE, 1988; FROEHLICH et al., 1998).

Uma vez que o arcabouço fornece um esqueleto de sistema que pode ser usado de diferentes maneiras, o desenvolvedor de software pode personalizá-lo para produzir rapidamente novos sistemas (FAYAD; SCHMIDT, 1997). Na medida em que o desenvolvedor ganha experiência no arcabouço, ele poderá compreender outro projeto formado pelo mesmo arcabouço, pois conterá componentes semelhantes e analisará que os sistemas foram construídos de forma parecida. Isso permite compartilhar informações e alavancar o aprendizado no domínio, bem como a evolução de cada sistema. A reutilização da análise e do projeto podem se tornar uma grande recompensa (BIGGERSTAFF; RICHTER, 1987; JOHNSON; FOOTE, 1988). Ademais, muitas vezes se torna possível a criação de exemplos para novos desenvolvedores aprenderem suas propriedades e esclarecerem dúvidas frequentes.

Assim, os arcabouços se tornaram cada vez mais populares como um meio para automatizar o processo de desenvolvimento associado com domínios de sistemas complexos. Porém, eles devem ser bem projetados, testados e fornecer documentação útil e abrangente, a fim de obter uma adoção e um sucesso generalizado (JOHNSON; FOOTE, 1988).

Em contraste com as técnicas de reutilização anteriores, os arcabouços desempenham um papel fundamental porque visam reutilizar projetos de um nível mais alto para diversos domínios de aplicação e unidades de negócio (FAYAD; SCHMIDT, 1997). Logo, um arcabouço é mais abstrato e flexível que um componente, porém mais concreto e mais fácil para a reutilização do que um projeto feito do zero (JOHNSON; FOOTE, 1988; JOHNSON, 1997a).

A reutilização através de arcabouços muitas vezes significa adaptar alguns componentes ou módulos de acordo com a necessidade. Primeiro, um arcabouço fornece aos seus usuários um determinado conjunto de funcionalidades e em segundo lugar, os arcabouços permitem aos seus usuários personalizar ou modificar essas funcionalidades. Devido a sua natureza, ele abstrai com uma solução mais genérica vários problemas específicos. Nesse caso, uma estrutura que centralizada as funcionalidades e onde sistemas podem ser construídos com qualidade ao identificar as funcionalidades necessárias, torna-se num processo evolutivo e com um custo menor (FROEHLICH et al., 1998). Sendo nesse caso, ideal para apoiar as linhas de produtos de software, que são famílias de softwares relacionados (JOHNSON; FOOTE, 1988).

4.6.1 Classificação

Os arcabouços são empregados com sucesso nas mais variadas áreas da computação, tais como interfaces gráficas do usuário, infraestrutura de sistemas e *middleware* de integração. Os arcabouços estão cada vez mais populares nas organizações e podem ser classificados em três categorias (JOHNSON; FOOTE, 1988; FAYAD; SCHMIDT, 1997; ADAIR, 1995):

- **Arcabouços de aplicação empresarial (ou de domínio):** são designados para o desenvolvimento de sistemas orientados a objeto que abordam um domínio de aplicações em particular, como na telecomunicação, na indústria ou no setor financeiro, e formam a base das atividades de negócios corporativos;
- **Arcabouços de infraestrutura (ou de sistema):** são indicados para simplificar o desenvolvimento de infraestruturas de sistemas eficientes, como interfaces gráficas de usuário, sistemas operacionais, estruturas de comunicação ou ferramentas de processamento de linguagem. Esses arcabouços são usados por organizações, de preferência internamente, que trabalham no desenvolvimento de software e que não são vendidos diretamente aos clientes;

- **Arcabouços de integração de *middleware* (ou de suporte):** são comumente usados para integrar componentes e sistemas distribuídos ou *middleware*. Eles são projetados para aumentar a capacidade dos desenvolvedores de software para modularizar, reutilizar e estender sua infraestrutura de software para funcionar perfeitamente em um ambiente distribuído. Exemplos comuns incluem *middleware* orientado a mensagem e bancos de dados transacionais.

Um arcabouço de domínio pode ser mais oneroso para se desenvolver ou comprar, em relação aos demais tipos. Porém, eles podem fornecer um retorno substancial de investimento, porque eles podem suportar diretamente o desenvolvimento de sistemas e novos produtos para o usuário final (JOHNSON; FOOTE, 1988; FAYAD; SCHMIDT, 1997).

4.6.2 Métodos Gancho

Os arcabouços são expressos em uma linguagem de programação, onde eles geralmente consistem um conjunto de classes que colaboraram e bibliotecas que compõem um projeto reutilizável. Certos métodos dessas classes não são totalmente especificados ou são abstratos, descritos por padrões de projetos. Desta forma, uma parte do arcabouço é constante e encapsulada, e através destes métodos, são expostos detalhes que variam entre instâncias do arcabouço e podem ser adaptados ou estendidos para fornecer uma funcionalidade específica (FROEHLICH et al., 1998).

Uma instância do arcabouço fornece os detalhes que estão faltando, isto é, uma parte do arcabouço deve ser codificada pelo desenvolvedor, porque é uma parte específica para o seu sistema. Essa parte variável, um conjunto de métodos de classes ou uma interface do arcabouço, que precisa ser implementada para atender aos novos sistemas é chamado de método gancho, em inglês *hook methods* (PREE, 1995). Então, é um emparelhamento de uma subclasse concreta com cada classe abstrata do arcabouço para fornecer uma implementação completa (FROEHLICH et al., 1998).

Os métodos gancho fornecem informações sobre as formas de como o arcabouço pode ser usado, tanto em termos do que ele oferece quanto do que ele espera. Os dois principais mecanismos da orientação a objetos que proveem os métodos gancho são: a herança e a composição. A fim de estarem em conformidade com a especificação do arcabouço, os sistemas instanciados devem preencher estes métodos de maneira adequada e respeitar o projeto do arcabouço, para garantir que as restrições não sejam violadas (FAYAD; SCHMIDT, 1997; TALIGENT, 1995).

4.6.3 Caixa-Branca e Caixa-Preta

Independentemente do seu escopo, os arcabouços também podem ser classificados pelas técnicas utilizadas para ampliá-los, que variam entre arcabouços de caixa-branca e arcabouços de caixa-preta. Os arcabouços de caixa-branca dependem fortemente do uso da herança de classes para atingir a extensibilidade. As funcionalidades existentes no arcabouço são reutilizadas e estendidas herdando de classes que compõem a base do arcabouço e ao codificar os métodos gancho pré-definidos. O desenvolvedor de sistemas deve entender a hierarquia de classes, e depois, derivar as novas classes que devem ser religadas ao arcabouço (JOHNSON; FOOTE, 1988; FAYAD; SCHMIDT, 1997).

Em contraste, os arcabouços caixa-preta são estruturados utilizando a composição e a delegação de objetos. Eles antecipam suas extensões ao definir as suas interfaces para instanciar novos objetos e atender às necessidades de um sistema (FAYAD; SCHMIDT, 1997). Eles fornecem um conjunto de componentes que podem ser ligados em conjunto e o desenvolvedor não precisa se preocupar sobre como esses componentes realizam suas tarefas individuais (JOHNSON; FOOTE, 1988).

Alguns estudos sugerem que os arcabouços comecem como caixa-branca, pois são principalmente personalizados e reutilizados por meio da especialização de classes. Assim, os arcabouços de caixa-branca são de interesse principalmente por duas razões (JOHNSON; FOOTE, 1988): (1) arcabouços de caixa-branca já são amplamente utilizados; e (2) nem todo arcabouço inevitavelmente evolui para um arcabouço de caixa-preta. Por exemplo, alguns arcabouços são retirados de uso antes de evoluir para a fase de caixa-preta. Outros arcabouços estão em domínios de aplicação que não são compreendidos num grau suficiente para oferecer suporte ao arcabouço de caixa-preta.

Por outro lado, quando a parte variável do arcabouço se estabiliza e é totalmente compreendida, muitas vezes é apropriado evoluir para um arcabouço de caixa-preta. Muitos arcabouços combinam ambas as características, fornecendo prontamente classes utilizáveis, bem como classes abstratas para que se possam instanciar subclasses específicas do sistema, também chamados de caixa-cinza (JOHNSON; FOOTE, 1988; YANG et al., 1998; ROBERTS; JOHNSON, 1996).

4.6.4 Inversão de Controle

Tradicionalmente, um desenvolvedor reutiliza componentes de uma biblioteca, escrevendo um programa principal que chama os componentes sempre que necessário. O

desenvolvedor decide quando chamar os componentes e é responsável pela estrutura geral e fluxo de controle do sistema.

Em um arcabouço, o sistema principal é reutilizado, e o desenvolvedor decide o que é conectado a ele e como os novos componentes serão conectados. O código do desenvolvedor é chamado pelo código do arcabouço. O arcabouço determina a estrutura e o fluxo de controle do sistema. Isso é chamado de inversão de controle, sendo uma das características principais dos arcabouços, onde o controle do fluxo da execução é administrado pelo arcabouço, não pelo sistema (JOHNSON; FOOTE, 1988; FROEHLICH et al., 1998; JOHNSON, 1997a).

4.6.5 Processo de Desenvolvimento

Um arcabouço deve ser concebido para ser reutilizável desde o seu princípio, como qualquer outro software reutilizável. Não é considerado adequado para um sistema do domínio se tornar um arcabouço de uma forma direta (FROEHLICH et al., 1998).

No entanto, o desenvolvimento de um arcabouço é uma tarefa difícil, pois são mais difíceis de projetar que os sistemas e as ferramentas particulares (GAMMA et al., 1995). Outra dificuldade reside na compreensão do domínio do problema, encontrar as abstrações apropriadas e observar as características de diversas aplicações pertencentes a este domínio (FAYAD; SCHMIDT; JOHNSON, 1999). Os arcabouços exigem, mais do que qualquer outro software, mudança e refatoração (OPDYKE, 1992).

A maioria dos trabalhos relacionados ao processo de desenvolvimento concorda que a construção de um arcabouço é essencialmente um processo iterativo no qual ele é definido, testado e refinado um número indefinido de vezes (FROEHLICH et al., 1998; BOSCH et al., 1997). Especificamente, diferentes autores expressaram diferentes fases do processo de desenvolvimento:

- Mattsson (2000) descreve as fases de análise do domínio, projeto e instanciação do arcabouço;
- Adair (1995) apresenta as fases de análise do domínio, projeto, implementação e incrementação do arcabouço;
- Pree (1995) define as fases de identificação de classes e métodos gancho, projeto do arcabouço, aplicação de padrões de projeto e adaptação do arcabouço;

- Bosch e outros (1997) indica as fases de análise de domínio, projeto, implementação, teste e criação de instâncias de teste, e a documentação do arcabouço;
- Froehlich e outros (1998) estabelece as fases de análise, projeto, implementação, teste e refinamento do arcabouço;
- Yang e outros (1998) descreve as fases de análise do domínio e projeto, implementação e teste do arcabouço baseadas na orientação a objeto e no uso da Linguagem de Modelagem Unificada, em inglês *Unified Modeling Language* (UML).

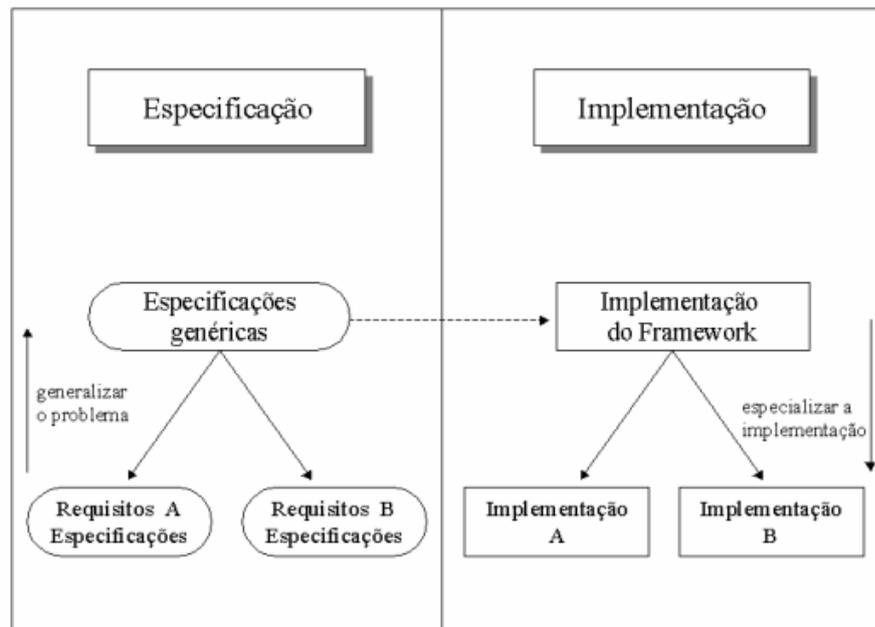


Figura 15: Especificação e implementação com arcabouços (adaptado de D'SOUZA; WILLS, 1999)

A Figura 15 ilustra os conceitos de especificação e implementação referentes ao processo de construção de um arcabouço a partir de uma visão mais generalizada dentre todos os processos listados e suas respectivas instâncias (D'SOUZA; WILLS, 1999).

Uma análise comparativa e mais detalhada entre as principais abordagens pode ser verificada nos trabalhos de Silva (2000), Franca (2000) e Rheinheimer (2002). Apesar dos estudos evidenciados abordarem várias propostas para o processo de desenvolvimento, esse trabalho adota a abordagem retratada por Yang e outros (1998).

Yang e outros (1998) discutem que estas abordagens se concentram nas fases de análise e projeto do arcabouço, só fornecendo esboços de modo genérico ao invés de estabelecerem orientações e tarefas concretas. Eles destacam principalmente que as

abordagens não aplicam notações específicas para o desenvolvimento do arcabouço. Uma comparação e avaliação das outras abordagens são apresentadas em seu trabalho.

Na proposta de Yang e outros (1998), por outro lado, é definida uma abordagem para desenvolvimento de arcabouços orientados a objeto através de notações e semânticas baseadas na UML. Dessa forma, um processo prático é descrito com tarefas essenciais para o desenvolvimento, na aplicação de notações UML e instruções para realizar cada tarefa. Assim, um conjunto de artefatos e diagramas é gerado para constituir uma documentação completa do arcabouço.

Os benefícios de linguagens de modelagem estão bem estabelecidos na Engenharia de Software. O objetivo dessas linguagens é obter um conjunto de conceitos, notações e regras comumente aceitas, a fim de expressar melhor os problemas e soluções. O uso de uma linguagem de modelagem imita a flexibilidade de uma linguagem natural, em que um vocabulário, sintaxe e gramática permitem expressar ideias complexas (OMG, 2009).

A UML é uma linguagem de modelagem bem conhecida e amplamente aceita para a modelagem de produtos de software. Ela traz uma notação comum entre os envolvidos para que eles possam debater sobre o projeto de um sistema. Isso aumenta a chance de haver uma interpretação adequada, evitando originar problemas na fase de implementação e, conseqüentemente, aumentar os custos do projeto. Portanto, a UML é uma notação extensível que inclui definições de diagramas de modelagem para as diversas fases do projeto (BOOCH; RUMBAUGH; JACOBSON, 1999). Uma revisão abrangente sobre a UML é encontrada em no trabalho de Fowler (2004).

Em vista do processo de desenvolvimento empregado neste trabalho, ele é dividido em quatro fases na seguinte maneira:

- a) **Fase de análise:** especificação de requisitos a partir de projetos da literatura, elaboração do diagrama de casos de uso do arcabouço proposto, e descrição dos casos de uso;
- b) **Fase de projeto:** elaboração do diagrama de classes e de sequência, aplicação de padrões de projeto, revisão do projeto, e documentação;
- c) **Fase de implementação:** implementação do arcabouço e criação da primeira instância;
- d) **Fase de teste:** realização de teste de unidade e de integração do arcabouço.

Na fase de análise é definido que a especificação de requisitos do arcabouço deve ser feita através da identificação e extração das funcionalidades comuns de um conjunto de

sistemas similares que pertencem ao mesmo domínio. Na fase de projeto é necessário identificar as principais classes e os métodos gancho do arcabouço, determinando o seu tipo em caixa-branca ou caixa-preta. O processo de desenvolvimento deve ser aplicado repetidamente para que o arcabouço evolua de maneira incremental na medida em que cada ciclo do processo é concluído. Cada ciclo aborda um conjunto de requisitos para adicionar novas funções ao arcabouço, procedendo-se através da análise, projeto, implementação e teste (YANG et al., 1998).

5 JEMF: Um Arcabouço para o Desenvolvimento de Sistemas Móveis no Domínio de Gestão Emergências

Java Emergency Management Framework (JEMF) é um arcabouço de desenvolvimento que se baseia em especificações abordadas em trabalhos prévios no domínio de emergências e no desenvolvimento de novas especificações. Neste capítulo são definidos e descritos os requisitos e as análises básicas necessárias para atender os aspectos de desenvolvimento de sistemas de apoio à gestão de emergências para dispositivos móveis.

5.1 Metodologia

Esta seção tem como objetivo detalhar os passos que foram necessários para a execução desta pesquisa. O desenvolvimento deste trabalho foi dividido em três etapas, explicadas conforme a Figura 16.

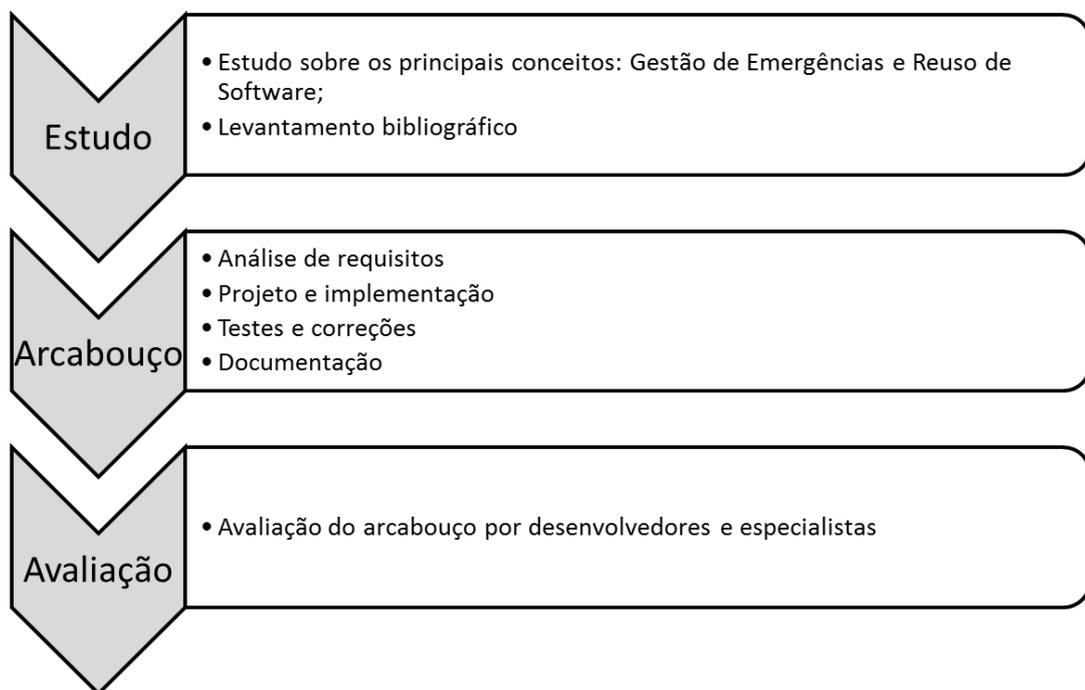


Figura 16: Etapas da pesquisa

Na etapa Estudo o objetivo foi formalizar o conhecimento em Gestão de Emergências, Sistemas Colaborativos e Reuso de Software. Uma revisão sistemática da literatura foi realizada em artigos e bibliotecas digitais da área da computação, em anais de conferências

internacionais e nacionais, em trabalhos desenvolvidos pelo Grupo de Engenharia do Conhecimento (PPGI/UFRJ), em documentos dos Corpos de Bombeiros Militares no Brasil e da Agência Federal de Gestão de Emergências dos Estados Unidos, dentre outros.

Na etapa Arcabouço o objetivo foi especificar e desenvolver o arcabouço proposto. O primeiro passo na concepção de um arcabouço é a determinação de seu domínio. Isso ajuda a definir o seu escopo e as funcionalidades básicas que ele deve proporcionar. Uma técnica que pode ser usada para ajudar neste processo é a análise de variabilidade. Isto envolve avaliar os diversos componentes de sistemas através de um número de cenários e a identificação de características variáveis e comuns. Cenários podem incluir descrições gerais, modelos de processo formal, ou, idealmente, ferramentas reais desenvolvidas para o domínio pretendido. Esta análise resulta na identificação dos padrões recorrentes do arcabouço e suas características comuns. Dessa forma, ele incorpora o conhecimento dos sistemas do domínio. Os detalhes de baixo nível são deixados para os usuários do arcabouço, uma vez que estas são as características que variam entre os sistemas no domínio (FROEHLICH et al., 1998). Assim, inicialmente foi realizada uma análise de sistemas de informação móveis para gestão de emergências, conforme foi apresentado no Capítulo 3 deste documento.

Devido à crescente importância de arcabouços, alguns estudos empenharam esforços em definir um processo para o seu desenvolvimento. Nesses estudos é possível identificar propostas para abordagens e linhas gerais a serem seguidas na análise e projeto de um arcabouço. Conforme detalhado no capítulo anterior, o processo de desenvolvimento adotado para a etapa Arcabouço segue a abordagem apresentada por Yang e outros (1998). Já nesse capítulo, são apresentadas as especificações e os artefatos por meio de representações gráficas baseadas na UML, como por exemplo, diagramas de casos de uso e identificação dos principais atores do domínio. Em seguida, foi elaborado o modelo estrutural do arcabouço, representado por diagramas de classes. Esse modelo foi obtido a partir da verificação de quais classes eram necessárias para a realização dos casos de uso e a partir de suas descrições. Por fim, foram definidos diagramas de sequência para os casos de uso, compondo o modelo dinâmico do projeto. Portanto, este capítulo visa abordar o desenvolvimento do arcabouço.

Em resumo, os passos para o desenvolvimento do arcabouço foram os seguintes: (1) definição do escopo e especificações; (2) definição do modelo funcional para o domínio estudado (diagrama de casos de uso e descrições); (3) definição do modelo estático para o domínio estudado (diagrama de classes); (4) definição do modelo dinâmico para o domínio estudado (diagramas de sequência); (5) documentação; (6) implementação; e (7) teste.

A última etapa tem por objetivo realizar a avaliação do arcabouço. Isso consiste em validar a proposta apresentada neste trabalho por meio de exercícios para desenvolvedores de software com o uso do arcabouço e também pela crítica de especialistas do domínio efetuadas para este projeto. Todos os detalhes dessa etapa são abordados no Capítulo 6.

5.2 Escopo

Segundo Engelbrecht, Borges e Vivacqua (2011), um procedimento habitual que visa aumentar a disponibilidade de informações sobre uma emergência é a criação de postos de comando perto de áreas impactadas. Esta proximidade conduz para uma tomada de decisões mais apropriada devido à avaliação da situação atual de forma rápida das operações de resposta.

Conforme descrito no Capítulo 2, e reconhecido por Engelbrecht, Borges e Vivacqua (2011), é possível destacar uma divisão entre dois grupos que compõem a estrutura de C2 e que possuem funções diferentes na fase de resposta aos desastres. O primeiro grupo é chamado de comando e é responsável pela avaliação de riscos e tomada de decisões táticas e operacionais, bem como acompanhar o progresso de todas as atividades de resposta. Nesse grupo, um indivíduo é destacado para a posição de comandante e situa-se geralmente nos postos de comando ou no centro de controle. O segundo grupo é denominado agentes de resposta, primeiros respondedores ou equipe de operação. Esse grupo é composto por bombeiros, policiais, médicos e voluntários que realizam as atividades na linha de frente da emergência.

A colaboração e a troca de informação entre estes grupos são fundamentais para o sucesso das operações de emergência. Por este motivo, o uso da tecnologia e sistemas de computação colaborativos pode melhorar a consciência situacional dos envolvidos e resultar em melhores decisões. Engelbrecht, Borges e Vivacqua (2011) destacam em seu trabalho o uso de uma mesa digital como instrumento tecnológico que apoia o comando na gestão da emergência. Eles também salientam que o uso de dispositivos móveis pelas equipes de operação é importante para a transmissão de informações para o posto de comando para a compreensão da situação atual das atividades em andamento. A Figura 17 apresenta o ambiente típico entre os dois grupos discutidos.

O escopo deste trabalho concentra-se no apoio ao desenvolvimento de soluções móveis para dispositivos que auxiliem o trabalho dos primeiros respondedores, uma vez que eles fazem a inserção de dos dados da emergência na fase de resposta.



Figura 17: Ambiente de resposta à emergência

5.3 Especificação do Arcabouço

Esta seção aborda a especificação de requisitos elaborados para o arcabouço proposto e compreende as fases de análise e projeto da metodologia adotada.

5.3.1 Modelagem de Análise

De acordo com Pressman (2006), modelos são criados para obter uma melhor compreensão da entidade real a ser construída. Quando a entidade a ser construída é um software, o modelo deve ser capaz de modelar a informação que o software transforma, as funções que possibilitam que as transformações ocorram e o comportamento do sistema quando a transformação está se desenvolvendo. Os modelos quando criados fazem uso de

uma notação gráfica que descreve as informações, o processamento, o comportamento do sistema e outras características, usando ícones distintos e reconhecíveis, como também, informações descritivas podem ser oferecidas mediante o uso de uma linguagem natural ou de uma linguagem especializada que descreve as exigências.

5.3.1.1 Requisitos Não Funcionais

“Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento” (SOMMERVILLE, 2011). Os requisitos do JEMF foram obtidos a partir da análise das iniciativas anteriores abordadas no Capítulo 3. A partir da classificação das funcionalidades dos sistemas de informação, foram extraídas as funcionalidades genéricas mais relevantes que constituem o arcabouço aqui proposto. No Apêndice A é possível observar com maior detalhamento as funcionalidades de cada sistema retratado no Capítulo 3.

Os requisitos não funcionais são restrições aos serviços ou funções oferecidas pelo sistema e, na maioria das vezes, aplicam-se ao sistema como um todo. Através destas soluções computacionais discutidas anteriormente é possível destacar alguns dos requisitos não funcionais mais difundidos e que são necessários aos sistemas de informação móveis desse domínio:

- **Comunicação:** suporte ao compartilhamento de informação entre equipes de comando e operação. Acesso e coleção de informações sobre o desastre podem ajudar a organizar as operações de reposta, aumentar a colaboração e melhorar a segurança dos envolvidos;
- **Entrada e saída de dados:** A entrada de dados precisa ser simples e rápida para não tomar tempo hábil para realização de tarefas. A saída de dados deve facilitar a visualização dos dados armazenados para que o usuário tenha uma melhor compreensão de dados importantes;
- **Integridade:** as tomadas de decisão dependem da integridade dos dados fornecidos. Portanto, o sistema deve representar a confiabilidade da fonte, e a credibilidade da informação nas estratégias e meios de transferência dos dados;
- **Interoperabilidade:** o sistema deve ser aberto para troca de informações com outros sistemas. Assim como as informações devem ser exportáveis ou importáveis em um padrão que possa ser utilizado por outros sistemas de informação;

- **Disponibilidade:** Os dados devem estar disponíveis a todos, em todos os lugares e a qualquer momento, permitindo aos usuários trabalharem também de forma assíncrona e oferecer opções para sincronizar as equipes de operação e comando no momento necessário.

5.3.1.2 Requisitos Funcionais

Os requisitos funcionais são declarações de serviços ou funções que o sistema deve fornecer e se comportar em determinadas situações (SOMMERVILLE, 2011). Dentre as diversas funções providas pelos projetos supracitados no Capítulo 3, selecionamos as funcionalidades mais relevantes que fazem parte do JEMF. Elas representam um conjunto de ações fundamentais que os sistemas de informação em gestão de emergências devem apoiar para um gerenciamento e tomada de decisão adequada pelos agentes nos atendimentos aos desastres. O Quadro 2 mostra a associação de cada funcionalidade do JEMF com a classificação apresentada no Capítulo 3. Na Figura 18 é apresentada a associação de cada funcionalidade do JEMF com os aspectos do modelo 3C da Colaboração.

As funcionalidades não contempladas nesta versão do arcabouço servirão como referência para pontos de extensão em futuras versões. A definição de novos requisitos específicos e funcionais é de competência de cada novo projeto a ser desenvolvido, portanto eles podem variar de acordo com diferentes características e objetivos das organizações dentro do domínio.

Quadro 2: Associação entre as categorias de funcionalidades gerais e do JEMF.

<i>Funcionalidades Gerais (NILSSON; STØLEN, 2011)</i>	<i>Funcionalidades do JEMF</i>
Visão Operacional	Gerenciar Emergência
Gestão de Recursos	Gerenciar Recurso
Planos e Ações	Gerenciar Missão Gerenciar Tarefa
Monitoramento	Não contemplado por depender de características específicas para cada projeto (por exemplo, nível de radiação nuclear, sensores, etc.).
Gestão da Comunicação	Gerenciar Mensagem
Serviço de Informação	Gerenciar Documento Compartilhado
Transmissão	Gerenciar Mensagem
Mecanismo Especial de Interação	Não contemplado por depender de características específicas para cada projeto (por exemplo, reconhecimento de voz, etc.).
Detalhes do Incidente	Gerenciar Vítima Gerenciar Testemunha Gerenciar Unidade de Saúde

Análise Automatizada	Gerenciar Ponto de Interesse Gerenciar Localização
Registro	Não contemplado

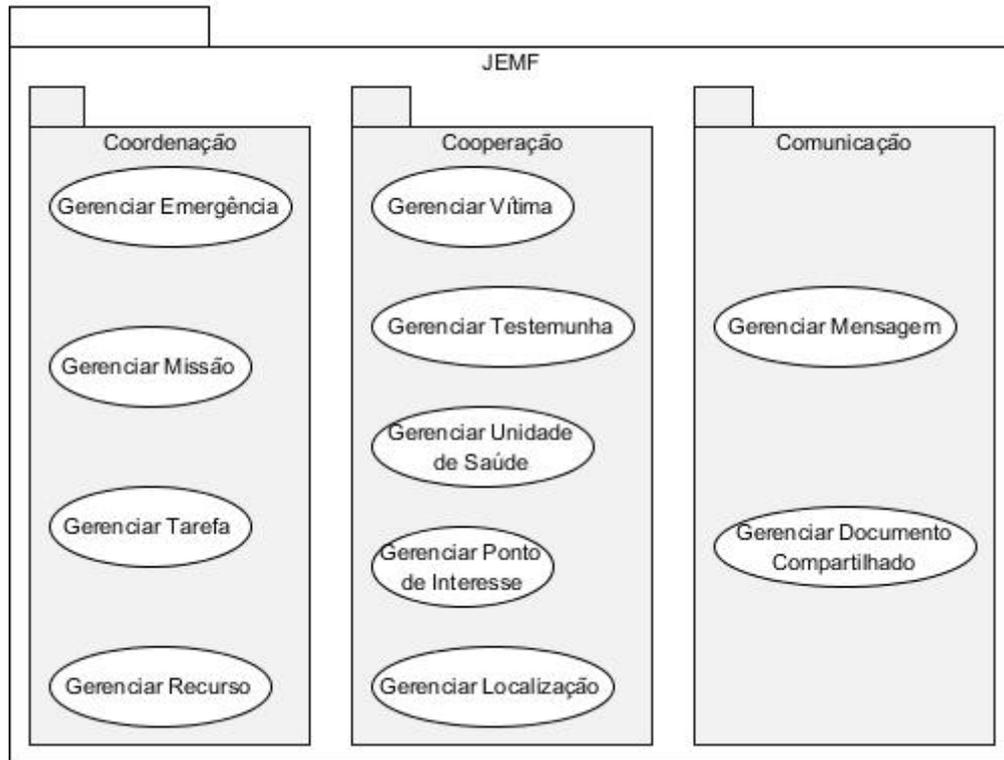


Figura 18: Principais funcionalidades estabelecidas para o JEMF

A seguir, detalhamos o objetivo das funcionalidades para a manutenção dos dados que englobam as atividades de resposta, sendo que cada uma é composta pelas operações de inclusão, consulta, alteração e exclusão de dados:

- **Gerenciar Emergência:** Funcionalidade que gerencia os dados do evento principal do desastre ocorrido, ou seja, serve para registrar qual o tipo, a intensidade, o local e a data da emergência que estará sendo tratada. Por exemplo, um incêndio urbano;
- **Gerenciar Missão:** Funcionalidade que gerencia os dados de macro atividades destinadas a um grupo de agentes que poderão ser registradas pelos gestores de crise. Por exemplo, atividades de combate ao incêndio na região A, busca e salvamento de vítimas na região B, retirada de cidadãos da região de risco C, etc;
- **Gerenciar Tarefa:** Funcionalidade que gerencia os dados de atividades

individuais de cada agente de resposta alocado numa missão. Por exemplo, agente 1 responsável pela atividade de rescaldo, agente 2 responsável pela atividade de estabilização de vítimas, agente 3 responsável pela operação de equipamentos ou veículos de resgate, etc;

- **Gerenciar Unidade de Saúde:** Funcionalidade que gerencia os dados das entidades de saúde que fazem parte da resposta à emergência. Por exemplo, registro de ambulâncias, tendas de atendimento e hospitais que poderão receber vítimas do desastre;
- **Gerenciar Vítima:** Funcionalidade que gerencia os dados inerentes a cada vítima socorrida no processo de resposta;
- **Gerenciar Testemunha:** Funcionalidade que gerencia os dados de possíveis testemunhas do evento ocorrido, permitindo adquirir informações que poderão ser importantes para a execução de atividades pelos agentes;
- **Gerenciar Recurso:** Funcionalidade que gerencia os dados dos equipamentos disponíveis para suporte das atividades realizadas pelos agentes em campo;
- **Gerenciar Mensagem:** Funcionalidade que gerencia os dados que serão enviados e recebidos entre cada agente em campo, incluindo mensagens de texto, imagem ou áudio;
- **Gerenciar Documento Compartilhado:** Funcionalidade que gerencia os dados que estarão compartilhados e considerados relevantes a todos os agentes em campo durante o período de resposta ao incidente, como planos de ação, documentos, etc;
- **Gerenciar Localização:** Funcionalidade que gerencia os dados da localização do agente em campo;
- **Gerenciar Ponto de Interesse:** Funcionalidade que gerencia os dados de localização de pontos estratégicos, como uma infraestrutura danificada pelo evento ou qualquer localidade que seja considerada relevante no processo de resposta.

5.3.1.3 Casos de Uso

Uma emergência pode demandar atores de resposta com diferentes níveis de autoridade, funções e especialidades. Os sistemas de informação devem abranger funcionalidades para cada perfil de pessoa e para que elas possam exercer suas responsabilidades da melhor maneira possível com o apoio desses sistemas. As vítimas e testemunhas envolvidas nos incidentes também estão presentes nesse contexto, mas diferentemente dos agentes de resposta, geralmente elas podem fazer parte do objetivo das tarefas de resgate e não são usuários dos sistemas (Figura 19).

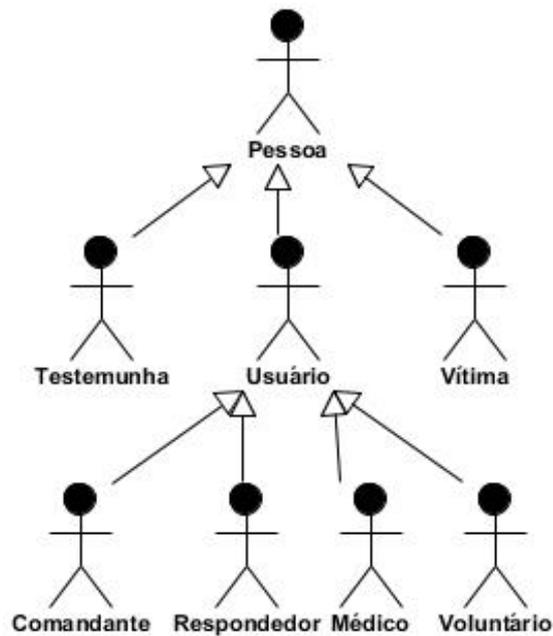


Figura 19: Principais atores envolvidos em Gestão de Emergências

As características e responsabilidades desses atores que trabalham com dispositivos móveis são descritas a seguir:

- **Comandante:** Ator responsável pela gerência da resposta às emergências. Ele define e atribui atividades para os demais usuários, concentra a maior parte das informações relevantes durante a resposta e realiza a tomada das decisões no local do incidente com a visão geral do contexto situacional. Exemplos: Comandante de incidente, oficial de polícia;
- **Respondedor:** Ator responsável pela execução das atividades de resposta ao incidente. O respondedor recebe as suas atividades atribuídas pelo comandante e as executa de acordo com sua especialidade. Ele também captura as informações contextuais *in loco* e compartilha com os demais atores. Seu principal objetivo é a estabilização do incidente e o resgate de vítimas. O

respondedor engloba a maior parte das funções de resposta em campo. Exemplos: Bombeiro, policial, observador, motorista de veículo, piloto de helicóptero, etc;

- **Médico:** Ator responsável pela execução das atividades de atendimento médico as pessoas vitimadas. Ele detém a capacidade de avaliar e estabilizar a saúde das pessoas que sofreram algum dano pelo incidente. Exemplos: Médico, paramédico, enfermeiro e outras profissões relacionadas à assistência em saúde pública;
- **Voluntário:** Ator responsável pela execução das atividades de resposta com menor prioridade ou risco. Ele oferece suporte ao comandante e respondedor na realização das ações que demandam maior quantidade de pessoas e podem auxiliar com conhecimentos especializados. Exemplos: Analista, técnico, empregados de companhias de luz, água e telecomunicações, agente de organizações não governamentais (ONGs), etc;
- **Vítima:** Representa a entidade que foi afetada pelo incidente, englobando desde uma pessoa até uma comunidade. As vítimas necessitam das ações de resgate do combatente, médico ou voluntário. Mas, geralmente não fazem parte do grupo de usuários dos sistemas de informação. Exemplo: Cidadão (qualquer indivíduo);
- **Testemunha:** Representa a entidade que testemunhou o incidente. Ela pode fornecer informações relevantes sobre o fato ocorrido para os demais atores responsáveis pela resposta à emergência. Da mesma forma que a vítima, geralmente não fazem parte do grupo de usuários dos sistemas de informação. Exemplos: Cidadão (qualquer indivíduo).

As funcionalidades estabelecidas para o JEMF estão organizadas por funcionalidades gerais e funcionalidades específicas. As funcionalidades gerais são destinadas a todos os atores envolvidos no processo de resposta, pois são funções fundamentais que os sistemas de emergências precisam apoiar, tais como a gestão de tarefas, recursos, pontos de interesse, mensagens e documentos compartilhados. As funções específicas são destinadas para cada ator de acordo com o seu nível de responsabilidade dentro da equipe de resposta. O comandante utiliza o gerenciamento de dados da emergência, missão e unidade de saúde. O respondedor usa o gerenciamento de dados da missão, vítima e testemunha. O médico usa o gerenciamento de dados da missão, vítima e unidade de saúde. O voluntário utiliza o

gerenciamento de dados da testemunha. Entretanto, cada novo projeto poderá definir o propósito de cada de funcionalidade segundo seus critérios específicos.

Os diagramas de casos de uso descrevem o uso pretendido do sistema modelado (DOUGLASS; CERNOSEK, 1997). Eles identificam os atores envolvidos em uma interação com o sistema, nomeia o tipo dessa interação, e suplementa com informações adicionais por descrições textuais (SOMMERVILLE, 2011). Um conjunto de casos de uso e suas respectivas descrições foram elaborados de acordo com as funcionalidades já indicadas na subseção anterior. O exemplo apresentado a seguir refere-se à funcionalidade Gerenciar Emergência (Figura 20). As descrições dos casos de uso foram realizadas e detalhadas no Apêndice B, seguindo o exemplo das operações de Incluir, Alterar, Consultar e Excluir da funcionalidade Gerenciar Emergência.

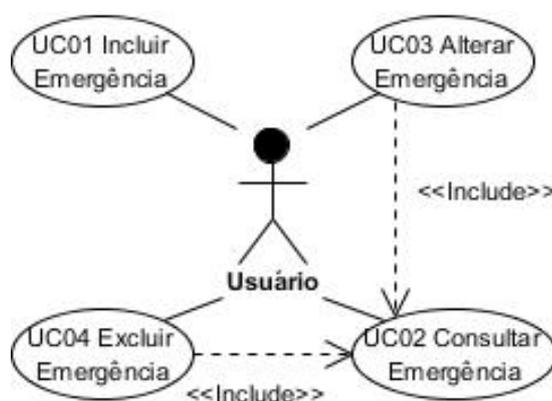


Figura 20: Casos de Uso da funcionalidade Gerenciar Emergência

5.3.2 Modelagem de Projeto

O projeto de software é o processo pelo qual os requisitos são traduzidos numa representação do software. Projetar a arquitetura do sistema, identificar os principais objetos do sistema e desenvolver modelos de projeto são algumas das atividades criativas nesse processo. O projeto de classes transforma modelos de análise em realizações de classes de projeto e nas estruturas de dados citando os requisitos necessários que serão exigidos para se implementar o software (PRESSMAN, 2006; SOMMERVILLE, 2011).

5.3.2.1 Diagrama de Classes

Depois de definir os requisitos e os casos de uso, foram definidos os diagramas de classe para a construção do arcabouço. Os diagramas de classes descrevem várias classes de objetos e suas relações e associações, incluindo herança e agregação (DOUGLASS; CERNOSEK, 1997). Alguns estudos indicam que um arcabouço deve ser evoluído a partir de

classes gerais e usar herança nos sistemas individuais (JOHNSON; FOOTE, 1988; FAYAD; SCHMIDT, 1997).

A herança permite aos desenvolvedores construir novas classes e alterar o código em um ambiente orientado a objeto, simplesmente herdando o comportamento desejado de uma classe existente e substituindo apenas os métodos que são diferentes na subclasse. A definição de muitas subclasses para especificar as diferenças de regras de negócio pode ser necessária, mas as funções comuns podem ser abstraídas e reutilizadas de forma conveniente.

O JEMF é dividido em dois pacotes: o Núcleo (em inglês *Core*) e o Móvel (em inglês *Mobile*). A seguir serão detalhadas as características principais de cada pacote.

5.3.2.1.1 Pacote Núcleo

O pacote Núcleo é onde se concentram as classes fundamentais do arcabouço e representam as entidades que envolvem uma resposta de emergência. A Figura 21 apresenta em um único diagrama todas as principais classes do modelo de domínio, associando os aspectos de Coordenação, Cooperação e Comunicação. Espera-se que essas classes do núcleo sofram uma menor quantidade alterações, pois se tratam de classes com maior nível de abstração (Figura 22). A partir dessas classes poderão ser estendidas as classes concretas que formam o conjunto de funcionalidades descritas na subseção 5.3.1.2.

A Figura 23 apresenta as principais classes do pacote Núcleo, onde uma emergência (*Emergency*) necessita de serviços de resposta (*Service*). Esses serviços podem ser fornecidos por entidades de resposta (*Entity*) às emergências ou podem ser requisitados por entidades afetadas. Uma entidade pode ser uma organização (*Organization*), por exemplo, Corpo de Bombeiros, ou pode ser uma pessoa (*Person*), por exemplo, um bombeiro.

