

**FAI – CENTRO DE ENSINO SUPERIOR EM GESTÃO TECNOLOGIA
E EDUCAÇÃO**

CURSO DE SISTEMAS DE INFORMAÇÃO

**ÂNGELA FONSECA FARIA
CAROLINE RIBEIRO OLIVEIRA
MARCOS ANTÔNIO DE SOUSA
RODRIGO MANSUR GONÇALVES**

**SAFEWALK - SISTEMA DE SOFTWARE PARA AUXÍLIO NA
ESCOLHA DE PERCURSOS PÚBLICOS**

SANTA RITA DO SAPUCAÍ - MG

2019

**FAI – CENTRO DE ENSINO SUPERIOR EM GESTÃO TECNOLOGIA
E EDUCAÇÃO**

CURSO DE SISTEMAS DE INFORMAÇÃO

**ÂNGELA FONSECA FARIA
CAROLINE RIBEIRO OLIBEIRA
MARCOS ANTÔNIO DE SOUSA
RODRIGO MANSUR GONÇALVES**

**SAFEWALK - SISTEMA DE SOFTWARE PARA AUXÍLIO NA
ESCOLHA DE PERCUSOS PÚBLICOS**

Projeto de final de curso apresentado à FAI – Centro de Ensino Superior em Gestão, Tecnologia e Educação, como requisito parcial para obtenção do Título de Bacharel em Sistemas de Informação, sob a orientação da profa. Eunice Gomes de Siqueira.

SANTA RITA DO SAPUCAÍ - MG

2019

FOLHA DE APROVAÇÃO

HISTÓRICO DE REVISÕES

Data	Versão	Autor	Descrição
10/12/19	2.10	Ângela Fonseca Faria	Versão corrigida.
15/11/19	2.9	Ângela Fonseca Faria	Correção da Fase 4.
07/11/19	2.8	Eunice G. Siqueira	Correção parcial da Fase 4.
18/10/19	2.7	Ângela Fonseca	Correção da Fase 3, incremento do Capítulo 6, adição do Capítulo 8 e complemento da Conclusão.
07/10/19	2.6	Eunice G. Siqueira	Correção parcial das Fases 1, 2 e 3.
15/08/19	2.5	Todos os membros da equipe	Complemento da Introdução e Conclusão.
12/08/19	2.4	Ângela Fonseca	Elaboração do Capítulo 7.
09/08/19	2.3	Caroline Ribeiro	Adição de subseções no Capítulo 6.
07/08/19	2.2	Ângela Fonseca	Incremento de subseções no Capítulo 3,4 e 5.
08/06/19	2.1	Todos os membros da equipe	Correção parcial da Fase 2.
23/05/19	2.0	Caroline Ribeiro	Elaboração do Capítulo 6.
20/05/19	1.9	Todos os membros da equipe	Revisão dos Capítulos 4 e 5.
08/05/19	1.8	Eunice G. Siqueira	Correção parcial da Fase 1.
10/04/19	1.7	Todos os membros da equipe	Correção da documentação da Fase 1.
28/03/19	1.6	Todos os membros da equipe	Revisão do documento para entrega.
10/03/19	1.5	Marcos Antônio	Elaboração do Capítulo 5.
08/03/19	1.4	Ângela Fonseca	Elaboração do Capítulo 4.
08/03/19	1.3	Rodrigo Mansur	Elaboração do Capítulo 3.
07/03/19	1.2	Caroline Ribeiro	Elaboração do Capítulo 2.
07/03/19	1.1	Ângela Fonseca	Elaboração do Capítulo 1.
05/03/19	1.0	Ângela Fonseca	Criação do documento base.

RESUMO

A violência nas vias urbanas é uma questão de grande preocupação da sociedade, pois afeta a vida da maioria das pessoas. Suas principais causas vão desde a exclusão social até a corrupção e o tráfico de drogas. Este documento apresenta o Projeto *SafeWalk* que visou desenvolver um sistema de software para propiciar um ambiente que possa contribuir com a segurança da população, especialmente, durante percursos em vias públicas. Assim, a metodologia adotada para desenvolver o projeto foi a realização da pesquisa bibliográfica, gerência com referência ao guia de conhecimentos do *Project Management Institute* (PMI), especificação e modelagem baseada na *Unified Modeling Language* (UML), construção e testes de software orientado a objetos. Como resultado do projeto, todos os objetivos foram atendidos conforme foram definidos e o sistema está pronto para avaliação e uso das partes interessadas.

Palavras-chave: Segurança pública. Vida urbana. Aplicação Java Web. Sistemas de Informação.

LISTA DE FIGURAS

FIGURA 1 - Tela do aplicativo Malalai	20
FIGURA 2 - Detalhes do aplicativo Malalai	21
FIGURA 3 - Tela do aplicativo bSafe	22
FIGURA 4 - Tela do aplicativo Circle of 6	23
FIGURA 5 - Gráfico da violência no período entre 2014 e 2016 no Brasil.....	26
FIGURA 6 - Matriz Poder X Interesse	29
FIGURA 7 - Modelo Incremental	30
FIGURA 8 - Visão geral do gerenciamento do escopo do projeto – PMBOK – 5ª. ed.....	34
FIGURA 9 - Visão geral do gerenciamento do tempo do projeto – PMBOK – 5ª. ed.....	36
FIGURA 10 - Visão geral do gerenciamento da integração do projeto – PMBOK – 5ª. ed.	40
FIGURA 11 - Visão geral do gerenciamento da qualidade do projeto – PMBOK – 5ª. ed.	45
FIGURA 12 - Visão geral do gerenciamento dos riscos do projeto – PMBOK – 5ª. ed.....	47
FIGURA 13 - Processo de comunicação entre os componentes do sistema SafeWalk	72
FIGURA 14 - Exemplo de ícones utilizados no sistema	75
FIGURA 15 - Página inicial do SafeWalk	76
FIGURA 16 - Exemplo de ícones	77
FIGURA 17 - Exemplo de alteração de dados	78
FIGURA 18 - Exemplo de ícones conhecidos	78
FIGURA 19 - Exemplo de classe de complexidade $O(1)$	81
FIGURA 20 - Exemplo de classe de complexidade $O(n)$	81

LISTA DE QUADROS

QUADRO 1 - Cronograma de entregas do projeto	31
QUADRO 2 - Esforços planejados e realizados por fase	38
QUADRO 3 - Propostas de mudanças para o projeto SafeWalk	45
QUADRO 4 - Estimativas de Esforços	59
QUADRO 5 - Design Patterns no projeto SafeWalk	65
QUADRO 6 - Documentos relevantes para testes.	82
QUADRO 7 - Equipamentos a serem utilizados para a realização dos testes.....	82
QUADRO 8 - Softwares a serem utilizados para a realização dos testes.....	83
QUADRO 9 - Identificação dos itens a serem testados	83
QUADRO 10 - Requisitos e casos de teste	84
QUADRO 11 - Matriz de Responsabilidades.....	87
QUADRO 12 - Treinamentos Previstos	88
QUADRO 13 - Tarefas previstas na implantação	88
QUADRO 14 - Documentos de apoio à implantação.....	89

LISTA DE ABREVIATURAS E SIGLAS

ABNT - Associação Brasileira de Normas Técnicas

API - *Application Programming Interface*

CASE - *Computer Aided Software Engineering*

CEP - Código de Endereçamento Postal

CPF - Cadastro de Pessoa Física

DER - Diagrama de Entidade e Relacionamento

EAP - Estrutura Analítica de Projeto

FAI - FAI – Centro de Ensino Superior em Gestão Tecnologia e Educação

FBSP - Fórum Brasileiro de Segurança Pública

GB - *Gigabytes*

GHZ - *GigaHertz*

GPS - *Global Positioning System*

HTML - *HyperText Markup Language*

HTTPS - *HyperText Transfer Protocol Secure*

IDE - *Integrated Development Environment*

IOS - *Iphone Operating System*

IP - *Internet Protocol*

IPEA - Instituto de Pesquisa Econômica Aplicada

JSP - *JavaServer Pages*

JSTL - *JavaServer Pages Standard Tag Library*

JVM - *Java Virtual Machine*

MER - Modelo de Entidade e Relacionamento

ODS - Objetivos de Desenvolvimento Sustentável

PCU - Pontos por Casos de Uso

PF - Pontos por Função

PMBOK - *Project Management Body of Knowledge*

PMI - *Project Management Institute*

PNG - *Portable Network Graphics*

RAM - *Random Access Memory*

RNF - Requisito não Funcional

RF - Requisito Funcional

SGBD - Sistema Gerenciador de Banco de Dados

SMS - *Short Message Service*

SQL - *Structured Query Language*

SSP - Secretaria de Segurança Pública

TB - *Terabytes*

TCP - *Ethernet Transmission Control Protocol*

UML - *Unified Modeling Language*

WBS - *Work breakdown structure*

SUMÁRIO

1 INTRODUÇÃO	14
2 REVISÃO BIBLIOGRÁFICA	16
2.1 SEGURANÇA PÚBLICA	16
2.2 VIOLÊNCIA CONTRA A MULHER	17
2.3 PROCEDIMENTOS DE SEGURANÇA RECOMENDADOS PARA PERCURSOS PÚBLICOS	18
2.4 TRABALHOS RELACIONADOS	19
2.4.1 <i>Malalai</i>	19
2.4.2 <i>bSafe</i>	21
2.4.3 <i>Circle of 6</i>	22
3 OBJETIVO DO PROJETO.....	24
3.1 FORMULAÇÃO DO PROBLEMA	24
3.2 OBJETIVOS	25
3.3 JUSTIFICATIVA	25
3.4 PÚBLICO ALVO	26
3.5 NÍVEIS DE DECISÃO E GRUPOS FUNCIONAIS ATENDIDOS	27
3.6 CONSIDERAÇÕES SOBRE OS OBJETIVOS DO DESENVOLVIMENTO SUSTENTÁVEL (ODS).....	27
4 GERÊNCIA DO PROJETO	28
4.1 PLANO DE PROJETO.....	28
4.1.1 Partes Interessadas.....	28
4.1.2 Modelo de Ciclo de Vida	30
4.1.3 Recursos Necessários.....	31
4.2 ÁREAS DE CONHECIMENTO	33
4.2.1 Gestão do Escopo.....	33
4.2.2 Gestão do Tempo	35
4.2.3 Gestão da Integração	39
4.2.4 Gestão da Qualidade	45
4.2.5 Gestão dos Riscos	46
5 ESPECIFICAÇÃO E ANÁLISE DOS REQUISITOS	48
5.1 REQUISITOS DO SISTEMA DE SOFTWARE	48

5.1.1 Requisitos Funcionais	48
5.1.2 Requisitos não Funcionais	53
5.2 ANÁLISE DOS REQUISITOS	57
5.2.2 Modelo Conceitual dos Dados	57
5.2.3 Modelo Inicial de Interface Usuário	58
5.3 MÉTRICAS PARA ESTIMATIVA DE ESFORÇO	58
6 ARQUITETURA E PROJETO DO SISTEMA DE SOFTWARE	60
6.1 VISÃO ESTRUTURAL	60
6.1.1 Diagrama de Pacotes	60
6.1.2 Diagrama de Classes	61
6.1.3 Diagrama de Objetos	61
6.2 VISÃO COMPORTAMENTAL	61
6.2.1 Projeto das Interações Entre Objetos.....	62
6.2.2 Diagrama de Atividades	63
6.3 VISÃO DOS DADOS	63
6.3.1 Modelo Lógico	63
6.3.2 Dicionário de Dados do Modelo Lógico	63
6.4 VISÃO FÍSICA.....	64
6.4.1 Diagrama de Componentes	64
6.4.2 <i>Frameworks</i> Adotados.....	64
6.4.3 <i>Design Patterns</i> Aplicados.....	65
6.4.4 Convenções e Guias para Codificação.....	66
6.5 PROJETO DO SISTEMA DISTRIBUÍDO	67
6.5.1 Procedimentos para Tratamentos dos Desafios.....	68
6.6 PROJETO DA INTERAÇÃO HUMANO-COMPUTADOR	72
6.6.1 Perfil do Usuário.....	73
6.6.2 Aspecto Visual da Interface do Usuário.....	74
6.6.3 Heurísticas de Usabilidade	75
6.6.4 Análise de Complexidade Algorítmica	80
7 PLANO DE TESTES	82
7.1 FINALIDADE	82
7.2 ESCOPO	82
7.2.1 Referências a Documentos Relevantes	82
7.2.2 Ambiente para a Realização dos Testes	82
7.3.3 Descrição dos Casos de Testes.....	84

7.4 RESULTADOS DOS TESTES	84
7.4.1 Histórico de Realização	84
7.4.2 Resultados	85
8 PLANO DE IMPLANTAÇÃO	86
8.1 METODOLOGIA	86
8.1.2 Matriz de Responsabilidades	86
8.2 TREINAMENTO PREVISTO	87
8.3 CRONOGRAMA DE IMPLANTAÇÃO	88
8.4 DOCUMENTOS DE APOIO À IMPLANTAÇÃO	88
8.5 VISÃO DA IMPLANTAÇÃO	89
9 CONCLUSÃO	90
APÊNDICE A - ESTRUTURA ANALÍTICA DE PROJETO	95
APÊNDICE B - CRONOGRAMA	96
APÊNDICE C - RELATÓRIO DE DESEMPENHO	97
APÊNDICE D - LISTA DE VERIFICAÇÃO DA QUALIDADE	98
APÊNDICE E - DIAGRAMA DE CASOS DE USO	99
APÊNDICE F - DESCRIÇÃO DOS CASOS DE USO	100
APÊNDICE G - DIAGRAMA ENTIDADE E RELACIONAMENTO	101
APÊNDICE H - MODELO DE INTERFACE DO USUÁRIO	102
APÊNDICE I - ACOMPANHAMENTO DOS RISCOS	103
APÊNDICE J - DIAGRAMA DE PACOTES	104
APÊNDICE K - DIAGRAMA DE CLASSES	105
APÊNDICE L - DIAGRAMA DE OBJETOS	106
APÊNDICE M - MODELO LÓGICO DOS DADOS	107
APÊNDICE N - ENTREVISTA DO USUÁRIO	108
APÊNDICE O - DIAGRAMAS DE SEQUÊNCIA	109
APÊNDICE P - DIAGRAMA DE VISÃO GERAL DE INTERAÇÃO	110
APÊNDICE Q - DIAGRAMAS DE ATIVIDADES	111
APÊNDICE R - DICIONÁRIO DE DADOS DO MODELO LÓGICO	112
APÊNDICE S - PONTOS POR CASOS DE USO E PONTOS POR FUNÇÃO	113
APÊNDICE T - <i>SCRIPTS</i> DDL E DML	114
APÊNDICE U - DIAGRAMA DE COMPONENTES	115
APÊNDICE V - CASOS DE TESTE E HISTÓRICO DE REALIZAÇÃO	116
APÊNDICE W - DIAGRAMA DE IMPLANTAÇÃO	117
ANEXO A – MANUAL DE INSTALAÇÃO APACHE TOMCAT E JRE	119

ANEXO B - MANUAL DE INSTALAÇÃO POSTGRESQL	120
--	------------

1 INTRODUÇÃO

A violência urbana é um problema que atinge a maioria das cidades brasileiras. Trata-se, além disso, de um fenômeno que acontece também em nível mundial. O crescimento da violência, em suas múltiplas modalidades - crime comum, crime organizado, violência doméstica e a violação de direitos humanos - vêm se constituindo em uma das preocupações principais da sociedade brasileira nas duas últimas décadas. A falta de segurança pública é notória e a cada ano as estatísticas de violência aumentam. Em cada parte do Brasil, os governos locais não conseguem combater de forma eficaz a criminalidade que toma conta das cidades.

De acordo com Dantas (2019), em 2016, pela primeira vez na história, o número de homicídios no Brasil superou a casa dos 60 mil em um ano. Segundo o Atlas da Violência de 2018, produzido pelo Instituto de Pesquisa Econômica Aplicada (Ipea) e o Fórum Brasileiro de Segurança Pública (FBSP), o número de 62.517 assassinatos cometidos no país, em 2016, coloca o Brasil em um patamar 30 vezes maior do que o da Europa. Na década de 2000, 553 mil brasileiros perderam a vida por morte violenta, em torno de 153 mortes por dia (IPEA, 2018).

Diante disso, este projeto, denominado *SafeWalk*, visou desenvolver um sistema de software que contribua para a segurança da população, especialmente, durante trajetos efetuados em vias públicas.