Os serviços e as entidades de resposta podem utilizar os recursos materiais (*Resource*) disponíveis durante a emergência. Assim como eles podem enviar ou receber informações (*Information*) entre si. Uma entidade contém um dado para contato (*Contact*), por exemplo, número de telefone ou e-mail que podem ser usados para a troca de informações.

Uma emergência, um serviço, um recurso e uma entidade podem conter uma localização (*Location*). Essa localização pode ser formada por dados de logradouro, dados de posicionamento global ou dados para uma identificação personalizada (*Address*, *Position* ou *PlaceIdentifier*).

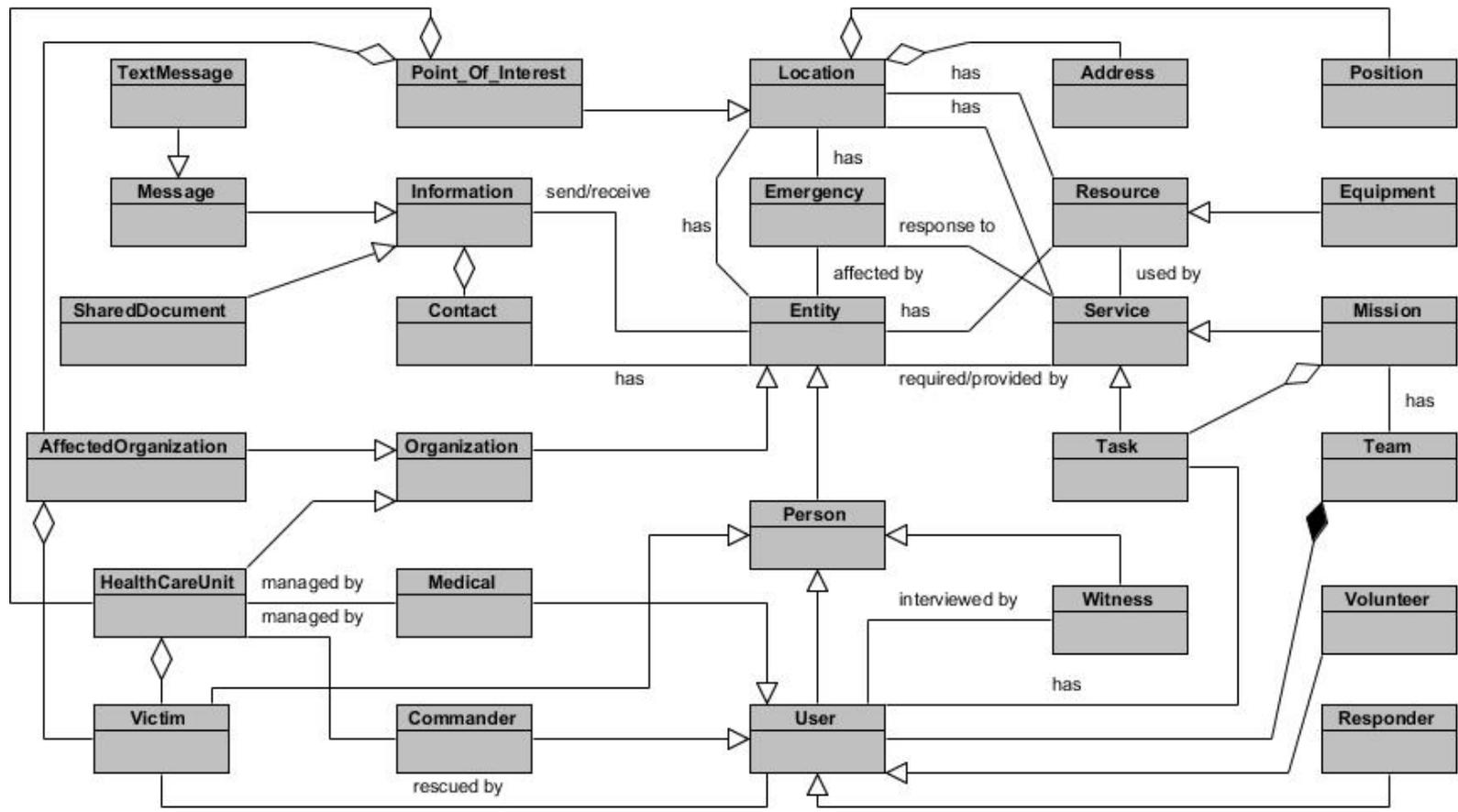


Figura 21: Diagrama de classes do modelo de domínio

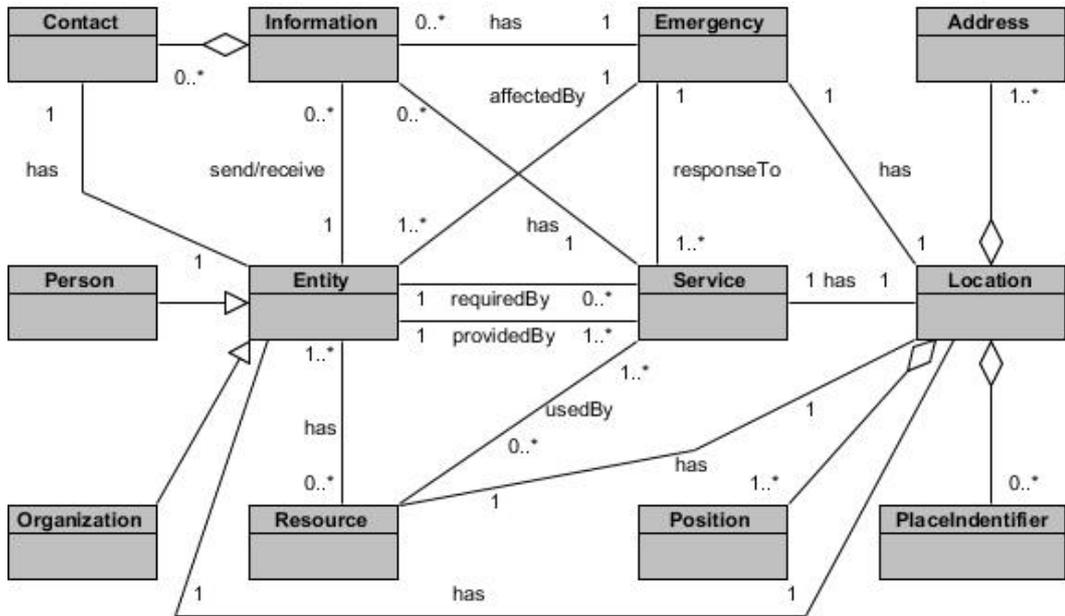


Figura 22: Pacote Núcleo do JEMF destacando as principais classes (abstratas)

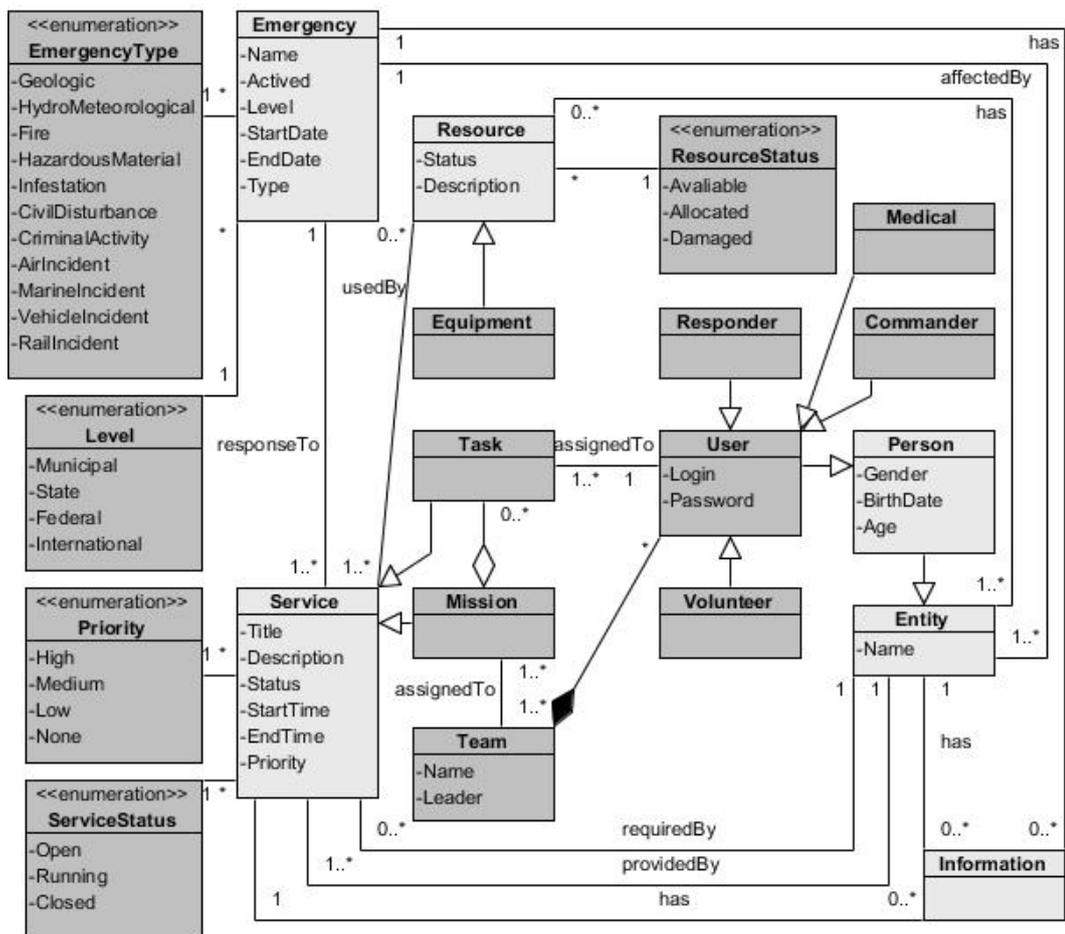


Figura 23: Pacote Núcleo do JEMF destacando classes do aspecto de coordenação

A Figura 23 apresenta um conjunto de classes que se referem ao aspecto de coordenação. Além das classes principais já mencionadas, são destacadas algumas especializações das mesmas. Um serviço pode ser especializado em missão (*Mission*) ou tarefa (*Task*). Essa diferença é encontrada em alguns sistemas de informação que trabalham com a alocação de pessoa em serviços de resposta de formas distintas. A missão retrata a designação de um serviço para um grupo de pessoas, por exemplo, o grupo A – resgate de vítimas – para o salvamento de vítimas numa emergência. Já a tarefa é retratada como um serviço para cada pessoa separadamente, por exemplo, o bombeiro B deve executar a atividade de apagar o incêndio e o bombeiro C deve verificar a reserva de água disponível para atuar na emergência.

Uma pessoa pode ser especializada em usuários (*User*) de um sistema de informação para resposta aos desastres. Dentre os principais usuários estão: (1) o comandante (*Commander*), representando a pessoa com maior nível de autoridade e que possui a responsabilidade de gestão de todos os serviços de resposta à emergência; (2) o combatente ou respondedor (*Responder*), representando o agente que executa os serviços designados pelo comandante; (3) o médico (*Medical*), representando o agente que possui conhecimentos na área médica para serviços específicos nessa área; e (4) o voluntário (*Volunteer*), representando o agente que realiza serviços com menor risco ou prioridade. Caso necessário, novas especializações de usuário podem ser criadas para atender outras funcionalidades. Uma tarefa pode ser designada a um desses tipos de usuários. Entretanto, a missão é preciso ser designada a uma equipe de usuários (*Team*).

Um recurso pode ser especializado em equipamentos (*Equipment*), representando os materiais utilizados pelos agentes durante execução dos serviços, como mangueira, pá, machado, dentre outros. Caso necessário, novas especializações de recurso podem ser criadas, por exemplo, suprimentos ou veículos de resposta.

As Figura 24 e Figura 25 também destacam as especializações das classes principais descritas anteriormente. A Figura 24 apresenta um conjunto de classes que se referem ao aspecto de comunicação e a Figura 25 ao aspecto de cooperação.

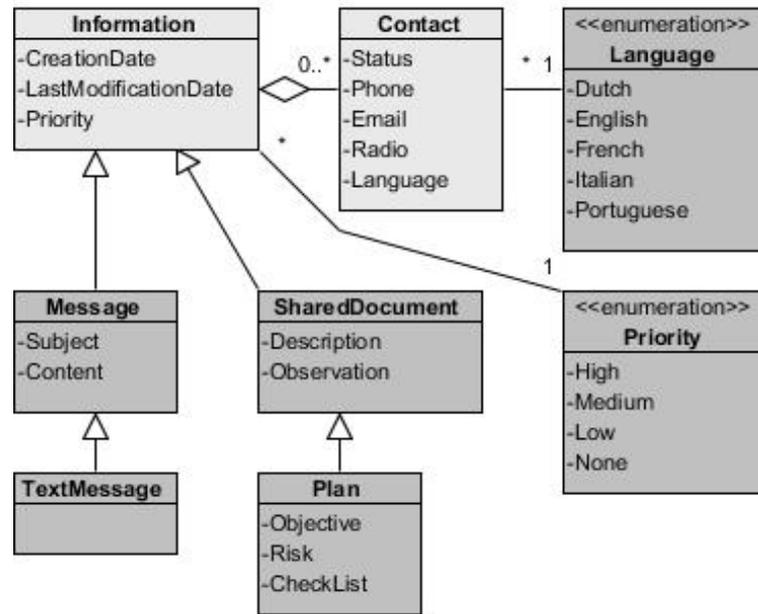


Figura 24: Pacote Núcleo do JEMF destacando as classes para o aspecto de comunicação

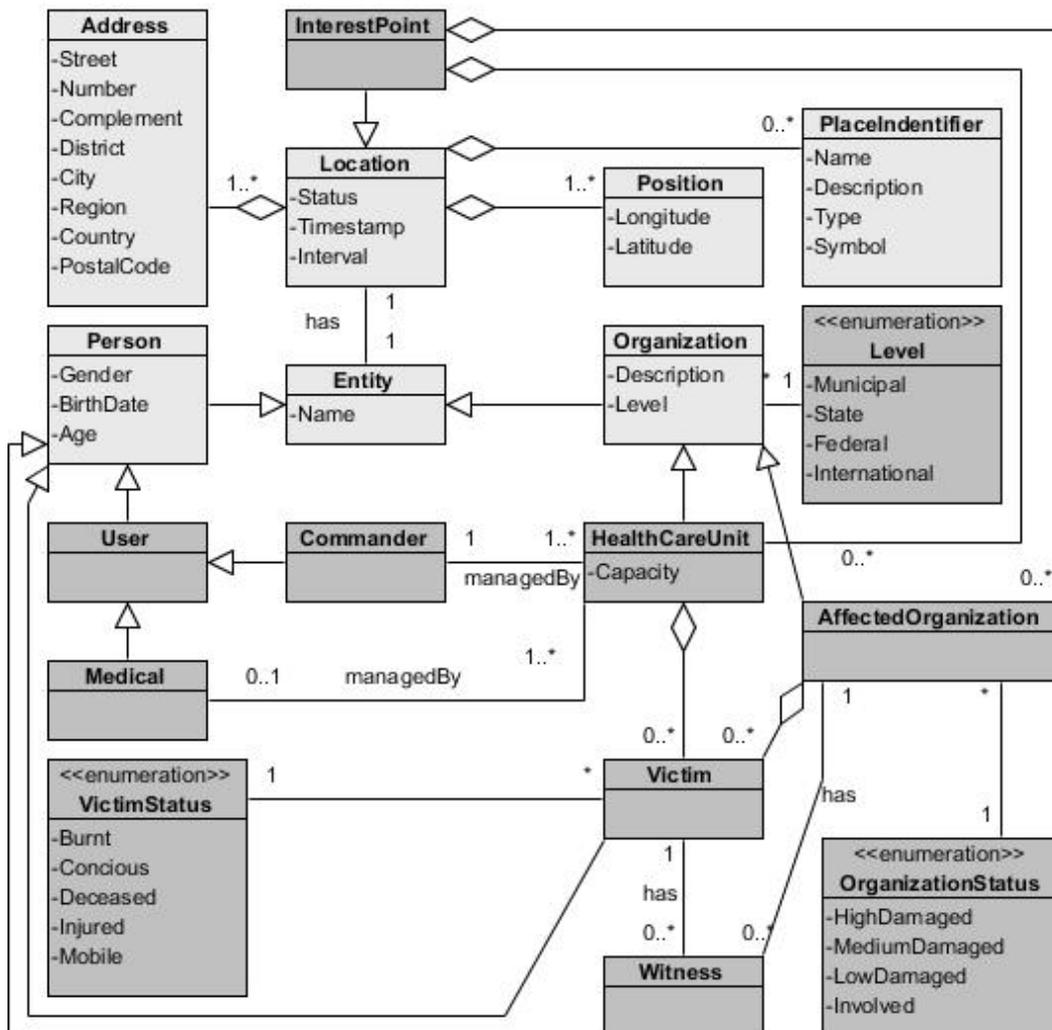


Figura 25: Pacote Núcleo do JEMF destacando as classes para o aspecto de cooperação

Na Figura 24 são apresentadas duas especializações para a classe informação: mensagem (*Message*) e documento compartilhado (*SharedDocument*). A classe mensagem representa os dados que os agentes podem trocar com outros agentes ou organizações. A classe mensagem de texto (*TextMessage*) representa uma especialização que contém as propriedades para esse tipo de mensagem. A classe documento compartilhado representa os documentos que estarão disponíveis entre todos os agentes ou organizações. A classe plano de resposta (*Plan*) contém propriedades específicas para um dos documentos mais importantes que podem ser compartilhados entre agentes numa situação de resposta.

Caso necessário, novas especializações de mensagem podem ser elaboradas, como mensagens para áudio ou vídeo. Já para a classe de documento compartilhado, podem-se estender novas classes com outras propriedades, por exemplo, relatórios.

Na Figura 25 são evidenciadas outras duas especializações para a classe pessoa. A primeira é a vítima (*Victim*), representando as pessoas que foram afetadas pela emergência e necessitam de alguma assistência pelos agentes de resposta. A segunda é a testemunha (*Witness*), representando as pessoas que presenciaram e observaram a ocorrência da emergência e que podem informar dados relevantes sobre o evento.

Na Figura 25 também são observadas duas especializações para a classe organização, onde sua característica principal é definida em função da sua perspectiva ao desastre. A classe unidade de saúde (*HealthCareUnit*) representa as entidades que recebem as vítimas, envolvendo os hospitais ou posto de saúde, no sentido de resgatar vítimas e atenuação das consequências do desastre na parte médica. A classe organização afetada (*AffectedOrganization*) representa as entidades que sofreram avarias ou consequências de um desastre ocorrido, envolvendo qualquer estrutura física, como edifícios ou casas. As classes unidade de saúde e organização afetada podem compor um ponto de interesse (*InterestPoint*), especializando a classe localização.

5.3.2.1.2 Pacote Móvel

O pacote Móvel é destinado às classes concretas que estão atreladas com as propriedades disponibilizadas pela plataforma móvel, como componentes, padrões, e recursos. Dessa maneira, elas poderão ser reutilizadas para integrar a arquitetura de software de um novo sistema móvel. Este pacote contém as operações para apoiar a gestão de dados e permite que os desenvolvedores modifiquem e incrementem métodos oferecidos por essas classes concretas ou agreguem com outras classes da plataforma móvel. A plataforma móvel definida

para este trabalho é o Google Android (ANDROID, 2014a) e será esclarecida na subseção 5.4.1 adiante.

A Figura 26, a Figura 27 e a Figura 28 apresentam as principais classes do pacote Móvel, focando como exemplo a funcionalidade Gerenciar Emergência. Entretanto, a estrutura e as relações entre as classes genéricas desse pacote para as outras funcionalidades são praticamente as mesmas. Nessas figuras são destacados também os padrões de projetos aplicados para apoiar a reutilização de software (notas de observação).

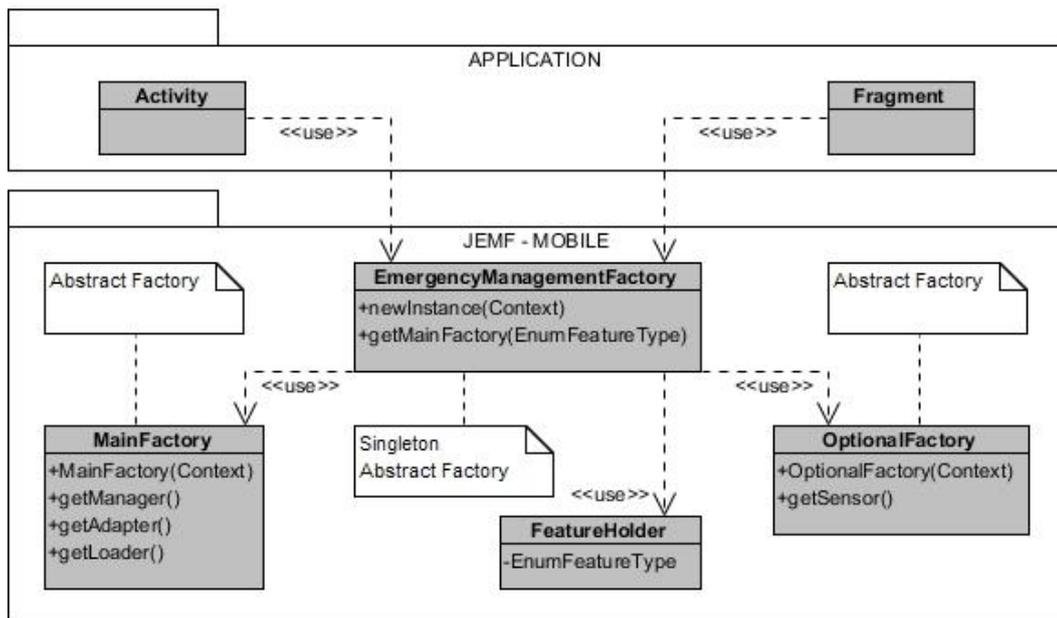


Figura 26: Pacote Móvel do JEMF destacando as superclasses e os padrões de projetos (Parte 1)

Na Figura 26 são apontadas na parte superior duas classes nativas do Android – *Activity* e *Fragment* – que representam como um sistema móvel faz uso do arcabouço. Uma *Activity*⁴ é uma classe focada na interação com o usuário do sistema, sendo essas as classes que servem como janelas (telas) na interface do usuário e gerencia os componentes visuais (Android, 2014c). Uma classe *Fragment*⁵ representa uma parte da interface ou um comportamento que compõem uma *Activity*. Dessa forma, é possível combinar vários *Fragments* em uma única *Activity* para criar uma interface mais completa, ou seja, um *Fragment* funciona como uma seção modular (parte) de uma *Activity* (Android, 2014c). Ainda nessa figura, são destacadas três principais classes e que serão detalhadas a seguir.

⁴ Activity: <http://developer.android.com/reference/android/app/Activity.html>.

⁵ Fragment: <http://developer.android.com/guide/components/fragments.html>.

A classe *EmergencyManagementFactory* tem por objetivo oferecer uma interface abstrata para acessar todas as funcionalidades disponibilizadas pelo arcabouço, administrando a criação de todos os objetos concretos e oferecendo um ponto global de acesso aos novos sistemas móveis.

A classe *OptionalFactory* na versão atual do arcabouço é considerada apenas um gancho para as versões futuras do mesmo, onde espera-se que sejam criadas funções para acessar a API do Android para a manipulação de dados dos recursos de um dispositivo móvel, por exemplo os seus sensores, para apoiar as funcionalidades já evidenciadas neste capítulo. Espera-se também que essa classe possa futuramente gerenciar, se necessário, o acesso às bibliotecas desenvolvidas por terceiros.

A classe *MainFactory* tem por finalidade gerenciar a criação de objetos concretos para as funcionalidades especificadas para o arcabouço. Logo, a partir da notificação do tipo da funcionalidade (Emergência, Missão, etc.) pela instância (novo sistema móvel), o arcabouço poderá manipular os seus objetos concretos para atender as funções requisitadas por ela, como salvar, excluir e carregar dados. Além disso, minimiza a declaração de classes do arcabouço nessa instância para realizar estas mesmas funções.

A classe *FeatureHolder* é uma classe do tipo *Enumeration* (Java, 2014d) que mantém os tipos de todas as funcionalidades disponibilizadas no arcabouço. Assim, por meio dessa classe o arcabouço identifica qual é a funcionalidade que está em operação, segundo a requisição do sistema móvel.

Na Figura 27, a classe *ManagerFactory* tem por função gerenciar a instanciação dos objetos da classe *Manager* de forma a estabelecer somente uma interface para tal operação. A classe *Manager* é uma das principais classes de todo o arcabouço, pois ela define as funções fundamentais para inserir, alterar, excluir e carregar dados do banco de dados. As partes genéricas de todas essas funções foram codificadas nessa classe, de maneira que as partes variantes para atender separadamente cada funcionalidade estabelecida (Emergência, Missão, etc.) foram criadas interfaces que permitem a redefinição para atender as características individuais.

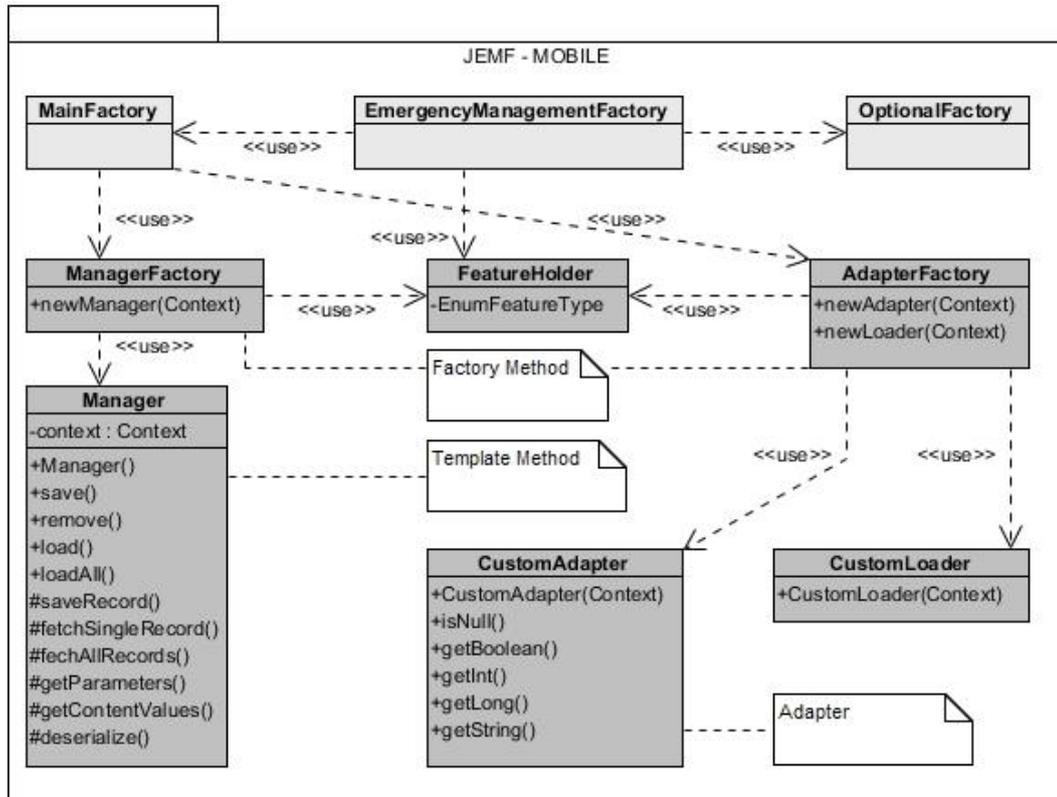


Figura 27: Pacote Móvel do JEMF destacando as superclasses e os padrões de projetos (Parte 2)

A classe *AdapterFactory* tem por objetivo administrar a criação dos objetos através de interfaces para as classes *CustomCursorAdapter* e *CustomLoader*. A classe *CustomCursorAdapter* constitui uma nova interface para a classe *CursorAdapter*⁶ do Android. Por sua vez, o *CursorAdapter* é um adaptador que expõe os dados de um *Cursor*⁷, que mantém os dados carregados do banco de dados para um componente *ListView*⁸, responsável por exibir esses dados em uma lista de itens com rolagem vertical (Android, 2014c).

A classe *CustomLoader* é uma extensão da classe *Loader*⁹ do Android. O *Loader* executa o carregamento assíncrono de dados, e enquanto está ativo, ele deve monitorar a fonte de seus dados e entregar novos resultados quando o conteúdo é alterado (Android, 2014c). Portanto, as classes *CustomCursorAdapter* e *CustomLoader* necessitam trabalhar em conjunto para preservar os dados carregados sempre atualizados no sistema móvel de maneira apropriada.

⁶ CursorAdapter: <http://developer.android.com/reference/android/widget/CursorAdapter.html>.

⁷ Cursor: <http://developer.android.com/reference/android/database/Cursor.html>.

⁸ ListView: <http://developer.android.com/reference/android/widget/ListView.html>.

⁹ Loader: <http://developer.android.com/reference/android/content/Loader.html>.

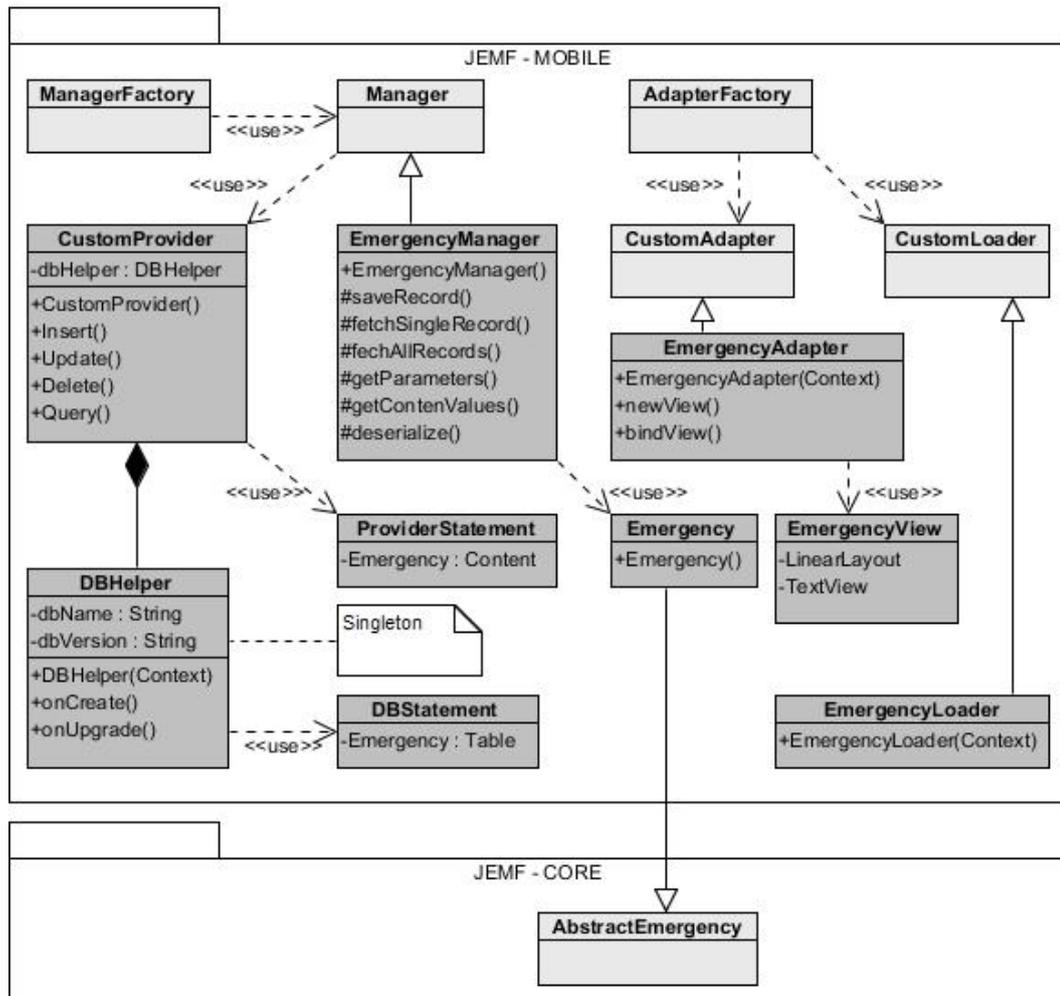


Figura 28: Pacote Móvel do JEMF destacando a classes que compõem a funcionalidade Gerenciar Emergência

Na Figura 28, a classe *CustomProvider* é uma especialização da classe *ContentProvider*¹⁰ do Android. O *ContentProvider* encapsula os dados e através de uma interface única os compartilha para um ou mais sistemas móveis. Por exemplo, os dados de contatos são utilizados por várias aplicações e devem ser armazenado por um único *ContentProvider* (Android, 2014c). A classe *DBHelper* é uma especialização da classe *SQLiteOpenHelper*¹¹. O *SQLiteOpenHelper* é uma classe auxiliar para gerenciar a criação e a versão atual do banco de dados (Android, 2014c). A classe *DBStatement* também é uma classe auxiliar e que organiza todas as declarações das consultas, em inglês *queries*, para

¹⁰ ContentProvider: <http://developer.android.com/reference/android/content/ContentProvider.html>.

¹¹ SQLiteOpenHelper: <http://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html>.

criação e exclusão das tabelas do banco de dados. Já a classe *ProviderStatement* tem uma função similar, porém é para a declaração de propriedades para o *CustomProvider*.

Ainda na Figura 28 é destacado um conjunto de classes que compõem a funcionalidade Gerenciar Emergência: *EmergencyManager*, *EmergencyAdapter* e *EmergencyLoader*. Essas classes herdam das classes genéricas já mencionadas (*Manager*, *CustomAdapter* e *CustomLoader*) as propriedades e funções básicas, e torna-se necessário apenas codificar as partes específicas para trabalhar com os dados da emergência. Além disso, a classe *EmergencyView* detém a codificação de uma interface básica com os componentes visuais já pré-estabelecidos para a funcionalidade Gerenciar Emergência.

Deste modo, para adicionar uma nova funcionalidade é preciso criar respectivamente novas especializações dessas superclasses em destaque na Figura 28. Caso necessário, todas as classes são passíveis de personalização para atender novos requisitos. A Figura 29 exemplifica a instanciação das classes para algumas outras funcionalidades.

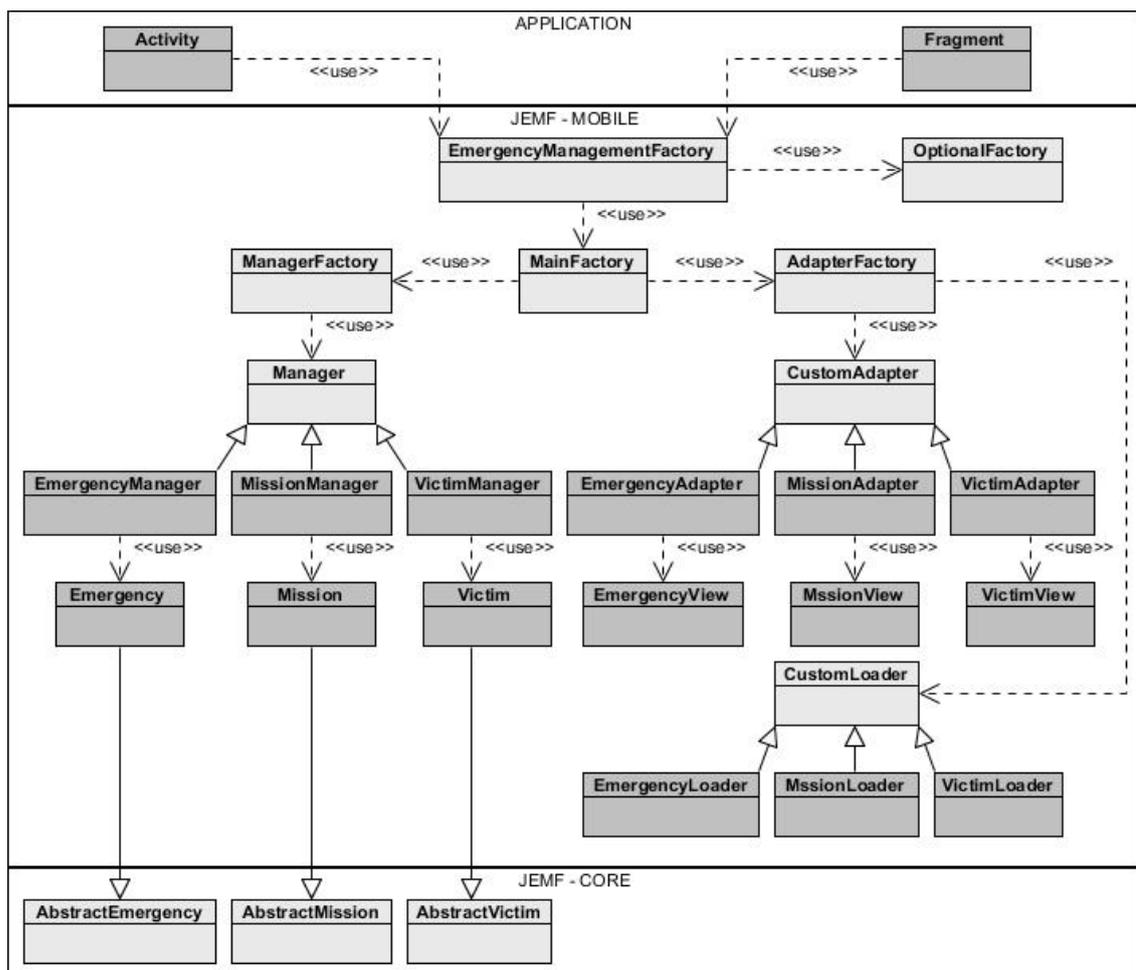


Figura 29: Pacote Móvel do JEMF destacando as classes que compõem as funcionalidades Gerenciar Emergência, Gerenciar Missão e Gerenciar Vítima

Na parte inferior da Figura 28 é apresentado a classe *AbstractEmergency* do pacote Núcleo. A funcionalidade Gerenciar Emergência demanda que seja criada uma classe do tipo *JavaBean* (Java, 2014b). Nesse caso, é definida a classe *Emergency* a partir da concretização da classe *AbstractEmergency* com o objetivo de estabelecer as propriedades da emergência e suas interfaces através dos métodos *getter* e *setter* para trabalhar de forma adequada com a plataforma móvel.

5.3.2.2 Padrões de Projeto

Nesta subseção são destacados e brevemente descritos os padrões de projetos que foram aplicados para aumentar a quantidade de código reutilizável nas superclasses já apresentadas na subseção anterior.

Os padrões de projeto criacionais abstraem o processo de instanciação de objetos, ou seja, eles ajudam a fazer um sistema independente onde seus objetos são criados, compostos e representados. Um padrão criacional usa a herança para variar a classe que é instanciada e traz dois temas recorrentes nesses padrões. O primeiro, eles encapsulam o conhecimento sobre quais classes concretas que o sistema utiliza. O segundo, eles escondem como as instâncias dessas classes são criadas e associadas (GAMMA et al., 1995). Os padrões criacionais usados neste projeto foram o Abstract Factory, o Factory Method e o Singleton.

O Abstract Factory fornece uma interface para criar objetos genéricos, removendo as dependências de classes concretas para os clientes que criam esses objetos (GAMMA et al., 1993). O principal objetivo dele é oferecer uma interface para a criação de uma família de objetos relacionados, sem especificar explicitamente suas classes para uma aplicação cliente (GAMMA et al., 1995). O Abstract Factory determina qual é o tipo do objeto concreto que se deseja criar, e depois de criá-lo, retorna um ponteiro abstrato desse objeto para o cliente. O fato desse padrão retornar um ponteiro do objeto criado significa que o cliente não tem conhecimento do tipo real do objeto. Isto implica que não há nenhuma necessidade para a inclusão de quaisquer declarações de classe relacionadas com o tipo concreto na aplicação cliente, pois ela acessa em todos os momentos o ponteiro abstrato. Os objetos do tipo concreto, criados pela fábrica, são acessados pela aplicação cliente apenas através da interface abstrata.

O Factory Method “permite que classes base criem instâncias de objetos dependentes de subclasse” (GAMMA et al., 1993). Ele estabelece uma interface para a criação de objetos, mas deixa que as subclasses decidam qual classe será instanciada, referindo-se ao objeto recém-criado através de uma interface comum (GAMMA et al., 1995).

O Singleton define um objeto que fornece acesso único aos serviços e variáveis bem conhecidas (GAMMA et al., 1993). Ele é responsável por fornecer um ponto global de acesso a um objeto, certificando-se de que apenas seja criada uma instância da sua classe. Neste caso, a mesma instância pode ser usada por várias vezes e em vários locais, sendo impossível invocar diretamente o construtor da classe (GAMMA et al., 1995).

Os padrões estruturais lidam com a forma como as classes e os objetos são compostos para criar estruturas maiores. Esses padrões usam a herança para compor interfaces ou implementações e são úteis para que classes desenvolvidas independentemente possam trabalhar juntas (GAMMA et al., 1995). O padrão estrutural aplicado neste projeto foi o Adapter.

O Adapter “faz o protocolo de uma classe estar em conformidade com o protocolo de outra classe” (GAMMA et al., 1993). Como qualquer adaptador no mundo real, ele é usado para ser uma interface, uma ponte entre dois objetos que não poderiam de outra forma trabalhar em conjunto. Ele converte a interface de uma classe para outra interface, permitindo que classes com interfaces diferentes e incompatíveis se comuniquem (GAMMA et al., 1995).

Os padrões comportamentais estão envolvidos com a atribuição de responsabilidades entre os objetos e algoritmos. Eles descrevem os padrões de comunicação entre as classes e os objetos, assim como caracterizam o fluxo de controle em tempo de execução. Esses padrões se concentram na forma como os objetos estão interligados e usam a herança para distribuir o comportamento entre as classes (GAMMA et al., 1995). O padrão estrutural usado neste projeto foi o Template Method.

O Template Method “implementa um algoritmo abstrato, adiando passos específicos para métodos da subclasse” (GAMMA et al., 1993). Ele define o esqueleto de um algoritmo de uma operação, repassando algumas partes do algoritmo para as subclasses. Dessa maneira, ele permite que as subclasses redefinam determinadas etapas do algoritmo sem deixá-los mudar ou replicar toda a estrutura do algoritmo (GAMMA et al., 1995).

5.3.2.3 Diagramas de Sequência

O diagrama de sequência descreve a passagem de mensagens com o fluxo de tempo em um sistema e inclui anotações para requisitos específicos de sincronismo (DOUGLASS; CERNOSEK, 1997). Um conjunto de diagramas de sequência foi elaborado de acordo com as funcionalidades já abordadas e visa detalhar as interações entre as classes do arcabouço, seguindo as descrições já definidas nesta etapa, modelagem de projeto (Seção 5.3.2.1). Todos esses diagramas são detalhados no Apêndice C, seguindo as principais operações

estabelecidas para as funcionalidades: Incluir, Alterar, Consultar e Excluir.

5.3.2.4 Modelo do Banco de Dados

O projeto de banco de dados é um processo de modelagem de entidades do mundo real que contém informações selecionadas necessárias para um projeto de software. O Modelo Entidade-Relacionamento (MER) é usado em muitos desenvolvimentos de banco de dados e, assim como o UML, existem algumas notações gráficas que podem representá-lo. Esse modelo, inclusive, pode parecer semelhante ao diagrama de classes da UML (JEWETT, 2006).