A metodologia adotada para desenvolver o projeto *SafeWalk* foi a realização da pesquisa bibliográfica, gerência com referência ao guia de conhecimentos do *Project Management Institute* (PMI), especificação e modelagem baseada na *Unified Modeling Language* (UML), construção e testes de software orientado a objetos.

Sendo assim, este documento descreve o projeto e contém os capítulos apresentados a seguir, além desta introdução.

O Capítulo 2 apresenta a revisão bibliográfica e todo embasamento necessário para a realização do projeto.

O Capítulo 3 apresenta o problema encontrado, os objetivos, a justificativa do projeto, público alvo, níveis de decisão e grupos funcionais atendidos.

O Capítulo 4 apresenta o plano e a gerência deste projeto e mostra os artefatos criados durante a gestão do escopo, do tempo, da integração, da qualidade e dos riscos.

O Capítulo 5 apresenta os requisitos funcionais e os requisitos não funcionais do sistema de software *SafeWalk*, o modelo de casos de uso, o modelo conceitual de dados e de interface de usuário e as estimativas de esforço.

O Capítulo 6 apresenta as visões estrutural, comportamental, dos dados, física e ainda padrões, convenções e guias para codificação. Além de os desafios para desenvolvimento de um sistema distribuído e o projeto da interação humano-computador.

O Capítulo 7 apresenta o plano de testes. Nele constam finalidade, escopo, especificação dos casos de testes e aborda a execução detalhada dos testes durante o processo de construção do software para promover boa funcionalidade da aplicação e confiabilidade no desempenho.

No Capítulo 8, plano de implantação, apresentam-se a metodologia utilizada para a implantação, treinamentos, cronogramas, documentos de apoio e suporte à implantação.

No Capítulo 9 apresenta-se a conclusão do projeto, na qual os membros da equipe relatam os objetivos alcançados na elaboração do projeto e expõem as conclusões obtidas durante sua execução.

Este documento ainda contém 24 apêndices ordenados de A ao X, com os diagramas e características técnicas do projeto e 2 anexos, ordenados de A a B.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo tem por objetivo expor conceitos e fundamentos que são necessários para o entendimento do projeto, bem como citar trabalhos relacionados que apresentam características semelhantes ao projeto desenvolvido.

Desta forma, as seções a seguir apresentam uma visão da segurança pública no Brasil, dados da violência contra a mulher, os procedimentos de segurança recomendados para percurso público e os trabalhos relacionados ao tema proposto.

2.1 SEGURANÇA PÚBLICA

A segurança é um incontestável princípio para a preservação dos direitos humanos. No Brasil, diante de uma sociedade amedrontada, a violência urbana corrobora para problemas sociais. Sob essa ótica, manifesta-se a necessidade de políticas públicas e educadoras a fim de atenuar tal problemática. A princípio, quando se trata de violência, raramente a atenção se volta para o processo de formação educacional da pessoa, que é algo tão importante.

Visto posto, a violência urbana persiste como um dos mais graves problemas sociais no Brasil, totalizando mais de 1 milhão de vítimas fatais nos últimos 24 anos (a partir de 1992). A taxa de mortes por agressão saltou de 22,2 no ano de 1990 para 28,3 por 100 mil habitantes em 2013, com variações importantes entre diferentes estados (MACHADO et al., 2016).

Todavia, é passível de engano quem acredita que o fenômeno da violência urbana está restrito aos grandes centros. Esse problema pode ser observado também em pequenos centros urbanos, em todo o país, onde as manchetes dos jornais mostram um aumento no número de assaltos, homicídios e outros atos de violência, o que deixa as populações locais apreensivas. Isso comprova que a violência tem tomado proporções alarmantes e é configurada como um “morbus social”¹ que carece de uma solução urgente (GOMES, 2016).

Desta maneira, o resultado de um cenário de grande violência urbana é, principalmente, a baixa qualidade de vida de todos os cidadãos. A perda prematura de vidas é o custo mais vultoso para o país. Isso é refletido em um sentimento de insegurança crescente,

¹ Morbus Social quer dizer doença social (LEXICO, 2019).

principalmente entre as camadas menos favorecidas da população. Muitos policiais civis e militares também são vítimas de homicídios, em serviço ou fora dele. Além de um impacto direto na vida daqueles que vivem essa violência, com altas taxas de assassinatos e outros crimes contra a pessoa, é possível observar também a mácula na imagem do país no cenário internacional. Esse grave problema social apresenta inúmeros desdobramentos, que podem ir da insatisfação popular com o serviço de segurança pública, passando pelos altos custos de ações paliativas por parte do governo e perdas econômicas como a redução do turismo e das viagens de negócio (MONTE, 2006).

2.2 VIOLÊNCIA CONTRA A MULHER

A violência contra a mulher é um grave problema social no Brasil e no mundo, apesar das iniciativas de políticas públicas em torno da questão. Todavia, os avanços são poucos, embora, desde 1990, a Organização Mundial de Saúde já reconheça a violência contra a mulher como um problema de saúde pública que exige dos governantes políticas públicas mais eficientes no combate e prevenção do fenômeno. Além de causar sofrimento físico e psíquico à mulher e, conseqüentemente, a seus filhos e família, esse tipo de violência é também uma violação dos direitos humanos (ARAÚJO, 2008).

Sob essa ótica, as consequências nas vidas das mulheres que sofrem ou sofreram algum tipo de violência são caracterizadas de várias formas, trazendo um grande impacto no cotidiano das vítimas, como pânico, desvalorização pessoal, desespero, sensação de abandono e distúrbio do estresse pós-traumático, podendo chegar inclusive a casos de homicídios. Tendo em vista os direitos humanos presentes no Código Penal e na lei federal nº 11.340, conhecida como Lei Maria da Penha (BRASIL, 2006), toda mulher tem direito ao respeito e à dignidade, igualdade, liberdade de associação, liberdade de professar a religião e às próprias crenças (SILVA et al., 2010).

A violência contra a mulher é referida de diversas formas desde a década de 1950, sendo designada como violência intrafamiliar, violência contra a mulher e violência doméstica. Na década de 1990, os estudos passaram a tratar essas relações de poder, em que a mulher em qualquer faixa etária é submetida e subjugada, como violência de gênero (GUEDES; GOMES, 2014).

Diante disso, as formas de violência contra mulher são divididas em: violências físicas, psicológicas, patrimoniais, sexuais e sociais. Para combater e prevenir a violência contra a mulher é preciso que o Estado possa cumprir com sua obrigação que está prevista na Constituição Federal de 1988 que cria mecanismos de combate à violência no âmbito das relações familiares. É preciso também que o Estado passe a investir em políticas públicas de segurança, saúde, educação, assistência social, habitação, dentre outras. Além da importância de o Estado somar à sua legislação interna as normas penais, civis, administrativas e outras que sejam necessárias, que estão previstas na convenção sobre a eliminação de todas as formas de discriminação contra a mulher (CAVALCANTE, 2016)

2.3 PROCEDIMENTOS DE SEGURANÇA RECOMENDADOS PARA PERCURSOS PÚBLICOS

A sensação de insegurança nas vias é cada dia maior. O número crescente de assaltos e casos de violência amedronta a população e a sensação de medo é quase constante. Pensando nisso, a Secretaria de Segurança Pública de São Paulo (SSP/SP) criou um manual de segurança do cidadão para ajudar a prevenir ou enfrentar diversas situações, como acidentes, afogamentos, assaltos, entre outras. Ao todo são 50 ações contendo situações diversas para cada uma delas. Nesse manual, além dos procedimentos básicos nas várias situações enfrentadas no dia-a-dia, são citadas as formas de condutas e medidas que cada cidadão deve adotar para aumentar sua segurança.

Desta maneira, em relação aos percursos públicos, algumas orientações são (POLÍCIA MILITAR DE SÃO PAULO, 2018):

- a) manter sempre atenção na rua, no metrô, no ônibus, em centros comerciais, entre outros;
- b) ter especial atenção às pessoas que estão ao redor, pois os assaltantes valem-se principalmente do fator surpresa e da desatenção para atacarem suas vítimas;
- c) não andar sozinho(a), especialmente, à noite;
- d) evitar passar por locais desertos e ou pouco iluminados;

- e) evitar cortar caminho por becos, vielas, ruas desconhecidas, terrenos, construções, entre outros;
- f) conhecer o local onde se mora e trabalha. Certificar-se da localização de unidades da polícia, bombeiro, hospitais, telefones públicos, lojas 24 h, dentre outras;
- g) procurar variar horários e fazer rotas diferentes para deslocar-se de casa ao trabalho ou escola;
- h) evitar pontos de parada em locais escuros e isolados.

As orientações da Polícia Militar constam também para outras situações que envolvam espaços públicos, como práticas de esportes, caminhadas, ida à escola, dentre outras.

2.4 TRABALHOS RELACIONADOS

Nesta seção são apresentados os aplicativos que se assemelham às funcionalidades propostas pelo sistema de software *SafeWalk*. Uma vez que, o projeto possui como diferencial em relação aos demais, o fato de possuir interação com o poder público por meio da funcionalidade “Manter o Cadastro de Áreas Inseguras”, onde os responsáveis poderão mapear qual área está desprotegida para o cidadão. Visto posto, são citadas 3 soluções que visam auxiliar na segurança da população que se encontra em situações de vulnerabilidade em espaços públicos. São elas: *Malalai*, *Circle of 6* e *bSafe*.

2.4.1 *Malalai*

A plataforma virtual *Malalai* foi desenvolvida por Priscila Gama, uma arquiteta mineira, após aumento explícito de relatos de vítimas sofrendo abusos e violência nas ruas. Está disponível para os sistemas operacionais Android e iOS com licença de uso gratuita. O aplicativo permite que usuários compartilhem suas rotas com alguém em sua lista de contatos, para que eles possam acompanhar em um mapa todo o seu deslocamento ao longo do trajeto. Também é possível notificar essa pessoa quando se chega ao destino ou enviar um alerta em caso de emergência, sempre informando a localização da pessoa que se desloca. As principais características do aplicativo *Malalai* são (MALALAI, 2017):

- a) analisar os pontos positivos e negativos das rotas publicadas;
- b) consultar informações das vias públicas;
- c) possibilitar a amigos ou parentes monitorar o trajeto que está sendo percorrido pela pessoa na via;
- d) acionar algum conhecido, enviado sua localização atual, no caso de uma situação de risco à segurança pessoal.

A Figura 1 mostra uma visão geral das telas do aplicativo. Nela é possível observar as telas que contêm o mapa, bem como o percurso definido pelo usuário.



FIGURA 1 - Tela do aplicativo *Malalai*
FONTE: adaptado de *Malalai* (2017)

A Figura 2 mostra uma visão detalhada da tela de relatos do aplicativo. Nela é possível avaliar um problema em uma determinada via, definido um ponto no mapa.



FIGURA 2 - Detalhes do aplicativo *Malalai*
 FONTE: adaptado de *Malalai* (2017)

2.4.2 *bSafe*

O *bSafe* é um aplicativo de segurança pessoal que tem as mulheres como público alvo. Por meio dele, as usuárias criam uma “rede de segurança” com pessoas de sua confiança, que são notificadas em caso de emergência ou situações de insegurança. O aplicativo foi criado por Rich Larsen e Charlen Larsen, com versões gratuita e paga para Android e iOS. As principais características do aplicativo *bSafe* são (BSAFE, 2019):

- a) **alarme:** envia local para contatos pré-definidos, automaticamente, informações sobre onde a mulher está, obtido a partir do Sistema de Posicionamento Global (GPS), e o que está acontecendo, pois aciona o vídeo e áudio do dispositivo.
- b) **“Me siga”:** a usuária convida seus tutores para segui-la ao vivo. Eles serão capazes de vê-la através do mapa mostrado pelo aplicativo no dispositivo móvel. Quando ela chegar em casa com segurança, seus tutores receberão uma mensagem de aviso;
- c) **chamada falsa:** a usuária do aplicativo recebe uma ligação de um de seus tutores, sendo aplicável nos casos em que ela precisa sair de situações desagradáveis ou ameaçadoras;
- d) **rede de segurança:** é possível formar uma rede de segurança pessoal com amigos e familiares que receberão os alertas quando a usuária achar necessário.

A Figura 3 mostra uma visão geral da tela inicial do aplicativo. Nela é possível observar a rede de contato de cada usuário, bem como enviar convites para os amigos e assim criar uma rede de segurança.

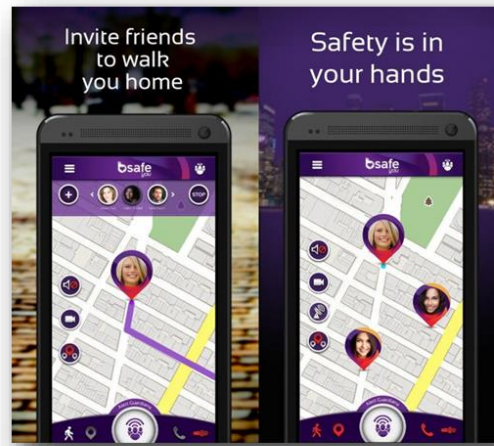


FIGURA 3 - Tela do aplicativo *bSafe*
FONTE: adaptada de *bSafe* (2019)

2.4.3 *Circle of 6*

A plataforma virtual *Circle of 6* foi lançada nos Estados Unidos, em 2008. Por meio de um aplicativo disponível para Android ou IOS é possível definir até seis contatos de pessoas de confiança para solicitar socorro, via Serviço de Mensagens Curtas (SMS), em uma situação desconfortável ou de risco. As principais características do aplicativo *Circle of 6* são (CIRCLE OF 6, 2019):

- a) adicionar até seis pessoas para que possam participar do círculo;
- b) acesso direto a informações sobre sexualidade, relacionamentos e segurança;
- c) envio e troca de mensagens, por meio de um *chat* virtual;
- d) envio de SMS com a localização atual ao círculo de amigos selecionados.

A Figura 4 mostra a tela inicial do aplicativo. Nela é possível definir até 6 contatos diferentes que podem ser utilizados caso o usuário necessite solicitar socorro.

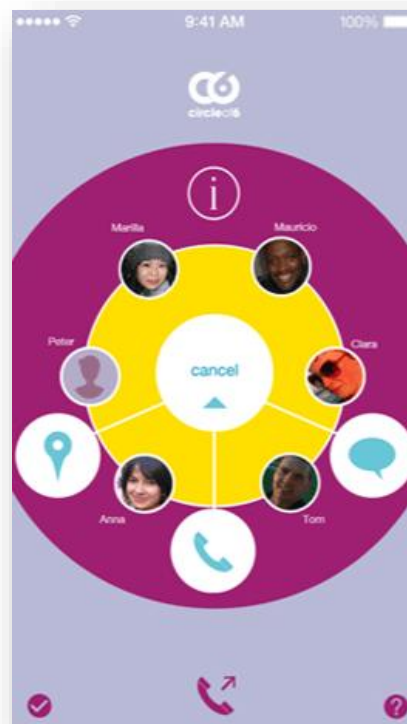


FIGURA 4 - Tela do aplicativo *Circle of 6*
FONTE: adaptada de *Circle of 6* (2019)

3 OBJETIVO DO PROJETO

Neste capítulo é apresentada a formulação do problema, pela qual se mostra a necessidade de realização deste projeto, os objetivos gerais e específicos traçados, a justificativa, o público alvo e os níveis de decisão e grupos funcionais atendidos.

3.1 FORMULAÇÃO DO PROBLEMA

Um dos quesitos mais importantes na vida de uma pessoa é o seu direito de ir e vir com segurança. A liberdade de locomoção, de forma legal e em tempos de paz, é um dos desdobramentos do direito à liberdade, ao respeito e à dignidade humana.

O fenômeno da violência urbana tem se tornado um dos principais problemas para os governos e se constitui em um desafio para todos. Fatos execráveis de homicídios, sequestros e estupros têm sido divulgados com frequência pelos meios de comunicação e afetam cada vez mais as famílias brasileiras.

Os problemas relacionados com o aumento das taxas de criminalidade, a sensação de insegurança, as dificuldades relacionadas à reforma das instituições da administração da justiça criminal, a superpopulação nos presídios, a incapacidade preventiva das instituições destinadas aos jovens infratores, rebeliões, fugas, os altos custos operacionais do sistema carcerário, a precariedade dos instrumentos de investigação criminal e das perícias policiais, além da morosidade judicial, entre tantos outros problemas, comprometem o bem estar social e a democracia (CALHEIROS, 2014).

Desta maneira, sabendo como as ruas e cidades se tornaram perigosas, observou-se a necessidade desenvolver um sistema de software no qual os usuários poderão avaliar se o lugar por onde passaram é seguro ou não, a fim de alertar outras pessoas sobre as condições do local escolhido como percurso. Assim se as avaliações não forem positivas, a pessoa que ali quer passar poderá procurar outra via mais segura para seguir seu trajeto.