Cada classe é definida por seu conjunto de atributos, tanto na UML quanto no MER, também chamados de propriedades em algumas linguagens OO. Cada atributo é um pedaço de informação que caracteriza cada um dos membros desta classe no banco de dados. Juntos, eles fornecem a estrutura para formar as tabelas do banco de dados ou os objetos em código-fonte. Em uma linguagem de programação OO, cada classe é instanciada com objetos dessa classe. Na construção de um banco de dados relacional, cada classe é traduzida pela primeira vez em um esquema de modelo relacional. Quando o banco de dados é construído, cada esquema de relação torna-se a estrutura de uma tabela.

O trabalho de Jewett (2006) é uma importante referência para o desenvolvimento do modelo de banco de dados deste projeto, pois traz de maneira clara e objetiva as principais questões entre a modelagem UML e o projeto de banco de dados relacional. Dentre os temas do trabalho, podem-se destacar os que foram essenciais para o arcabouço proposto, como é o caso da especialização de classes, que representa um subconjunto de classes de uma superclasse. Além disso, a agregação de classes, as classes do tipo enumeração e outros temas são exemplificados para formação de uma solução final consistente.

O diagrama a seguir apresenta o MER estabelecido para o arcabouço. No Apêndice D são descritos os atributos das principais tabelas desse diagrama ilustrado na Figura 30.

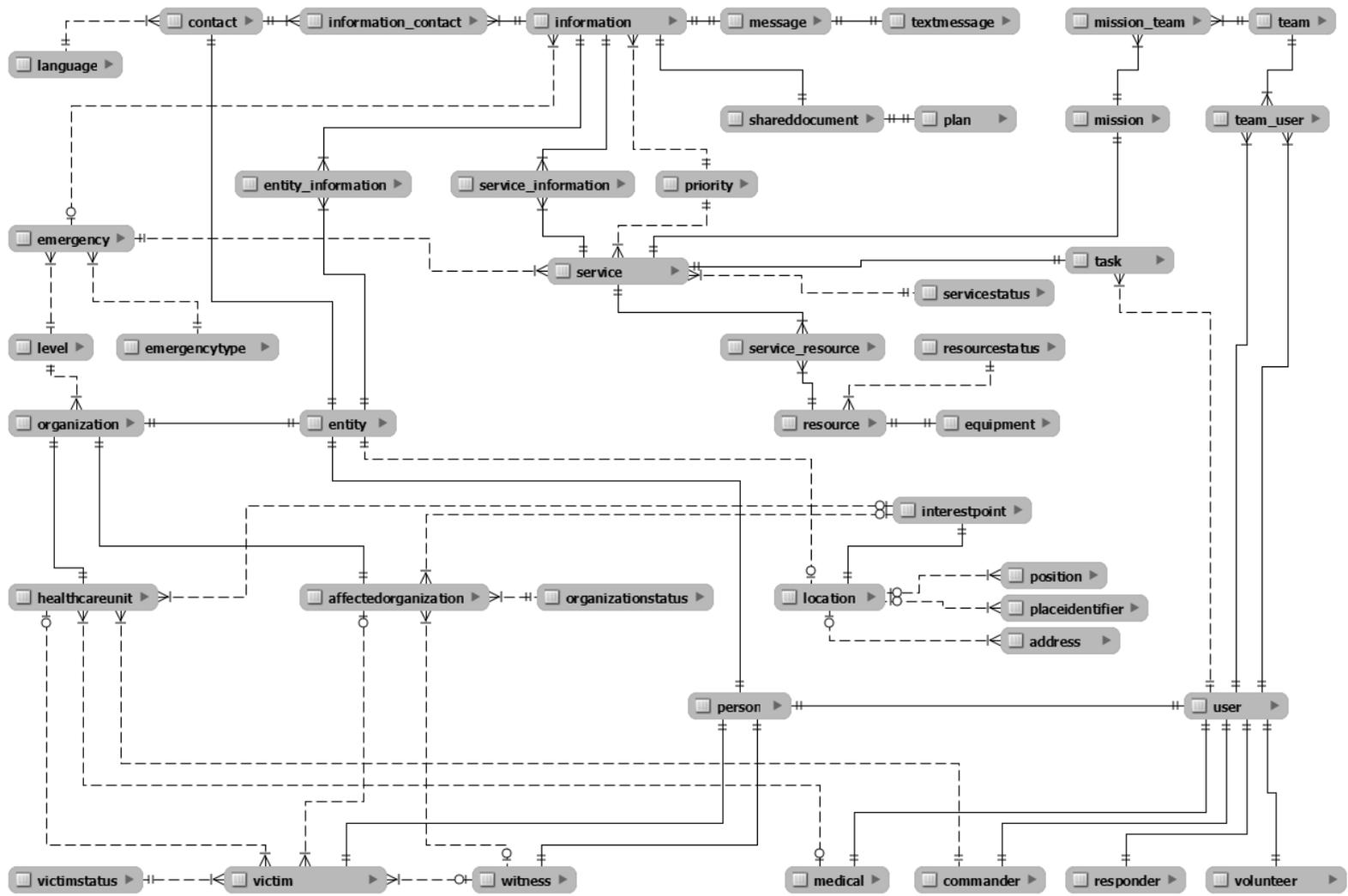


Figura 30: Diagrama do Modelo Entidade-Relacionamento do JEMF

5.4 Implementação do Arcabouço

Esta seção tem por objetivo apresentar a fase de implementação da metodologia adotada.

5.4.1 Plataforma Móvel

O JEMF foi desenvolvido para a plataforma Google Android (Android, 2014a), que é um dos sistemas operacionais móveis mais difundidos hoje em dia (IDC, 2012). A plataforma Android é uma coleção de software que inclui um sistema operacional e um número de bibliotecas de alto nível que simplificam a tarefa de se comunicar com o sistema operacional. Ele também inclui vários aplicativos que os usuários de smartphones utilizam, como telefone, cliente de e-mail, gerenciador de contatos, mapas, navegador Web, calendário e assim por diante.

Tudo no ambiente de desenvolvimento Android, bem como todos os aplicativos incluídos, pode ser programado com uma combinação de códigos escritos em Java e arquivos XML, graças ao seu Kit de Desenvolvimento de Software (Android, 2014e), em inglês *Software Development Kit* (SDK). Este SDK traduz em tempo de execução o código Java e o XML escritos pelo desenvolvedor em uma linguagem que o sistema operacional e o dispositivo entendem.

A base sobre a qual é construído o Android foi codificada para o sistema operacional Linux. O Linux e seus serviços gerenciam o telefone físico, dando acesso as suas principais funções aos aplicativos, como tela sensível ao toque, armazenar dados na memória, segurança dos dados, vários receptores e transmissores de rede, sensor de GPS e muito mais.

O Android possui também um número de bibliotecas que fornecem outros serviços para trabalhar com gráficos, formatos de áudio e vídeo, e funções personalizadas do mais amplo uso para as aplicações, por exemplo, fazer chamadas, enviar mensagens, capturar imagens com a câmera, etc. Isso permite ao desenvolvedor criar aplicativos que possam acessar todas as principais funções e sensores do dispositivo e trabalhar com os dados gerados por eles. Outra principal característica, o Android é um projeto de código aberto que pode incentivar os desenvolvedores a modificá-lo e construí-lo para atender às suas necessidades e redistribuir as alterações disponíveis para todos os outros interessados.

O JEMF está configurado para o funcionamento a partir da versão 4.0.3 do Android, mais conhecido em inglês como *Ice Cream Sandwich*. Conforme apresentado na Figura 31 é evidenciado que a maior parte dos dispositivos atualmente utiliza o Android a partir da versão

4.0.3, o que possibilita o JEMF operar em qualquer versão superior. Esta versão corresponde a API nível 15, em inglês *API Level 15*, da plataforma Android, sendo esse identificador usado pelos aplicativos para interagir com a plataforma (Android, 2014g). Cada versão sucessiva do Android pode incluir atualizações para a API que ele oferece. Essas atualizações para a API são projetadas para que uma nova versão permaneça compatível com as versões anteriores, ou seja, a maioria das alterações na API é aditiva e introduzem novas ou substituem funcionalidades. Como partes da API são atualizadas, as partes mais antigas se tornam obsoletas, mas não são removidas, para que aplicativos existentes ainda possam usá-las.

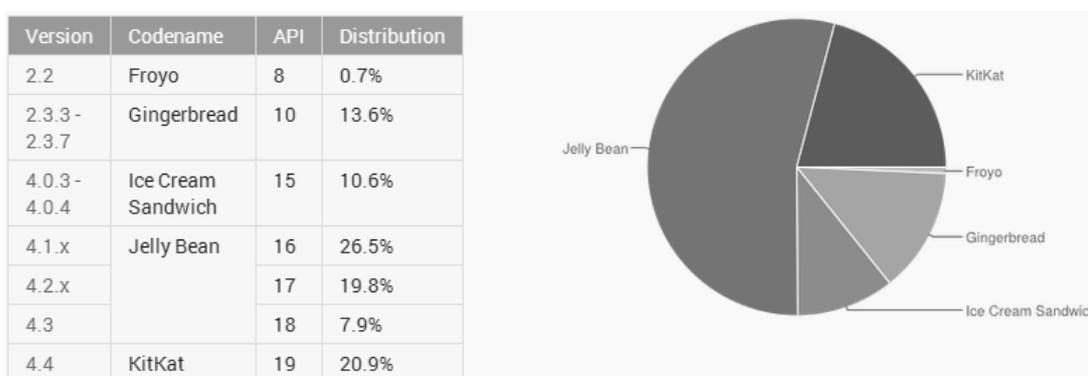


Figura 31: Percentual de distribuição de dispositivos móveis para cada versão da plataforma Android (Android, 2014d)

Portanto, o arcabouço proposto é baseado na plataforma Android, visto que ele representa a melhor alternativa para este trabalho e também pela sua adoção com maior frequência como plataforma escolhida pelos trabalhos mais recentes em Gestão de Emergência, conforme visto nas análises do Capítulo 3 (Seção 3.13).

5.4.2 Programação-por-diferença

Uma vez que o JEMF é considerado um arcabouço do tipo caixa-branca, a principal característica evidenciada na programação orientada a objetos nesse contexto é a herança de classe. Por meio da herança, a classe pode herdar as operações e a estrutura interna de uma superclasse. Essa classe pode adicionar novas operações, mas não pode excluir as operações herdadas, apenas redefini-las. Uma das vantagens que a herança proporciona é a reutilização de código, visto que uma classe pode começar com o código compartilhado de sua superclasse.

A programação-por-diferença (JOHNSON; FOOTE, 1988) foi adotada para implementar as classes do arcabouço e outros desenvolvedores podem da mesma forma usá-la com o código do JEMF em seus sistemas móveis. O desenvolvedor define uma nova classe

escolhendo uma classe intimamente relacionada como sua superclasse e descreve as diferenças entre a classe nova e a antiga. Isto é, quando é necessário criar uma nova classe deve-se criar uma subclasse de uma classe já disponibilizada pelo arcabouço e substituir os métodos que são diferentes, específicos para o novo sistema. Então, as principais classes abstratas contêm as partes comuns de código e métodos originais podem ser definidos uma vez no JEMF, permitindo que todas as subclasses contenham somente a parte específica.

Outro benefício da herança de classe é a possibilidade do código-fonte original permanecer intacto, mesmo que extensões sejam feitas pelos desenvolvedores. Nesse caso, torna-se menos propenso que a alteração realizada por um desenvolvedor afete o código de outro.

Ambos os pacotes (núcleo e móvel) podem ser personalizados para oferecer suporte a novos sistemas, porque todas as classes e código estão disponíveis para os desenvolvedores, uma vez que este arcabouço é definido como código aberto. Tanto o código do arcabouço e seus exemplos estão totalmente documentados para que eles entendam as suas propriedades facilmente.

5.4.3 Código Aberto

O JEMF é um projeto de código aberto e licenciado sob a Licença Pública Geral (GNU, 2014b), em inglês *General Public License* (GPL). A GPL é uma designação da licença para software livre no âmbito do Projeto GNU da organização Free Software Foundation (GNU, 2014c). A licença GPL foi originalmente publicada em Janeiro de 1989. No entanto, o arcabouço é baseado na GPL Versão 3 (ou apenas GPLv3), seguindo a versão final que foi lançada em 29 de Junho de 2007.

Em termos gerais, a GPL baseia-se em quatro liberdades (GNU, 2014a):

- **Liberdade nº 0:** a liberdade de executar o programa, para qualquer propósito;
- **Liberdade nº 1:** a liberdade de estudar como o programa funciona e adaptá-lo para as suas necessidades. O acesso ao código-fonte é um pré-requisito para esta liberdade;
- **Liberdade nº 2:** a liberdade de redistribuir cópias de modo que se possa ajudar ao seu próximo (desenvolvedor);
- **Liberdade nº 3:** a liberdade de aperfeiçoar o programa, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie deles. O acesso ao código-fonte é um pré-requisito para esta liberdade.

Portanto, estas liberdades asseguram que os softwares sejam distribuídos e reaproveitados. Porém, os direitos do autor são mantidos de modo a não permitir que essa informação seja usada de uma maneira que limite as liberdades originais. A licença não autoriza que o código seja apoderado por outra pessoa, ou que sejam impostos sobre ele restrições que impeçam que seja distribuído da mesma forma que foi adquirido. Ademais, a GPL está redigida em inglês e atualmente nenhuma tradução é aceita como válida. Esse argumento previne dos riscos de introdução de erros de tradução que poderiam deturpar a essência da licença. Deste modo, neste trabalho se manteve a obrigatoriedade de distribuir o texto oficial em inglês com o código-fonte do arcabouço.

5.4.4 Gerência de Configuração

Para a gerência de configuração deste projeto foi adotado o Apache Subversion (SVN, 2014). Ele é o sistema de controle de versão usado para manter as versões atuais e históricas de arquivos tais como o código-fonte e a documentação do projeto. Esse sistema permite rastrear as alterações feitas durante o desenvolvimento do software com um controle do histórico do projeto, facilitando analisar e resgatar as versões antigas dos arquivos. Ele também permite separar o projeto em linhas de desenvolvimento para implementações paralelas, apoiando o trabalho em equipe sobre o mesmo conjunto de arquivos ao mesmo tempo.

O código-fonte e a documentação do arcabouço estão disponíveis na Web para que qualquer desenvolvedor possa baixar e utilizar este projeto (JEMF, 2014a). Uma cópia deste projeto se encontra disponibilizado também na plataforma GitHub (JEMF, 2014b).

5.4.5 Ambiente de Desenvolvimento

O ambiente de desenvolvimento recomendado para a implementação de sistemas com o uso do JEMF é o Android SDK em conjunto com o Android Developer Tools (ADT) (Android, 2014b). O ADT é um plug-in que fornece um conjunto de ferramentas integradas com o Eclipse (Eclipse, 2014a) para desenvolvimento em Java. Ele oferece acesso a muitos recursos que ajudam todo o desenvolvimento de aplicativos Android, principalmente para a codificação e a prototipagem de interfaces de usuário. Assim, o ADT constitui-se como um ambiente de desenvolvimento bem estabelecido por oferecer recursos específicos para construção de aplicativos de forma rápida e organizada.

Além do ADT, é recomendada a instalação do plug-in Subversive (Eclipse, 2014b) no Eclipse para fornecer suporte ao desenvolvimento de projetos armazenados em repositórios

Apache Subversion, habilitando o acesso ao código-fonte e a documentação do arcabouço de maneira simples.

5.5 Teste do Arcabouço

“O objetivo do teste de software é descobrir erros” (PRESSMAN, 2006). Mas para alcançar esse objetivo, primeiro deve ser planejado e executado uma série de passos de testes, onde cada passo é apoiado por estratégias que compreendem o projeto de casos de teste.

Uma estratégia de teste de software integra métodos e atividades que resultam na construção bem-sucedida de software. Por isso, a estratégia deve prover um roteiro que descreve os passos que são conduzidos como parte do teste, e depois de executados sistematicamente, podem-se avaliar os dados e tempo que foram obtidos.

A estratégia também estabelece testes de baixo nível, que são necessários para verificar se um pequeno segmento de código-fonte foi corretamente implementado, bem como testes de alto nível, que validam as principais funções do sistema com base nos seus requisitos. A qualidade do software pode ser avaliada e confirmada quando ocorre a aplicação adequada de métodos e ferramentas de teste (PRESSMAN, 2006).

Esta seção tem por objetivo apresentar a fase de teste da metodologia adotada.

5.5.1 Teste de Unidade e de Integração

Os testes de unidade e de integração concentram-se na verificação funcional de um componente e na incorporação de componentes em uma estrutura de software. O teste de unidade tem por objetivo verificar a menor unidade de projeto do software – a classe, o componente ou módulo de software. Segundo Radatz, Geraci e Katki (1990) é “testar unidades individuais de software”. A lógica interna de processamento e as estruturas de dados dentro dos limites de um componente são testados para descobrir os erros. Esse tipo de teste pode ser conduzido em paralelo para diversos componentes.

Antes que qualquer teste seja iniciado, o teste de fluxo de dados através da interface da classe ou do componente deve ser considerado para verificar se está adequado. Caso contrário, qualquer outro teste pode ser questionável. Quando um componente com alta coesão é projetado, isto é, quando um pequeno número de funções é implementada por um componente, o número de casos de teste é reduzido e assim os erros são descobertos mais facilmente. Nos casos onde os componentes não podem ser testados no nível de unidade, o teste pode ser adiado até o passo de teste de integração.

Segundo Pressman (2006), o teste de integração é “uma técnica sistemática para construir a arquitetura do software enquanto, ao mesmo tempo, conduz testes para descobrir erros associados às interfaces”. Radatz, Geraci e Katki (1990) definem o teste de integração como “teste em que os componentes de software são combinados e testados para avaliar a interação entre eles”.

A realização do teste de integração para sistemas OO é baseada no caminho de execução, ou seja, cada caminho de execução é testado individualmente ao integrar um conjunto de classes que correspondem a uma entrada ou um evento do sistema. Para testar grupos inteiros de classes, é empregado o uso de pseudocontroladores durante o teste de operações do sistema, por exemplo, eles substituem a interface do usuário de modo que testes de funcionalidades possam ser conduzidos quando a interface ainda não foi totalmente implementada ou mesmo antes da sua implementação.

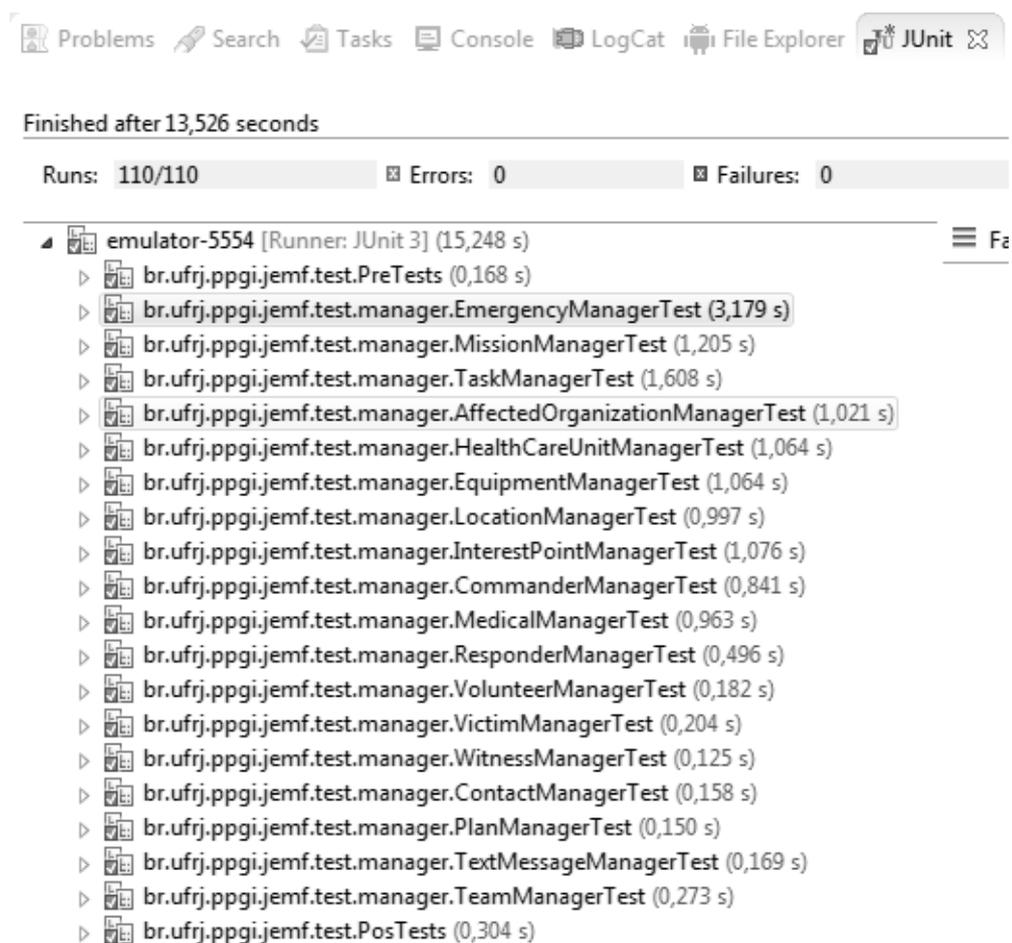


Figura 32: Mensagens de teste bem-sucedido do JEMF pelo JUnit

Durante a fase de teste deste trabalho, um conjunto de casos de teste foi desenvolvido para capturar e diminuir os problemas e erros no código-fonte do JEMF. Os testes de unidade e de integração foram definidos de acordo com as descrições dos casos de uso apresentados no Apêndice B. Tal atividade foi apoiada com o uso do arcabouço JUnit (GAMMA; BECK, 1999). Com o JUnit, a criação e manutenção de testes se tornam fáceis, além de fornecer uma interface para definir e executar conjuntos de testes automatizados. Isso diminui o esforço ao reutilizar recursos de teste e faz com que os processos de testes complexos sejam mais gerenciáveis. Os conjuntos de testes da plataforma Android são baseados em JUnit (Android, 2014f), visto que é um arcabouço simples, bem projetado, implementado em Java, disponível gratuitamente e amplamente adotado. Enfim, os resultados dos testes malsucedidos geraram manutenções no código-fonte do arcabouço até que todos os problemas encontrados fossem corrigidos. Isso conduziu aos resultados bem-sucedidos para as funcionalidades do JEMF demonstrados na Figura 32.

5.5.2 Documentação

Dado que um arcabouço é uma técnica importante e mais completa no reuso de software, ele pode ser mais difícil de aprender. Isso significa que ele requer uma melhor documentação e um maior tempo de aprendizado do que outras técnicas. Identificar os padrões de projetos e outros detalhes apenas lendo o seu código-fonte não é o mais adequado (JOHNSON, 1997b).

Ao invés dos desenvolvedores seguirem diretamente por esse caminho, o arcabouço precisa anteriormente ser introduzido de forma responsável para que os eles possam aprender gradualmente sobre os seus requisitos básicos, funcionalidades e componentes. Na prática, isso pode ser realizado por meio de sessões de apresentação, aplicativos de exemplo, ampla documentação e tutoriais (FROEHLICH et al., 1998).

A fim de reduzir o esforço de aprendizagem, torna-se desejável ter um “livro de receitas”, também conhecido em inglês como *cookbook*, que deve incluir uma coleção de receitas que ilustram o uso do arcabouço ao estabelecer os passos a serem seguidos para produzir um sistema ou parte de um sistema (WEINAND; GAMMA; MARTY, 1989).

Neste trabalho foi elaborado um manual que apoiará os desenvolvedores de sistemas a alavancar os conceitos do domínio e fornecer conhecimento com receitas ao trazer uma compilação dos principais tópicos do JEMF. Mais especificamente, nesse manual são encontradas instruções para configuração do ambiente de desenvolvimento, instruções para

um novo sistema móvel instanciar o arcabouço, exemplos para codificação e uso das principais funcionalidades e métodos oferecidos, dentre outras orientações.

Além do manual, o próprio código-fonte está todo documentado com avisos e sugestões para facilitar a compreensão dos desenvolvedores. Através desses comentários foi gerado e anexado a este projeto também um Javadoc (Java, 2014c). Essa ferramenta é outra forma usada para documentação que possibilita a realização de consultas a todas as classes do JEMF em formato HTML, colaborando para viabilizar sua utilização.

5.5.3 Instanciação

Além de uma documentação completa para auxiliar o processo de aprendizagem, também foram desenvolvidos sistemas móveis de exemplo para demonstrar o funcionamento do JEMF num cenário real de implementação. Esse apoio adicional permite ver como as diferentes classes do arcabouço se encaixam e como as funcionalidades providas são executadas. A Figura 33 apresenta algumas interfaces da funcionalidade Gerenciar Emergência de um dos sistemas móveis de exemplo que acompanham o JEMF.

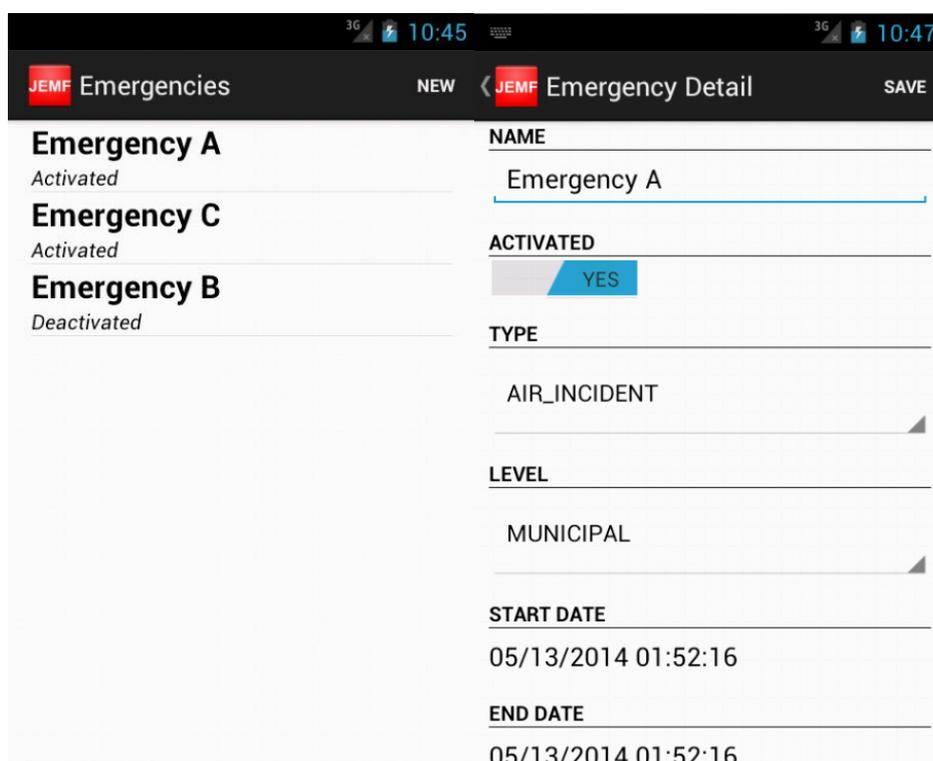


Figura 33: Interfaces do sistema móvel de exemplo do arcabouço

A fim de obter sucesso na adoção generalizada do JEMF, estes exemplos são distribuídos com código-fonte do arcabouço para ajudar os desenvolvedores a começarem

rapidamente o desenvolvimento de um novo sistema. Os exemplos podem inclusive auxiliá-los na decisão de adotar ou recusar o uso deste arcabouço, de acordo com os objetivos e necessidades de seus projetos. Ademais, os exemplos permitiram executar uma investigação prévia e testes com o objetivo de analisar a estabilidade do arcabouço antes de oferecê-lo aos desenvolvedores.

6 Avaliação do Arcabouço

Este capítulo tem por objetivo abordar a avaliação qualitativa proposta para o arcabouço apresentado neste trabalho. Esta avaliação se baseia em dois enfoques que são discutidos separadamente. O primeiro enfoque trata de uma avaliação realizada junto a desenvolvedores de software. O segundo enfoque é dedicado a uma avaliação aplicada junto a especialistas do domínio. Então, através das análises geradas, foram obtidas as correções, perspectivas de uso e as contribuições deste arcabouço para o desenvolvimento de sistemas móveis no domínio de Gestão de Emergências.

6.1 Avaliação por Desenvolvedores de Software

Esta avaliação se consiste na disponibilização do JEMF para um grupo de desenvolvedores de software e na implementação de sistemas móveis com o uso do JEMF. Assim, perguntas foram aplicadas por meio de questionários para cada participante. O questionário é uma técnica de aquisição de conhecimento que possibilitou validar este trabalho de maneira simples e objetiva.

Os questionários definidos para a avaliação do arcabouço por desenvolvedores de software são:

- Questionário de Caracterização do Participante;
- Questionário de Avaliação dos Exercícios do Arcabouço;
- Questionário de Avaliação Geral do Arcabouço.

No Apêndice E estão descritos todos os questionários e os exercícios definidos para esta avaliação.

6.1.1 Caracterização do Participante

O questionário de caracterização do participante é o primeiro a ser aplicado, e através dele, espera-se obter um perfil do entrevistado ao investigar os seguintes itens:

- Quanto a sua escolaridade;
- Quanto a sua experiência em programação orientada a objetos e na linguagem de programação Java;
- Quanto a sua experiência em desenvolvimento para dispositivos móveis, qual a plataforma experimentada (Android, iOS ou Windows), e quais ferramentas usadas para implementação de sistemas móveis.

6.1.2 Exercícios de Avaliação do Arcabouço

Este experimento compreende determinados exercícios na qual o participante deve desenvolver um novo sistema para a gestão de emergências a partir da descrição de um cenário simples de utilização do JEMF. Esses exercícios foram baseados nas especificações do projeto Sistema de Emergência, detalhado no Capítulo 3. De modo geral, são as seguintes tarefas:

- 1) Os participantes inicialmente analisam a descrição do novo sistema para identificar as suas principais funcionalidades e seus requisitos;
- 2) Os participantes são convidados a analisar a proposta de projeto para o novo sistema através de um diagrama de classes, e finalmente;
- 3) Os participantes devem implementar o novo sistema com uso do arcabouço e verificar quando é necessário modificar ou estender as funcionalidades já providas pelo arcabouço.

Isto posto, os exercícios definidos para a avaliação do arcabouço visam examinar alguns dos principais aspectos almejados por arcabouços de software (RADATZ; GERACI; KATKI, 1990; ADAIR, 1995; FAYAD; SCHMIDT, 1997; SCHMIDT; BUSCHMANN, 2003):

- **Reusabilidade:** facilidade na qual um software pode ser usado em mais de um sistema de computador. As interfaces estáveis fornecidas pelos arcabouços aprimoram a reutilização através da definição de componentes genéricos que podem ser reaplicados para criar novos sistemas. Ao aproveitar o conhecimento do domínio, pode-se evitar a recodificação e revalidação de soluções comuns, bem como produzir melhoria na qualidade e na confiabilidade do software;
- **Flexibilidade:** facilidade na qual um software pode ser modificado para uso em sistemas ou ambientes diferentes daqueles para os quais foi especificamente projetado. A flexibilidade permite que abstrações sejam usadas em diferentes contextos para suportar uma variedade de requisitos;
- **Extensibilidade:** facilidade na qual um software pode ser modificado para aumentar a sua capacidade funcional. A extensibilidade apoia atualizações e adições para atender novos requisitos ao fornecer métodos gancho, onde os softwares clientes podem personalizar um arcabouço ao derivar novas classes.

A cada exercício executado foi estabelecido um aumento gradual na obtenção de conhecimento e aprendizado sobre o arcabouço, e conseqüentemente, um aumento no grau de dificuldade dos exercícios até a conclusão de todo o experimento.

6.1.2.1 Avaliação da Reusabilidade do Arcabouço

Este exercício tem por objetivo averiguar a reutilização do arcabouço proposto. Espera-se obter a experiência do desenvolvedor quanto ao uso do arcabouço e seus primeiros passos:

- a) O desenvolvedor deve criar um novo projeto móvel para a plataforma Android;
- b) O desenvolvedor deve importar do arcabouço e a fazer a conexão com o seu novo projeto;
- c) O desenvolvedor deve implementar uma funcionalidade identificada na especificação com o uso de uma instância do arcabouço. Nesse exemplo, instanciar a funcionalidade para gestão da emergência que já é provida pelo arcabouço.

6.1.2.2 Avaliação da Flexibilidade do Arcabouço

Este exercício tem por objetivo averiguar a flexibilidade do arcabouço proposto. Espera-se obter a experiência do desenvolvedor quanto à manipulação das funções e propriedades do arcabouço:

- d) O desenvolvedor deve realizar o registro de dados através das funções disponibilizadas pelo arcabouço para a funcionalidade implementada no exercício anterior (Exercício C). Com isso, avaliar a facilidade do acionamento das operações para incluir, alterar, excluir dados, por exemplo, da emergência;
- e) O desenvolvedor deve realizar alterações de propriedades da funcionalidade do arcabouço para que atenda aos requisitos do novo sistema, por exemplo, modificar um atributo já existente para a emergência, assim como adicionar um novo atributo.

6.1.2.3 Avaliação da Extensibilidade do Arcabouço

Este exercício tem por objetivo averiguar a expansão do arcabouço proposto. Espera-se obter a experiência do desenvolvedor quanto à criação de novas funcionalidades para o arcabouço:

- f) O desenvolvedor deve implementar uma nova funcionalidade, por exemplo, para a gestão de suprimentos (recursos materiais como água, comida, remédio,

etc.). Notadamente, propor a codificação de classes a partir da herança dos componentes existentes no arcabouço, como classes para as camadas de visão (*layout*), controle (manipulação e carregamento de dados) e modelo (JavaBean e tabela para o banco de dados).

6.1.3 Avaliação de Experiência Geral do Arcabouço

O questionário de experiência geral é o último a ser aplicado, e por meio dele, espera-se obter o parecer do entrevistado após o uso prático do arcabouço proposto ao questionamento dos itens:

- Quanto à facilidade de uso do arcabouço e de aprendizagem no domínio de Gestão de Emergências;
- Quanto à documentação do arcabouço;
- Quanto à experiência geral, aos erros encontrados durante o uso do arcabouço, e as dúvidas e sugestões para este projeto.

6.1.4 Resultados Coletados dos Desenvolvedores

Nesta seção são examinados os resultados obtidos na avaliação por desenvolvedores de software, conforme definido nas subseções anteriores (6.1.1, 6.1.2 e 6.1.3).

Esta avaliação foi aplicada para um conjunto total de quatro desenvolvedores de sistemas para dispositivos móveis. Segundo os resultados coletados em atividades práticas e questionários, eles possuem diferentes níveis de experiência entre os temas fundamentais abordados neste trabalho. Nos próximos gráficos são apresentadas detalhadamente as conclusões dos desenvolvedores participantes.

A Figura 34 mostra que os desenvolvedores, em sua maioria, são considerados juniores em função do tempo de experiência em programação orientada a objetos e Java, entre um mês a doze meses. Apenas um desenvolvedor possui mais de três anos e, então, é considerado como desenvolvedor pleno.

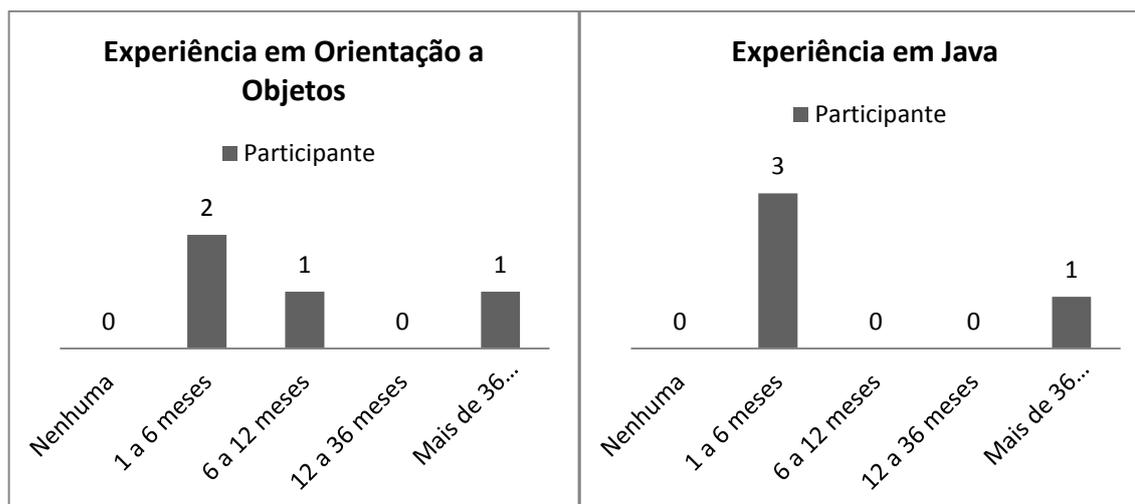


Figura 34: Resultados sobre a experiência dos desenvolvedores em OO e Java

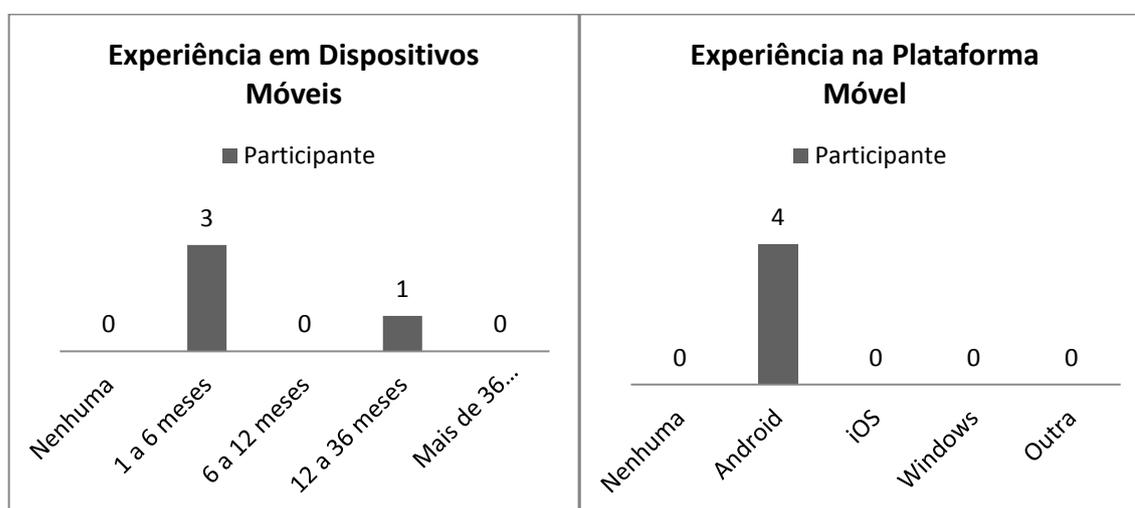


Figura 35: Resultados sobre a experiência dos desenvolvedores em Dispositivos Móveis e Plataforma

A Figura 35 atesta a experiência dos participantes no desenvolvimento de sistemas móveis, seguindo os resultados dos gráficos anteriores quanto aos níveis de experiência. Este gráfico também afirma que os participantes possuem conhecimento exclusivo na plataforma Android. Ademais, eles descreveram nas respostas coletadas que o ambiente de desenvolvimento preferencial para esta plataforma foi o ADT, já retratado na seção 5.4.5.

Todos os participantes citaram experiência em outros arcabouços, APIs e ferramentas para a plataforma Android, por exemplo, TheMovieDB¹², ActionBarSherlock¹³, Weka¹⁴ e

¹² TheMovieDB: <https://www.themoviedb.org/documentation/api>.

¹³ ActionBarSherlock: <http://actionbarsherlock.com/>.

¹⁴ Weka: <http://www.cs.waikato.ac.nz/ml/weka/>.

jAudio¹⁵. Eles usaram essas ferramentas para o desenvolvimento de projetos particulares e foram importantes para adquirir conhecimentos e superar desafios encontrados por quem trabalha nesta área, como a importação dos arcabouços e bibliotecas em seus sistemas móveis. Esses sistemas são dedicados aos mais diferentes propósitos, desde sistemas de busca em banco de dados até sistemas de processamento de sinais de áudio.

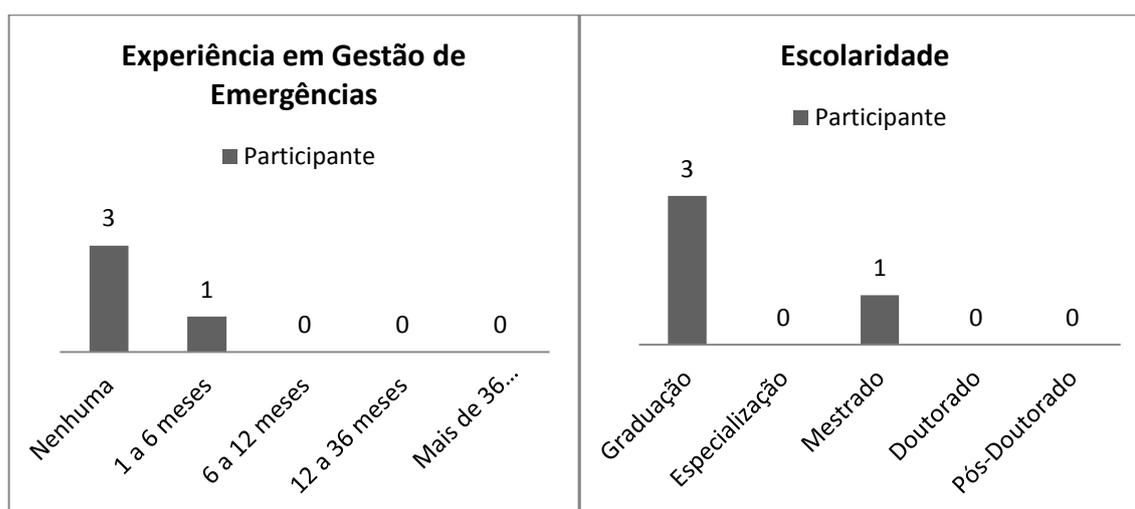


Figura 36: Resultados sobre a experiência dos desenvolvedores em Gestão de Emergências e sua escolaridade

Contudo, nota-se na Figura 36 que apenas um participante possui experiência no domínio de Gestão de Emergências, ou seja, antes da realização desta avaliação os demais desconheciam os conceitos, objetivos e desafios desse tema. Ainda neste gráfico, percebe-se que o maior número dos desenvolvedores são graduandos ou graduados, e com o destaque para o desenvolvedor com maior experiência que é mestrando ou mestre em Computação.

A Figura 37 apresenta a compilação dos resultados dos exercícios de avaliação do arcabouço (6.1.2). O Exercício A lida com a criação do novo sistema móvel que foi determinado para realizar o uso prático do arcabouço, onde todos os participantes concluíram que a atividade é simples. No Exercício B, os participantes fizeram a importação do JEMF para esse novo sistema móvel, e metade dos participantes entendeu que é uma atividade fácil e a outra metade deles considerou muito fácil.

¹⁵ jAudio: <http://jaudio.sourceforge.net/>.