3.2 OBJETIVOS

O objetivo geral deste projeto final de curso é desenvolver um sistema de software que possa ajudar na segurança da população. A ideia consiste em uma aplicação onde uma condição insegura em um local público possa ser informada e assinalada por meio de mapas geográficos projetados pelo sistema.

Assim, é possível que as pessoas avaliem ruas, parques e praças de acordo com critérios estabelecidos. Os usuários poderão consultar o mapa do local desejado e adicionar dados visando facilitar a busca de informações fornecidas entre si, de uma forma colaborativa.

Como objetivos específicos têm-se:

- a) divulgar relatos sobre percursos públicos às pessoas que utilizam o sistema;
- b) coletar informações sobre percursos públicos com a colaboração dos usuários do sistema;
- c) garantir a confidencialidade e a atualização das informações fornecidas pelos usuários que serão exibidas no mapa;
- d) aplicar o ciclo de vida da informação, que vai desde sua criação até o seu descarte;
- e) avaliar o nível de satisfação dos utilizadores com o resultado obtido;
- f) garantir que o ambiente do sistema seja o mais íntegro possível, identificando e analisando prontamente cada denúncia que possa existir em relação ao comportamento de seus usuários.

3.3 JUSTIFICATIVA

O Anuário Brasileiro de Segurança Pública 2017 apontou que, nos anos de 2014, 2015 e 2016, os registros de violência atingiram as marcas de 50.544, 46.873 e 44.080 (para tentativa de homicídio), assim como 575.957, 631.383 e 602.628 (para lesão corporal), além de 50.438, 47.461 e 49.497 (para estupro) e 7.846, 7.211 e 6.548 (para tentativa de estupro), respectivamente (FBSP, 2017, p. 42-46).

A Figura 5 mostra esses dados no período trianual.

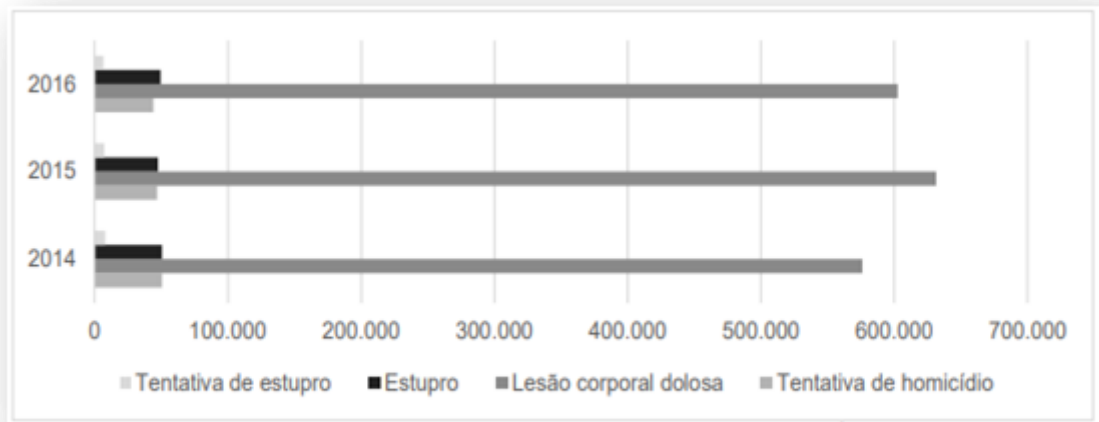


FIGURA 5 - Gráfico da violência no período entre 2014 e 2016 no Brasil
FONTE: adaptado FBSP (2017)

Assim, partindo deste ponto, um sistema para facilitar a obtenção de informações sobre a segurança das vias públicas pode contribuir com a diminuição dos índices de violência. As pessoas poderão ficar a par das ocorrências e, a partir delas, solicitar providências do poder público.

3.4 PÚBLICO ALVO

O sistema de software *SafeWalk* tem como público alvo, as pessoas que querem andar nas ruas com mais segurança, sabendo se no seu trajeto há algum ponto relatado como de condição insegura e, assim, tomarem a decisão sobre qual o melhor caminho a percorrer até o seu destino.

As pessoas que transitam pelas vias poderão consultar as informações sobre a segurança dos locais pelos quais precisarão passar e, caso esses locais sejam mal avaliados por outras pessoas, elas poderão contatar as autoridades responsáveis e solicitarem providências para que as condições de segurança dessas vias melhorem.

Além disso, no modelo de negócio do *SafeWalk* está previsto, inicialmente, a comercialização de licenças para as prefeituras da região do sul de Minas Gerais, fazendo com que o sistema possa ser implantado tanto para o cidadão quanto para o poder público,

gratuitamente. E, assim, aumentando sua abrangência de acordo com o *feedback* dos usuários. Ademais para manter o engajamento dos usuários, o *SafeWalk* trabalha de maneira colaborativa e possui como diferencial a possibilidade de cadastro das áreas inseguras mapeadas pelo próprio poder público.

3.5 NÍVEIS DE DECISÃO E GRUPOS FUNCIONAIS ATENDIDOS

O sistema atuará no nível operacional, pois o próprio usuário poderá visualizar e analisar se o caminho que ele deseja percorrer é seguro e assim segui-lo. Os grupos funcionais são diversos, pois se trata de um sistema onde usuários poderão avaliar as vias para que outros usuários possam ver se elas são seguras ou não. A quantidade e diversidade das avaliações serão proporcionais à quantidade cadastrada pelos usuários.

3.6 CONSIDERAÇÕES SOBRE OS OBJETIVOS DO DESENVOLVIMENTO SUSTENTÁVEL (ODS)

Os Objetivos de Desenvolvimento Sustentável (ODS) (ou Objetivos Globais para o Desenvolvimento Sustentável) são uma coleção de 17 grandes metas globais estabelecidas pela Assembleia Geral das Nações Unidas. Esses 17 Objetivos foram propostos a partir do sucesso dos Objetivos de Desenvolvimento do Milênio, incluindo novos temas, como a mudança global do clima, desigualdade econômica, inovação, consumo sustentável, paz e justiça, entre outras prioridades.

Desta maneira, visando contribuir com os objetivos levantados na Agenda 2030, por meio do desenvolvimento do projeto *SafeWalk*, destaca-se o Objetivo 11 - Tornar as cidades e os assentamentos humanos inclusivos, seguros, resilientes e sustentáveis, será alcançado. Uma vez que a ideia principal do projeto consiste em uma aplicação web pela qual o usuário poderá relatar uma condição insegura em um local público, informada e assinalada por meio de mapas geográficos projetados pelo sistema.

4 GERÊNCIA DO PROJETO

A gerência de projetos engloba as atividades de abertura, planejamento, execução, monitoramento e controle e encerramento do projeto, com o intuito de assegurar que tal empreendimento tenha seus objetivos alcançados.

Segundo Sommerville (2011, p. 414):

O gerenciamento de projetos de software é uma parte essencial da engenharia de software. Os projetos precisam ser gerenciados, pois a engenharia de software profissional está sempre sujeita a orçamentos organizacionais e restrições de cronograma. O trabalho do gerente de projetos é garantir que o projeto de software atenda e supere essas restrições, além de oferecer softwares de alta qualidade.

Neste contexto, este capítulo aborda o plano de gerenciamento deste projeto juntamente com a gerência efetiva realizada sobre ele. No plano de projeto, trata-se da definição das partes interessadas, do modelo de ciclo de vida adotado para o projeto dos recursos necessários para sua realização e das estimativas de esforço para o seu efetivo desenvolvimento. Na seção das áreas de conhecimento, são abordadas questões que englobam a gestão do escopo, do tempo, da integração, da qualidade e dos riscos, conforme o *Project Management Body of Knowledge (PMBOK)*, 5ª edição, publicada pelo PMI.

4.1 PLANO DE PROJETO

Nesta seção são apresentados os interessados do projeto, utilizando para isto uma matriz de Poder x Interesse, o modelo de processo adotado, e também recursos humanos, de *hardware* e de *software* necessários para o desenvolvimento.