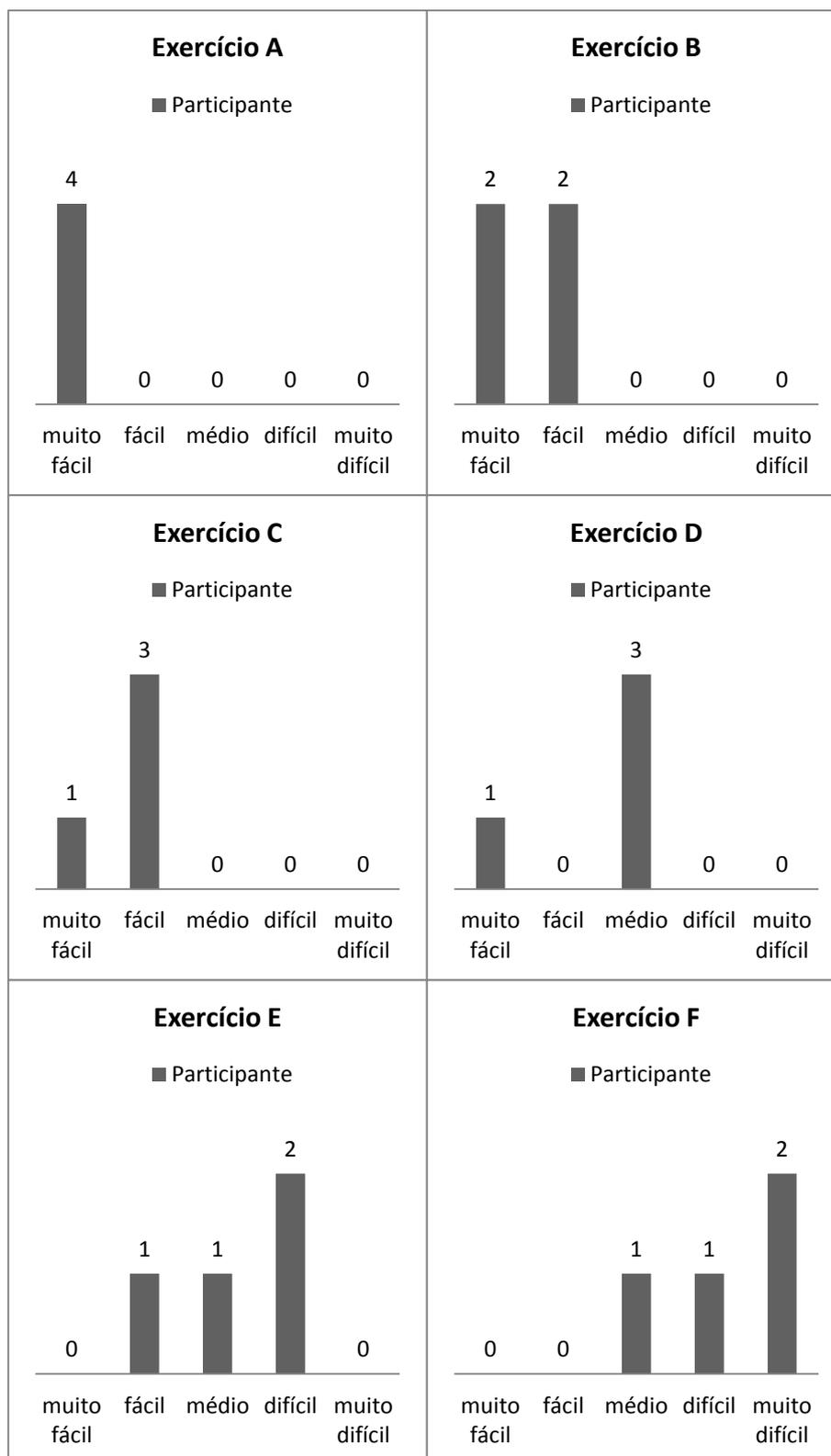


Figura 37: Resultados sobre a execução dos exercícios práticos do JEMF pelos desenvolvedores

O Exercício C visa implementar uma das funcionalidades especificadas para o novo sistema, e a maioria (3 entre 4) assinalou que é uma atividade fácil e apenas um deles (1 entre

4) disse que foi muito fácil. No Exercício D, os participantes implementaram as principais operações da funcionalidade proposta (incluir, alterar e excluir dados), e a maior parte deles (3 entre 4) achou que a atividade possui uma dificuldade mediana, contra a menor parte deles (1 entre 4) que confirmou ser uma atividade muito fácil. No Exercício E, os participantes implementaram alterações nas propriedades da funcionalidade proposta do exercício anterior e houve maior distribuição dos resultados, sendo que os participantes entenderam que essa atividade foi: uma parte fácil (1 entre 4), uma parte intermediária (1 entre 4) e a maior parte difícil (2 entre 4). Todavia o Exercício F foi considerado o mais desafiador, pois tem por objetivo criar uma nova funcionalidade para o sistema móvel com o apoio do arcabouço. Apenas uma minoria dos participantes (1 entre 4) indicou que a atividade foi mediana, em oposição aos que disseram ser difícil (1 entre 4) e muito difícil (2 entre 4).

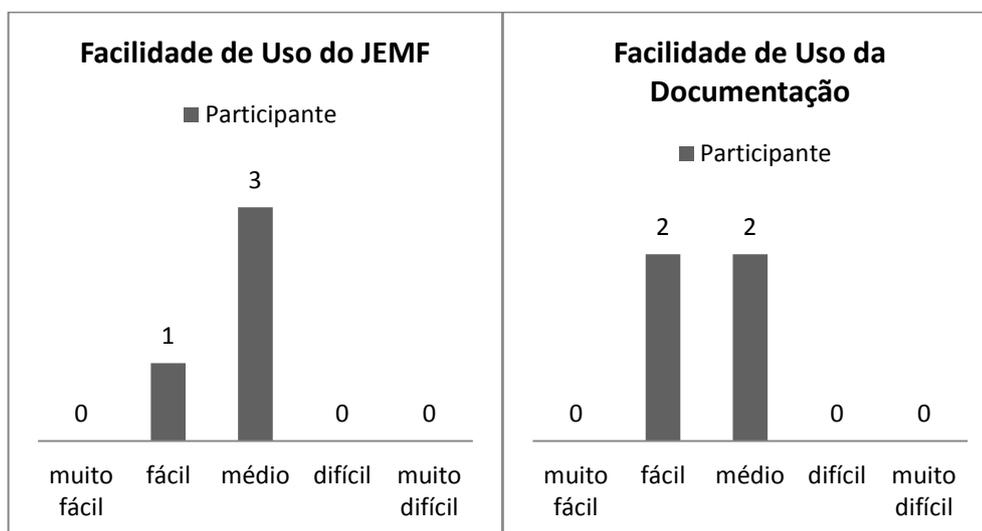


Figura 38: Resultados sobre a facilidade de uso do JEMF e sua documentação

A Figura 38 mostra que a maior parte dos desenvolvedores indicou que a facilidade de uso do arcabouço, de maneira geral, foi intermediária. Já a facilidade de uso da documentação ficou equilibrada entre mediana e fácil.

Na Figura 39 fica constatado que a facilidade no desenvolvimento de um novo sistema móvel para o domínio de Gestão de Emergências variou entre fácil e difícil, com maior incidência para uma facilidade intermediária.

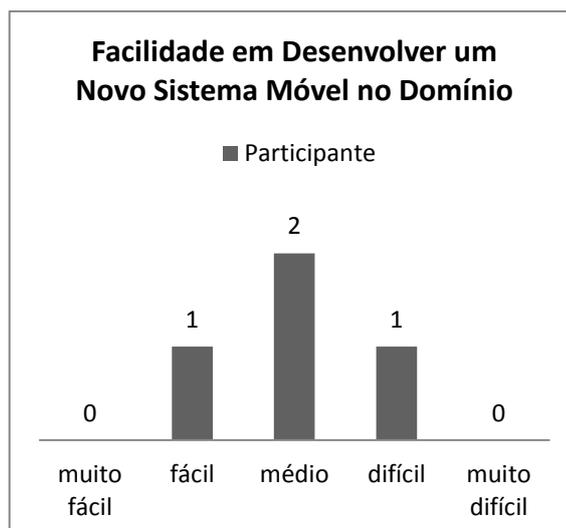


Figura 39: Resultados sobre a facilidade no desenvolvimento de sistema móvel para a gestão de emergências

As dificuldades relatadas se limitaram a encontrar suporte na documentação quando foi necessário incluir novos atributos e funcionalidades no sistema, bem como nas instruções relacionadas com o banco de dados. Houve também uma dificuldade na instalação do plug-in Subversive no ambiente de desenvolvimento (previamente citado na Seção 5.4.5). Dessa forma, os participantes sugeriram algumas melhorias quanto a esses assuntos na documentação do arcabouço.

Além do mais, uma sugestão descrita pelo participante mais experiente foi com relação às classes do tipo *View* (Seção 5.3.2.1.2) que poderiam ser definidas em arquivos XML, conforme o padrão estabelecido pela plataforma Android. Entretanto, durante a fase de implementação do arcabouço, essa opção não foi adotada devido à forma como o Android trata de arquivos XML importados de outros projetos, bibliotecas ou arcabouços, incluindo demasiadamente referências desses arquivos e podendo então ocorrer conflitos na identificação de cada componente implementado para a camada de visão de um novo sistema móvel.

Por fim, os participantes declaram que não possuem experiência suficiente no domínio de Gestão de Emergências para identificar se alguma funcionalidade vital no tema não foi atendida pelo JEMF. Mas, eles afirmam que o arcabouço é completo para o que ele se propõe a atender pelo conjunto de funcionalidades dispostas para a criação de sistemas móveis para o domínio.

6.2 Avaliação por Especialistas do Domínio

Esta avaliação se consiste na análise crítica geral do arcabouço por um grupo de especialistas no domínio de Gestão de Emergências. Cada especialista recebeu um questionário com perguntas para contribuir com suas observações ao realizarem uma revisão do artigo deste projeto: “*JEMF: A Framework for the Development of Mobile Systems for Emergency Management*” (MACHADO et al., 2014). No Apêndice E está detalhado o questionário definido para esta avaliação.

6.2.1 Avaliação Crítica do Arcabouço

Mediante o questionário de avaliação crítica por especialistas, espera-se obter o parecer do entrevistado após a leitura do artigo ao questionamento dos itens:

- Quanto à abordagem proposta para o apoio no desenvolvimento de sistemas móveis no domínio de gestão de emergências;
- Quanto à aplicabilidade, utilidade e adaptabilidade do arcabouço proposto em novos projetos de resposta a emergências;
- Quanto aos benefícios e deficiências encontradas no arcabouço proposto, e as dúvidas e sugestões para este projeto;
- Quanto a sua escolaridade, sua experiência em estudos ou trabalhos no domínio, e seu envolvimento em projetos para sistemas colaborativos na gestão de emergências.

6.2.2 Resultados Coletados dos Especialistas

Nesta seção são examinados os resultados coletados na avaliação feita por especialistas do domínio, sendo aplicada para um conjunto total de três participantes. Segundo os resultados obtidos pelo questionário, eles possuem opiniões distintas sobre a abordagem estabelecida neste trabalho. Nos próximos gráficos são apresentadas detalhadamente as conclusões dos especialistas participantes.

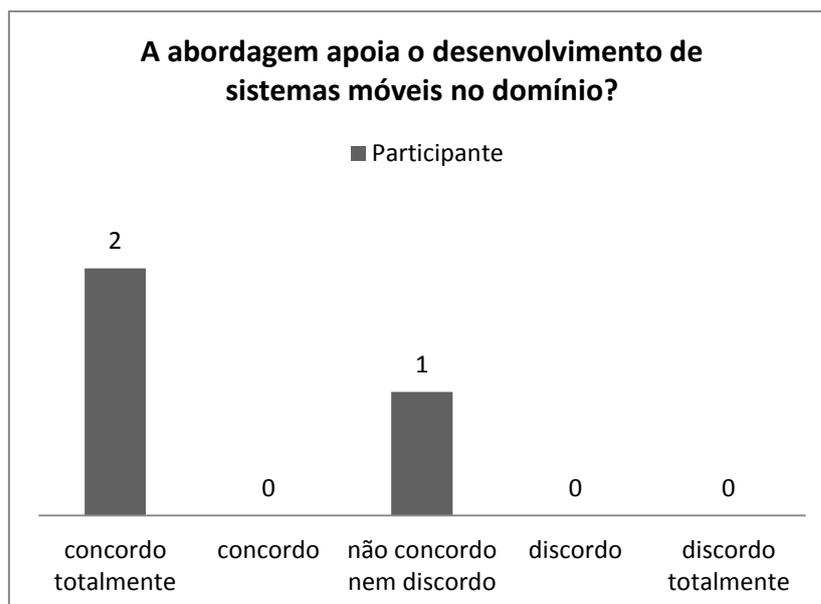


Figura 40: Resultados sobre o apoio da abordagem para o desenvolvimento de sistemas móveis no domínio

A Figura 40 mostra que a maioria dos especialistas concorda totalmente com a abordagem proposta para suportar o desenvolvimento de sistemas móveis no domínio de Gestão de Emergências, enquanto a minoria deles não concorda nem discorda.

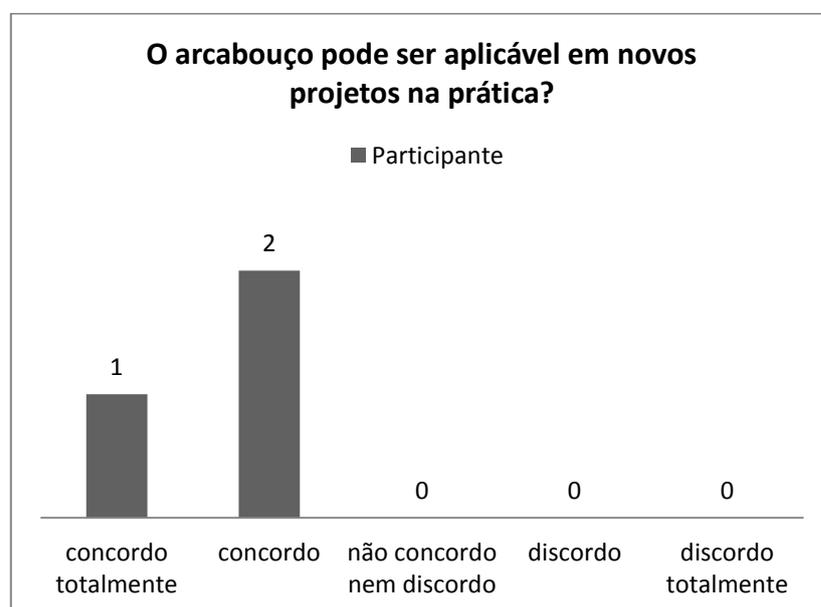


Figura 41: Resultados sobre a aplicabilidade do JEMF em novos projetos

Na Figura 41 é revelado que os especialistas concordam ou concordam totalmente que o arcabouço pode ser aplicável na prática em novos projetos de sistemas no domínio.

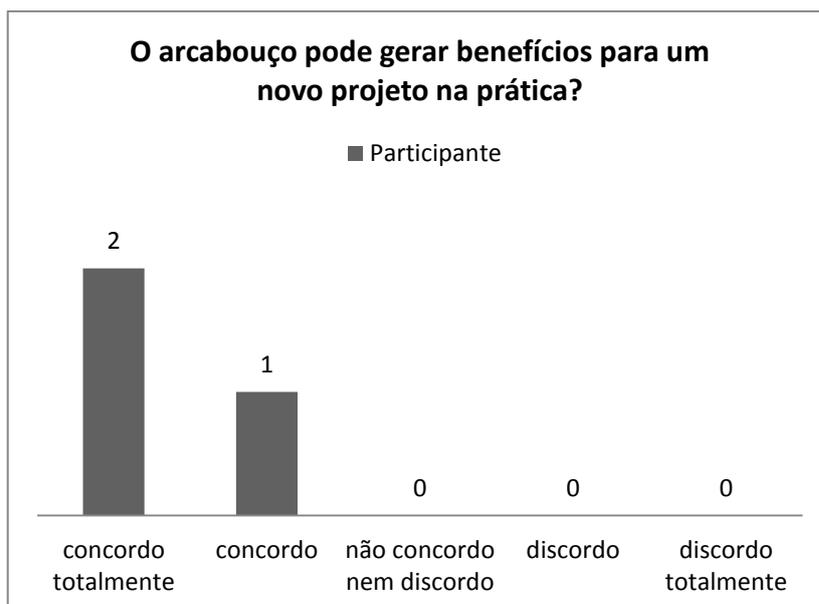


Figura 42: Resultados sobre a geração de benefícios pelo JEMF em novos projetos

A Figura 42 destaca que a maior parte dos especialistas concorda totalmente que o arcabouço pode gerar benefícios para um novo projeto de sistema no domínio, e a menor parte deles concorda que o arcabouço pode ser útil.

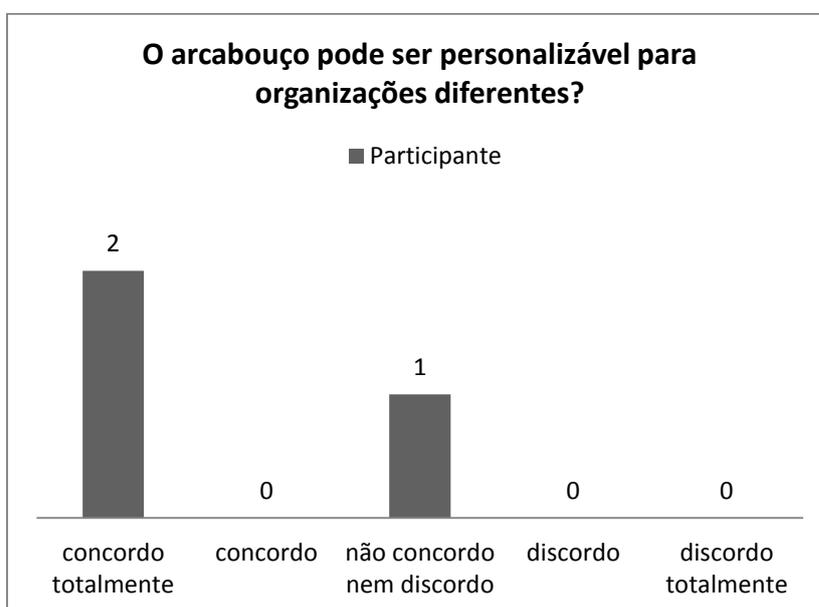


Figura 43: Resultados sobre a personalização do JEMF em organizações de resposta a emergências

A Figura 43 assinala que a maioria dos especialistas concorda totalmente que o arcabouço pode ser personalizável para atender as diversas organizações de respostas a emergências, como Corpo de Bombeiros, Polícia Militar e outras. Os demais especialistas não concordam nem discordam nesse quesito.

No que concerne às deficiências descritas pelos especialistas, foi observada a ausência de mecanismos que garantam a autonomia da utilização de sistemas móveis em situações onde não há conectividade com a Internet, sendo isto uma condição essencial para plataformas que operam sobre um cenário de emergência.

Outra questão observada pelos especialistas foi com relação ao acesso da literatura para analisar os requisitos dos sistemas de informação que serviram como base na fase de análise e projeto deste trabalho. Eles contestam que os requisitos desses sistemas podem ser de difícil obtenção em sua totalidade. Contudo, essa dificuldade foi superada com uma busca exaustiva dos artigos e teses que abordaram cada um dos projetos descritos no Capítulo 3, além de ser revisado criteriosamente sob um conjunto de funcionalidades previstas para esses sistemas do domínio, conforme ressaltado na Seção 3.13.1 e no Apêndice A dessa dissertação.

Por outro lado, os especialistas afirmaram que a abordagem proposta é válida, pois tem por objetivo agilizar o processo de desenvolvimento e evitar o consumo de tempo em atividades que podem gerar retrabalho a cada novo projeto no domínio. Logo, a criação do arcabouço traz uma simplificação desse processo por meio da reutilização e aumenta a qualidade de software, que é pertinente ao tema.

Os especialistas sugeriram que a documentação deve ser detalhada e acessível aos desenvolvedores, diminuindo eventuais dificuldades no uso do arcabouço. Outra sugestão dada se refere à inclusão de uma funcionalidade que permita a visualização do estado geral da emergência ao abranger as demais funcionalidades, ou seja, uma sumarização das funcionalidades providas nessa primeira versão do arcabouço.

A Figura 44 apresenta o nível de escolaridade dos especialistas, sendo todos doutorandos ou doutores em Computação. Essa figura destaca também todos os especialistas possuem pelo menos um ano de experiência em trabalhos e/ou estudos na Gestão de Emergências. Mais especificamente, um dos participantes atua como pesquisador na área de visualização de informações e técnicas de interação humano-computador para apoiar a colaboração em situações de crise. Outro especialista atua na pesquisa de avaliação de planos de emergência com o uso de testes de software e na adaptação de planos de forma dinâmica e automática. O último especialista trabalha em atividades de base e campo há mais de 10 anos através do Centro de Apoio Científico em Desastres – Universidade Federal do Paraná, colaborando em dezenas de missões de resposta a emergências no Brasil e na América Latina. Esse especialista já atuou no provimento e manutenção de sistemas de informação em parceria

com o Corpo de Bombeiros do Estado do Rio de Janeiro e na elaboração de Planos Diretores Municipais para o mapeamento de áreas de riscos.

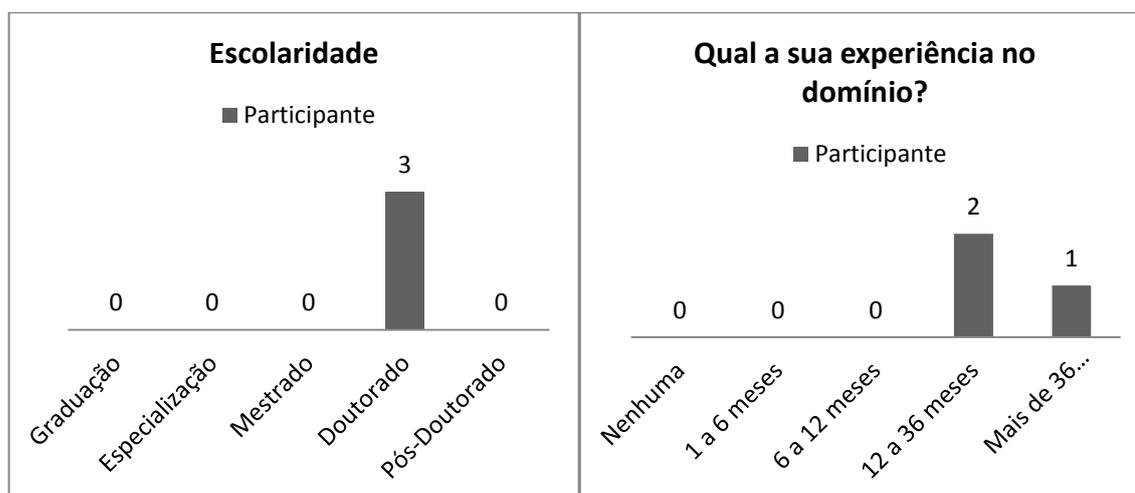


Figura 44: Resultados sobre a escolaridade dos especialistas e sua experiência no domínio

6.3 Conclusões da Avaliação do Arcabouço

As avaliações apresentadas nesse capítulo foram importantes para a realização de análises críticas e validação da abordagem proposta neste trabalho por desenvolvedores terceiros e especialistas experientes no domínio. Isso não somente permitiu constatar as perspectivas de uso e as contribuições adquiridas pelo arcabouço, mas também as correções e melhorias sugeridas que visam colaborar com a manutenção deste projeto.

Notam-se nos resultados verificados entre o Exercício D e o Exercício F uma diferença clara de experiências entre os desenvolvedores. Os desenvolvedores menos experientes passaram a relatar maior dificuldade na execução dos exercícios práticos à medida que foi exigido um maior grau de interação no uso do JEMF. Assim, pode-se conferir que os aspectos examinados de flexibilidade e extensibilidade no uso do arcabouço estão atrelados com o conhecimento e experiências prévias adquiridas com o tempo em programação orientada a objetos, na linguagem de programação Java e na plataforma Android. Conseqüentemente, o desenvolvedor com maior experiência relatou menor dificuldade sobre os mesmos aspectos. Já o aspecto de reusabilidade demonstrou-se o mais satisfatório entre os desenvolvedores, pois eles desempenharam melhor os exercícios definidos para a avaliação desse aspecto (entre o Exercício A e C). Uma visão geral da facilidade de uso do arcabouço pode ser considerada pelo resultado apresentado na Figura 38.

Apesar dos participantes não terem conhecimentos prévios no domínio para realização dos exercícios propostos, ficou constatado que de maneira geral os desafios encontrados no desenvolvimento de sistemas com o apoio do arcabouço foram considerados medianos,

conforme visto na Figura 39. Portanto, mesmo desenvolvedores com pouca experiência nos temas discutidos são capazes de usar tal ferramental para se beneficiar da reutilização de software e apoiá-los no desenvolvimento de novos sistemas móveis para o domínio.

Os especialistas defendem que deve ser investida uma maior experimentação do arcabouço ao elaborar um caso de uso onde um sistema móvel criado com apoio do arcabouço seja usado por equipes de operação em situações reais de resposta a emergências. Com isso, podem-se validar outros aspectos inerentes ao sistema móvel, por exemplo, a usabilidade e o desempenho, em função dos componentes e funcionalidades fornecidas pelo arcabouço. Um dos especialistas destaca também que seja investido um esforço na expansão do arcabouço para apoiar a conectividade entre essas equipes, uma vez que o escopo atualmente estabelecido não trata dos desafios de autonomia e de rede para o caso da ausência de comunicação.

Enfim, a avaliação proposta para este trabalho prevê a validação dos principais aspectos almejados no reuso de software, respeitando o escopo de atuação definido para o arcabouço.

7 Trabalhos Relacionados

Junto com a grande expansão no uso de sistemas de informação no domínio de emergências, a comunidade já tem trabalhado no sentido de enfrentar os desafios de reutilização de software com ferramentas e arcabouços. Por este motivo, neste capítulo é descrito alguns dos principais trabalhos anteriores relacionados ao tema.

7.1 ESCAPE

ESCAPE é um arcabouço de *middleware* sensível ao contexto para a gestão de emergências (TRUONG et al., 2007). Dependendo do contexto de cenários específicos, as tarefas realizadas durante as situações de emergência podem ser alteradas e os serviços de resposta devem ser capazes de usar informações de contexto para ajustar a colaboração entre os envolvidos. As informações de contexto podem afetar substancialmente nas situações de resposta, pois elas estão relacionadas com as entidades em cada local da emergência, como as equipes de resposta que detêm informações sobre tarefas de cada membro, o status de dispositivos, redes, etc., mas também as entidades afetadas pela emergência, tais como vítimas e infraestruturas. Especificamente, este arcabouço apoia os seguintes níveis de conhecimento: individual, equipe, local, resposta e situação. Esses níveis de conhecimento fornecem uma estrutura detalhada de informações de contexto para serem utilizadas em diferentes propósitos.

Truong e outros (2007) apresentaram o projeto e a implementação de serviços de gestão de contexto dentro do ESCAPE, que pode suportar várias equipes trabalhando em diferentes locais e dentro de situações de respostas distintas, para capturar e compartilhar vários tipos de informações de contexto. O ESCAPE é capaz de gerenciar informações relevantes, que é descrito por esquemas XML e fornece as informações de contexto para todos os dispositivos clientes. As equipes são equipadas com diversos tipos de dispositivos e capacidades diferentes, tais como PDAs e laptops, que fazem a coleta das informações. Técnicas e modelos baseados em Arquiteturas Orientadas a Serviços (SOA) foram empregados na gestão e disseminação de informações de contexto, ou seja, *Web Services* são usados para publicação e recuperação dessas informações.

Assim, este arcabouço de gerenciamento de contexto deve ser flexível ou ser facilmente adaptável para lidar com informações especificadas por diferentes modelos em

vários níveis de abstração. Um dos principais motivos para o desenvolvimento deste arcabouço é o suporte na comunicação distribuída para o projeto WORKPAD, previamente apresentado no Capítulo 3.

7.2 CIMS

A comunidade de GE sempre trabalhou para a definição dos requisitos essenciais que as tecnologias de informação e comunicação devem apoiar na resposta e recuperação de emergências. Uma destas necessidades é a troca de informações entre múltiplas agências e órgãos de emergência para uma ação coordenada e o compartilhamento da capacidade de resposta em grandes crises.

Sistemas de Gestão de Informação de Crise, em inglês *Crisis Information Management Systems* (CIMS), tem por objetivo fornecer um conjunto completo de funções que abordam as exigências da comunidade de GE (IANNELLA; ROBINSON; RINTA-KOSKI, 2007). Um arcabouço para esse tipo de sistema foi desenvolvido e, com isso, permitiu-se capturar e classificar as funções e serviços que estabelecem uma interoperabilidade nas arquiteturas desses sistemas de informação.

O conjunto de funcionalidades de um CIMS poderá variar bastante, refletindo as necessidades das equipes de resposta que usam esse sistema e o seu nível de especialização. Este conjunto pode ser dividido em três camadas. A primeira camada incluem funções de controle direto das emergências, como o suporte para o gerenciamento de incidentes, de recursos, de notificações e também da consciência situacional. A segunda camada compreende as funções de suporte para os serviços da primeira camada, incluindo o gerenciamento de documentos, de recursos financeiros e de relatórios. Na terceira camada são encontradas as funções de núcleo, como serviços de autenticação, de autorização e geoespaciais.

Dessa forma, um arcabouço de CIMS suporta os processos utilizados nos centros de coordenação de crise que concentram diferentes órgãos e jurisdições, apoiando fundamentalmente uma estrutura hierárquica de comando e controle.

7.3 HLMP API

HLMP API é uma interface de programação que implementa o protocolo de roteamento denominado High Level MANET Protocol (HLMP), e vários mecanismos de troca de mensagens e consciência situacional que são normalmente exigidos pelos sistemas móveis colaborativos (RODRÍGUEZ-COVILI et al., 2011). A HLMP API foi projetada

especificamente para apoiar o trabalho colaborativo móvel usando redes móveis Ad Hoc, em inglês *Mobile Ad Hoc Network* (MANET), e facilitar o desenvolvimento de um conjunto de serviços e funcionalidades de comunicação. As infraestruturas de comunicação que apoiam esse tipo de trabalho devem fornecer serviços adequados para permitir que os usuários interajam num ambiente distribuído.

Criar serviços de apoio à comunicação em cenários de trabalho colaborativo móveis pode se tornar um grande desafio para os desenvolvedores. A necessidade de implementar uma infraestrutura de comunicação como parte do sistema móvel conduz os desenvolvedores a concentrar esforços em questões básicas de projetos de *groupware*, tais como o encaminhamento e a entrega de mensagens, identificação de nós numa rede, etc. Esses serviços de comunicação podem ajudar a garantir que a coordenação e a colaboração sejam mantidas durante atividades de grupo.

A HLMP API é composta por dois componentes principais: um núcleo e os plug-ins. O núcleo implementa os mecanismos para suportar os procedimentos de interoperabilidade do processo de comunicação, de compartilhamento de dados e funcionalidades delegadas para os sistemas operacionais. Por outro lado, o componente de plug-ins contém especificações para os protocolos de comunicação de *groupware* e os mecanismos de consciência situacional, utilizando os serviços prestados pelo núcleo.

Uma contribuição importante da HLMP API é a qualidade dos serviços de comunicação que ela fornece. A fim de demonstrar esta característica, o desempenho da API foi comparado com outros protocolos de roteamento reconhecidos para redes MANET. Os resultados obtidos mostram que a HLMP API é capaz de fornecer um suporte de comunicação adequado entre dispositivos móveis (computadores portáteis ou PDAs), independentemente do tipo de mobilidade que os usuários adotem. Os autores adotaram uma estratégia para apoiar a reutilização dos serviços de comunicação já implementados por meio dessa API, permitindo aos desenvolvedores concentrar em necessidades secundárias, pois as questões importantes de *groupware* já foram solucionadas.

7.4 Sahana

Sistemas de informação que proporcionam o acesso oportuno a informações relevantes são fundamentais nas operações de resposta a emergências. Quanto mais rápido uma equipe de operação é capaz de coletar, analisar, divulgar e atuar sobre informações confiáveis da emergência, mais eficaz será a resposta, as necessidades serão atendidas e maior será o benefício para as populações afetadas (CURRION; SILVA; VAN DE WALLE, 2007). As

organizações de resgate podem usar uma variedade de ferramentas e recursos tecnológicos, mas grande parte delas precisa lidar com limitações pela capacidade de recursos humanos, técnicos e financeiros para desenvolver e gerenciar essas ferramentas, criando lacunas no processo de resposta.

Segundo Currión, Silva e Van De Walle (2007), muito poucos países e organizações comprometem recursos suficientes para a GE e os investimentos na fase de mitigação permanece tímido em boa parte do mundo, uma vez que há sempre projetos de maior prioridade que precisam de financiamento pelos governos das nações. A abordagem de desenvolvimento de softwares de código aberto se alinha com os princípios proclamados pelas organizações humanitárias. Idealmente, tais sistemas devem ser compartilhados, desenvolvidos e mantidos de forma livre a nível mundial. Esses fatores permitem que um sistema de informação seja configurado e adaptado rapidamente, de modo que se incorpore à situação específica de cada país, pois o código-fonte está disponível para qualquer organização.

O Sistema de Gestão de Desastres Sahana foi desenvolvido com o propósito de oferecer um conjunto de componentes que foram projetados para resolver os principais desafios de coordenação de resgate e conta com uma variedade de serviços de informação disponibilizados via Web. Os serviços importantes são: registro de organizações, vítimas de desastres e acampamentos médicos; gestão de voluntários e requisições de socorro; módulo de mensagens; módulo de visão operacional e outros. A equipe de desenvolvimento do sistema trabalha para apoiar as exigências e necessidades das organizações de resposta, em parceria com o governo e organizações não governamentais, agregando experiências e conhecimentos no domínio para a concepção e desenvolvimento desses componentes.

7.5 Discussão

Os autores do arcabouço ESCAPE afirmam que a maioria dos *middleware* de gerenciamento de informações de contexto existentes são direcionados para ambientes pequenos e fechados. Portanto, eles identificaram a falta de um *middleware* para situações de emergência, uma vez que o ambiente em que as tarefas realizadas de apoio para situações de emergência são altamente dinâmicas e não estruturadas. Segundo Truong e outros (2007), durante a realização de teste foi averiguado que a comunicação baseada em TCP pode ser instável quando o volume de dados transferidos aumenta. Nesse caso, quando o número de membros da equipe de resgate é grande e há a necessidade de transmitir uma quantidade de dados superior, os PDAs usados não suportam o processamento apropriado das informações

nesse tipo de cenário. Com isso, outro problema identificado está relacionado aos aspectos da qualidade da informação de contexto, como incompletude, duplicação e inconsistência. Uma vez que as informações são coletadas através de dispositivos móveis e esses não possuem recursos suficientes, as informações produzidas pelos sensores podem proporcionar diferentes fragmentos de dados em tempos diferentes e os métodos de agregação dessas informações são prejudicados, apesar de seguirem o mesmo modelo de informação de contexto. Ademais, o ESCAPE é um arcabouço que envolve a implementação de *Web Services* via SOAP para o projeto WORKPAD.

Iannella, Robinson e Rinta-Koski (2007) abordam alguns requisitos para CIMS e como as tecnologias devem atender a esses requisitos, defendendo que é preciso o desenvolvimento de ferramentas, estruturas e terminologias para enfrentar as situações de crise. A GE é tipicamente uma atividade dinâmica que envolve equipes de pessoas distribuídas de uma variedade de organizações. Isso demandou que normas e funcionalidades comuns fossem projetadas para que o arcabouço proposto apoie a distribuição de informações, tanto entre diferentes implementações de CIMS, bem como entre CIMS e outros tipos de software. O uso de padrões abertos de informação colabora para uma terminologia comum e a interoperabilidade entre as organizações. Mas, a GE é uma disciplina que não segue regras bem definidas, nem todas as hipóteses podem ser modeladas e atendidas devido sua complexidade. Além disso, foi apresentada uma instância do arcabouço que aborda a implementação de um sistema para a notificação de incidentes através de dispositivos com recursos limitados, tais como PDAs e telefones celulares, e baseado em arquitetura SOA. Assim, foi apresentado um arcabouço que cobre as funcionalidades e serviços de alto nível prestados por CIMS. Entretanto, o foco deste arcabouço está relacionado com arquiteturas de interoperabilidade, ou seja, não aborda detalhadamente sobre as especificidades dos recursos providos pela computação móvel na GE.

A HLMP API (RODRÍGUEZ-COVILI et al., 2011) é uma solução típica para lidar com os desafios de comunicação em ambientes de trabalho colaborativo móveis. Essa arquitetura de software fornece um conjunto de serviços já implementados, que são expostos através de uma interface aos desenvolvedores, sem ter que lidar com detalhes de programação de baixo nível. A camada superior dessa API é capaz de criar funcionalidades de trabalho colaborativo usando o paradigma de troca de mensagem de sistemas distribuídos. Esta API pode ser facilmente ajustada para atender requisitos específicos de comunicação para apoiar um trabalho colaborativo. Os resultados de testes experimentais demonstraram que a API

funciona adequadamente em cenários com grande mobilidade dos usuários, o que favorece um desempenho aceitável durante qualquer colaboração. No entanto, o gerenciamento de informações de rede adiciona uma sobrecarga extra de comunicação quando ocorre uma transferência de arquivos. Os autores destacam que a API não foi projetado para suportar um grande número de nós para uma rede móvel que exercem atividades de colaboração, mas para apoiar o trabalho de baixo acoplamento entre uma pequena quantidade de participantes. Isso é um fator que precisa ser analisado no domínio de emergências, pois em uma situação de resposta o total de agentes pode chegar a algumas dezenas de pessoas, tornando inviável a comunicação entre elas. Apesar disso, vários sistemas foram desenvolvidos com a utilização da HLMP API como é o caso do MobileMap, relatado no Capítulo 3.

Sahana (CURRION; SILVA; VAN DE WALLE, 2007) não é um arcabouço ou um *middleware*, mas sua principal característica é ser um sistema de código aberto. Isso permite que outras nações reutilizem e adaptem este sistema para atender as especificidades de sua localidade e apoiar organizações de resgate que não possuem investimentos suficientes oferecidas pelo seu governo. Embora haja um custo inicial na criação de um sistema de gestão de emergências, ele não tem nenhum custo para ser adquirido e o custo de manutenção é considerado baixo. Deste modo, trata do problema de falta de financiamento ou de quaisquer restrições administrativas que exigem um processo de aquisição, geralmente longo, de uma licença de software de alto custo que poderiam atender ao processo de resposta. Ele é compartilhado de forma gratuita com o objetivo de criar oportunidades para que organizações de emergências possam construir e manter esse tipo de ferramental, poupando vidas em países que não têm mão de obra qualificada ou conhecimentos necessários para desenvolver sistemas de colaboração e tomada de decisão. Ademais, Sahana é um sistema Web e depende de uma rede confiável para compartilhar as informações de emergência.

O arcabouço aqui proposto diferencia-se dos trabalhos já citados, pois atuará como suporte na construção de sistemas por meio de um arcabouço de software para a computação móvel, abordando um escopo diferente dos outros projetos que se basearam em arquiteturas de software Web. Em vista disso, a maior parte dos trabalhos se empenhou em resolver problemas heterogêneos de rede e desafios particulares de comunicação. Eles também preconizam que os sistemas móveis podem apenas consumir serviços Web, tornando sua autonomia pequena ao exigir um prestador de serviços e um desempenho adequado de redes de comunicação, ou seja, proporciona uma arquitetura centralizada. Isso não é o mais adequado para um ambiente de resposta devido à disponibilidade de redes de comunicação

que podem ficar inacessíveis, dependendo da intensidade de danos que uma emergência pode provocar em determinado local. Ou seja, o sistema móvel está apenas disponível aos usuários quando o serviço Web está funcionando corretamente. Neste trabalho o objetivo é produzir sistemas nativos para a plataforma móvel e que não sejam dependentes unicamente de serviços Web.

Outra diferença dos trabalhos já citados é que projetos futuros podem ajudar a expandir esta estrutura de software, onde novas especificações genéricas podem ser compartilhadas dentro da comunidade de pesquisa do domínio de Gestão de Emergências. Portanto, o JEMF colabora com o desenvolvimento de novos sistemas e qualquer alteração de projeto é possível devido uma característica importante, o seu código aberto. Além disso, o objetivo do JEMF é atuar na construção de sistemas para dispositivos móveis atuais, como smartphones e tablets.

8 Conclusão

Os dispositivos móveis se transformaram em um importante instrumento para a realização de tarefas no dia-a-dia das pessoas. Estes dispositivos proporcionaram o surgimento de sistemas que se aprimoram pela combinação da mobilidade com os mais variados tipos de recursos. Isso cria oportunidades de consulta e atualização de informações a qualquer momento e em quase todos os lugares. O uso destes dispositivos na Gestão de Emergências não é diferente. Eles estão se tornando cada vez mais comuns ao suportar simultaneamente diferentes tipos de usuários e organizações que atuam nesse domínio. Essa área de pesquisa aliada com a Engenharia de Software tem concentrado estímulos para explorar as capacidades desses dispositivos por meio do desenvolvimento de novos sistemas móveis. O aumento do número de sistemas nesse contexto traz ênfase para as metodologias e técnicas que apoiam a reutilização de software, assegurando benefícios tanto para o desenvolvedor quanto para o produto final.

A literatura reporta diversas técnicas que possibilitam uma significativa redução dos esforços de desenvolvimento ao permitir que artefatos do processo de criação de um sistema (requisitos, estrutura lógica, código, etc.) sejam reaproveitados, sem que haja perda da qualidade e a elevação dos custos. Dentre as técnicas mais reconhecidas para tal propósito está a adoção de arcabouços orientados a objetos. Através dos arcabouços podem-se obter diferentes sistemas sem a necessidade do retrabalho na análise, projeto e codificação a cada novo software, ou seja, um arcabouço é um projeto parcial e implementação de um determinado domínio a partir do qual os desenvolvedores produzem sistemas para um problema específico. Assim, o arcabouço visa reunir um conjunto de classes cooperantes aliadas ao emprego de padrões de projetos, sendo responsável pelas funcionalidades genéricas relacionadas a esse domínio.

Esta dissertação envolveu o desenvolvimento de um arcabouço para sistemas móveis de resposta a emergências. Inicialmente, o trabalho apresentou os conceitos, as fases e os desafios encontrados no domínio de Gestão de Emergências (Capítulo 2). A partir das soluções de software relacionadas ao tema, foi feito um levantamento e análise de características comuns dos sistemas móveis que favorecem a colaboração entre as equipes durante as situações de emergências, de modo a cobrir todas as perspectivas de uso desses sistemas móveis no domínio (Capítulo 3). Em seguida, também foram relacionados às

principais técnicas de reuso de software para o conhecimento básico sobre os termos utilizados neste trabalho (Capítulo 4).

Ao considerar as características identificadas, foi construído o arcabouço com base numa metodologia de desenvolvimento de software que concilia a orientação a objeto e a UML. A primeira etapa envolveu a definição dos requisitos e descrição de todos os artefatos que compõem as fases de análise e projeto do arcabouço. Outra etapa da metodologia é a implementação do conjunto de classes que formam o arcabouço proposto. A última etapa dedicou à instanciação e testes, demonstrando que o arcabouço alcançou o propósito previsto para o desenvolvimento de sistemas móveis (Capítulo 5). Após, foi abordada uma avaliação qualitativa para desenvolvedores de software e especialistas do domínio com o intuito de validar a proposta deste trabalho (Capítulo 6). No final, foram discutidos alguns trabalhos relacionados ao tema (Capítulo 7).

Diante do exposto, ao analisar os resultados conquistados pelo JEMF é notável uma redução das complexidades técnicas para o desenvolvimento de sistemas no domínio de Gestão de Emergências, aprimorando esse processo para garantir um menor esforço e maior qualidade de software. Em conclusão, este arcabouço de software contribui para a construção de novos sistemas móveis por meio da reutilização de software.

8.1 Contribuições

Este trabalho aponta que a abordagem de desenvolvimento com arcabouços é muito promissora no que se refere à concepção de sistemas móveis para o gerenciamento de informações nas operações de emergência. Nesta pesquisa foram definidos alguns princípios essenciais e uma solução apropriada para remover a complexidade de construção deles. Assim sendo, a pesquisa produziu um conjunto de resultados que engloba os objetivos estabelecidos nesta proposta. A seguir são enfatizadas as contribuições concretas deste trabalho:

- Um arcabouço para o desenvolvimento de software no domínio de Gestão de Emergências. Esta estrutura se caracteriza pela reusabilidade de requisitos, projeto e implementação de funcionalidades comuns para a criação de softwares com o objetivo de serem usados durante situações de respostas às emergências;
- A aplicação de padrões de projetos para o reuso de experiência de projeto do arcabouço, levando a obtenção de estruturas de classes bem organizadas e mais adequadas na manutenção e expansão das funcionalidades providas;

- A elaboração de um manual para auxiliar o uso do arcabouço por desenvolvedores terceiros, baseando-se nas principais características do ambiente de desenvolvimento, dos conceitos do domínio e na instanciação de novos sistemas móveis abordadas neste estudo;
- A distribuição em conjunto do código-fonte do arcabouço mais outros três subprojetos: um projeto de teste do arcabouço e dois projetos de exemplo de utilização do arcabouço para ajudar no aprendizado dos desenvolvedores terceiros. Todo o código-fonte é de livre distribuição, uma vez que é caracterizado como código aberto;
- A obtenção de opiniões de desenvolvedores e especialistas, levando em consideração a avaliação dos aspectos de reusabilidade, flexibilidade e extensibilidade, além de observar a facilidade de uso do arcabouço.

O arcabouço constitui um ferramental de suporte à produção de software para a plataforma Android e representa uma contribuição para o desenvolvimento de sistemas móveis nesse domínio. Este arcabouço padroniza os sistemas gerados a partir dele, podendo proporcionar um aumento na produtividade do desenvolvimento. O objetivo é permitir que ele seja empregado em cada novo software criado para dispositivos móveis modernos, por exemplo, smartphones e tablets.

8.2 Limitações

De modo a tornar a proposta factível, a principal limitação encontrada é em relação ao escopo atual do arcabouço. Para isso, foi delimitado o seu contexto de atuação no desenvolvimento de software exclusivamente para a gestão de informações pelas equipes de operação e líderes que trabalham na linha de frente durante as situações de crise. Apesar dos benefícios gerados pelos sistemas de informação destinados a esses agentes, a infraestrutura necessária para o domínio em questão exige também que sejam elaborados mecanismos que proporcionem o compartilhamento das informações através dos mais variados tipos de rede, por exemplo, redes de Internet sem fio, redes de telefonia móvel e redes MANET, que complementam a comunicação com o contexto que tem a responsabilidade da gestão e visão estratégica de todas as ações e do aparato necessário para conter um desastre, o centro de comando e controle. Portanto, o cenário atual ainda não permite a integração entre esses dois contextos já apresentados no Capítulo 5 (Seção 5.2), dada a complexidade das tecnologias envolvidas.

Outra limitação se refere à estrutura presente que não contempla a variedade de recursos e sensores que estão à disposição nos dispositivos móveis, faltando ainda o desenvolvimento de mecanismos para acessá-los e integrá-los de maneira apropriada com as funcionalidades estabelecidas para o arcabouço. Além disso, deve-se considerar a criação de interfaces flexíveis para a importação e funcionamento com outros tipos de ferramentas, bibliotecas e até outros arcabouços que possam expandir ou aprimorar, em conjunto, o desenvolvimento de sistemas para a gestão de emergências.

A versão vigente deste arcabouço está associada com a versão 4.0.3 da plataforma Android (também conhecida em inglês como *Ice Cream Sandwich*). Dessa forma, os testes especificados para o arcabouço atendem somente esse nível ou aos níveis superiores constituídos para essa plataforma móvel. Tal característica limita a construção de sistemas móveis para dispositivos que funcionam com versões inferiores, sendo necessário realizar alterações no arcabouço (estritamente no Pacote Móvel – Seção 5.3.2.1.2) para se adaptar as versões mais antigas da plataforma Android.

Por último, ainda não foi feito uma avaliação experimental do ganho de produtividade obtido com o uso deste arcabouço. Seria necessário promover experimentações envolvendo grupos de desenvolvedores distintos que seriam submetidos ao desenvolvimento de sistemas com níveis de dificuldades equivalentes, com e sem a utilização do arcabouço, seguindo as mesmas considerações de requisitos e projeto para esses novos sistemas móveis. A avaliação do tempo de desenvolvimento deve produzir uma avaliação quantitativa dos benefícios que podem ser obtidos a partir do uso do arcabouço, mas para isso seria necessário um maior período de avaliação e um registro preciso das atividades a serem executadas.

8.3 Trabalhos Futuros

A técnica de desenvolvimento de software por meio de arcabouços orientado a objetos tornou-se comum, pois é um recurso reutilizável que constitui a base para uma série de sistemas no mesmo domínio. Tal como acontece com todos os softwares, os arcabouços tendem a evoluir. Com seu uso, novas funcionalidades podem ser mais facilmente integradas nesses sistemas do que pelo método tradicional de desenvolvimento. Conforme previsto por Roberts e Johnson (1996), existe um longo caminho na evolução de um arcabouço. Uma vez que a primeira versão do arcabouço tenha sido lançada, faz-se importante ter um controle adequado das novas versões para que não causem impactos negativos nas suas instâncias ou aumente os custos dessa estrutura.

Na classificação apresentada Capítulo 3 (Seção 3.13.1) percebe-se que os atuais sistemas para a gestão de emergências não abrangem a exploração de mecanismos especiais de interação para suportar outras formas de interação avançada para os dispositivos móveis, por exemplo, interfaces multimodais, que podem auxiliar em novas perspectivas de uso com a adoção de reconhecimento de voz e gestos multitoque. Machado e outros (2013) preveem que o uso de desse tipo de interface agregará novas oportunidades que podem inclusive reutilizar ações e comandos que os agentes de resposta já estão habituados a lidar por muito tempo na comunicação via rádio e, portanto, poderá aumentar a usabilidade dos sistemas e diminuir possíveis empecilhos pelos usuários mais experientes na adoção de novas tecnologias. Então, trabalhos futuros poderão incrementar essa pesquisa ao implementar novas versões do arcabouço tanto para atender esses mecanismos citados quanto no uso em sistemas móveis para dispositivos vestíveis, como relógios de pulso, óculos, pulseiras inteligentes, entre outros.

Outra possibilidade é a experimentação em conjunto das equipes de operação em situações reais que poderá trazer novos requisitos visando complementar ou corrigir os já propostos para a composição das funcionalidades atuais do arcabouço. Com isso, será permitido averiguar outros aspectos não abordados neste estudo, como o desempenho, a segurança, e a eficiência originada pelos sistemas de informações para a gestão de emergências. Existe, portanto, um potencial que não foi explorado para avaliar as diferentes características em função do mais variado tipo de organizações e *stakeholders* envolvidos no domínio.

Finalmente, sugere-se o aperfeiçoamento do arcabouço para assistir os escopos desktop e Web, trazendo novas funcionalidades e componentes que alcancem um número maior de tipos de sistemas que possam ser desenvolvidos dentro da abordagem proposta. Isso proporcionará a ampliação do escopo deste arcabouço a um nível onde constituirá uma solução mais completa para o domínio e, assim, compor um ambiente onde será possível à reutilização de software para construção de sistemas e infraestruturas que auxiliem no trabalho e apoiem a colaboração entre os agentes de campo e o centro de comando e controle. Ademais, possibilitará sua adequação a outros domínios de maior afinidade com os conceitos associados à estrutura de Comando e Controle.

Referências

ADAIR, D. **Building object-oriented frameworks**. [S. l.]: AIXpert, 1995.

ANDROID. **Android**. 2014. Disponível em: <http://www.android.com/>. Acesso em: abr. 2014.

_____. **Android ADT Bundle**. 2014. Disponível em: <http://developer.android.com/tools/help/adt.html>. Acesso em: abr. 2014.

_____. **App Components**. 2014. Disponível em: <http://developer.android.com/guide/components/index.html>. Acesso em: abr. 2014.

_____. **Dashboards**. 2014. Disponível em: <http://developer.android.com/about/dashboards/index.html#platform>. Acesso em: abr. 2014.

_____. **Get the Android SDK**. 2014. Disponível em: <http://developer.android.com/sdk/index.html>. Acesso em: abr. 2014.

_____. **Testing Fundamentals**. 2014. Disponível em: http://developer.android.com/tools/testing/testing_android.html. Acesso em: abr. 2014.

_____. **What is API Level?**. 2014. Disponível em: ANDROID <http://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>. Acesso em: abr. 2014.

ANGERMANN, M. et al. DMT - An integrated disaster management tool. In: INTERNATIONAL CONFERENCE ON DISASTER MANAGEMENT AND HUMAN HEALTH RISK: REDUCING RISK, IMPROVING OUTCOMES, 1., 2009, New Forest. **Proceedings...** Ashurst Lodge: WIT Press, 2009.

ARAUJO, F. C. S. **Um Framework para desenvolvimento de sistemas em dispositivos móveis no apoio a gestão de emergência**. 2012. 134 f. Dissertação (Mestrado em Informática) -- Instituto de Matemática, Instituto Tércio Pacitti, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2012.

BACHMAN, F. et al. **Technical concepts of component-based software engineering**. Pittsburgh: Software Engineering Institute / Carnegie Mellon University, 2000. 2. ed. (CMU/SEI-2000-TR-008).

BARR, J. L. et al. Current domain challenges in the emergency response community. In: INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS FOR CRISIS RESPONSE AND MANAGEMENT, 8., 2011, Lisbon. **Proceedings...** [S. l.]: ISCRAM, 2011.

BIGGERSTAFF, T. J. ; RICHTER, C. Reusability framework, assessment, and directions. **IEEE Software**, Los Alamitos, v. 4, n. 2, p. 41-49, 1987.

BOOCH, G. ; RUMBAUGH, J. ; JACOBSON, I. **The Unified modeling language user guide**. Reading: Addison-Wesley, 1999.

BOSCH, J. et al. **Object-oriented frameworks-problems & experiences**. Ronneby: University of Karlskrona/Ronneby, 1997. (HKR-RES-97/9-SE).

CARROLL, J. Emergency management on a grand scale: a bureaucrat's analysis. **Handbook of crisis and emergency management**. Ali Farazmand, Boca Raton: CRC Press, v. 28, p. 463-480, 2001.

CARTER, W. N. Disaster management: a disaster manager's handbook. In: W. Nick Carter **Disaster management: a disaster manager's handbook**. Manila: ADB, 1991.

CDC. **Crisis and emergency risk communication**. Atlanta: Centers For Disease Control And Prevention, Department of Health and Human Services, 2002.

CHESHIRE, M. **What is an api? your guide to the internet business (r)evolution**. Barcelona: 3scale Networks S.L., 2011.

CURRION, P. ; SILVA, C. ; VAN DE WALLE, B. Open source software for disaster management. **Communications of the ACM**, New York, v. 50, n. 3, p. 61-65, 2007.

CUTTER, S. L. **Living with risk: the geography of technological hazards**. London: Edward Arnold, 1993.

D'SOUZA, D. ; WILLS, A. C. **Catalysis: objects, components, and frameworks with UML**. Reading: Addison-Wesley, 1998. (Object Technology Series).

DHS. **National Incident Management System Integration Center**. Washington: DHS Management Directive System, Department of Homeland Security, 2004.

_____. **National Infrastructure Protection Plan**. Washington: Department of Homeland Security, 2009.

_____. **National Response Plan**. Washington: Department of Homeland Security, 2004.

_____. **Target Capabilities List - A Companion to the National Preparedness Guidelines**. Washington: Department of Homeland Security, 2007.

DINIZ, V. B. **Uma Abordagem para definição de sistemas de gestão de conhecimento no tratamento de emergências**. 2006. 194 f. Dissertação (Mestrado em Informática) -- Instituto de Matemática, Instituto Tércio Pacitti, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.

DOD. **Dictionary of Military and Associated Terms**. Washington: Joint Doctrine Division, Department of Defense, 2010.

_____. **United States Government Interagency Domestic Terrorism Concept of Operations Plan**. Washington: Department of Defense, 2001.

DOMBROWSKY, W. R. Again and again: is a disaster what we call a disaster?. **International Journal of Mass Emergencies and Disasters**, Stillwater, v. 13, n. 3, p. 241-254, 1995.

DOTARS. **Natural disasters in Australia: reforming mitigation, relief and recovery arrangements**. Canberra: Department of Transport and Regional Services, 2002.

DOUGLASS, B. ; CERNOSEK, G. **Unified modeling language for real-time systems design**. [S. l.]: Rational Software Corporation, 1997.

DRABEK, T. E. ; HOETMER, G. J. **Emergency management: principles and practice for local government**. Washington: International City Managers Association, 1991.

ECLIPSE. **Eclipse**. 2014. Disponível em: <https://www.eclipse.org/>. Acesso em: abr. 2014.

_____. **Subversive**. 2014. Disponível em: <http://www.eclipse.org/subversive/>. Acesso em: abr. 2014.

EEA. **EEA Multilingual environmental glossary**. Copenhagen: European Environmental Agency, 2007.

ELLIS, C. A. ; GIBBS, S. J. ; REIN, G. Groupware: some issues and experiences. **Communications of the ACM**, New York, v. 34, n. 1, p. 39-58, 1991.

ENGELBRECHT, A. ; BORGES, M. ; VIVACQUA, A. S. Digital tabletops for situational awareness in emergency situations. In: INTERNATIONAL CONFERENCE ON COMPUTER SUPPORTED COOPERATIVE WORK IN DESIGN, 15., 2011, Lausanne. **Proceedings...** Piscataway: IEEE, 2011, p. 669-676.

FAYAD, M. E. ; SCHMIDT, D. C. ; JOHNSON, R. E. **Building application frameworks: object-oriented foundations of framework design**. Hoboken: John Wiley & Sons, Inc., 1999.

FAYAD, M. ; SCHMIDT, D. C. Object-oriented application frameworks. **Communications of the ACM**, New York, v. 40, n. 10, p. 32-38, 1997.

FEMA. **Livestock in disasters - Unit 4**. Washington: Emergency Management Institute, Federal Emergency Management Agency, 2002.

_____. **Local multi-hazard mitigation planning guidance**. Washington: Emergency Management Institute, Federal Emergency Management Agency, 2008.

_____. **Principles of emergency management supplement**. Washington: Emergency Management Institute, Federal Emergency Management Agency, 2007.

_____. **What is emergency management? and what are the principles of emergency management**. background think piece for the emergency management roundtable meeting. Washington: Emergency Management Institute, Federal Emergency Management Agency, 2007.

FERREIRA, A. F. E. **Um modelo de apoio a percepção situacional na resposta a emergências**. 2011. 202 f. Dissertação (Mestrado em Informática) -- Instituto de Matemática, Instituto Tércio Pacciti, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2011.

FHWA. **Evacuation transportation management task five: operational concept**. Washington: Federal Highway Administration, Department of Transportation, 2006.

FOWLER, M. **UML distilled: a brief guide to the standard object modeling language**. Reading: Addison-Wesley Professional, 2004.

FRANCA, L. P. A. **Um Processo para a construção de geradores de artefatos**. 2000. 180 f. (Tese de Doutorado) -- Departamento de Informática, PUC-Rio, Rio de Janeiro, 2000.

FROEHLICH, G. et al. **Designing object-oriented frameworks**. Edmonton: University of Alberta, 1998.

GAMMA, E. et al. **Design patterns: elements of reusable software architecture**. Reading: Addison-Wesley, 1995.

_____. **Design patterns: abstraction and reuse of object-oriented design**. Berlin Heidelberg: Springer, 1993.

GAMMA, E. ; BECK, K. JUnit: A cook's tour. **Java Report**, Zurich, v. 4, n. 5, p. 27-38, 1999.

GNU. **O que é o software livre?**. 2014. Disponível em: <http://www.gnu.org/philosophy/free-sw.html>. Acesso em: mar. 2014.

_____. **Questões Feitas Frequentemente (FAQ) sobre a GNU GPL**. 2014. Disponível em: <http://www.gnu.org/licenses/gpl-faq.pt-br.html>. Acesso em: mar. 2014.

_____. **The GNU operating system**. 2014. Disponível em: <http://www.gnu.org/gnu/gnu.html>. Acesso em: mar. 2014.

GUHA-SAPIR, D. et al. **Annual disaster statistical review 2011: the numbers and trends**. Brussels: Centre for Research on the Epidemiology of Disasters (CRED), 2012.

GUNN, S. W. A. The language of disasters. **Prehospital and Disaster Medicine**, [S. l.], v. 5, n. 4, p. 373-376, 1990.

HADDOW, G. ; BULLOCK, J. ; COPPOLA, D. P. **Introduction to emergency management**. Oxford: Butterworth-Heinemann, 2007.

HHS. **Medical surge capacity and capability handbook**. Washington: Office of Public Health Emergency Preparedness, Department of Health and Human Services, 2004.

HSC. **National planning scenarios**. Washington: Homeland Security Council, 2006.

HUMAYOUN, S. R. et al. Designing mobile systems in highly dynamic scenarios: the workpad methodology. **Knowledge, Technology & Policy**, Houten, v. 22, n. 1, p. 25-43, 2009.

IANNELLA, R. ; ROBINSON, K. ; RINTA-KOSKI, O. Towards a framework for crisis information management systems (CIMS). In: THE INTERNATIONAL EMERGENCY MANAGEMENT SOCIETY, 14., 2007, Trogir. **Proceedings...** Trogir: TIEMS, 2007.

IBARRA, M. et al. A mobile collaborative application to reduce the radio traffic in urban emergencies. In: COMPUTER SUPPORTED COOPERATIVE WORK IN DESIGN, 16., 2012, Wuhan. **Proceedings...** Piscataway: IEEE, 2012. p. 358-365.

IDC. **Android marks fourth anniversary since launch with 75.0% market share in third quarter, according to IDC**. IDC Press. 2012. Disponível em: <http://www.idc.com/getdoc.jsp?containerId=prUS23771812#.UO8kkuQiG1U>. Acesso em: maio 2013.

JAVA. **Java**. Oracle Technology Network, Oracle. 2014. Disponível em: <http://docs.oracle.com/javase/8/docs/index.html>. Acesso em: fev. 2014.

_____. **JavaBean**. 2014. Disponível em: <http://stackoverflow.com/questions/3295496/what-is-a-javabean-exactly>. Acesso em: fev. 2014.

_____. **Javadoc tool**. 2014. Disponível em: <http://www.oracle.com/technetwork/articles/java/index-jsp-135444.html>. Acesso em: fev. 2014.

_____. **Java enumeration**. 2014. Disponível em: <http://docs.oracle.com/javase/7/docs/api/java/util/Enumeration.html>. Acesso em: fev. 2014.

JEMF. **Java emergency management framework**. 2014. Disponível em: <https://code.google.com/p/jemf/>. Acesso em: set. 2014.

_____. **Java emergency management framework no GitHub**. 2014. Disponível em: <https://github.com/marcusfelipetm/jemf/>. Acesso em: nov. 2014.

JEWETT, T. **Database design with UML and SQL**. 2006. Disponível em: <http://www.tomjewett.com/dbdesign/dbdesign.php>. Acesso em: mar. 2014.

JOHNSON, R. E. Components, frameworks, patterns. **ACM SIGSOFT Software Engineering Notes**, New York, v. 22, n. 3, p. 10-17, 1997.

_____. Frameworks = (components + patterns). **Communications of the ACM**, New York, v. 40, n. 10, p. 39-42, 1997.

JOHNSON, R. E. ; FOOTE, B. Designing reusable classes. **Journal of Object-Oriented Programming**, Denville, v. 1, n. 2, p. 22-35, 1988.

JONES, R. W. **Critical incident protocol: a public and private partnership**. East Lansing: Michigan State University, 2000. Disponível em: <http://www.cip.msu.edu/cip.pdf>. Acesso em: nov. 2013.

JOSHI, S. G. **Exploring map-based interfaces for mobile solutions in emergency work**. 2011. 223 f. (Dissertação de Mestrado) -- Department of Informatics, University of Oslo, Oslo, 2011.

KANCHANASUT, K. et al. DUMBONET: a multimedia communication system for collaborative emergency response operations in disaster-affected areas. **International Journal of Emergency Management**, London, v. 4, n. 4, p. 670-681, 2007.

KIENZLE, J. ; GUELFY, N. ; MUSTAFIZ, S. Crisis management systems: a case study for aspect-oriented modeling. In: S. Katz et al. (Eds.), **Transactions on aspect-oriented software development VII**. Berlin Heidelberg: Springer, 2010. p. 1-22. (LNCS 6210).

LUYTEN, K. et al. A situation-aware mobile system to support fire brigades in emergency situations. In: R. Meersman, Z. Tari, P. Herrero et al. (Eds.), **On the Move to Meaningful Internet Systems**. 2006. Berlin Heidelberg: Springer, 2006. p. 1966-1975. (LNCS 4278).

MACHADO, M. F. T. et al. JEMF: A framework for the development of mobile systems for emergency management. In: N. Baloian et al. (Eds.), **Collaboration and Technology**. Berlin Heidelberg: Springer International Publishing, 2014. p. 239-254. (LNCS 8658).

_____. O uso integrado de interface multimodal e dispositivos móveis em gestão de emergências. In: BRAZILIAN SYMPOSIUM ON HUMAN FACTORS IN COMPUTING SYSTEMS, 12., 2013, Manaus. **Proceedings...** Porto Alegre: SBC, 2013, p. 325-326.

MATTSSON, M. **Evolution and composition of object-oriented frameworks**. 2000. 231 f. (Tese de Doutorado) -- Department of Software Engineering and Computer Science, University of Karlskrona/Ronneby, Ronneby, 2000.

MEISSNER, A. et al. MIKoBOS - a mobile information and communication system for emergency response. In: INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS FOR CRISIS RESPONSE AND MANAGEMENT, 3., 2006, Newark. **Proceedings...** [S. l.]: ISCRAM, 2006. p. 92-101.

NGA. **Comprehensive emergency management: a governor's guide**. Washington: National Governors' Association, Defense Civil Preparedness Agency, 1979.

NILSSON, E. G. ; STØLEN, K. Generic functionality in user interfaces for emergency response. In: AUSTRALIAN COMPUTER-HUMAN INTERACTION CONFERENCE, 23., 2011, Canberra. **Proceedings...** New York: ACM, 2011. p. 233-242.

NIMPUNO, K. ; HILMAN, R. **Disaster management glossary**. Nairobi: United Nations Centre for Human Settlements (Habitat). Disaster and Emergency Reference Center, Delft, 1998.

OMG. **Unified modeling language infrastructure specification**: version 2.2. [S. l.]: Object Management Group, 2009.

OPDYKE, W. F. **Refactoring object-oriented frameworks**. 1992. 151 f. (Tese de Doutorado) -- Department of Computer Science, University of Illinois at Urbana-Champaign, 1992.

PADILHA, R. P. **Apoio à colaboração entre equipes de comando e de operações na resposta a emergências: uma proposta utilizando computação móvel**. 2010. 125 f. Dissertação (Mestrado em Informática) -- Instituto de Matemática, Instituto Tércio Pacitti, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2010.

PAELKE, V. et al. Designing multi-modal map-based interfaces for disaster management. In: INTERNATIONAL CONFERENCE ON ADVANCES IN COMPUTER-HUMAN INTERACTIONS, 5., 2012, Valencia. **Proceedings...** [S. l.]: Open Digital Library, 2012. p. 95-100.

PIMENTEL, M. ; FUKS, H. **Sistemas colaborativos**. Rio de Janeiro: Elsevier, 2011.

PORFIRIEV, B. N. Disaster and disaster areas: methodological issues of definition and delineation. **International Journal of Mass Emergencies and Disasters**, Stillwater, v. 13, n. 3, p. 285-304, 1995.

PREE, W. **Design patterns for object-oriented software development**. Reading: Addison-Wesley, 1995.

PRESSMAN, R. S. **Engenharia de software: uma abordagem prática**. 6. ed. São Paulo: MacGraw Hill, 2006.

_____. **Software engineering: a practitioner's approach**. 5. ed. New York: McGraw Hill, 2001.

RADATZ, J. ; GERACI, A. ; KATKI, F. **IEEE standard glossary of software engineering terminology**. New York: IEEE, Std., 1990. (Std 610.12-1990).

RHEINHEIMER, L. R. **JLearningServices: um framework para serviços síncronos em ambientes para EAD**. 2002. 79 f. (Tese de Doutorado) -- Centro de Ciências Exatas e Tecnológicas, Universidade do Vale do Rio dos Sinos, São Leopoldo, 2002.

ROBERTS, D. ; JOHNSON, R. **Evolving frameworks**. pattern languages of program design, 1996. v. 3.

RODRÍGUEZ-COVILI, J. et al. A communication infrastructure to ease the development of mobile collaborative applications. **Journal of Network and Computer Applications**, London, v. 34, n. 6, p. 1883-1893, 2011.

SCHÄCHINGER, U. et al. NOAH - A mobile emergency care system. **European Journal of Medical Research**, London, v. 5, n. 1, p. 13-18, 2000.

SCHMIDT, D. C. ; BUSCHMANN, F. Patterns, frameworks, and middleware: their synergistic relationships. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 25., 2003, Portland. **Proceedings...** Washington: IEEE, 2003. p. 694-704.

SEITO, A. I. et al. **A Segurança contra incêndio no Brasil**. São Paulo: Projeto Editora, 2008.

SHARMA, R. et al. Speech-gesture driven multimodal interfaces for crisis management. **Proceedings of the IEEE**, Piscataway, v. 91, n. 9, p. 1327-1354, 2003.

SILVA, A. J. D. **Um Sistema de computação móvel para suporte ao primeiro respondedor em emergência radiológica**. 2012. 131 f. Dissertação (Mestrado em Informática) -- Instituto de Matemática, Instituto Tércio Pacitti, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2011.

SILVA, R. P. **Suporte ao desenvolvimento e uso de frameworks e componentes**. 2000. 262 f. (Tese de Doutorado) -- Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000.

SOMMERVILLE, I. **Engenharia de software**. Tradução André Maurício de Andrade Ribeiro. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

SOUCHAUD, S. ; FUSCO, W. População e ocupação do espaço: o papel das migrações no Brasil. **REDES - Rev. Des. Regional**, Santa Cruz do Sul, v. 17, n. 2, p. 5-17, 2012.

SVN. **Apache Subversion**. 2014. Disponível em: <http://subversion.apache.org/>. Acesso em: abr. 2014.

SZYPERSKI, C. ; GRUNTZ, D. ; MURER, S. **Component software: beyond object-oriented programming**. 2. ed. Boston: Addison-Wesley/ACM Press, 2002.

TALIGENT. **The Power of frameworks: for windows and OS/2 developers**. Reading: Addison-Wesley, 1995.

TANGEN, S. **ISO/PAS 22399 societal security: guideline for incident preparedness and operational continuity management**. Geneva: ISO Management Systems, 2008. p. 5.

TRUONG, H. et al. ESCAPE - An adaptive framework for managing and providing context information in emergency situations. In: Kortuem, G., Finney, J., Lea, R., Sundramoorthy, V. (Eds.). European Conference Smart Sensing and Context (EuroSSC). Berlin Heidelberg: Springer, 2007. pp. 207-222. (LNCS 4793)

TUROFF, M. et al. The design of a dynamic emergency response management information system (DERMIS). **Journal of Information Technology Theory and Application (JITTA)**, Clear Water Bay, v. 5, n. 4, p. 3, 2004.

UNISDR. **Terminology on disaster risk reduction**. Geneva: United Nations International Strategy for Disaster Reduction, 2009. Disponível em: <http://www.unisdr.org/we/inform/terminology>. Acesso em: jan. 2014.

UOL. Tragédia em Santa Maria (RS) é 2º incêndio mais mortal da história do Brasil. **Portal de Notícias Universo Online**, 2013. Disponível em: <http://noticias.uol.com.br/ultimas-noticias/valor/2013/01/27/tragedia-em-santa-maria-rs-e-2-incendio-mais-mortal-e-5-maior-tragedia-da-historia-do-brasil.htm>. Acesso em: abr. 2013.

WAUGH, W. L. **Living with hazards, dealing with disasters**: an introduction to emergency management. New York: ME Sharpe, 1999.

WEINAND, A. ; GAMMA, E. ; MARTY, R. Design and implementation of ET++, a seamless object-oriented application framework. **Structured Programming**, Berlin, v. 10, n. 2, p. 63-87, 1989.

YANG, Y. J. et al. A UML-based object-oriented framework development methodology. In: ASIA PACIFIC SOFTWARE ENGINEERING CONFERENCE, 1998, Taipei. **Proceedings...** Piscataway: IEEE, 1998, p. 211-218.

ZHANG, Z. ; LI, Q. An open urban emergency decision support system. **The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences**, Reston, v. 37, n. B4, p. 1123-1128, 2008.

Apêndices

APÊNDICE A – QUADRO DE FUNCIONALIDADES DOS SISTEMAS DE INFORMAÇÃO NA GESTÃO DE EMERGÊNCIAS

Este apêndice tem por objetivo apresentar as funcionalidades levantadas dos sistemas de informação na Gestão de Emergências descritos no Capítulo 3. A partir desse levantamento, foram identificadas e estabelecidas as principais funcionalidades que compõem o arcabouço JEMF (Quadro A.1).

Quadro A.1: Classificação dos sistemas de informação de acordo com as categorias de funcionalidade (NILSSON; STØLEN, 2011).

<i>Funcionalidade</i>	<i>Projeto</i>											
	NOAH	MIKo- BOS	Geo- BIPS	DUM- BONET	DMT	WORK- PAD	SisC2- Celular	CCCMS	Mobile- Map	EMER- GENCY	Sis.Emer- gência	ProRad
<i>Visão Operacional</i>												
Registrar emergência	Sim										Sim	
Visualizar situação geral da resposta			Sim		Sim				Sim	Sim		Sim
Enviar relatório de situação de atual	Sim	Sim					Sim					
<i>Gestão de Recursos</i>												
Registrar recurso interno			Sim			Sim				Sim	Sim	
Alocar recurso interno							Sim	Sim		Sim		
Alocar recurso externo							Sim	Sim				
Visualizar recursos disponíveis									Sim	Sim		
Visualizar recursos alocados										Sim		
<i>Planos e Ações</i>												
Registrar missão						Sim						
Registrar tarefa			Sim			Sim	Sim	Sim				
Registrar dados contextuais						Sim		Sim				Sim
<i>Monitoramento</i>												
Visualizar status da equipe/ator			Sim									
Alertar equipe/ator sobre situação de risco			Sim						Sim			Sim
<i>Gestão da Comunicação</i>												

Enviar Mensagem Textual	Sim	Sim		Sim		Sim	Sim		Sim		Sim	
Receber Mensagem Textual	Sim			Sim		Sim	Sim		Sim		Sim	
Enviar Mensagem de Áudio			Sim	Sim								
Receber Mensagem de Áudio			Sim	Sim					Sim			
<i>Serviço de Informação</i>												
Acessar informação compartilhada	Sim	Sim			Sim	Sim			Sim			Sim
<i>Transmissão</i>												
Transmitir imagem	Sim	Sim	Sim	Sim					Sim		Sim	
<i>Mecanismo Especial de Interação</i>												
Interfaces multimodais												
Realidade aumentada												
<i>Detalhes do Incidente</i>												
Registrar vítima	Sim	Sim		Sim							Sim	Sim
Registrar testemunha								Sim				
Registrar infraestrutura afetada											Sim	
<i>Análise Automatizada</i>												
Visualizar localização de ator/equipe/recurso			Sim		Sim		Sim		Sim	Sim	Sim	
Registrar ponto de interesse					Sim				Sim			
Registrar dados geográficos			Sim			Sim						Sim
Acessar mapa					Sim		Sim		Sim	Sim	Sim	Sim
<i>Registro</i>												
Registrar histórico de interações entre atores/equipes									Sim			

APÊNDICE B – DIAGRAMAS DE CASOS DE USO E SUAS DESCRIÇÕES PARA AS FUNCIONALIDADES DO JEMF

Este apêndice tem por objetivo discorrer a descrição de casos de uso para as operações de Incluir, Alterar, Consultar e Excluir das funcionalidades estabelecidas para o arcabouço proposto.

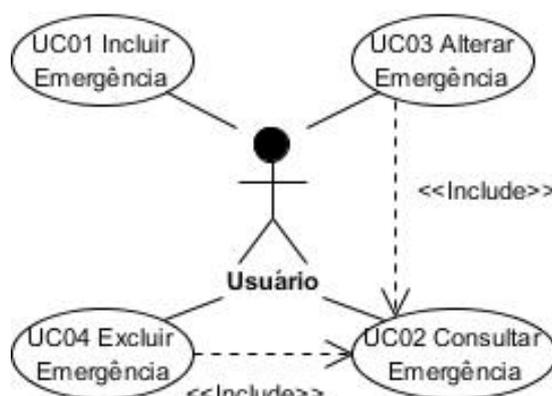


Figura B.1: Casos de Uso da funcionalidade Gerenciar Emergência

Quadro B.1: Caso de Uso – Incluir Emergência

UC01 – Incluir Emergência	
Objetivo	Incluir uma nova Emergência.
Atores	Usuário (Ex. Comandante, Médico).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Emergência; 2. O Ator seleciona a opção incluir; 3. O Ator insere os dados da Emergência: nome, ativado, nível, data de início, data de término e tipo; 4. O Ator solicita a confirmação dos dados [FA1]; 5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma o cadastro de Emergência.
Fluxo Alternativo	1. [FA1] Desistência da inclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	Não possui.
Pós-Condições	Dados da emergência inseridos no sistema.
Regras de Negócio	[RN1] A emergência só poderá ser incluída pelos atores usuários, por exemplo, Comandante ou Médico.

Quadro B.2: Caso de Uso – Consultar Emergência

UC02 – Consultar Emergência	
Objetivo	Consultar uma emergência cadastrada.
Atores	Usuário (Ex. Comandante, Médico).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Emergência; 2. O Ator seleciona a opção consultar; 3. O sistema retorna os dados da consulta [FE1];
Fluxo Alternativo	Não possui.
Fluxo Exceção	1. [FE1] Nenhuma emergência cadastrada [MSG1];
Inclusões	Não possui.
Pós-Condições	Dados da emergência consultados.
Regras de Negócio	[RN1] A consulta da emergência só poderá ser visualizada pelos atores usuários, por exemplo, Comandante ou Médico.

Quadro B.3: Caso de Uso – Alterar Emergência

UC03 – Alterar Emergência	
Objetivo	Alterar uma emergência cadastrada.
Atores	Usuário (Ex. Comandante, Médico).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Emergência; 2. O Ator seleciona a opção alterar; 3. O Ator realiza consulta [UC02 – Consultar Emergência]; 3. O Ator altera os dados da emergência: nome, ativado, nível, data de início, data de término e tipo; 4. O Ator solicita a confirmação dos dados [FA1]; 5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma a alteração de emergência.
Fluxo Alternativo	1. [FA1] Desistência da alteração e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	UC02 – Consultar Emergência.
Pós-Condições	Dados da emergência atualizados.
Regras de Negócio	[RN1] A emergência só poderá ser alterada pelos atores usuários, por exemplo, Comandante ou Médico.

Quadro B.4: Caso de Uso – Excluir Emergência

UC04 – Excluir Emergência	
Objetivo	Excluir uma emergência cadastrada.
Atores	Usuário (Ex. Comandante, Médico).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro Emergência; 2. O Ator seleciona a opção excluir; 3. O Ator realiza consulta [UC02 – Consultar Emergência]; 4. O Ator exclui a emergência [MSG1] [FA1] [FE1]; 5. O sistema confirma a exclusão dos dados.
Fluxo Alternativo	1. [FA1] Desistência da exclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Caso a emergência tiver vínculos associados a outros cadastros ou restrições específicas, não poderá ser excluído [MSG1].
Inclusões	UC02 – Consultar Emergência.
Pós-Condições	Dados da emergência excluídos do sistema.
Regras de Negócio	[RN1] A emergência só poderá ser excluída pelos atores usuários, por exemplo, Comandante ou Médicos. [RN2] A emergência não poderá ter nenhum vínculo associado ou restrições específicas.

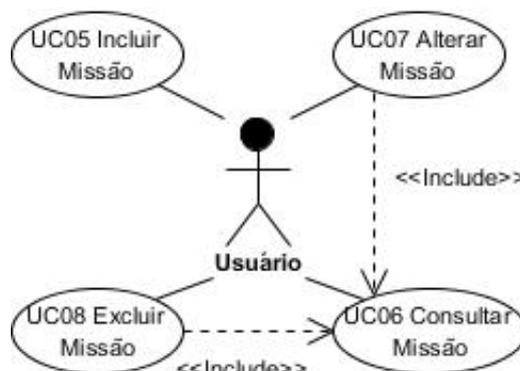


Figura B.2: Casos de Uso da funcionalidade Gerenciar Missão

Quadro B.5: Caso de Uso – Incluir Missão

UC05 – Incluir Missão	
Objetivo	Incluir uma nova missão.
Atores	Usuário (Ex. Comandante, Combatente, Médico).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Missão; 2. O Ator seleciona a opção incluir; 3. O Ator insere os dados da missão: título, descrição, status, data de início, data de término, prioridade; 4. O Ator solicita a confirmação dos dados [FA1];

	5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma o cadastro da missão.
Fluxo Alternativo	1. [FA1] Desistência da inclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	Não possui.
Pós-Condições	Dados da missão inseridos no sistema.
Regras de Negócio	[RN1] A missão só poderá ser incluída pelos atores usuários, por exemplo, Comandante, Combatente ou Médico.

Quadro B.6: Caso de Uso – Consultar Missão

UC06 – Consultar Missão	
Objetivo	Consultar uma missão cadastrada.
Atores	Usuário (Ex. Comandante, Combatente, Médico).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Missão; 2. O Ator seleciona a opção consultar; 3. O sistema retorna os dados da consulta [FE1];
Fluxo Alternativo	Não possui.
Fluxo Exceção	1. [FE1] Nenhuma missão cadastrada [MSG1];
Inclusões	Não possui.
Pós-Condições	Dados da missão consultados.
Regras de Negócio	[RN1] A consulta da missão só poderá ser visualizada pelos atores usuários, por exemplo, Comandante, Combatente ou Médico.

Quadro B.7: Caso de Uso – Alterar Missão

UC07 – Alterar Missão	
Objetivo	Alterar uma missão cadastrada.
Atores	Usuário (Ex. Comandante, Combatente, Médico).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Missão; 2. O Ator seleciona a opção alterar; 3. O Ator realiza consulta [UC06 – Consultar Missão]; 3. O Ator altera os dados da missão: título, descrição, status, data de início, data de término, prioridade; 4. O Ator solicita a confirmação dos dados [FA1]; 5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma a alteração da missão.

Fluxo Alternativo	1. [FA1] Desistência da alteração e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	UC06 – Consultar Missão.
Pós-Condições	Dados da missão atualizados.
Regras de Negócio	[RN1] A missão só poderá ser alterada pelos atores usuários, por exemplo, Comandante, Combatente ou Médico.

Quadro B.8: Caso de Uso – Excluir Missão

UC08 – Excluir Missão	
Objetivo	Excluir uma missão cadastrada.
Atores	Usuário (Ex. Comandante, Combatente, Médico).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Missão; 2. O Ator seleciona a opção excluir; 3. O Ator realiza consulta [UC06 – Consultar Missão]; 4. O Ator exclui a missão [MSG1] [FA1] [FE1]; 5. O sistema confirma a exclusão dos dados.
Fluxo Alternativo	1. [FA1] Desistência da exclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Caso a missão tiver vínculos associados a outros cadastros ou restrições específicas, não poderá ser excluído [MSG1].
Inclusões	UC06 – Consultar Missão.
Pós-Condições	Dados da missão excluídos do sistema.
Regras de Negócio	[RN1] A missão só poderá ser excluída pelos atores usuários, por exemplo, Comandante, Combatente ou Médico. [RN2] A missão não poderá ter nenhum vínculo associado ou restrições específicas.

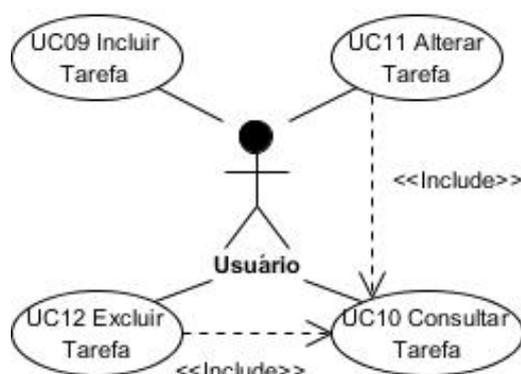


Figura B.3: Casos de Uso da funcionalidade Gerenciar Tarefa

Quadro B.9: Caso de Uso – Incluir Tarefa

UC09 – Incluir Tarefa	
Objetivo	Incluir uma nova tarefa.
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Tarefa; 2. O Ator seleciona a opção incluir; 3. O Ator insere os dados da tarefa: título, descrição, status, data de início, data de término, prioridade; 4. O Ator solicita a confirmação dos dados [FA1]; 5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma o cadastro da tarefa.
Fluxo Alternativo	1. [FA1] Desistência da inclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	Não possui.
Pós-Condições	Dados da tarefa inseridos no sistema.
Regras de Negócio	[RN1] A tarefa só poderá ser incluída pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário.

Quadro B.10: Caso de Uso – Consultar Tarefa

UC10 – Consultar Tarefa	
Objetivo	Consultar uma tarefa cadastrada.
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Tarefa; 2. O Ator seleciona a opção consultar; 3. O sistema retorna os dados da consulta [FE1];
Fluxo Alternativo	Não possui.
Fluxo Exceção	1. [FE1] Nenhuma tarefa cadastrada [MSG1];
Inclusões	Não possui.
Pós-Condições	Dados da tarefa consultados.
Regras de Negócio	[RN1] A consulta da tarefa só poderá ser visualizada pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário.

Quadro B.11: Caso de Uso – Alterar Tarefa

UC11 – Alterar Tarefa	
Objetivo	Alterar uma tarefa cadastrada.
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Tarefa; 2. O Ator seleciona a opção alterar; 3. O Ator realiza consulta [UC10 – Consultar Tarefa]; 3. O Ator altera os dados da tarefa: título, descrição, status, data de início, data de término, prioridade; 4. O Ator solicita a confirmação dos dados [FA1]; 5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma a alteração da tarefa.
Fluxo Alternativo	1. [FA1] Desistência da alteração e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	UC10 – Consultar Tarefa.
Pós-Condições	Dados da tarefa atualizados.
Regras de Negócio	[RN1] A tarefa só poderá ser alterada pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário.