4.1.1 Partes Interessadas

Parte interessada (*stakeholders*) é definida como pessoas e organizações, clientes, patrocinadores, organizações executoras e o público que estejam ativamente envolvidas no projeto ou cujos interesses possam ser afetados de forma positiva ou negativa pela execução ou término do projeto (PMI, 2013, p. 30). Além disso, Pressman (1997 apud SOMMERVILLE, 2011) define por interessado, todo aquele que de forma direta ou indireta se beneficiará do projeto que está sendo desenvolvido. Desta maneira, para o desenvolvimento deste projeto, foram identificadas 4 partes interessadas, a saber:

- a) **população (mantenha informada):** habitantes ou visitantes de um município que farão interação com o sistema;
- b) **instituição promotora (gerencie de perto):** FAI – Centro de Ensino Superior em Gestão, Tecnologia e Educação, entidade que define as datas marco, fornece os conhecimentos e orientações necessárias para o desenvolvimento do projeto. É a instituição onde estão matriculados os integrantes da equipe de desenvolvimento;
- c) **equipe de desenvolvimento (monitore):** formada por Ângela Fonseca Faria, Caroline Ribeiro Oliveira, Marcos Antônio de Sousa e Rodrigo Mansur Gonçalves. São estudantes do Curso de Sistemas de Informação e os responsáveis pelo desenvolvimento das atividades referentes às fases deste projeto;
- d) **orientadora (mantenha satisfeita):** profa. Eunice Gomes de Siqueira, responsável por orientar a equipe de desenvolvimento durante todas as fases;
- e) **gerente de projeto (gerencie de perto):** Ângela Fonseca Faria, responsável por identificar, estabelecer, coordenar e monitorar as atividades, tarefas e recursos necessários para que o desenvolvimento aconteça dentro do prazo previsto.

Os interessados do projeto *SafeWalk* estão expostos na matriz de poder e interesse apresentada na Figura 6.



FIGURA 6 - Matriz Poder X Interesse

FONTE: elaborado pelos autores (2019)

4.1.2 Modelo de Ciclo de Vida

De acordo com o PMI (2013), ao passo que todo o projeto possui um início e um fim definidos, as entregas e atividades específicas de cada um podem variar muito de acordo com suas especificidades. Dessa maneira, o ciclo de vida oferece uma estrutura básica para o gerenciamento do projeto, independentemente dos trabalhos específicos envolvidos em cada um. Assim, para que o projeto seja bem-sucedido, é essencial a utilização de modelos de processos.

Por meio de uma análise do problema como um todo, das datas previstas e das pessoas nele envolvidas, o modelo de ciclo de vida adotado é o Modelo Incremental, cujas atividades genéricas podem ser vistas na Figura 7.

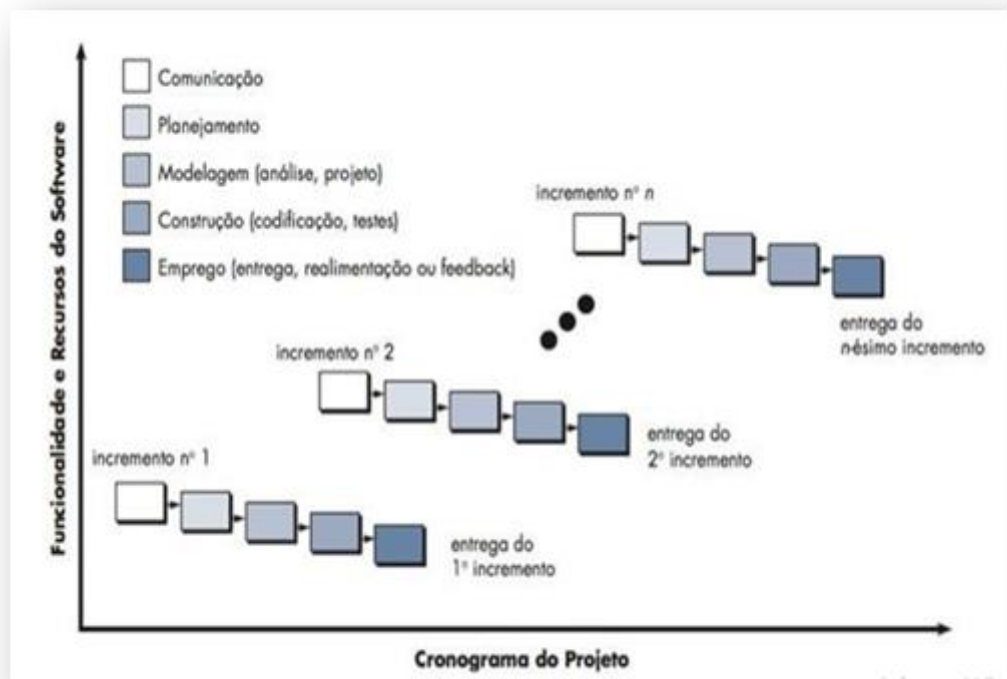


FIGURA 7 - Modelo Incremental
FONTE: Pressman (2013)

Esse modelo mostra-se adequado às necessidades do projeto, já que existem alguns pontos de incerteza que serão esclarecidos à sucessão do tempo e aprimoramentos consequentes da utilização de todo o sistema. Por conseguinte, o projeto possui dois incrementos entregues de acordo com o Quadro 1.

Incremento	Data de entrega	Descrição
I	06/04/2019	Entrega dos tópicos do projeto: revisão bibliográfica, objetivos do projeto e suas seções, gerência do projeto com as subseções "Plano de Projeto" e "Áreas de Conhecimento", especificação e análise dos requisitos com as subseções "Requisitos Funcionais" e "Análise dos Requisitos" e as referências.
II	01/06/2019	Revisão dos tópicos do Incremento I. Completar gerência do projeto. Entrega do tópico Arquitetura e Projeto e suas seções. Codificação de no mínimo 30% dos casos de uso essenciais.
III	07/09/2019	Correções apontadas pelos orientadores nas fases anteriores e complemento da arquitetura de projeto com as seções: visão física, padrões de projeto e projeto da interação humano-computador. Pelo menos 60% dos casos de uso essenciais do sistema implementados.
IV	02/11/2019	Entrega das correções e refinamentos propostos nas fases anteriores e a construção dos capítulos restantes. Os casos de uso essenciais do projeto deverão estar implementados e testados.

QUADRO 1 - Cronograma de entregas do projeto
 FONTE: elaborado pelos autores (2019)

4.1.3 Recursos Necessários

Nesta seção são mostrados os recursos humanos, de hardware e de software necessários para a realização deste projeto.

4.1.3.1 Recursos Humanos

A equipe de desenvolvimento deste projeto possui 4 integrantes que são graduandos em Sistemas de Informação. Os integrantes e as atividades desempenhadas no projeto são:

- a) Ângela Fonseca Faria: gerência do projeto, pesquisa, documentação e codificação;
- b) Caroline Ribeiro Oliveira: testes do sistema, pesquisa e documentação;
- c) Marcos Antônio de Sousa: análise de negócios, pesquisa, codificação e documentação;
- d) Rodrigo Mansur Gonçalves: projetista, codificação, pesquisa e documentação.

4.1.3.2 Recursos de Hardware

Os recursos de hardware necessários para o projeto são: 4 computadores cuja especificação mínima de processador deve ser i3 ou equivalente, disco rígido de 250 *Gigabytes* (GB), 2 GB de *Random Access Memory* (RAM) e acesso à Internet.

4.1.3.3 Recursos de Software

Para o desenvolvimento do projeto, faz-se necessária a utilização de alguns *softwares* tanto gerenciais como auxiliares na execução das atividades. Desse modo, os recursos de software necessários são:

- a) **Sistemas Operacionais**
 - a.1) Microsoft Windows 7 ou superior.
- b) **Ferramentas CASE ou de modelagem**
 - b.1) BrModelo 2.0;
 - b.2) Ideas Modeler Standard Edition 11+;
 - b.3) Microsoft Visio 2016;
 - b.4) StarUML 3.0+;
 - b.5) WBS Tool 5.1+;
 - b.6) Quant UX 1.9+.
- c) **Ferramentas de versionamento**
 - c.1) GitHub;
 - c.2) Dropbox;
 - c.3) Google Drive.
- d) **Ferramentas de desenvolvimento e *frameworks***
 - d.1) Bootstrap 4.0+;
 - d.2) NetBeans IDE 8.0+;
 - d.3) Microsoft Visual Code 1.0+;
 - d.4) PgAdmin4;
 - d.5) PostgreSQL 9+.