Quadro B.12: Caso de Uso – Excluir Tarefa

UC12 – Excluir Tarefa	
Objetivo	Excluir uma tarefa cadastrada.
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Tarefa; 2. O Ator seleciona a opção excluir; 3. O Ator realiza consulta [UC10 – Consultar Tarefa]; 4. O Ator exclui a tarefa [MSG1] [FA1] [FE1]; 5. O sistema confirma a exclusão dos dados.
Fluxo Alternativo	1. [FA1] Desistência da exclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Caso a tarefa tiver vínculos associados a outros cadastros ou restrições específicas, não poderá ser excluído [MSG1].
Inclusões	UC10 – Consultar Tarefa.
Pós-Condições	Dados da tarefa excluídos do sistema.
Regras de Negócio	[RN1] A tarefa só poderá ser excluída pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário. [RN2] A tarefa não poderá ter nenhum vínculo associado ou restrições específicas.

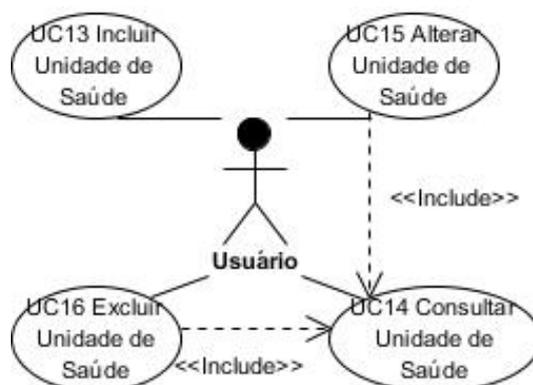


Figura B.4: Casos de Uso da funcionalidade Gerenciar Unidade de Saúde

Quadro B.13: Caso de Uso – Incluir Unidade de Saúde

UC13 – Incluir Unidade de Saúde	
Objetivo	Incluir uma nova Unidade de Saúde.
Atores	Usuário (Ex. Comandante, Médico).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Unidade de Saúde; 2. O Ator seleciona a opção incluir; 3. O Ator insere os dados da unidade de saúde: nome, descrição, nível, capacidade; 4. O Ator solicita a confirmação dos dados [FA1]; 5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma o cadastro da unidade de saúde.
Fluxo Alternativo	1. [FA1] Desistência da inclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	Não possui.
Pós-Condições	Dados da unidade de saúde inseridos no sistema.
Regras de Negócio	[RN1] A unidade de saúde só poderá ser incluída pelos atores usuários, por exemplo, Comandante ou Combatente.

Quadro B.14: Caso de Uso – Consultar Unidade de Saúde

UC14 – Consultar Unidade de Saúde	
Objetivo	Consultar uma Unidade de Saúde cadastrada.
Atores	Usuário (Ex. Comandante, Combatente).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Unidade de Saúde; 2. O Ator seleciona a opção consultar;

	3. O sistema retorna os dados da consulta [FE1];
Fluxo Alternativo	Não possui.
Fluxo Exceção	1. [FE1] Nenhuma unidade de saúde cadastrada [MSG1];
Inclusões	Não possui.
Pós-Condições	Dados da unidade de saúde consultados.
Regras de Negócio	[RN1] A consulta da unidade de saúde só poderá ser visualizada pelos atores usuários, por exemplo, Comandante ou Médico.

Quadro B.15: Caso de Uso – Alterar Unidade de Saúde

UC15 – Alterar Unidade de Saúde	
Objetivo	Alterar uma Unidade de Saúde cadastrada.
Atores	Usuário (Ex. Comandante, Médico).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Unidade de Saúde; 2. O Ator seleciona a opção alterar; 3. O Ator realiza consulta [UC14 – Consultar Unidade de Saúde]; 3. O Ator altera os dados da unidade de saúde: nome, descrição, nível, capacidade; 4. O Ator solicita a confirmação dos dados [FA1]; 5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma a alteração da unidade de saúde.
Fluxo Alternativo	1. [FA1] Desistência da alteração e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	UC14 – Consultar Unidade de Saúde.
Pós-Condições	Dados da unidade de saúde atualizados.
Regras de Negócio	[RN1] A unidade de saúde só poderá ser alterada pelos atores usuários, por exemplo, Comandante ou Médico.

Quadro B.16: Caso de Uso – Excluir Unidade de Saúde

UC16 – Excluir Unidade de Saúde	
Objetivo	Excluir uma Unidade de Saúde cadastrada.
Atores	Usuário (Ex. Comandante, Médico).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Unidade de Saúde; 2. O Ator seleciona a opção excluir; 3. O Ator realiza consulta [UC14 – Consultar Unidade de Saúde]; 4. O Ator exclui a unidade de saúde [MSG1] [FA1] [FE1]; 5. O sistema confirma a exclusão dos dados.

Fluxo Alternativo	1. [FA1] Desistência da exclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Caso a unidade de saúde tiver vínculos associados a outros cadastros ou restrições específicas, não poderá ser excluído [MSG1].
Inclusões	UC14 – Consultar Unidade de Saúde.
Pós-Condições	Dados da unidade de saúde excluídos do sistema.
Regras de Negócio	[RN1] A unidade de saúde só poderá ser excluída pelos atores usuários, por exemplo, Comandante ou Médico. [RN2] A unidade de saúde não poderá ter nenhum vínculo associado ou restrições específicas.

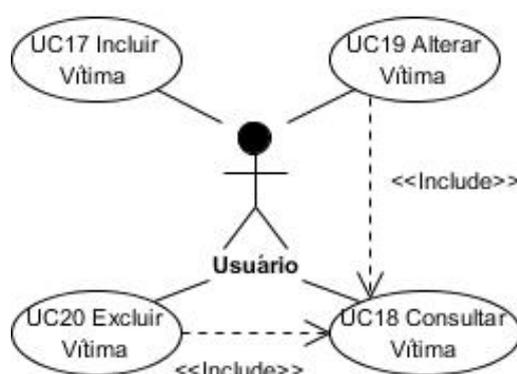


Figura B.5: Casos de Uso da funcionalidade Gerenciar Vítima

Quadro B.17: Caso de Uso – Incluir Vítima

UC17 – Incluir Vítima	
Objetivo	Incluir uma nova Vítima.
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Vítima; 2. O Ator seleciona a opção incluir; 3. O Ator insere os dados da vítima: nome, gênero, data de nascimento, idade, status; 4. O Ator solicita a confirmação dos dados [FA1]; 5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma o cadastro da vítima.
Fluxo Alternativo	1. [FA1] Desistência da inclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	Não possui.
Pós-Condições	Dados da vítima inseridos no sistema.
Regras de Negócio	[RN1] A vítima só poderá ser incluída pelos atores usuários, por

	exemplo, Comandante, Combatente, Médico ou Voluntário.
--	--

Quadro B.18: Caso de Uso – Consultar Vítima

UC18 – Consultar Vítima	
Objetivo	Consultar uma vítima cadastrada.
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Vítima; 2. O Ator seleciona a opção consultar; 3. O sistema retorna os dados da consulta [FE1];
Fluxo Alternativo	Não possui.
Fluxo Exceção	1. [FE1] Nenhuma vítima cadastrada [MSG1];
Inclusões	Não possui.
Pós-Condições	Dados da vítima consultados.
Regras de Negócio	[RN1] A consulta da vítima só poderá ser visualizada pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário.

Quadro B.19: Caso de Uso – Alterar Vítima

UC19 – Alterar Vítima	
Objetivo	Alterar uma vítima cadastrada.
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Vítima; 2. O Ator seleciona a opção alterar; 3. O Ator realiza consulta [UC18 – Consultar Vítima]; 3. O Ator altera os dados da vítima: nome, gênero, data de nascimento, idade, status; 4. O Ator solicita a confirmação dos dados [FA1]; 5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma a alteração da vítima.
Fluxo Alternativo	1. [FA1] Desistência da alteração e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	UC18 – Consultar Vítima.
Pós-Condições	Dados da vítima atualizados.
Regras de Negócio	[RN1] A vítima só poderá ser alterada pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário.

Quadro B.20: Caso de Uso – Excluir Vítima

UC20 – Excluir Vítima	
Objetivo	Excluir uma vítima cadastrada.
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Vítima; 2. O Ator seleciona a opção excluir; 3. O Ator realiza consulta [UC18 – Consultar Vítima]; 4. O Ator exclui a vítima [MSG1] [FA1] [FE1]; 5. O sistema confirma a exclusão dos dados.
Fluxo Alternativo	1. [FA1] Desistência da exclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Caso a vítima tiver vínculos associados a outros cadastros ou restrições específicas, não poderá ser excluído [MSG1].
Inclusões	UC18 – Consultar Vítima.
Pós-Condições	Dados da vítima excluídos do sistema.
Regras de Negócio	[RN1] A vítima só poderá ser excluída pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário. [RN2] A vítima não poderá ter nenhum vínculo associado ou restrições específicas.

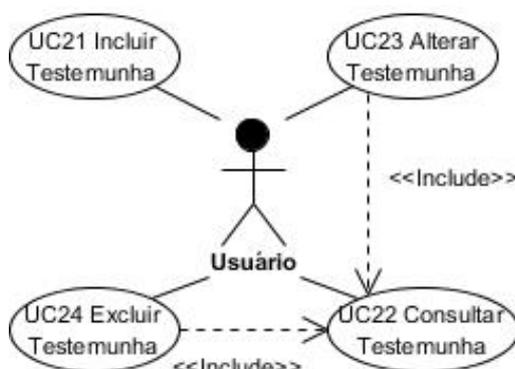


Figura B.6: Casos de Uso da funcionalidade Gerenciar Testemunha

Quadro B.21: Caso de Uso – Incluir Testemunha

UC21 – Incluir Testemunha	
Objetivo	Incluir uma nova Testemunha.
Atores	Usuário (Ex. Combatente, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Testemunha; 2. O Ator seleciona a opção incluir; 3. O Ator insere os dados da Testemunha: nome, gênero, data de nascimento, idade; 4. O Ator solicita a confirmação dos dados [FA1];

	5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma o cadastro da Testemunha.
Fluxo Alternativo	1. [FA1] Desistência da inclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	Não possui.
Pós-Condições	Dados da Testemunha inseridos no sistema.
Regras de Negócio	[RN1] A Testemunha só poderá ser incluída pelos atores usuários, por exemplo, Combatente ou Voluntário.

Quadro B.22: Caso de Uso – Consultar Testemunha

UC22 – Consultar Testemunha	
Objetivo	Consultar uma Testemunha cadastrada.
Atores	Usuário (Ex. Combatente, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Testemunha; 2. O Ator seleciona a opção consultar; 3. O sistema retorna os dados da consulta [FE1];
Fluxo Alternativo	Não possui.
Fluxo Exceção	1. [FE1] Nenhuma Testemunha cadastrada [MSG1];
Inclusões	Não possui.
Pós-Condições	Dados da Testemunha consultados.
Regras de Negócio	[RN1] A consulta da Testemunha só poderá ser visualizada pelos atores usuários, por exemplo, Combatente ou Voluntário.

Quadro B.23: Caso de Uso – Alterar Testemunha

UC23 – Alterar Testemunha	
Objetivo	Alterar uma Testemunha cadastrada.
Atores	Usuário (Ex. Combatente, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Testemunha; 2. O Ator seleciona a opção alterar; 3. O Ator realiza consulta [UC22 – Consultar Testemunha]; 3. O Ator altera os dados da Testemunha: nome, gênero, data de nascimento, idade; 4. O Ator solicita a confirmação dos dados [FA1]; 5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma a alteração da Testemunha.

Fluxo Alternativo	1. [FA1] Desistência da alteração e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	UC22 – Consultar Testemunha.
Pós-Condições	Dados da Testemunha atualizados.
Regras de Negócio	[RN1] A Testemunha só poderá ser alterada pelos atores usuários, por exemplo, Combatente ou Voluntário.

Quadro B.24: Caso de Uso – Excluir Testemunha

UC24 – Excluir Testemunha	
Objetivo	Excluir uma Testemunha cadastrada.
Atores	Usuário (Ex. Combatente, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Testemunha; 2. O Ator seleciona a opção excluir; 3. O Ator realiza consulta [UC22 – Consultar Testemunha]; 4. O Ator exclui a Testemunha [MSG1] [FA1] [FE1]; 5. O sistema confirma a exclusão dos dados.
Fluxo Alternativo	1. [FA1] Desistência da exclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Caso a Testemunha tiver vínculos associados a outros cadastros ou restrições específicas, não poderá ser excluído [MSG1].
Inclusões	UC22 – Consultar Testemunha.
Pós-Condições	Dados da Testemunha excluídos do sistema.
Regras de Negócio	[RN1] A Testemunha só poderá ser excluída pelos atores usuários, por exemplo, Combatente ou Voluntário. [RN2] A Testemunha não poderá ter nenhum vínculo associado ou restrições específicas.

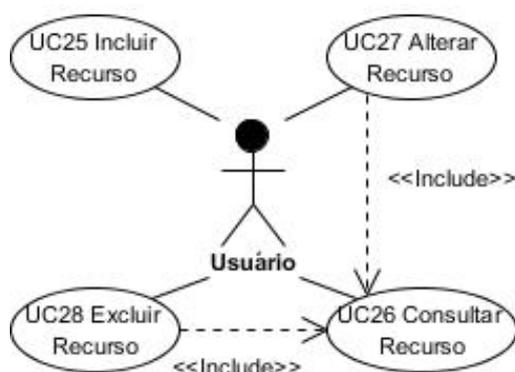


Figura B.7: Casos de Uso da funcionalidade Gerenciar Recurso

Quadro B.25: Caso de Uso – Incluir Recurso

UC25 – Incluir Recurso	
Objetivo	Incluir um novo Recurso (Ex. Equipamento).
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Recurso; 2. O Ator seleciona a opção incluir; 3. O Ator insere os dados do Recurso: nome, descrição, status; 4. O Ator solicita a confirmação dos dados [FA1]; 5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma o cadastro do Recurso.
Fluxo Alternativo	1. [FA1] Desistência da inclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	Não possui.
Pós-Condições	Dados do Recurso inseridos no sistema.
Regras de Negócio	[RN1] O Recurso só poderá ser incluído pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário.

Quadro B.26: Caso de Uso – Consultar Recurso

UC26 – Consultar Recurso	
Objetivo	Consultar um Recurso cadastrado (Ex. Equipamento).
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Recurso; 2. O Ator seleciona a opção consultar; 3. O sistema retorna os dados da consulta [FE1];
Fluxo Alternativo	Não possui.
Fluxo Exceção	1. [FE1] Nenhum Recurso cadastrado [MSG1];
Inclusões	Não possui.
Pós-Condições	Dados do Recurso consultados.
Regras de Negócio	[RN1] A consulta do Recurso só poderá ser visualizada pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário.

Quadro B.27: Caso de Uso – Alterar Recurso

UC27 – Alterar Recurso	
Objetivo	Alterar um Recurso cadastrado (Ex. Equipamento).
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Recurso; 2. O Ator seleciona a opção alterar; 3. O Ator realiza consulta [UC26 – Consultar Recurso]; 3. O Ator altera os dados do Recurso: nome, descrição, status; 4. O Ator solicita a confirmação dos dados [FA1]; 5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma a alteração do Recurso.
Fluxo Alternativo	1. [FA1] Desistência da alteração e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	UC26 – Consultar Recurso.
Pós-Condições	Dados do Recurso atualizados.
Regras de Negócio	[RN1] O Recurso só poderá ser alterado pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário.

Quadro B.28: Caso de Uso – Excluir Recurso

UC28 – Excluir Recurso	
Objetivo	Excluir um Recurso cadastrado (Ex. Equipamento).
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Recurso; 2. O Ator seleciona a opção excluir; 3. O Ator realiza consulta [UC26 – Consultar Recurso]; 4. O Ator exclui o Recurso [MSG1] [FA1] [FE1]; 5. O sistema confirma a exclusão dos dados.
Fluxo Alternativo	1. [FA1] Desistência da exclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Caso a Recurso tiver vínculos associados a outros cadastros ou restrições específicas, não poderá ser excluído [MSG1].
Inclusões	UC26 – Consultar Recurso.
Pós-Condições	Dados do Recurso excluídos do sistema.
Regras de Negócio	[RN1] O Recurso só poderá ser excluído pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário. [RN2] O Recurso não poderá ter nenhum vínculo associado ou restrições específicas.

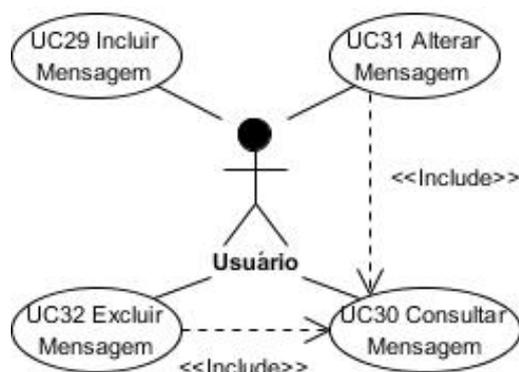


Figura B.8: Casos de Uso da funcionalidade Gerenciar Mensagem

Quadro B.29: Caso de Uso – Incluir Mensagem

UC29 – Incluir Mensagem	
Objetivo	Incluir uma nova Mensagem.
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Mensagem; 2. O Ator seleciona a opção incluir; 3. O Ator insere os dados da Mensagem: data de criação, data de ultima modificação, prioridade, assunto, conteúdo; 4. O Ator solicita a confirmação dos dados [FA1]; 5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma o cadastro da Mensagem.
Fluxo Alternativo	1. [FA1] Desistência da inclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	Não possui.
Pós-Condições	Dados da Mensagem inseridos no sistema.
Regras de Negócio	[RN1] A Mensagem só poderá ser incluída pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário.

Quadro B.30: Caso de Uso – Consultar Mensagem

UC30 – Consultar Mensagem	
Objetivo	Consultar uma Mensagem cadastrada.
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Mensagem; 2. O Ator seleciona a opção consultar; 3. O sistema retorna os dados da consulta [FE1];

Fluxo Alternativo	Não possui.
Fluxo Exceção	1. [FE1] Nenhuma Mensagem cadastrada [MSG1];
Inclusões	Não possui.
Pós-Condições	Dados da Mensagem consultados.
Regras de Negócio	[RN1] A consulta da Mensagem só poderá ser visualizada pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário.

Quadro B.31: Caso de Uso – Alterar Mensagem

UC31 – Alterar Mensagem	
Objetivo	Alterar uma Mensagem cadastrada.
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Mensagem; 2. O Ator seleciona a opção alterar; 3. O Ator realiza consulta [UC30 – Consultar Mensagem]; 3. O Ator altera os dados da Mensagem: data de criação, data de ultima modificação, prioridade, assunto, conteúdo; 4. O Ator solicita a confirmação dos dados [FA1]; 5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma a alteração da Mensagem.
Fluxo Alternativo	1. [FA1] Desistência da alteração e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	UC30 – Consultar Mensagem.
Pós-Condições	Dados da Mensagem atualizados.
Regras de Negócio	[RN1] A Mensagem só poderá ser alterada pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário.

Quadro B.32: Caso de Uso – Excluir Mensagem

UC32 – Excluir Mensagem	
Objetivo	Excluir uma Mensagem cadastrada.
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Mensagem; 2. O Ator seleciona a opção excluir; 3. O Ator realiza consulta [UC30 – Consultar Mensagem]; 4. O Ator exclui a Mensagem [MSG1] [FA1] [FE1]; 5. O sistema confirma a exclusão dos dados.

Fluxo Alternativo	1. [FA1] Desistência da exclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Caso a Mensagem tiver vínculos associados a outros cadastros ou restrições específicas, não poderá ser excluído [MSG1].
Inclusões	UC30 – Consultar Mensagem.
Pós-Condições	Dados da Mensagem excluídos do sistema.
Regras de Negócio	[RN1] A Mensagem só poderá ser excluída pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário. [RN2] A Mensagem não poderá ter nenhum vínculo associado ou restrições específicas.

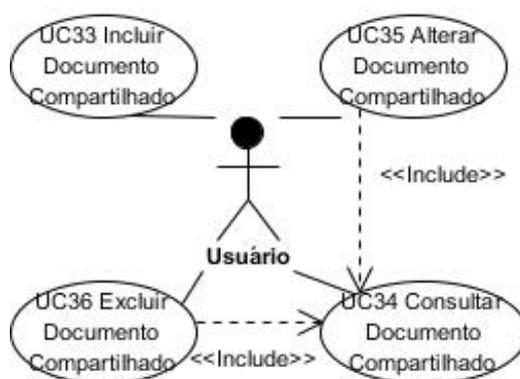


Figura B.9: Casos de Uso da funcionalidade Gerenciar Documento Compartilhado

Quadro B.33: Caso de Uso – Incluir Documento Compartilhado

UC33 – Incluir Documento Compartilhado	
Objetivo	Incluir um novo Documento Compartilhado (Ex. Plano de Resposta).
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Documento Compartilhado; 2. O Ator seleciona a opção incluir; 3. O Ator insere os dados do Documento Compartilhado: data de criação, data de ultima modificação, prioridade, descrição, observação; 4. O Ator solicita a confirmação dos dados [FA1]; 5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma o cadastro do Documento Compartilhado.
Fluxo Alternativo	1. [FA1] Desistência da inclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	Não possui.

Pós-Condições	Documento Compartilhado inserido no sistema.
Regras de Negócio	[RN1] O Documento Compartilhado só poderá ser incluído pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário.

Quadro B.34: Caso de Uso – Consultar Documento Compartilhado

UC34 – Consultar Documento Compartilhado	
Objetivo	Consultar um Documento Compartilhado cadastrado (Ex. Plano de Resposta).
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Documento Compartilhado; 2. O Ator seleciona a opção consultar; 3. O sistema retorna os dados da consulta [FE1];
Fluxo Alternativo	Não possui.
Fluxo Exceção	1. [FE1] Nenhum Documento Compartilhado cadastrado [MSG1];
Inclusões	Não possui.
Pós-Condições	Documento Compartilhado consultado.
Regras de Negócio	[RN1] A consulta do Documento Compartilhado só poderá ser visualizada pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário.

Quadro B.35: Caso de Uso – Alterar Documento Compartilhado

UC35 – Alterar Documento Compartilhado	
Objetivo	Alterar um Documento Compartilhado cadastrado (Ex. Plano de Resposta).
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Documento Compartilhado; 2. O Ator seleciona a opção alterar; 3. O Ator realiza consulta [UC34 – Consultar Documento Compartilhado]; 3. O Ator altera o Documento Compartilhado: data de criação, data de ultima modificação, prioridade, descrição, observação; 4. O Ator solicita a confirmação dos dados [FA1]; 5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma a alteração do Documento Compartilhado.
Fluxo Alternativo	1. [FA1] Desistência da alteração e solicitação do cancelamento do processo [MSG1].

Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	UC34 – Consultar Documento Compartilhado.
Pós-Condições	Dados do Documento Compartilhado atualizados.
Regras de Negócio	[RN1] O Documento Compartilhado só poderá ser alterado pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário.

Quadro B.36: Caso de Uso – Excluir Documento Compartilhado

UC36 – Excluir Documento Compartilhado	
Objetivo	Excluir um Documento Compartilhado cadastrado (Ex. Plano de Resposta).
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Documento Compartilhado; 2. O Ator seleciona a opção excluir; 3. O Ator realiza consulta [UC34 – Consultar Documento Compartilhado]; 4. O Ator exclui o Documento Compartilhado [MSG1] [FA1] [FE1]; 5. O sistema confirma a exclusão dos dados.
Fluxo Alternativo	1. [FA1] Desistência da exclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Caso o Documento Compartilhado tiver vínculos associados a outros cadastros ou restrições específicas, não poderá ser excluído [MSG1].
Inclusões	UC34 – Consultar Documento Compartilhado.
Pós-Condições	Dados do Documento Compartilhado excluídos do sistema.
Regras de Negócio	[RN1] O Documento Compartilhado só poderá ser excluído pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário. [RN2] O Documento Compartilhado não poderá ter nenhum vínculo associado ou restrições específicas.

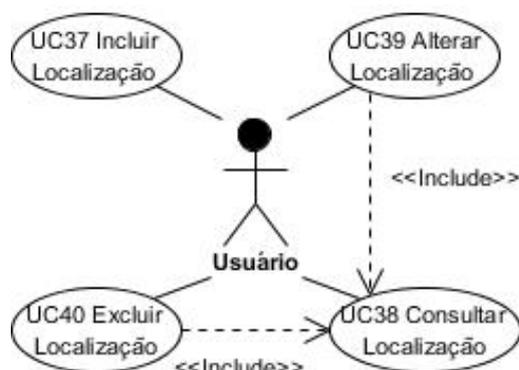


Figura B.10: Casos de Uso da funcionalidade Gerenciar Localização

Quadro B.37: Caso de Uso – Incluir Localização

UC37 – Incluir Localização	
Objetivo	Incluir uma nova Localização.
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Localização; 2. O Ator seleciona a opção incluir; 3. O Ator insere os dados da Localização: porção de tempo, intervalo, status; 4. O Ator solicita a confirmação dos dados [FA1]; 5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma o cadastro da Localização.
Fluxo Alternativo	1. [FA1] Desistência da inclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	Não possui.
Pós-Condições	Dados da Localização inseridos no sistema.
Regras de Negócio	[RN1] A Localização só poderá ser incluída pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário.

Quadro B.38: Caso de Uso – Consultar Localização

UC38 – Consultar Localização	
Objetivo	Consultar uma Localização cadastrada.
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Localização; 2. O Ator seleciona a opção consultar; 3. O sistema retorna os dados da consulta [FE1];

Fluxo Alternativo	Não possui.
Fluxo Exceção	1. [FE1] Nenhuma Localização cadastrada [MSG1];
Inclusões	Não possui.
Pós-Condições	Dados da Localização consultados.
Regras de Negócio	[RN1] A consulta da Localização só poderá ser visualizada pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário.

Quadro B.39: Caso de Uso – Alterar Localização

UC39 – Alterar Localização	
Objetivo	Alterar uma Localização cadastrada.
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Localização; 2. O Ator seleciona a opção alterar; 3. O Ator realiza consulta [UC38 – Consultar Localização]; 3. O Ator altera os dados da Localização: porção de tempo, intervalo, status; 4. O Ator solicita a confirmação dos dados [FA1]; 5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma a alteração da Localização.
Fluxo Alternativo	1. [FA1] Desistência da alteração e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	UC38 – Consultar Localização.
Pós-Condições	Dados da Localização atualizados.
Regras de Negócio	[RN1] A Localização só poderá ser alterada pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário.

Quadro B.40: Caso de Uso – Excluir Localização

UC40 – Excluir Localização	
Objetivo	Excluir uma Localização cadastrada.
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Localização; 2. O Ator seleciona a opção excluir; 3. O Ator realiza consulta [UC38 – Consultar Localização]; 4. O Ator exclui a Localização [MSG1] [FA1] [FE1]; 5. O sistema confirma a exclusão dos dados.

Fluxo Alternativo	1. [FA1] Desistência da exclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Caso a Localização tiver vínculos associados a outros cadastros ou restrições específicas, não poderá ser excluído [MSG1].
Inclusões	UC38 – Consultar Localização.
Pós-Condições	Dados da Localização excluídos do sistema.
Regras de Negócio	[RN1] A Localização só poderá ser excluída pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário. [RN2] A Localização não poderá ter nenhum vínculo associado ou restrições específicas.

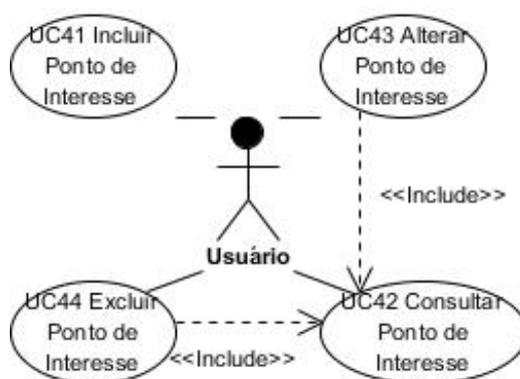


Figura B.11: Casos de Uso da funcionalidade Gerenciar Ponto de Interesse

Quadro B.41: Caso de Uso – Incluir Ponto de Interesse

UC41 – Incluir Ponto de Interesse	
Objetivo	Incluir um novo Ponto de Interesse.
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Ponto de Interesse; 2. O Ator seleciona a opção incluir; 3. O Ator insere os dados do Ponto de Interesse: porção de tempo, intervalo, status, nome; 4. O Ator solicita a confirmação dos dados [FA1]; 5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma o cadastro do Ponto de Interesse.
Fluxo Alternativo	1. [FA1] Desistência da inclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	Não possui.
Pós-Condições	Dados do Ponto de Interesse inseridos no sistema.
Regras de Negócio	[RN1] O Ponto de Interesse só poderá ser incluído pelos atores

	usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário.
--	--

Quadro B.42: Caso de Uso – Consultar Ponto de Interesse

UC42 – Consultar Ponto de Interesse	
Objetivo	Consultar um Ponto de Interesse cadastrado.
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Ponto de Interesse; 2. O Ator seleciona a opção consultar; 3. O sistema retorna os dados da consulta [FE1];
Fluxo Alternativo	Não possui.
Fluxo Exceção	1. [FE1] Nenhum Ponto de Interesse cadastrado [MSG1];
Inclusões	Não possui.
Pós-Condições	Dados do Ponto de Interesse consultados.
Regras de Negócio	[RN1] A consulta do Ponto de Interesse só poderá ser visualizada pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário.

Quadro B.43: Caso de Uso – Alterar Ponto de Interesse

UC43 – Alterar Ponto de Interesse	
Objetivo	Alterar um Ponto de Interesse cadastrado.
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Ponto de Interesse; 2. O Ator seleciona a opção alterar; 3. O Ator realiza consulta [UC42 – Consultar Ponto de Interesse]; 3. O Ator altera os dados do Ponto de Interesse: porção de tempo, intervalo, status, nome; 4. O Ator solicita a confirmação dos dados [FA1]; 5. O sistema efetua a validação dos campos [FE1]; 6. O sistema armazena os dados; 7. O sistema confirma a alteração do Ponto de Interesse.
Fluxo Alternativo	1. [FA1] Desistência da alteração e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Algum campo obrigatório possui valor inválido ou não foi preenchido [MSG2].
Inclusões	UC42 – Consultar Ponto de Interesse.
Pós-Condições	Dados do Ponto de Interesse atualizados.
Regras de Negócio	[RN1] O Ponto de Interesse só poderá ser alterado pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário.

Quadro B.44: Caso de Uso – Excluir Ponto de Interesse

UC44 – Excluir Ponto de Interesse	
Objetivo	Excluir um Ponto de Interesse cadastrado.
Atores	Usuário (Ex. Comandante, Combatente, Médico, Voluntário).
Pré-Condição	1. Ator com acesso ao sistema.
Fluxo Principal	1. O Ator seleciona a opção de cadastro de Ponto de Interesse; 2. O Ator seleciona a opção excluir; 3. O Ator realiza consulta [UC42 – Consultar Ponto de Interesse]; 4. O Ator exclui o Ponto de Interesse [MSG1] [FA1] [FE1]; 5. O sistema confirma a exclusão dos dados.
Fluxo Alternativo	1. [FA1] Desistência da exclusão e solicitação do cancelamento do processo [MSG1].
Fluxo Exceção	1. [FE1] Caso o Ponto de Interesse tiver vínculos associados a outros cadastros ou restrições específicas, não poderá ser excluído [MSG1].
Inclusões	UC42 – Consultar Ponto de Interesse.
Pós-Condições	Dados do Ponto de Interesse excluídos do sistema.
Regras de Negócio	[RN1] O Ponto de Interesse só poderá ser excluído pelos atores usuários, por exemplo, Comandante, Combatente, Médico ou Voluntário. [RN2] O Ponto de Interesse não poderá ter nenhum vínculo associado ou restrições específicas.

Quadro B.45: Mensagens de Casos de Uso

Mensagens	
[MSG1]	Mensagem de informação é emitida.
[MSG2]	Mensagem de correção é emitida.

APÊNDICE C – DIAGRAMAS DE SEQUÊNCIA PARA AS FUNCIONALIDADES DO JEMF

Este apêndice tem por objetivo apresentar os diagramas de sequências para os casos de uso descritos no Apêndice A para as operações de Incluir, Alterar, Consultar e Excluir das funcionalidades estabelecidas para o arcabouço JEMF.

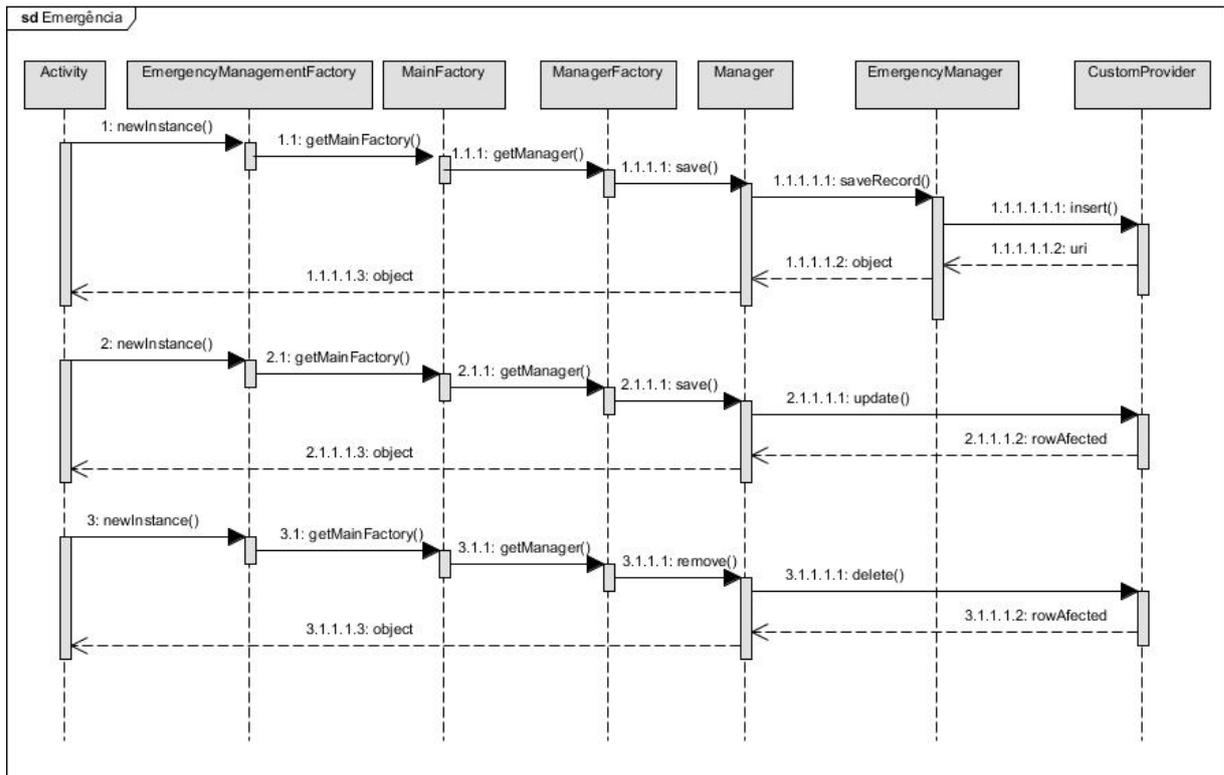


Figura C.1: Diagrama de sequência da funcionalidade Gerenciar Emergência (Parte 1)

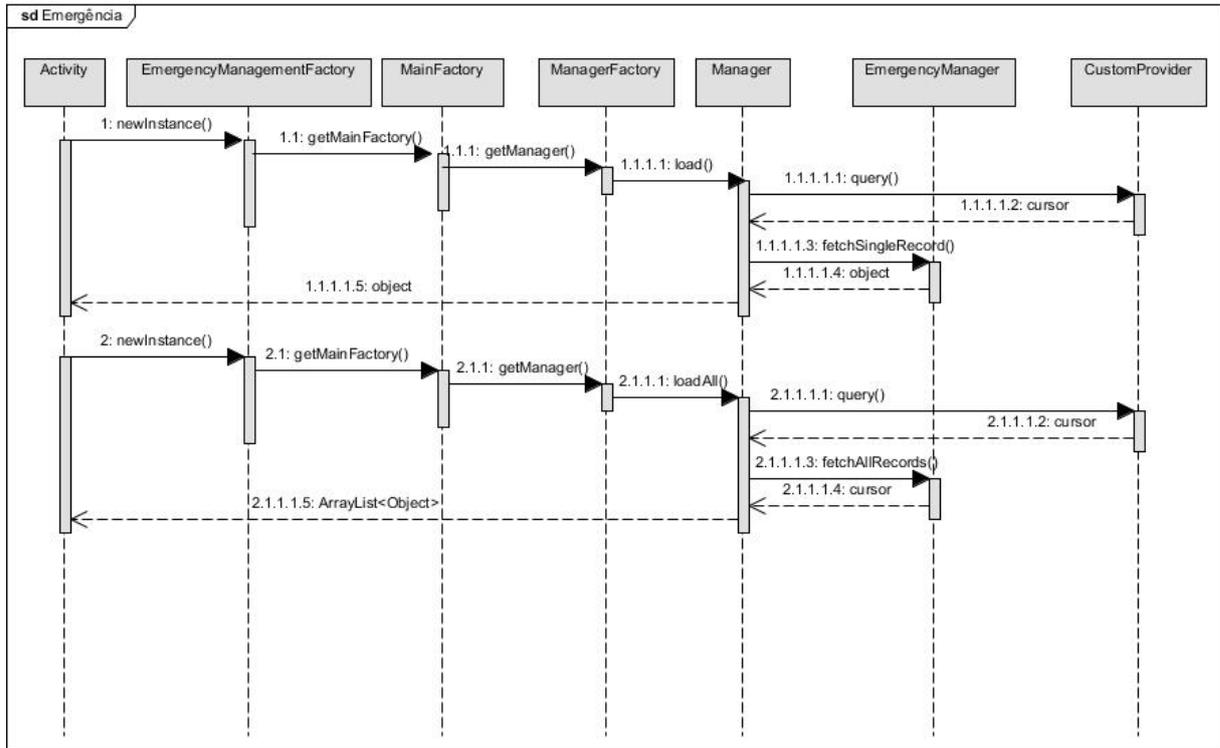


Figura C.2: Diagrama de sequência da funcionalidade Gerenciar Emergência (Parte 2)

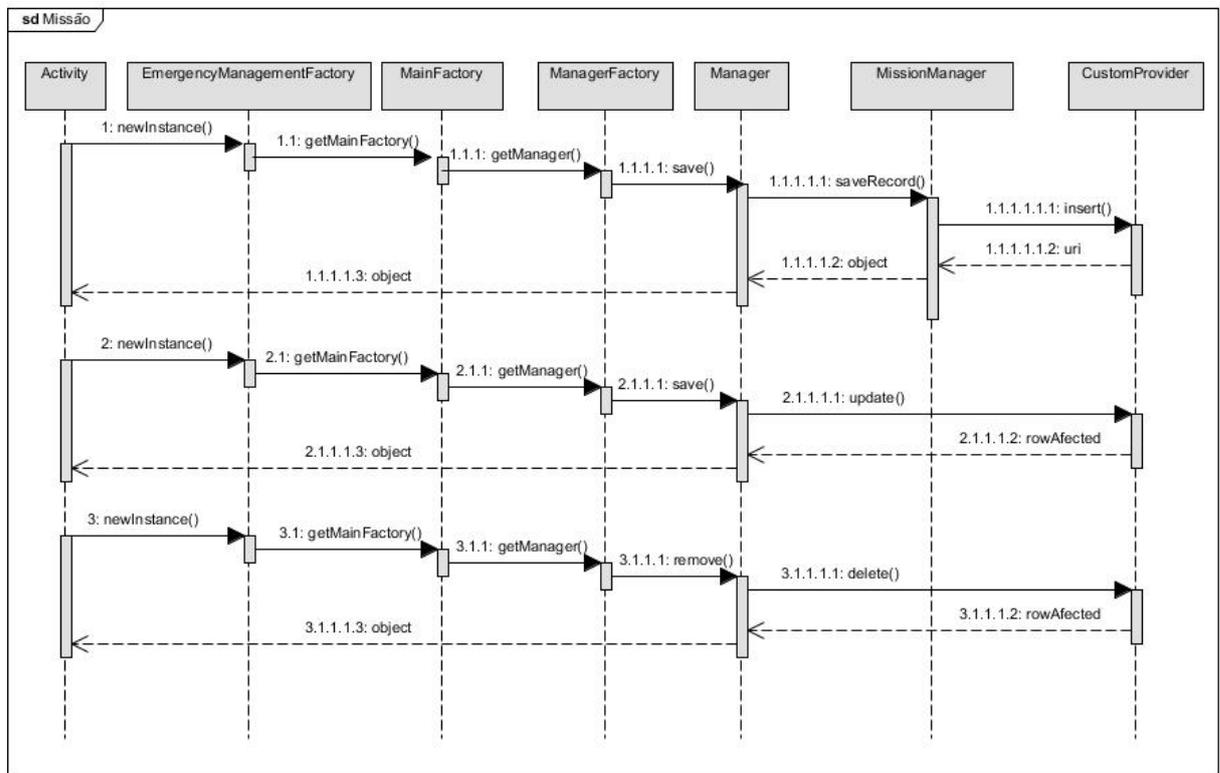


Figura C.3: Diagrama de sequência da funcionalidade Gerenciar Missão (Parte 1)

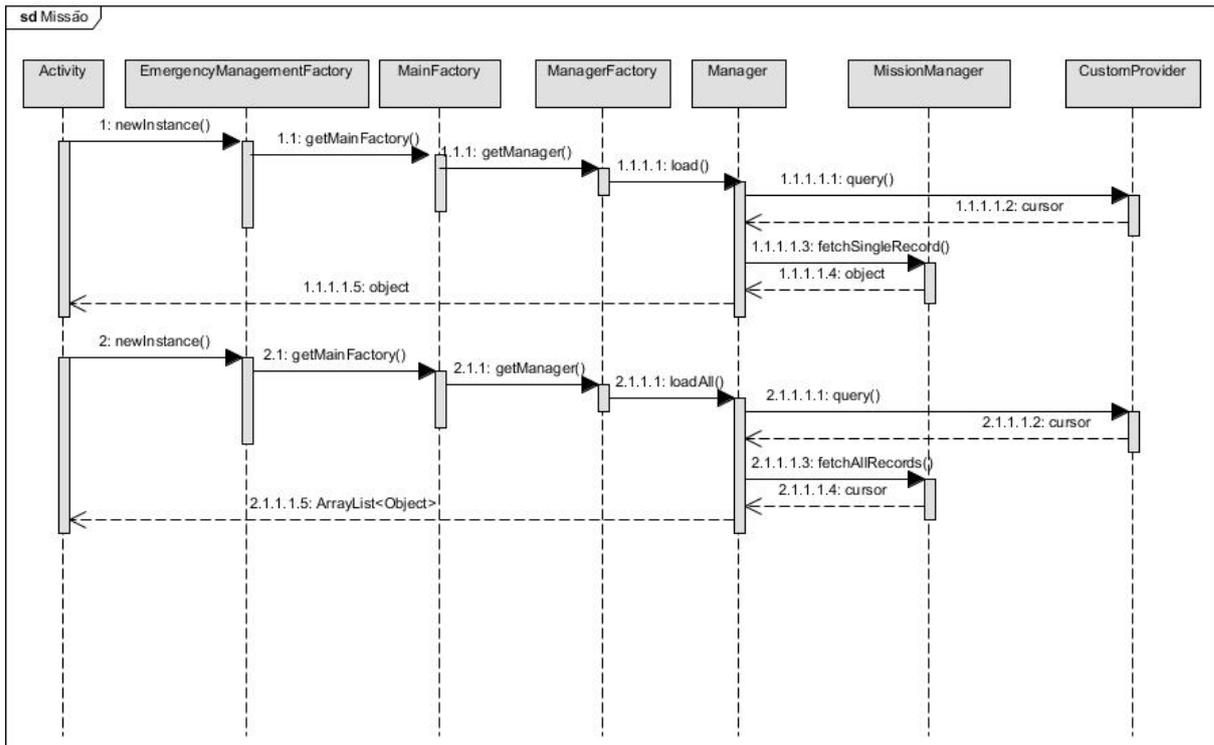


Figura C.4: Diagrama de sequência da funcionalidade Gerenciar Missão (Parte 2)

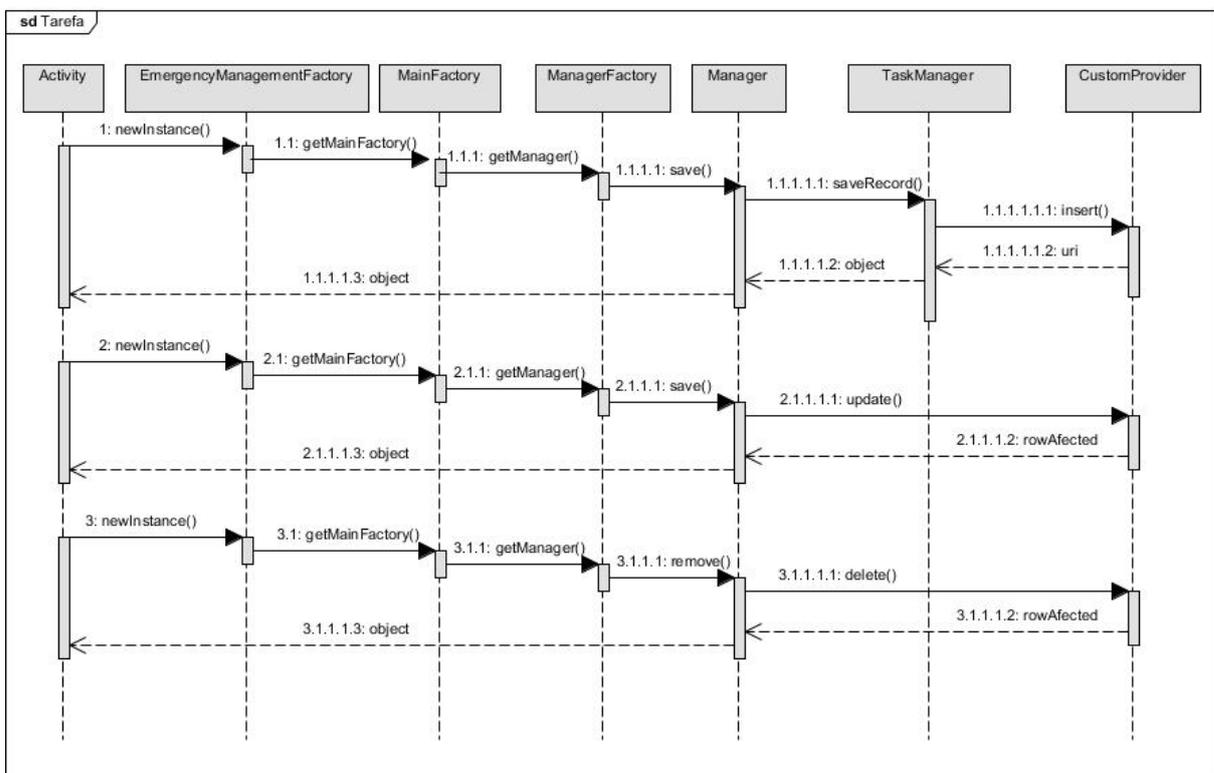


Figura C.5: Diagrama de sequência da funcionalidade Gerenciar Tarefa (Parte 1)

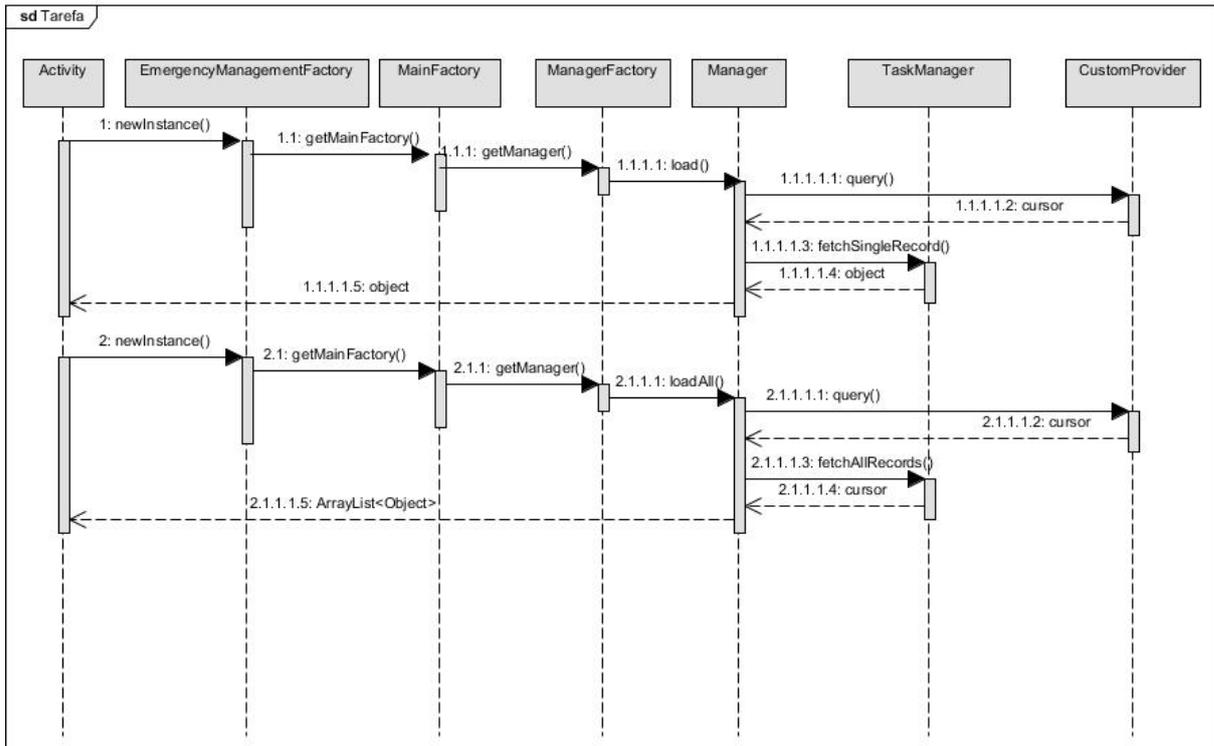


Figura C.6: Diagrama de sequência da funcionalidade Gerenciar Tarefa (Parte 2)

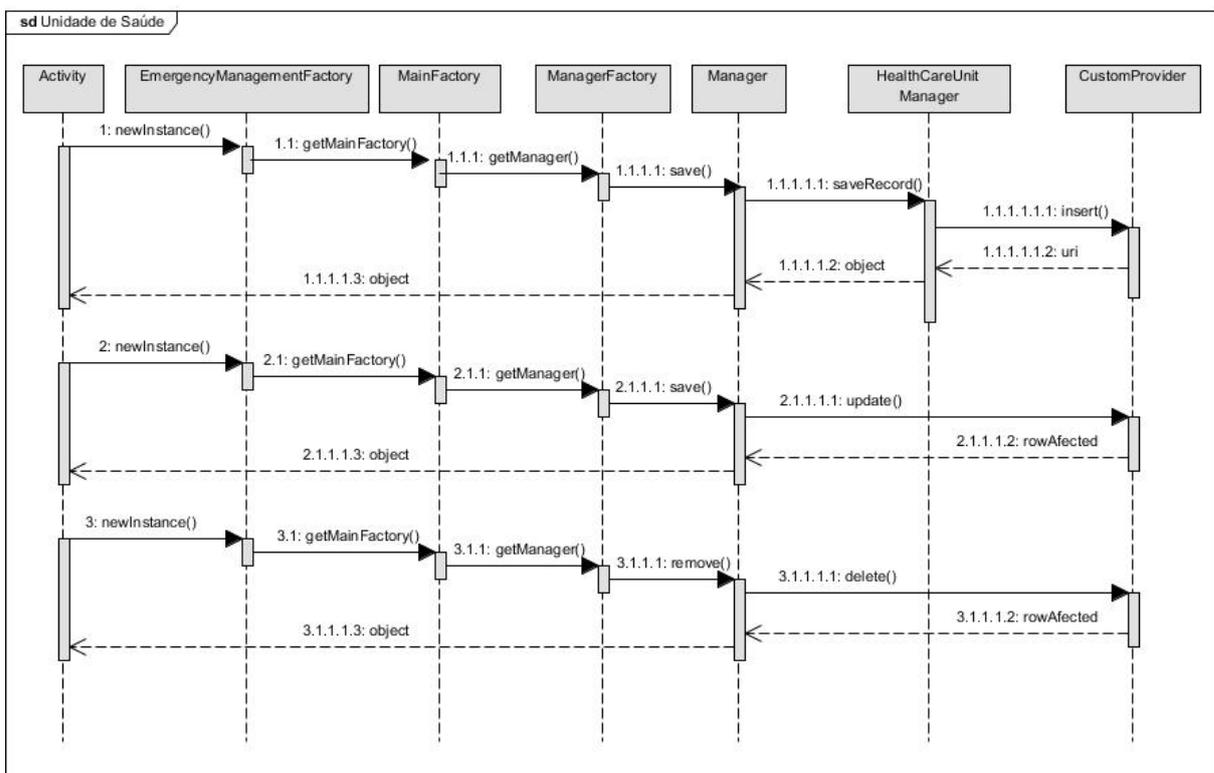


Figura C.7: Diagrama de sequência da funcionalidade Gerenciar Unidade de Saúde (Parte 1)

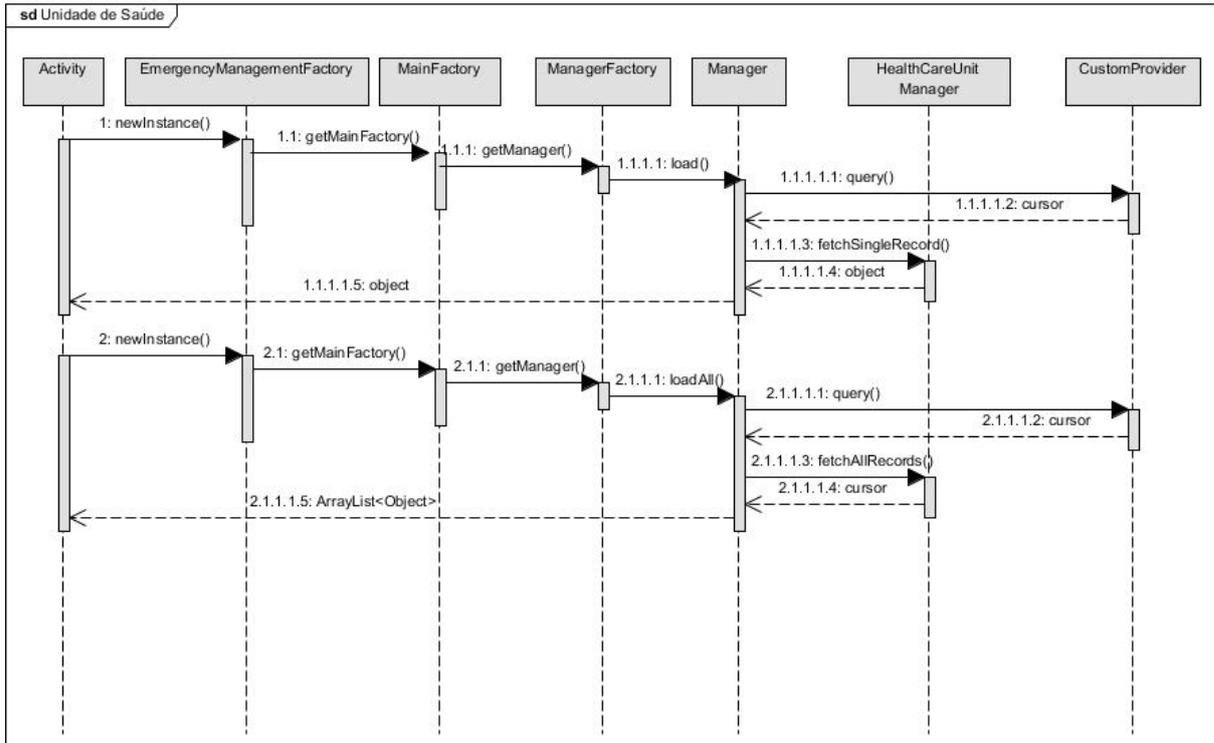


Figura C.8: Diagrama de sequência da funcionalidade Gerenciar Unidade de Saúde (Parte 2)

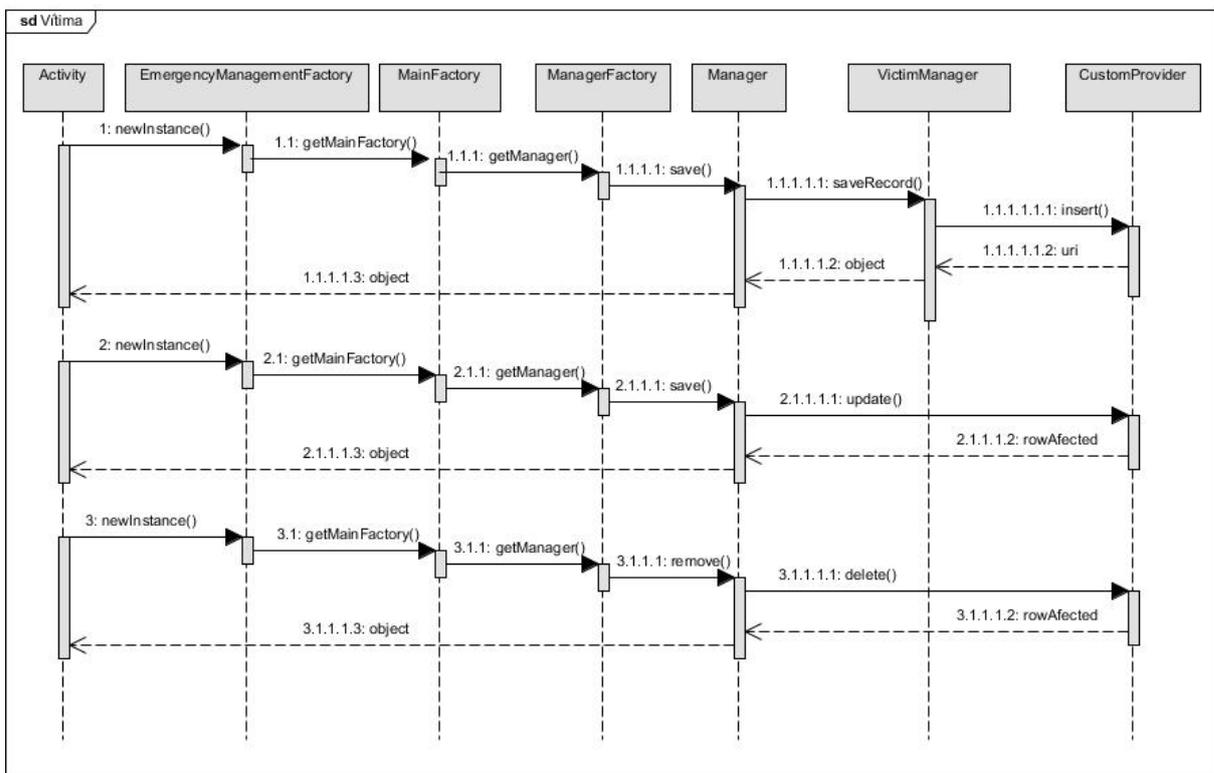


Figura C.9: Diagrama de sequência da funcionalidade Gerenciar Vítima (Parte 1)

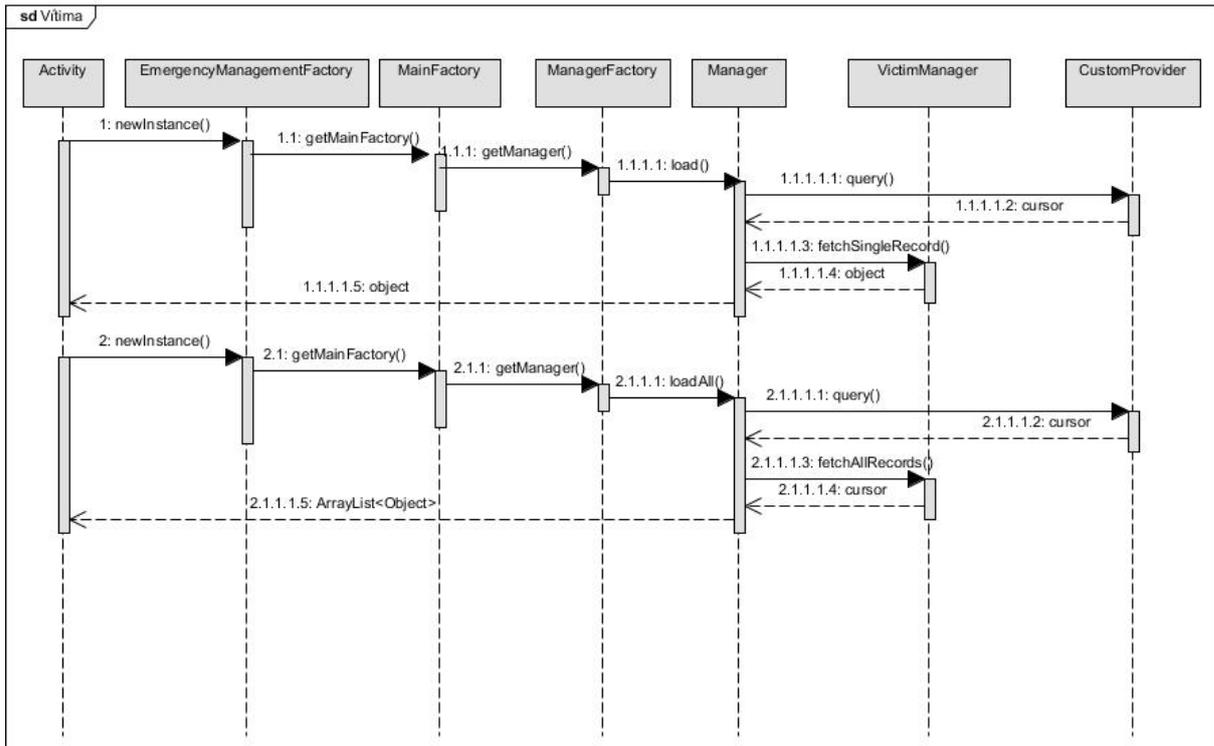


Figura C.10: Diagrama de seqüência da funcionalidade Gerenciar Vítima (Parte 2)

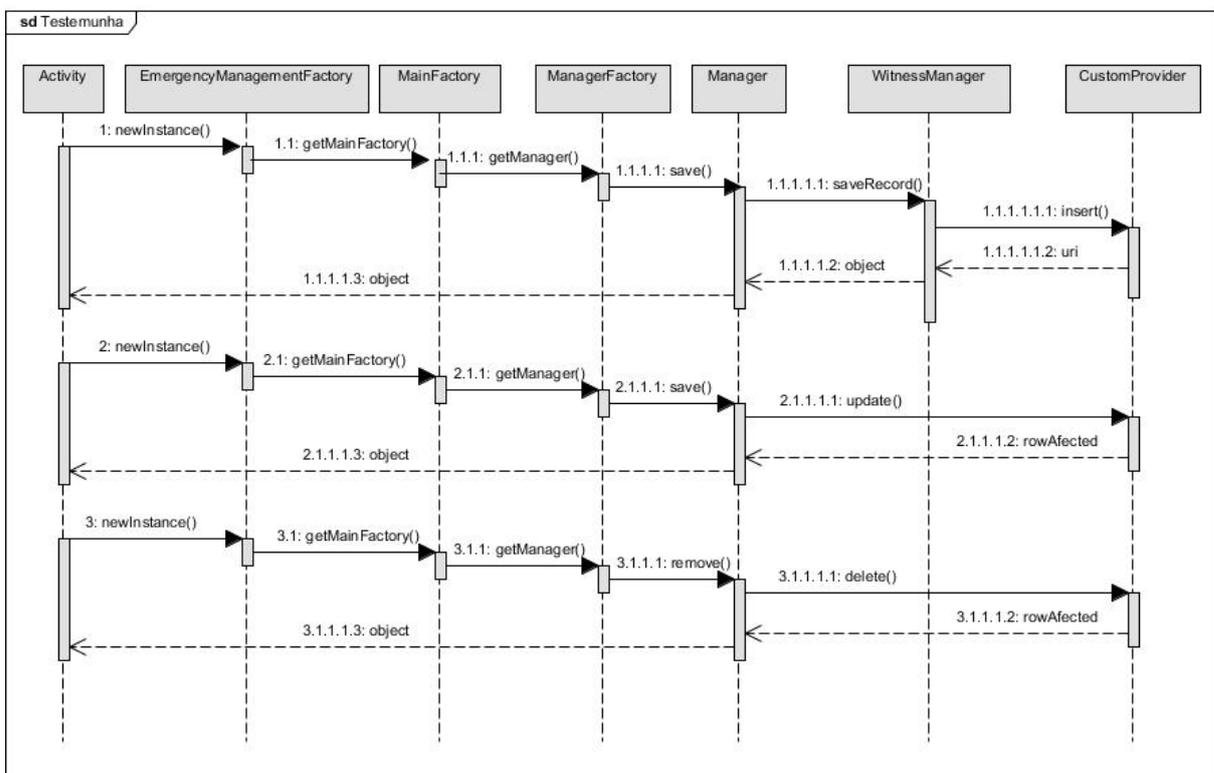


Figura C.11: Diagrama de seqüência da funcionalidade Gerenciar Testemunha (Parte 1)

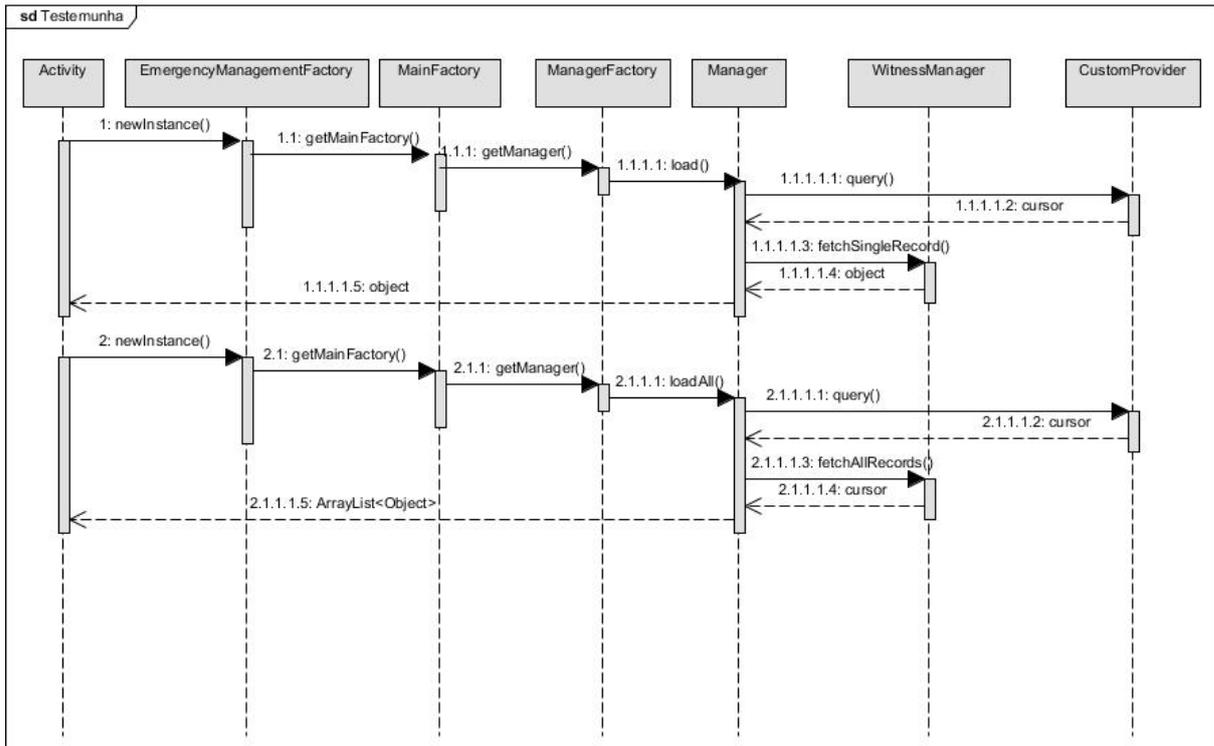


Figura C.12: Diagrama de sequência da funcionalidade Gerenciar Testemunha (Parte 2)

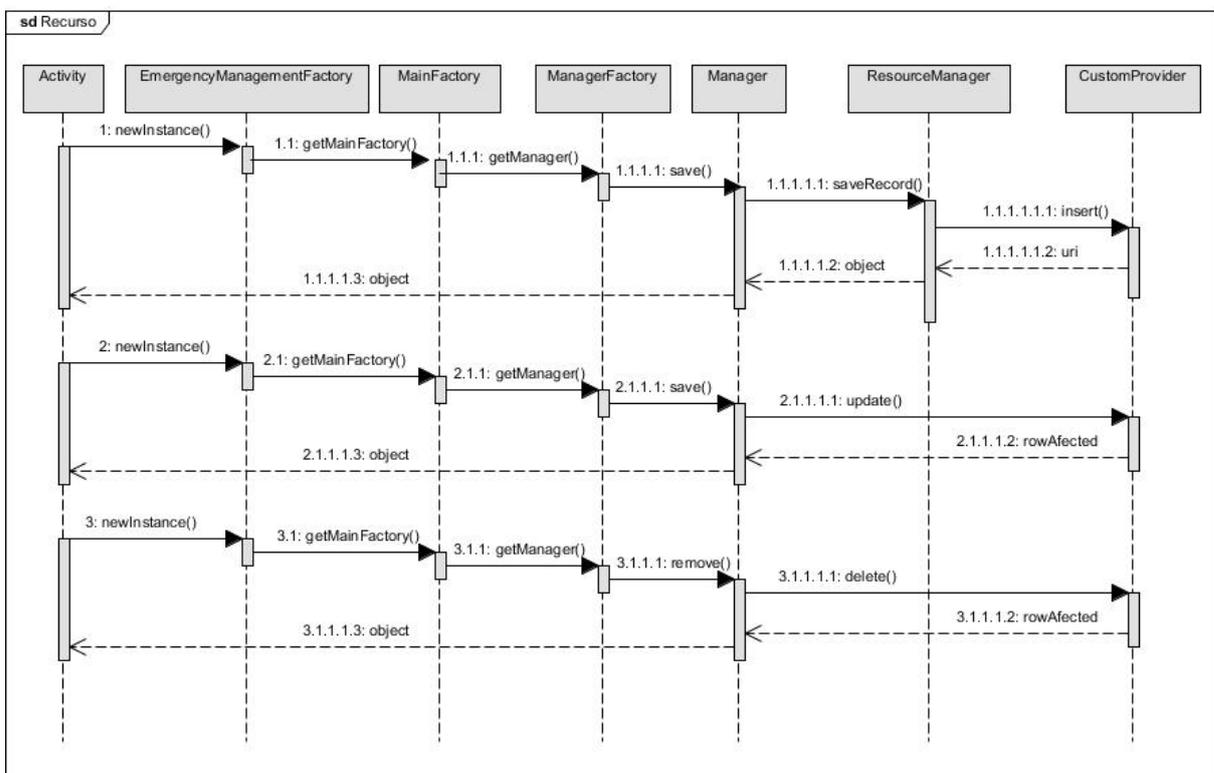


Figura C.13: Diagrama de sequência da funcionalidade Gerenciar Recurso (Parte 1)

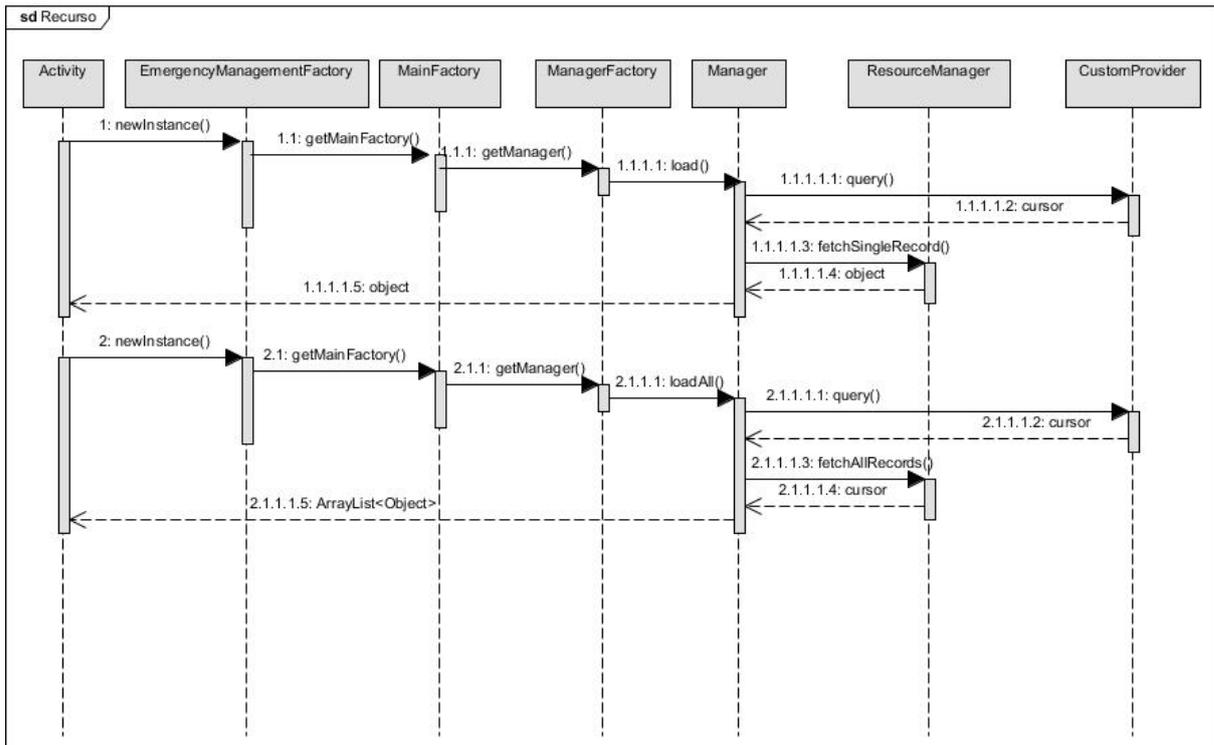


Figura C.14: Diagrama de sequência da funcionalidade Gerenciar Recurso (Parte 2)

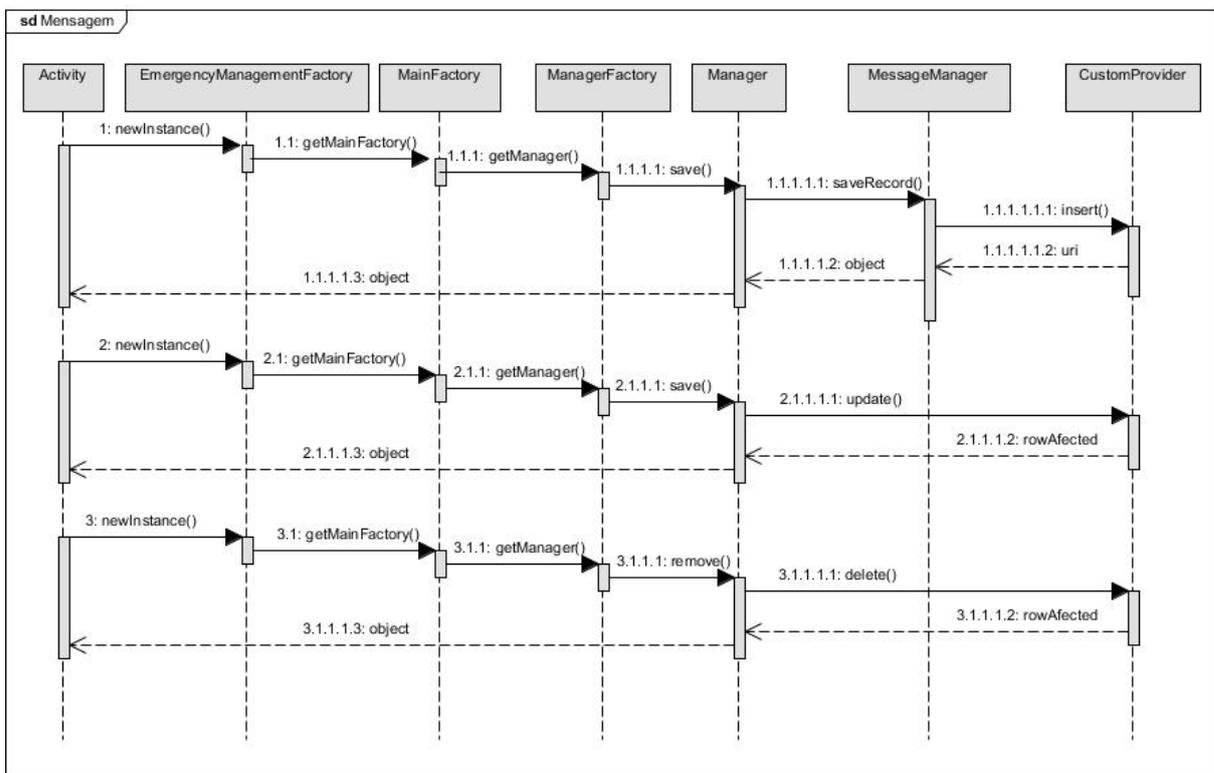


Figura C.15: Diagrama de sequência da funcionalidade Gerenciar Mensagem (Parte 1)

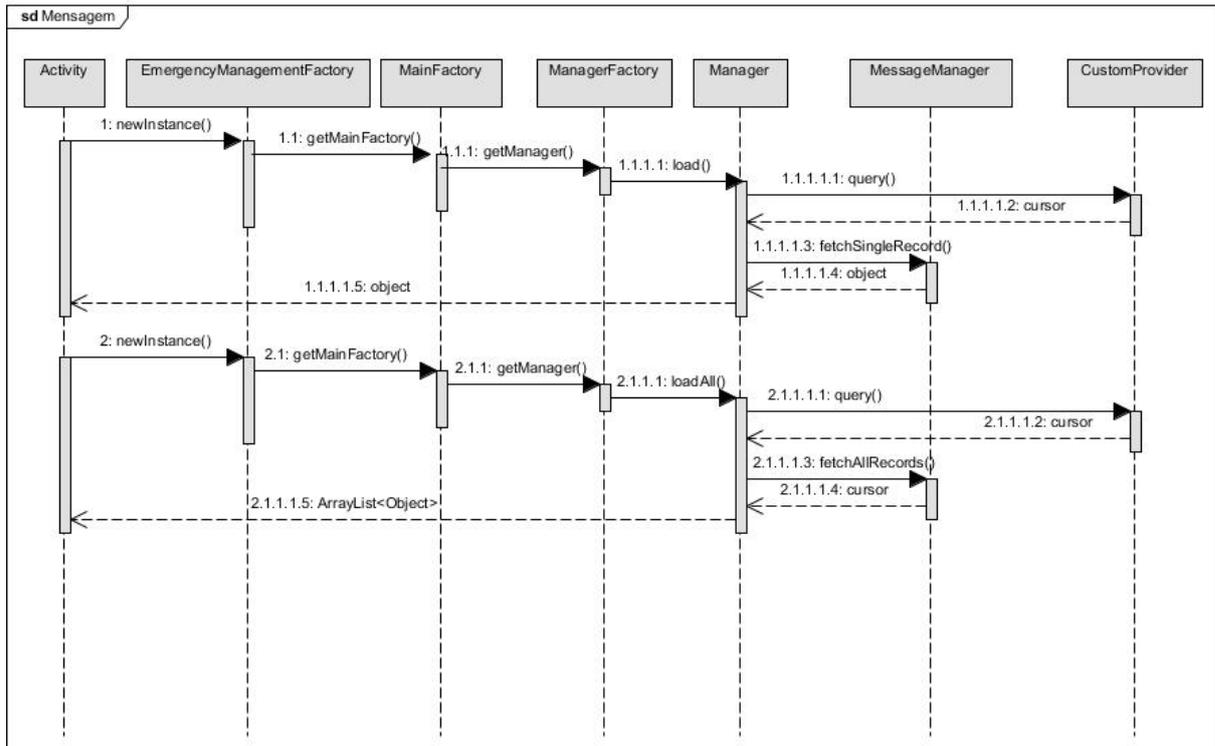


Figura C.16: Diagrama de sequência da funcionalidade Gerenciar Mensagem (Parte 2)

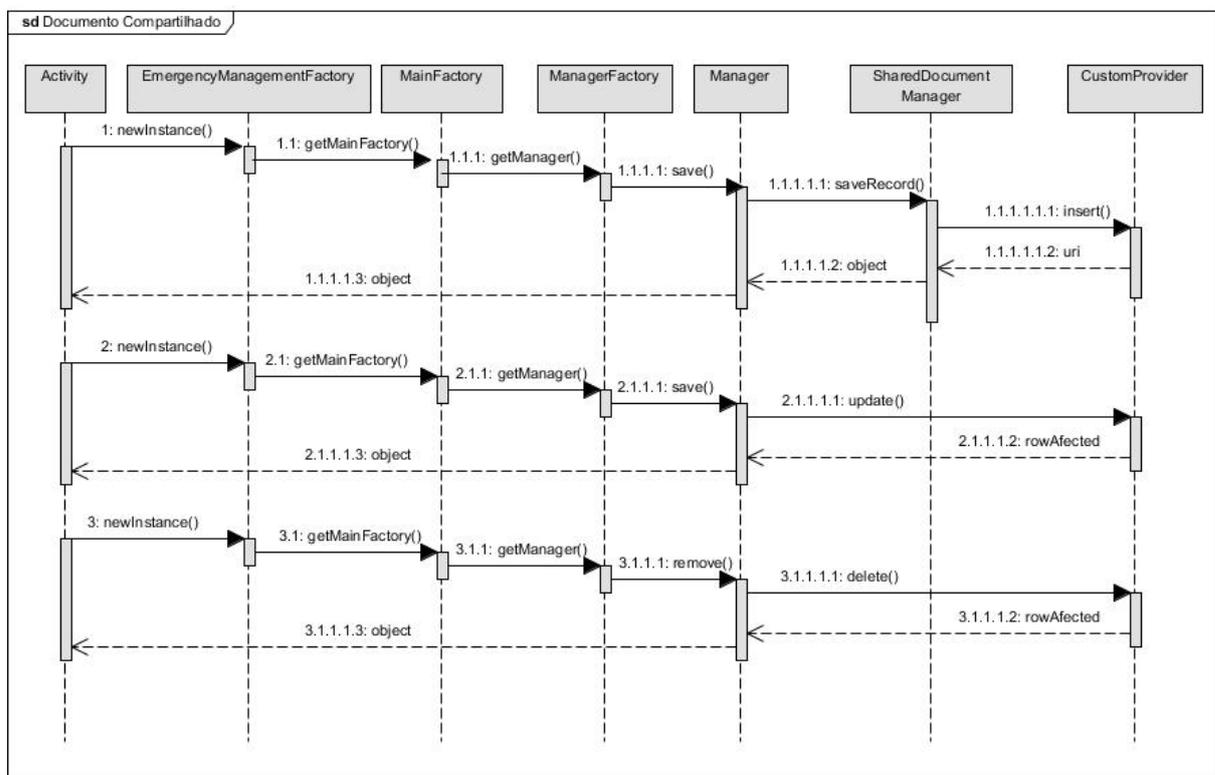


Figura C.17: Diagrama de sequência da funcionalidade Gerenciar Documento Compartilhado (Parte 1)

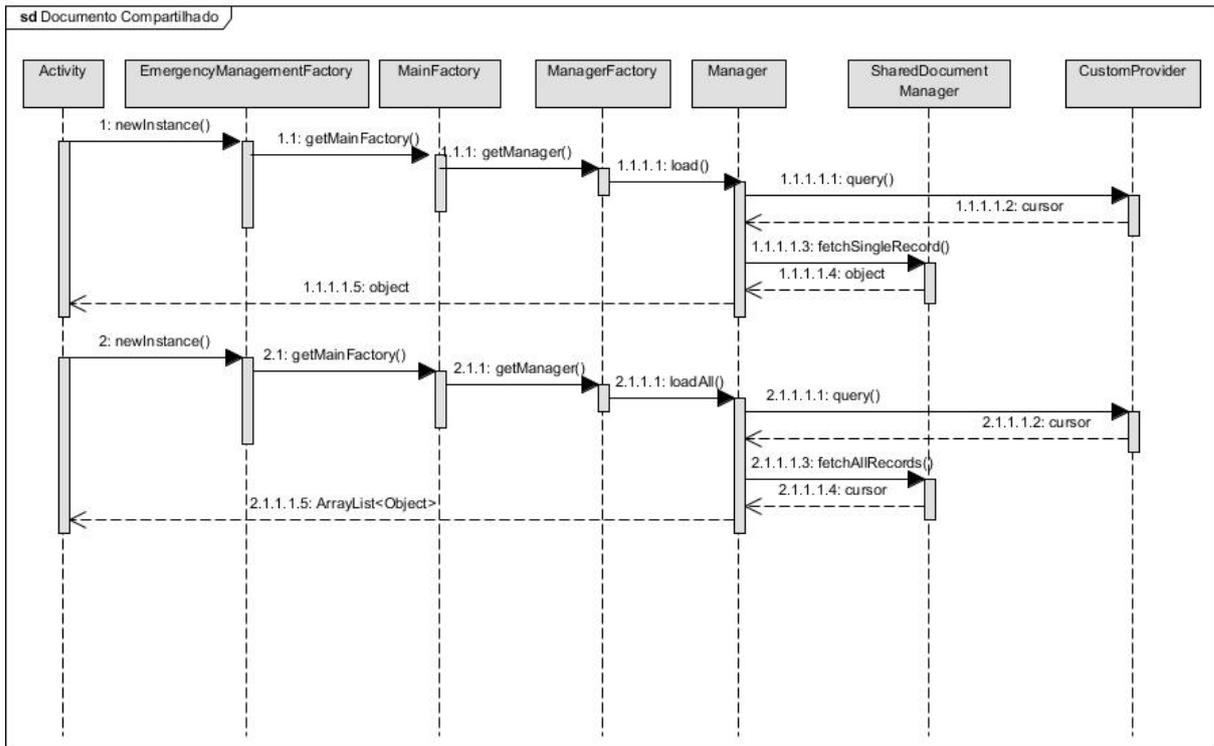


Figura C.18: Diagrama de sequência da funcionalidade Gerenciar Documento Compartilhado (Parte 2)