4.2 ÁREAS DE CONHECIMENTO

Esta seção aborda os principais aspectos das áreas de escopo, tempo, integração, qualidade e riscos dentro da gerência de projetos, apresentando em cada uma delas os artefatos produzidos durante o desenvolvimento do mesmo. Sendo este projeto um esforço integrado, a ausência dessas áreas de conhecimento poderá afetar negativamente a qualidade do projeto. As demais áreas previstas no PMBOK não são documentadas neste capítulo.

4.2.1 Gestão do Escopo

Escopo refere-se ao trabalho a ser realizado no âmbito do projeto. Sendo assim, o escopo pode estar ligado ao produto ou ao projeto.

De acordo com o PMI (2013, p. 105):

O gerenciamento do escopo do projeto inclui os processos necessários para assegurar que o projeto inclui todo o trabalho necessário e, apenas o necessário, para terminar o projeto com sucesso. O gerenciamento do escopo do projeto está relacionado principalmente com a definição e controle do que está e do que não está incluso no projeto.

A Figura 8 apresenta de forma geral a visão do gerenciamento do escopo do projeto, citando cada uma de suas entradas, ferramentas e técnicas e saídas esperadas.

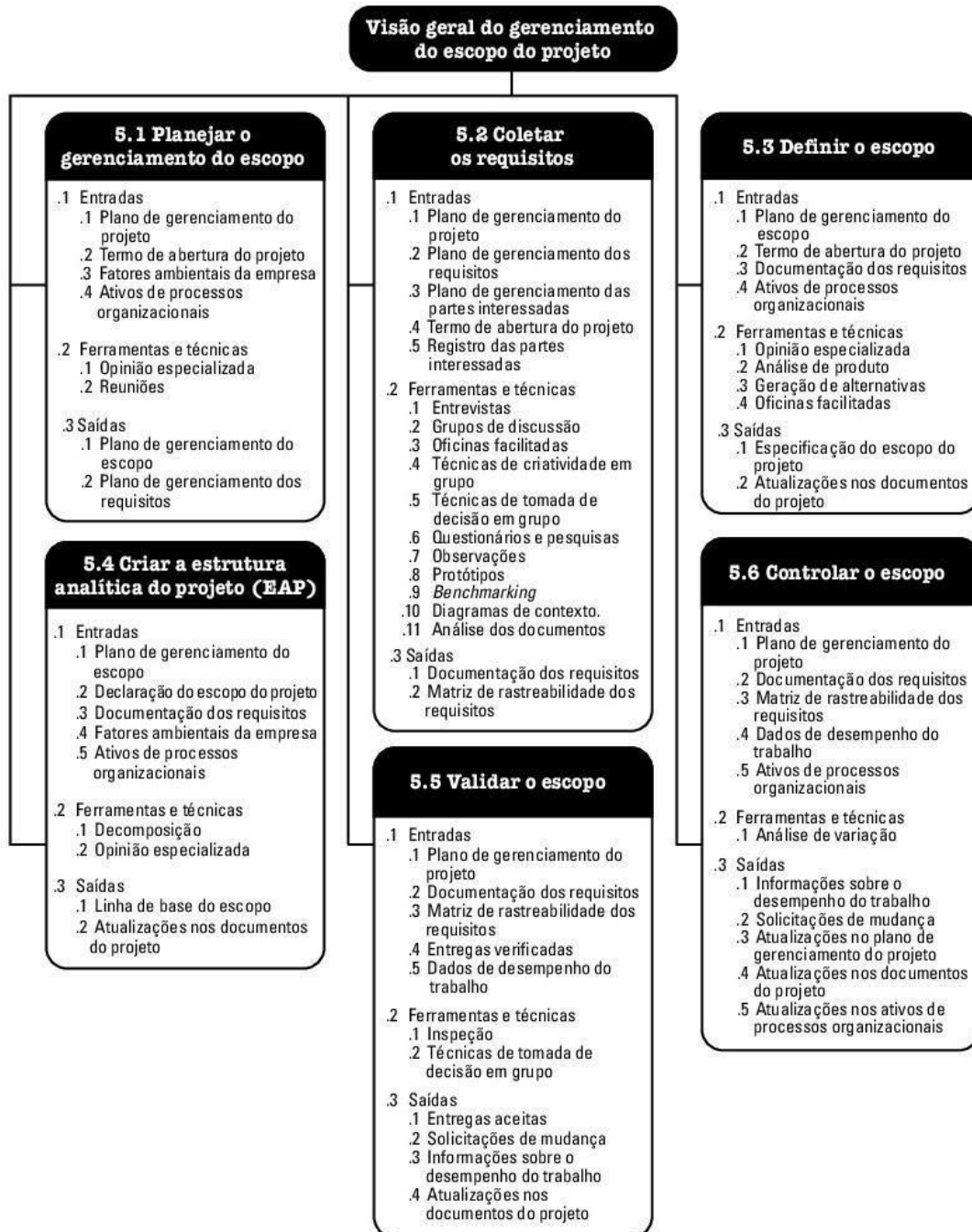


FIGURA 8 - Visão geral do gerenciamento do escopo do projeto – PMBOK – 5ª. ed.
 FONTE: PMI (2013, p. 106).

4.2.1.1 Estrutura Analítica do Projeto (EAP)

Conhecida pelo termo em inglês *Work Breakdown Structure* (WBS), a criação da EAP é um dos processos da gestão de escopo.

De acordo com Oliveira e Chiari (2014, p.47):

A EAP é uma decomposição do escopo do projeto em níveis hierárquicos, sendo eles: Projeto, no nível mais alto, seguido por fases (se o projeto tiver fases distintas), depois o produto ou entrega do projeto e por fim os pacotes de trabalho. A principal técnica utilizada para criar a EAP é decomposição. Por meio desta técnica o projeto será decomposto em produtos e, entregas e pacotes de trabalho que sejam mais fáceis de serem gerenciados. A EAP deve ser decomposta até o nível apropriado para definir um responsável, estimar o tempo e custo do pacote de trabalho.

Assim, como ferramenta visual, a EAP melhora a comunicação do projeto, a decomposição do trabalho e evita a omissão de entregas. A ferramenta é, ainda, uma fonte de avaliação do risco do projeto, particularizado em cada pacote de trabalho. Desse modo, a EAP do projeto encontra-se no Apêndice A este documento.

4.2.1.2 Dicionário EAP

O dicionário da EAP elucida os elementos presentes na sua estrutura, descrevendo os elementos com detalhes, propiciando mais entendimento dos pacotes de trabalho. Assim, ao visualizar a EAP por si só, podem surgir dúvidas quanto ao que cada elemento representa e o que está contido nele. Desta forma, cria-se esse dicionário que descreve com detalhes cada elemento.

O dicionário da EAP do projeto encontra-se no Apêndice B deste documento, descrito no campo "Anotações" do arquivo elaborado com o MS-Project.

4.2.2 Gestão do Tempo

Segundo Carvalho (2014, p. 97), o gerenciamento do tempo é composto por “processos que permitem o cumprimento dos prazos de acordo com o planejado, especialmente o término pontual do projeto”.

De acordo com o PMI (2013, p. 142):

Os processos de gerenciamento do tempo do projeto e suas ferramentas e técnicas associadas são documentados no plano de gerenciamento do cronograma. O plano de gerenciamento do cronograma é um plano auxiliar e integrado ao plano de gerenciamento do projeto através do processo desenvolver o plano de gerenciamento do projeto.

A Figura 9 apresenta de forma geral a visão do gerenciamento do tempo do projeto, citando cada uma de suas entradas, ferramentas e técnicas e saídas esperadas.