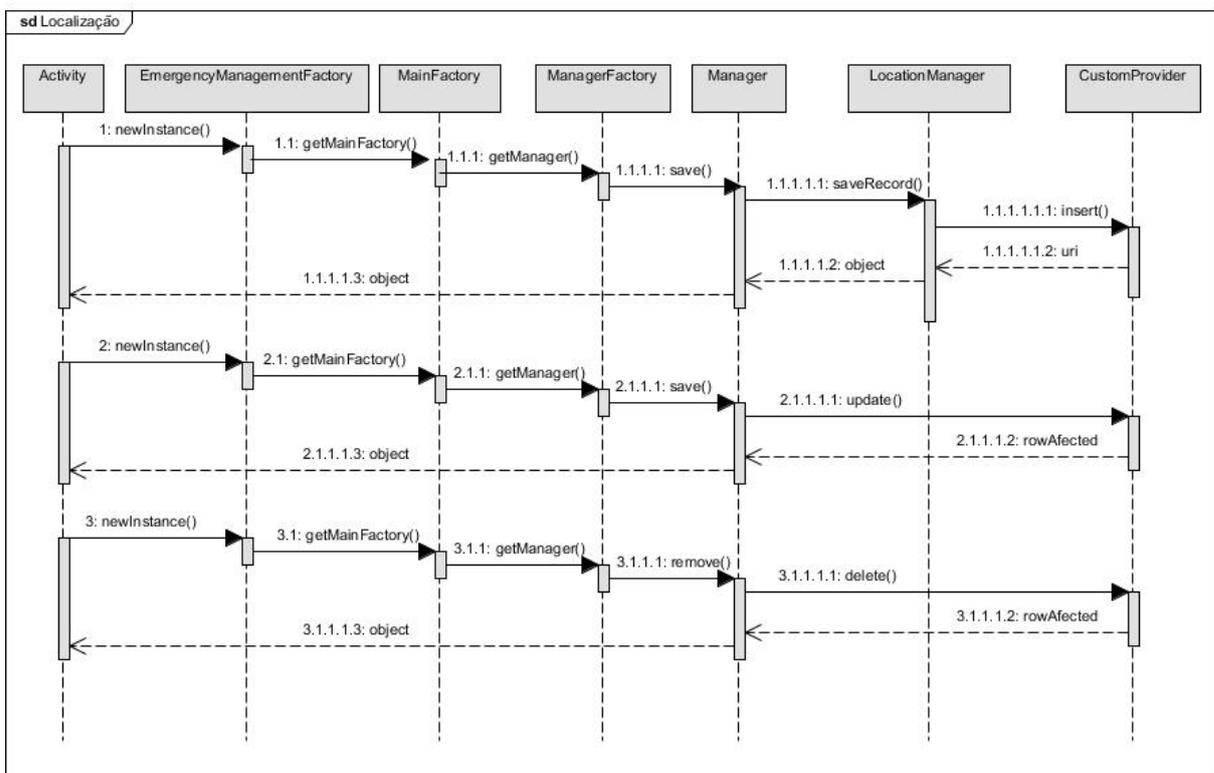


Figura C.19: Diagrama de sequência da funcionalidade Gerenciar Localização (Parte 1)

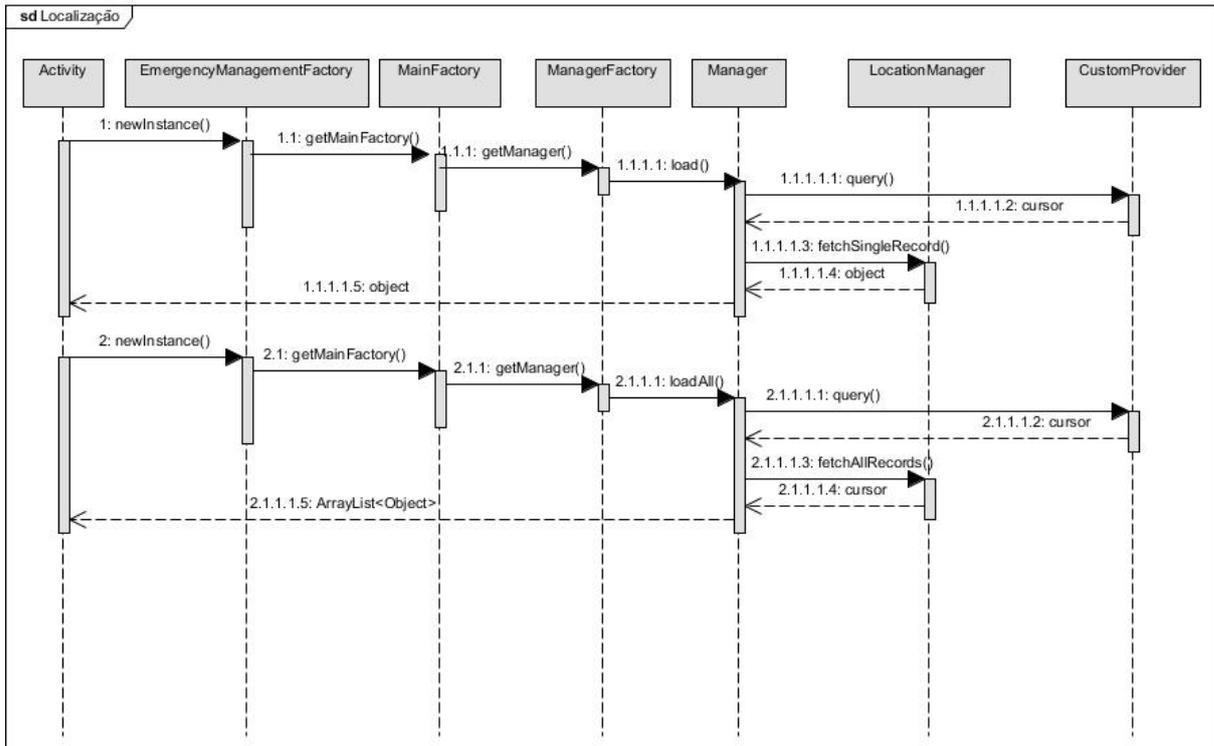


Figura C.20: Diagrama de seqüência da funcionalidade Gerenciar Localização (Parte 2)

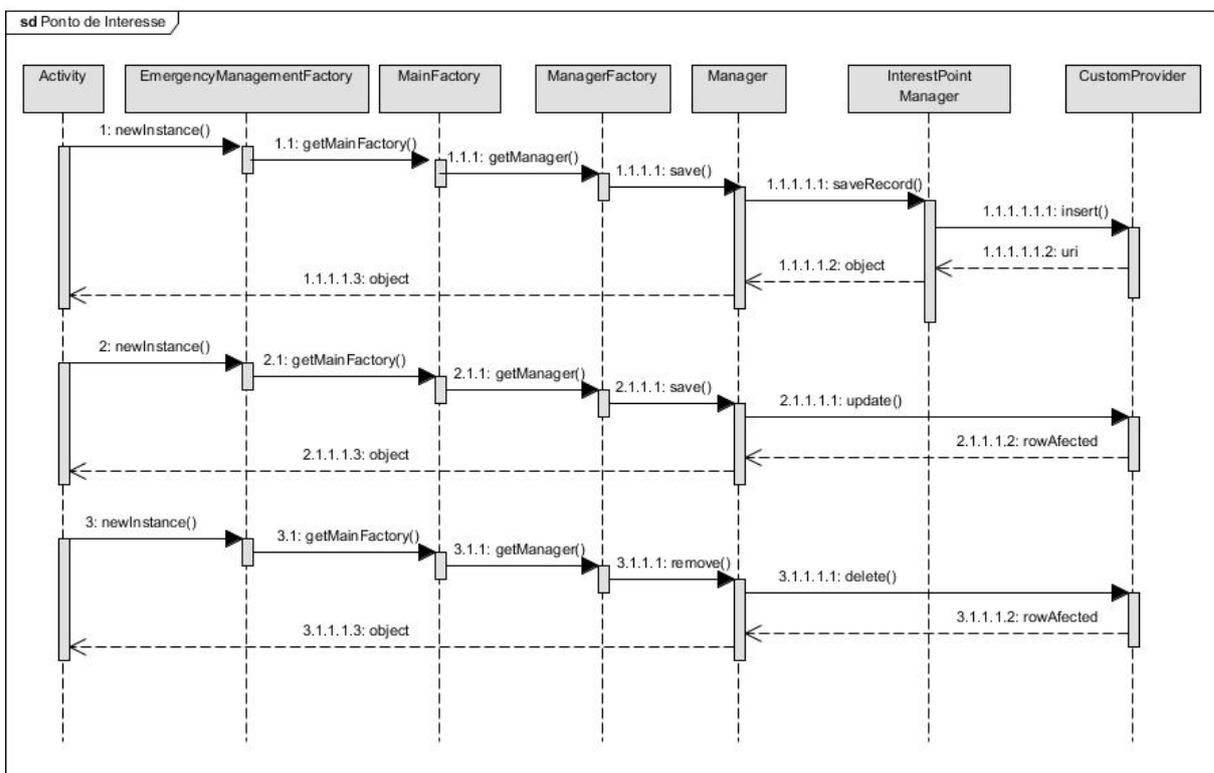


Figura C.21: Diagrama de seqüência da funcionalidade Gerenciar Ponto de Interesse (Parte 1)

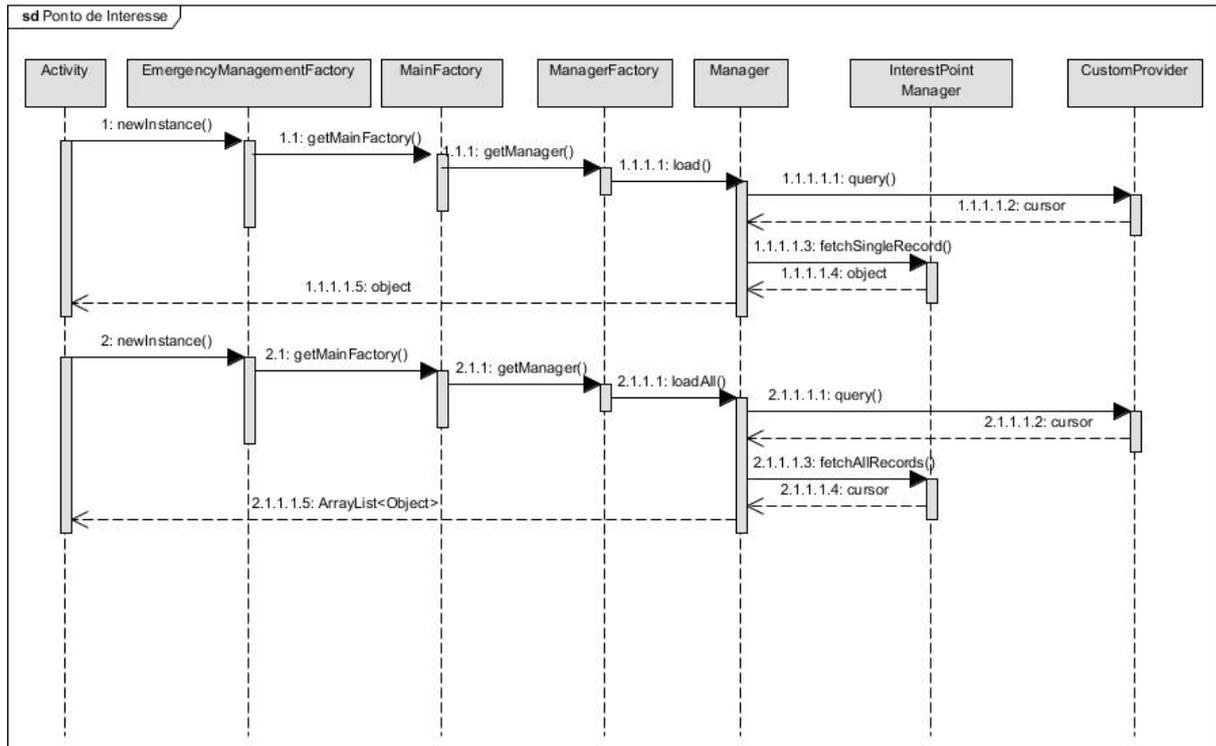


Figura C.22: Diagrama de sequência da funcionalidade Gerenciar Ponto de Interesse (Parte 2)

APÊNDICE D – DICIONÁRIO DE ATRIBUTOS DAS ENTIDADES DO MODELO DO BANCO DE DADOS

Este apêndice tem por objetivo apresentar os atributos das entidades do Modelo do Banco de Dados para o arcabouço proposto no Capítulo 5.

Quadro D.1: Entidade – Endereço

Entidade	Atributo	Valor	Descrição
address	_id	Integer	Chave Primária
	street	Text	Rua
	complement	Text	Complemento
	district	Text	Bairro
	city	Text	Cidade
	region	Text	Região
	country	Text	País
	postalcode	Text	CEP
idlocation	Integer	Identificador De Localização	

Quadro D.2: Entidade – Organização Afetada

Entidade	Atributo	Valor	Descrição
affected organization	_id	Integer	Chave Primária
	status	Text	Status
	idwitness	Integer	Identificador de Testemunha
	idinterestpoint	Integer	Identificador de Ponto de Interesse

Quadro D.3: Entidade – Comandante

Entidade	Atributo	Valor	Descrição
commander	_id	Integer	Chave Primária

Quadro D.4: Entidade – Contato

Entidade	Atributo	Valor	Descrição
contact	_id	Integer	Chave Primária
	identity	Integer	Identificador de Entidade
	phone	Text	Telefone
	email	Text	E-mail
	radio	Text	Rádio

	language	Text	Língua
--	----------	------	--------

Quadro D.5: Entidade – Emergência

Entidade	Atributo	Valor	Descrição
emergency	_id	Integer	Chave Primária
	name	Text	Nome
	activated	Integer	Ativado
	level	Text	Nível
	startdate	Text	Data de Início
	enddate	Text	Date de Término
	type	Text	Tipo

Quadro D.6: Entidade – Tipo de Emergência

Entidade	Atributo	Valor	Descrição
emergency type	type	Text	Chave Primária
	seq	Integer	Sequencial

Quadro D.7: Entidade – Entidade

Entidade	Atributo	Valor	Descrição
entity	_id	Integer	Chave Primária
	name	Text	Nome
	idlocation	Integer	Identificador de Localização

Quadro D.8: Entidade – Informação de Entidade

Entidade	Atributo	Valor	Descrição
entity_ information	identity	Integer	Chave Primária
	idinformation	Integer	Identificador de Informação

Quadro D.9: Entidade – Equipamento

Entidade	Atributo	Valor	Descrição
equipment	_id	Integer	Chave Primária

Quadro D.10: Entidade – Unidade de Saúde

Entidade	Atributo	Valor	Descrição
healthcareunit	_id	Integer	Chave Primária
	capacity	Text	Capacidade
	idcommander	Integer	Identificador de Comandante
	idmedical	Integer	Identificador de Médico
	idinterestpoint	Integer	Identificador de Ponto de Interesse

Quadro D.11: Entidade – Informação

Entidade	Atributo	Valor	Descrição
information	_id	Integer	Chave Primária
	creationdate	Text	Data de Criação
	lastmodificationdate	Text	Data de Última Modificação
	priority	Text	Prioridade
	idemergency	Integer	Identificador de Emergência

Quadro D.12: Entidade – Contato de Informação

Entidade	Atributo	Valor	Descrição
information_ contact	idinformation	Integer	Chave Primária
	idcontact	Integer	Identificador de Contato

Quadro D.13: Entidade – Ponto de Interesse

Entidade	Atributo	Valor	Descrição
interestpoint	_id	Integer	Chave Primária
	name	Text	Nome

Quadro D.14: Entidade – Língua

Entidade	Atributo	Valor	Descrição
language	language	Text	Chave Primária
	seq	Integer	Sequencial

Quadro D.15: Entidade – Nível

Entidade	Atributo	Valor	Descrição
level	level	Text	Chave Primária
	seq	Integer	Sequencial

Quadro D.16: Entidade – Localização

Entidade	Atributo	Valor	Descrição
location	_id	Integer	Chave Primária
	timestamp	Integer	Porção de Tempo
	interval	Integer	Intervalo
	status	Text	Status

Quadro D.17: Entidade – Médico

Entidade	Atributo	Valor	Descrição
medical	_id	Integer	Chave Primária

Quadro D.18: Entidade – Mensagem

Entidade	Atributo	Valor	Descrição
message	_id	Integer	Chave Primária
	subject	Text	Assunto
	content	Text	Conteúdo

Quadro D.19: Entidade – Missão

Entidade	Atributo	Valor	Descrição
mission	_id	Integer	Chave Primária

Quadro D.20: Entidade – Equipe de Missão

Entidade	Atributo	Valor	Descrição
mission_team	idmission	Integer	Chave Primária
	idteam	Integer	Identificador de Equipe

Quadro D.21: Entidade – Organização

Entidade	Atributo	Valor	Descrição
organization	_id	Integer	Chave Primária
	description	Text	Descrição
	level	Text	Nível

Quadro D.22: Entidade – Status da Organização

Entidade	Atributo	Valor	Descrição
organization status	status	Text	Chave Primária
	seq	Integer	Sequencial

Quadro D.23: Entidade – Pessoa

Entidade	Atributo	Valor	Descrição
person	_id	Integer	Chave Primária
	gender	Text	Gênero
	birthdate	Text	Data de Nascimento
	age	Integer	Idade

Quadro D.24: Entidade – Identificador de Local

Entidade	Atributo	Valor	Descrição
placeidentifier	_id	Integer	Chave Primária
	name	Text	Nome
	description	Text	Descrição
	type	Text	Tipo
	symbol	Text	Símbolo
	idlocation	Integer	Identificador de Localização

Quadro D.25: Entidade – Plano

Entidade	Atributo	Valor	Descrição
plan	_id	Integer	Chave Primária
	objective	Text	Objetivo
	risk	Text	Risco
	checkist	Text	Lista de Validação

Quadro D.26: Entidade – Posição

Entidade	Atributo	Valor	Descrição
position	_id	Integer	Chave Primária
	latitude	Integer	Latitude
	longitude	Integer	Longitude
	idlocation	Integer	Identificador de Localização

Quadro D.27: Entidade – Prioridade

Entidade	Atributo	Valor	Descrição
priority	priority	Text	Chave Primária
	seq	Integer	Sequencial

Quadro D.28: Entidade – Recurso

Entidade	Atributo	Valor	Descrição
resource	_id	Integer	Chave Primária
	name	Text	Nome
	status	Text	Status
	description	Text	Descrição

Quadro D.29: Entidade – Status do Recurso

Entidade	Atributo	Valor	Descrição
resourcestatus	status	Text	Chave Primária
	seq	Integer	Sequencial

Quadro D.30: Entidade – Respondedor

Entidade	Atributo	Valor	Descrição
responder	_id	Integer	Chave Primária

Quadro D.31: Entidade – Serviço

Entidade	Atributo	Valor	Descrição
service	_id	Integer	Chave Primária
	title	Text	Título
	description	Text	Descrição
	status	Text	Status
	startdate	Text	Data de Início
	enddate	Text	Date de Término
	priority	Text	Prioridade
	idemergency	Integer	Identificador de Emergência

Quadro D.32: Entidade – Informação de Serviço

Entidade	Atributo	Valor	Descrição
service_ information	idservice	Integer	Chave Primária
	idinformation	Integer	Identificador de Informação

Quadro D.33: Entidade – Recurso de Serviço

Entidade	Atributo	Valor	Descrição
service_ resource	idservice	Integer	Chave Primária
	idresource	Integer	Identificador de Recurso

Quadro D.34: Entidade – Status do Serviço

Entidade	Atributo	Valor	Descrição
servicestatus	status	Text	Chave Primária
	seq	Integer	Sequencial

Quadro D.35: Entidade – Documento Compartilhado

Entidade	Atributo	Valor	Descrição
shared document	_id	Integer	Chave Primária
	description	Text	Descrição
	observation	Text	Observação

Quadro D.36: Entidade – Tarefa

Entidade	Atributo	Valor	Descrição
task	_id	Integer	Chave Primária
	iduser	Integer	Identificador de Usuário

Quadro D.37: Entidade – Equipe

Entidade	Atributo	Valor	Descrição
team	_id	Integer	Chave Primária
	name	Text	Nome

Quadro D.38: Entidade – Usuário de Equipe

Entidade	Atributo	Valor	Descrição
team_user	idteam	Integer	Chave Primária
	idleader	Integer	Chave Primária

	idmember	Integer	Chave Primária
--	----------	---------	----------------

Quadro D.39: Entidade – Mensagem de Texto

Entidade	Atributo	Valor	Descrição
textmessage	_id	Integer	Chave Primária

Quadro D.40: Entidade – Usuário

Entidade	Atributo	Valor	Descrição
user	_id	Integer	Chave Primária
	login	Text	Login
	password	Text	Senha

Quadro D.41: Entidade – Vítima

Entidade	Atributo	Valor	Descrição
victim	_id	Integer	Chave Primária
	status	Text	Status
	idaffectedorganization	Integer	Identificador de Organiz. Afetada
	idhealthcareunit	Integer	Identificador de Localização
	idwitness	Integer	Identificador de Testemunha

Quadro D.42: Entidade – Status da Vítima

Entidade	Atributo	Valor	Descrição
victimstatus	status	Text	Chave Primária
	seq	Integer	Sequencial

Quadro D.43: Entidade – Voluntário

Entidade	Atributo	Valor	Descrição
volunteer	_id	Integer	Chave Primária

Quadro D.44: Entidade – Testemunha

Entidade	Atributo	Valor	Descrição
witness	_id	Integer	Chave Primária

APÊNDICE E – QUESTIONÁRIOS PARA A AVALIAÇÃO DO JEMF

Este apêndice tem por objetivo discorrer os questionários e os exercícios estabelecidos para a avaliação do arcabouço JEMF. Os questionários podem ser divididos de acordo com o seu propósito.

Os três primeiros questionários são para a avaliação do arcabouço proposto por desenvolvedores terceiros:

- Questionário de Caracterização do Participante;
- Questionário de Avaliação dos Exercícios do Arcabouço;
- Questionário de Avaliação Geral do Arcabouço.

O último questionário é para a avaliação do arcabouço proposto por especialistas do domínio de Gestão de Emergências:

- Questionário de Avaliação por Especialista do Domínio.

Questionário de Caracterização do Participante

Este questionário tem por objetivo obter a caracterização do participante que avaliará o arcabouço JEMF. Por favor, informe seu perfil de acordo com os seguintes critérios:

1) Escolaridade – Qual o seu nível de escolaridade (Completo ou Incompleto)?

Graduação Especialização Mestrado Doutorado Pós-Doutorado

2) Experiência OO – Qual a sua experiência em programação Orientada a Objetos (OO)?

Nenhuma 1 a 6 meses 6 a 12 meses 12 a 36 meses Mais de 36 meses

3) Experiência Java – Qual a sua experiência em programação na linguagem Java?

Nenhuma 1 a 6 meses 6 a 12 meses 12 a 36 meses Mais de 36 meses

4) Experiência DM – Qual a sua experiência em programação para Dispositivos Móveis (Smartphone / Tablet)?

Nenhuma 1 a 6 meses 6 a 12 meses 12 a 36 meses Mais de 36 meses

5) Plataforma DM – Qual a plataforma de Dispositivos Móveis você possui experiência?

Nenhuma Android Windows iOS Outra, _____

6) Quais ambientes de desenvolvimento (IDE) ou ferramentas (toolkits, frameworks, bibliotecas, APIs) para dispositivos móveis você possui experiência?

7) Experiência GE – Qual a sua experiência em estudos ou trabalhos relacionados com o domínio de Gestão de Emergências?

Nenhuma 1 a 6 meses 6 a 12 meses 12 a 36 meses Mais de 36 meses

8) Qual foi seu último projeto de desenvolvimento para dispositivos móveis? Relate brevemente sua experiência no projeto e os principais desafios.

Exercícios de Avaliação do Arcabouço

Este documento tem por objetivo aplicar um conjunto de exercícios propostos para avaliação do arcabouço JEMF junto ao participante.

1. Objetivo Principal

Desenvolver um sistema móvel com o uso do arcabouço JEMF a partir das especificações detalhadas neste documento.

2. Especificação do Sistema

O sistema móvel proposto para este exercício é baseado nas especificações estabelecidas no trabalho realizado por Fernanda Araújo para o sistema móvel chamado “Sistema de Emergência” (Araújo, 2012). Portanto, as especificações serão apresentadas de forma simplificada para contextualizar o participante no domínio de Gestão de Emergências e nos exercícios de avaliação.

2.1. Descrição do Sistema

O Sistema de Emergência é um sistema para o tratamento de resposta a emergência com o uso de dispositivos móveis. Ele é destinado exclusivamente para o contexto do Corpo de Bombeiros do Estado do Rio de Janeiro (CBMERJ). No caso do Rio de Janeiro, em emergências urbanas, os bombeiros são os principais responsáveis pela operação de resposta à emergência. Mas eles podem ser assistidos por outras instituições, como a polícia militar.

O sistema visa complementar o trabalho dos agentes no registro e troca de informações entre posto de comando, que poderá estar equipado com um computador (desktop), e equipe de operação, equipada com dispositivos móveis atuais (smartphones ou tablets), conforme exemplificado na Figura E.1. Este posto de comando precisa receber informações, das equipes que trabalham na operação, sobre o estado atual da situação e os resultados de suas atividades. Ele precisa tomar conhecimento sobre os acontecimentos e as ações desempenhadas pelas equipes. As equipes de operação podem também, eventualmente, receber informações quanto às tomadas de decisão pelo posto de comando. O comandante da equipe é quem fica recebendo as informações no posto de comando. Então, o ideal é que os agentes de combate menos atarefados, por exemplo, de segunda e terceira linha da hierarquia do Corpo de Bombeiros (Figura E.2), necessitam utilizar os dispositivos móveis para imputar todos os dados relevantes relativos à emergência.

O cenário escolhido considera que as equipes trabalharão em emergências de média escala, onde o objetivo geral do combate à emergência é estabilizar a situação, evitando perdas humanas e materiais. Nesta cidade, em geral, o clima é tropical por se tratar de uma região litorânea. Então, o clima da região merece atenção especial quanto às variações do vento, intensidade do sol e com o volume de chuvas. Com relação às chuvas, a cidade tem um grande histórico de emergências nas últimas décadas devido às consequências mais evidentes desse fenômeno, as enchentes e os desabamentos de terra. Dessa forma, a área de trabalho das equipes de socorro pode compreender todo o local onde ocorreu a emergência, sendo preciso ter informações sobre o estado atual da área atingida. Dentro desta área a maior prioridade dos bombeiros é buscar e salvar vítimas. Isso compreende a localização todas as pessoas, o

resgate delas e a realização do atendimento médico com o uso de suprimentos, como medicamentos ou alimentos, se necessário. Além disso, podem ser exigidas avaliações de infraestruturas afetadas, como prédios e casas, para evitar maiores riscos e outros incidentes.

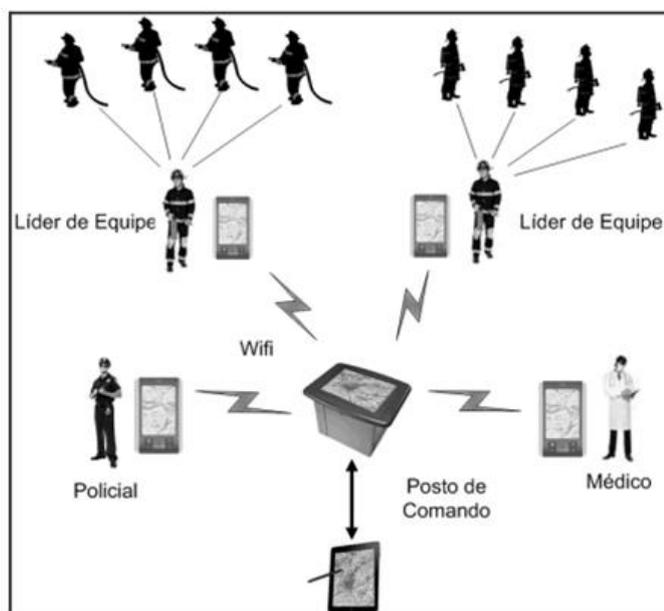


Figura E.1: Cenário de uso para equipes de operação com dispositivos móveis (Engelbrecht et al., 2011)

Oficiais Superiores				
Coronel	Tenente-coronel	Major		
Oficial Intermediário		Oficiais Subalternos		
Capitão		1º Tenente	2º Tenente	
Praça especial		Praça		
Aspirante		Subtenente		
Praças graduados				
1º Sargento	2º Sargento	3º Sargento	Cabo	Soldado

Figura E.2: Hierarquia do Corpo de Bombeiros (Araujo, 2012)

2.2. Requisitos e Modelagem do Sistema

Diante das características descritas sobre o sistema, é apresentado um esboço dos requisitos por meio de uma lista de necessidades para apoiar o propósito desta avaliação. Além disso, é exposto um diagrama de classes para contribuir com o entendimento e visualização dos dados e entidades do domínio de emergências.

Os requisitos deste sistema são:

- a) O sistema deve possuir uma tela de emergências já cadastradas e a opção de cadastrar novas emergências, editar ou excluir emergências registradas;
- b) O sistema deve possuir uma tela de vítimas já cadastradas e a opção de cadastrar novas vítimas, editar ou excluir vítimas registradas;
- c) O sistema deve possuir uma tela de infraestruturas afetadas já cadastradas e a opção de cadastrar novas infraestruturas afetadas, editar ou excluir infraestruturas registradas;
- d) O sistema deve possuir uma tela de socorristas já cadastrados e a opção de cadastrar novos socorristas, editar ou excluir socorristas registrados;
- e) O sistema deve possuir uma tela de suprimentos já cadastrados e a opção de cadastrar novos suprimentos, editar ou excluir suprimentos registrados;
- f) A seleção das opções será feita através da navegação em lista vertical dos registros cadastrados, utilizando o toque dos dedos na tela;
- g) O sistema deve atender emergências de média escala, como combate a incêndios, busca e salvamento, socorro de emergência em via pública, socorro florestal e meio-ambiente, etc.;
- h) O sistema deve ser desenvolvido para a plataforma Android.

Os casos de uso deste sistema são:

Quadro E.1 – Casos de uso do sistema proposto.

Entidade	Atributos
Incluir Emergência; Editar Emergência; Excluir Emergência; Carregar Emergência.	Nome, data, tipo, local, condições climáticas, observações.
Incluir Vítima; Editar Vítima; Excluir Vítima; Carregar Vítima.	Nome, idade, data, estado de saúde, observações.
Incluir Infraestrutura Afetada; Editar Infraestrutura Afetada; Excluir Infraestrutura Afetada; Carregar Infraestrutura Afetada.	Nome, tipo, grau do dano, observações.
Incluir Socorrista; Editar Socorrista; Excluir Socorrista; Carregar Socorrista.	Nome, idade, função, observações.
Incluir Suprimento; Editar Suprimento; Excluir Suprimento; Carregar Suprimento.	Nome, descrição, data de validade.

Diagrama de classes do sistema proposto:

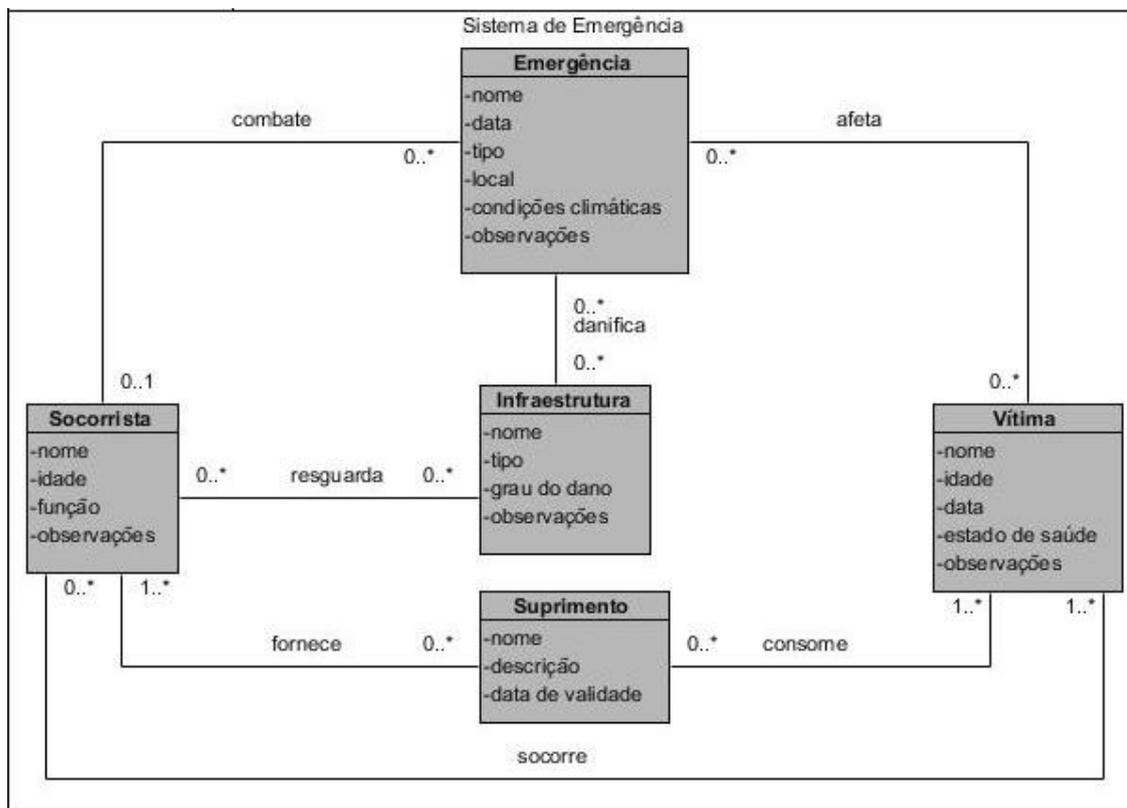


Figura E.3: Diagrama de classes para o Sistema de Emergência

3. Exercícios

Os exercícios de avaliação do arcabouço JEMF foram divididos em três etapas. Cada etapa foca no desenvolvimento de aspectos diferentes em relação ao uso do arcabouço: reusabilidade, flexibilidade e extensibilidade. Dessa forma, espera-se que o participante possa evoluir no desenvolvimento do sistema móvel proposto nesta avaliação. O manual disponibilizado com o arcabouço contribui para orientação dos participantes ao fornecer uma documentação detalhada das suas principais funcionalidades.

3.1. Etapa 1 – Reusabilidade

Na primeira etapa o objetivo principal é a instanciação do arcabouço JEMF para o sistema proposto. Espera-se obter a experiência do desenvolvedor quanto ao uso do arcabouço e seus primeiros passos.

Exercício (A): Crie um novo projeto de aplicativo para a plataforma Android.

Recomenda-se o uso do ambiente de desenvolvimento (IDE) Eclipse com o plug-in ADT. Para maiores detalhes de instalação da IDE, vide o manual do arcabouço. Crie um novo Workspace no Eclipse para uma melhor organização da avaliação. Este projeto será denominado “*Sistema de Emergência*” e deverá ser configurado para uma versão do Android igual ou superior a 4.0.3 (API Level 15).

Exercício (B): Importe o arcabouço JEMF para o Workspace e conecte-o ao projeto de aplicativo.

Uma vez criado o projeto de aplicativo, é necessário realizar a importação do arcabouço JEMF para o mesmo Workspace no Eclipse. No manual do arcabouço é definido o endereço para fazer o download do arcabouço e outros detalhes de importação.

Exercício (C): Implemente uma funcionalidade especificada para o Sistema de Emergência.

Na seção anterior (2.2) foram especificadas quatro funcionalidades para o Sistema de Emergência: Cadastro de Emergência, Cadastro de Vítima, Cadastro de Socorrista e Cadastro de Infraestrutura Afetada. O manual do arcabouço apresenta as suas principais funcionalidades e as classes que o compõem para criar uma instância do JEMF no projeto de aplicativo. Dentre essas funcionalidades do arcabouço, é possível fazer uma correlação com as funcionalidades para este sistema, ou seja, uma vez encontrado determinada funcionalidade já disponibilizado pelo arcabouço, poderá ser reutilizada pelo novo sistema, uma vez que a funcionalidade já foi implementada.

Implemente a funcionalidade para o cadastro de emergências com o uso do arcabouço JEMF. Recomenda-se que seja adicionado um componente *ListView* para a *Activity* principal do sistema, pois condiz com o requisito estabelecido neste documento e o arcabouço está preparado para trabalhar com esse componente. Dessa maneira, é possível o sistema fazer o carregamento de dados do banco de dados através do arcabouço.

3.2. Etapa 2 – Flexibilidade

Na segunda etapa o objetivo principal é a flexibilidade do arcabouço JEMF para o sistema proposto. Espera-se obter a experiência do desenvolvedor quanto à manipulação das funções e propriedades do arcabouço.

Exercício (D): Implemente as principais operações para o cadastro de emergências.

Uma vez instanciado o arcabouço JEMF no sistema proposto, é necessário implementar as principais operações especificadas para o cadastro de emergências: inclusão, edição, exclusão de dados. Recomenda-se analisar as classes que o compõem o arcabouço para executar o acionamento apropriado dessas operações, vide o manual.

Exercício (E): Implemente alterações de propriedades para o cadastro de emergências.

Após implementar as operações para o cadastro de emergências, é preciso realizar algumas alterações de propriedades para atender aos requisitos do sistema proposto. Ao analisar e confrontar os atributos do arcabouço com os atributos descritos neste documento percebe-se que alguns deles deverão ser adicionados no cadastro de emergências, por exemplo, condições climáticas e observações, que foram estabelecidos exclusivamente para o sistema e não estão presentes nos atributos do arcabouço. Recomenda-se analisar as classes que o compõem o arcabouço para a funcionalidade que gerencia as emergências e realizar as alterações apropriadas para atender os novos atributos, vide o manual.

3.3. Etapa 3 – Extensibilidade

Na terceira etapa o objetivo principal é a extensão do arcabouço JEMF para o sistema proposto. Espera-se obter a experiência do desenvolvedor quanto à criação de novas funcionalidades para o arcabouço.

Exercício (F): **Implemente uma nova funcionalidade.**

Ao analisar os requisitos do sistema proposto e as funcionalidades do arcabouço JEMF, é possível notar que o arcabouço não disponibiliza em sua versão atual uma opção para o cadastro de suprimentos. Apesar disso, o arcabouço já apresenta a entidade para recursos materiais (“*Resource*”), mas não há uma especialização para atender o requisito descrito neste documento. Sendo assim, é preciso realizar a adição dessa nova funcionalidade no arcabouço através da herança da classe de recursos materiais.

Implemente esta nova funcionalidade com a codificação de classes a partir da herança dos componentes existentes no arcabouço, como a classe entidade de para o pacote Núcleo e as classes do pacote Móvel.

- Pacote Núcleo
 - Criar a classe “*AbstractSupply*”, representando a entidade abstrata de suprimento.
- Pacote Móvel
 - Criar a classe “*Supply*”, representando a entidade concreta de suprimento (herança de recurso – “*Resource*”);
 - Criar a classe “*SupplyManager*” para a criação das operações de incluir, alterar e excluir específicas para a funcionalidade suprimento;
 - Criar a classe “*SupplyAdapter*” e “*SupplyLoader*” para a criação da operação de carregamento específico para a funcionalidade suprimento;
 - Criar a classe “*SupplyView*” para definição dos componentes de layout para a funcionalidade suprimento;
 - Alterar as classes “*DBStatement*” e “*DBHelper*” para a criação da tabela no banco de dados para a funcionalidade suprimento;
 - Alterar as classes “*ProviderStatement*” e “*CustomProvider*” para a criação do acesso ao banco de dados pela funcionalidade suprimento;
 - Alterar as classes “*ManagerFactory*”, “*AdapterFactory*”, “*FeatureHolder*” para o acesso das operações criadas neste exercício pelo aplicativo proposto.

Dica: Caso necessário, analise o código do arcabouço para as funcionalidades já disponibilizadas para melhor entendimento sobre a estrutura e os relacionamentos entre as suas principais classes.

Referências

Araujo, F. C. S. **Um framework para desenvolvimento de sistemas em dispositivos móveis no apoio a gestão de emergência.** (Dissertação de Mestrado) -- Programa de Pos-Graduação em Informática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil, 2012.

Engelbrecht, A.; Borges, M. R. S.; Vivacqua, A. S. Digital tabletops for situational awareness in emergency situations. In: **International Conference on Computer Supported Cooperative Work in Design (CSCWD)**. Los Alamitos: IEEE Computer Press, 15., v. 1, 2011. p. 669-676.

Questionário de Avaliação dos Exercícios do Arcabouço

Este questionário tem por objetivo obter uma avaliação dos exercícios após o participante desenvolver o sistema móvel – Sistema de Emergência – com o uso do arcabouço JEMF. Por favor, avalie de acordo com os seguintes critérios:

1) Exercício (A) – Quão fácil é executar o exercício (A)?

muito fácil fácil médio difícil muito difícil

2) Exercício (B) – Quão fácil é executar o exercício (B)?

muito fácil fácil médio difícil muito difícil

3) Exercício (C) – Quão fácil é executar o exercício (C)?

muito fácil fácil médio difícil muito difícil

4) Exercício (D) – Quão fácil é executar o exercício (D)?

muito fácil fácil médio difícil muito difícil

5) Exercício (E) – Quão fácil é executar o exercício (E)?

muito fácil fácil médio difícil muito difícil

6) Exercício (F) – Quão fácil é executar o exercício (F)?

muito fácil fácil médio difícil muito difícil

Questionário de Avaliação Geral do Arcabouço

Este questionário tem por objetivo obter uma avaliação geral após o participante utilizar o arcabouço JEMF. Por favor, avalie de acordo com os seguintes critérios:

1) Facilidade de uso – No geral, quão fácil é utilizar o JEMF?

() muito fácil () fácil () médio () difícil () muito difícil

2) Aprendizagem – Quão fácil é desenvolver um novo sistema móvel para o domínio de Gestão de Emergência?

() muito fácil () fácil () médio () difícil () muito difícil

3) Documentação – Quão fácil é buscar alguma informação na documentação do arcabouço no momento em que você precisou?

() muito fácil () fácil () médio () difícil () muito difícil

4) Você acha que faltou alguma funcionalidade no apoio à Gestão de Emergência na versão atual do arcabouço? Comente.

5) Relate as dificuldades ou os problemas encontrados durante todo o período de avaliação do arcabouço.

6) Você gostaria de acrescentar alguma sugestão, reclamação, etc.?

Questionário de Avaliação por Especialista do Domínio

Este questionário tem por objetivo obter uma avaliação geral do arcabouço JEMF por especialistas do domínio de Gestão de Emergências após a leitura do artigo:

MACHADO, M. F. T. et al. JEMF: A framework for the development of mobile systems for emergency management. In: N. Baloian et al. (Eds.), Collaboration and Technology. Berlin Heidelberg: Springer International Publishing, 2014. p. 239-254. (LNCS 8658).

Por favor, avalie de acordo com os seguintes critérios:

1) A abordagem proposta fornece um apoio claro e sistemático para o desenvolvimento de sistemas móveis para o domínio de Gestão de Emergências?

concordo totalmente concordo Não concordo nem discordo discordo discordo totalmente

2) O arcabouço pode ser aplicável em novos projetos para dispositivos móveis na prática?

concordo totalmente concordo Não concordo nem discordo discordo discordo totalmente

3) O arcabouço pode ser útil (gerando benefícios) para um novo projeto na prática?

concordo totalmente concordo Não concordo nem discordo discordo discordo totalmente

4) O arcabouço pode ser adaptável e personalizável para organizações com configurações diferentes (atividade principal, tamanho, cultura) de resposta à emergências?

concordo totalmente concordo Não concordo nem discordo discordo discordo totalmente

5) Você pode citar e explicar brevemente qualquer deficiência que você observou no arcabouço proposto?