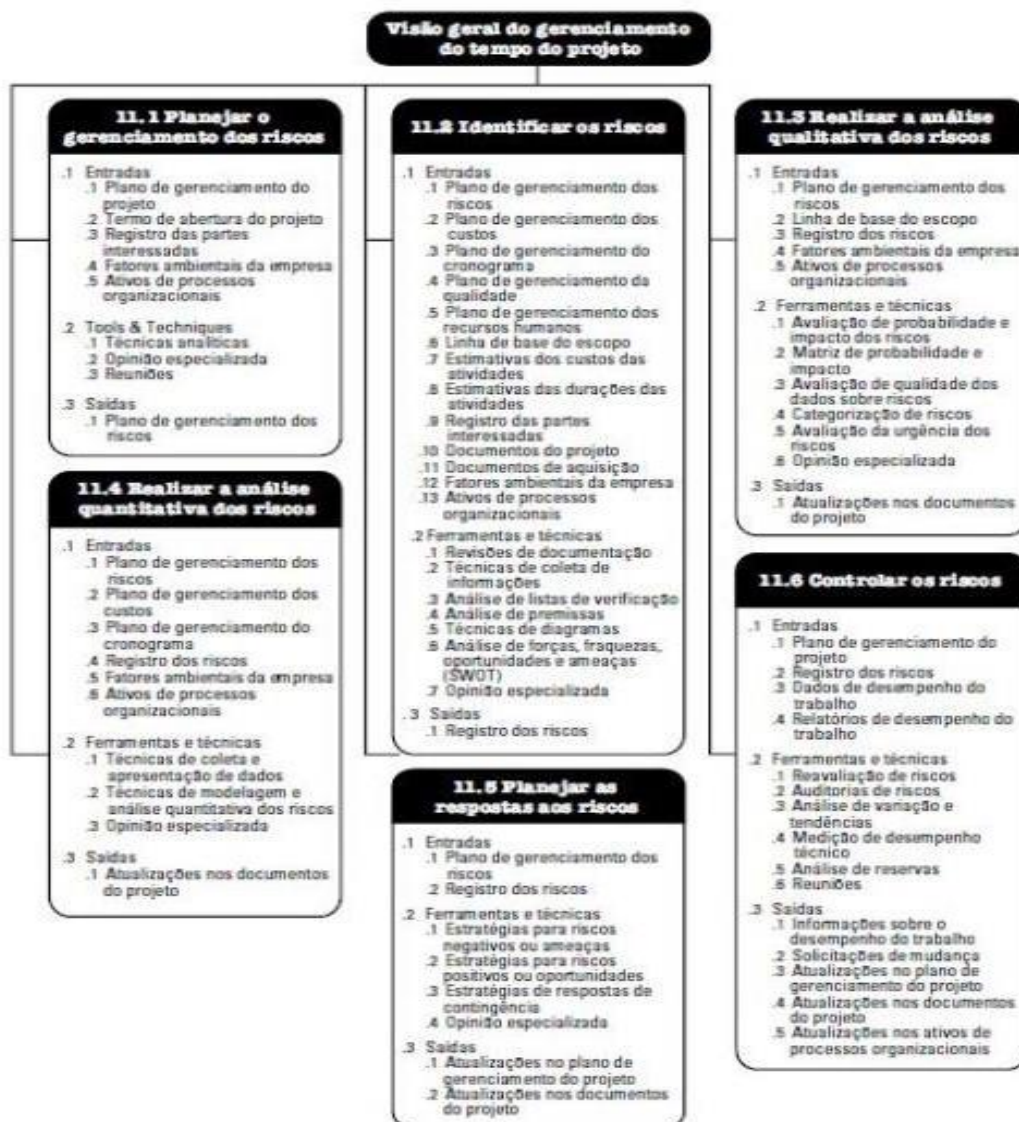


FIGURA 9 - Visão geral do gerenciamento do tempo do projeto – PMBOK – 5ª. ed.
FONTE: PMI (2013, p. 143)

Assim, as subseções apresentam os resultados dos processos realizados para a gestão do tempo deste projeto.

4.2.2.1 Lista de Atividades, Diagrama de Rede e Cronograma

A lista de atividades define todas as atividades previstas para cada pacote de trabalho previsto na EAP. Elas deverão ser iniciadas, executadas e finalizadas no tempo adequado de acordo com o cronograma.

De acordo com o PMI (2013, p. 152):

A lista de atividades é uma lista abrangente que inclui todas as atividades necessárias no projeto. A lista de atividades também inclui o identificador de atividades e uma descrição do escopo de trabalho de cada atividade em detalhe suficiente para assegurar que os membros da equipe do projeto entendam qual trabalho precisa ser executado.

Ainda, de acordo com o PMI (2013), o cronograma é criado por meio de um processo de análise das sequências das atividades, de suas durações, dos recursos necessários e das restrições. Um cronograma deve apresentar as datas de início e de término de cada atividade. Já um diagrama de rede do cronograma é “uma representação gráfica das relações lógicas também chamadas de dependências, entre as atividades do cronograma do projeto” (PMI, 2013, p. 159). Desta maneira, a lista de atividades, o cronograma e o diagrama de rede do projeto encontram-se no Apêndice B deste documento.

4.2.2.2 Quadro de Resumo de Esforços

O Quadro 2 apresenta o resumo de esforços para o desenvolvimento de cada fase deste projeto. Os valores mostrados no quadro foram obtidos do cronograma e do apontamento das horas efetivamente gastas no projeto.

	Fase 1	Fase 2	Fase 3	Fase 4
Estimativa <i>Ad-hoc</i>	196/196h	320/516h	250/776h	315/1081h
Dicionário EAP	147/147h	258/405h	210/615h	144/759h
Efetivo realizado	147/147h	255/402h	195/597h	144h/741h

QUADRO 2 - Esforços planejados e realizados por fase

Na Fase 1 deste projeto, devido à falta de experiência com relação ao planejamento das atividades, foram alocadas uma grande quantidade de horas de folga para a realização dos pacotes de trabalho. Desta forma, o efetivo realizado ficou abaixo do que foi planejado de forma *ad-hoc* e no dicionário da EAP.

Na Fase 2 deste projeto, houve uma melhoria significativa na estimativa de horas, pois houve um amadurecimento da equipe e do entendimento das atividades junto com uma melhor abordagem para a divisão das tarefas. Com o aumento na capacidade de visão foi possível a divisão dos tópicos de forma a refletir melhor a realidade, com isso, aconteceu um aumento das horas em relação à Fase 1, pois as mudanças acarretaram em um foco mais minucioso das atividades e de cada item da entrega. Dessa forma, a estimativa *ad-hoc*, o dicionário da EAP e o efetivo realizado ficaram mais próximos.

Na Fase 3 deste projeto, no planejamento das atividades, foi considerada a quantidade de atividades relacionadas à codificação do sistema que a equipe deveria implementar. Portanto, estimou-se de forma *ad-hoc* uma quantidade de 250h, tendo como base as fases anteriores e suas respectivas horas gastas. Durante a construção do cronograma, o dicionário da EAP apresentou uma previsão de 210 horas e após o fechamento da Fase 3, o efetivo realizado ficou com 195 h, próximo do planejado. Este número reflete um amadurecimento da equipe de projeto em relação ao planejamento, já que nas fases anteriores as estimativas foram maiores.

Na Fase 4 deste projeto, estimou-se de forma *ad-hoc* uma quantidade de 315h. Este esforço foi estimado com base nas atividades necessárias a serem realizadas nessa fase e nas

experiências em fases anteriores. A divisão das tarefas foi realizada de forma a não sobrecarregar nenhum membro. Ao decorrer da fase, os membros da equipe foram executando suas respectivas atividades e ao final da fase, o somatório do trabalho real foi de 144h. As estimativas feitas pelo dicionário da EAP e o efetivo realizado tiveram valores muito próximos o que indica que houve um bom planejamento e execução das atividades por parte da equipe. As atividades fluíram de acordo com o planejado, colaborando com o bom andamento do projeto e sua finalização.

4.2.3 Gestão da Integração

A gestão da integração tem extrema importância para o projeto, uma vez que por meio dela é feito o monitoramento do andamento do projeto, o controle das alterações e o controle do desempenho de suas atividades.

Segundo o PMI (2013, p. 63):

No contexto de gerenciamento de projetos, integração inclui características de unificação, consolidação, comunicação e ações integradoras que são essenciais para a execução controlada do projeto até a sua conclusão, a fim de gerenciar com sucesso as expectativas das partes interessadas e atender aos requisitos.

A Figura 10 apresenta de forma geral a visão do gerenciamento da integração do projeto, citando cada um de seus processos.



FIGURA 10 - Visão geral do gerenciamento da integração do projeto – PMBOK – 5ª. ed.

FONTE: PMI (2013, p. 92)

Desta maneira, para explicar como é o processo da gestão da integração no projeto *SafeWalk*, as subseções seguintes descrevem as atividades de monitoramento, controle de configuração e controle das mudanças, dispostas respectivamente na ordem citada.

4.2.3.1 Monitoramento

Depois de concluída a fase de planejamento do projeto, se faz necessário o monitoramento do mesmo a fim de averiguar o cumprimento dos objetivos planejados. O monitoramento permite ao gerente de projeto identificar possíveis desvios do que foi planejado, promovendo ações corretivas ou preventivas.

Uma vez iniciado o projeto, o monitoramento contínuo é necessário, porque fornece uma compreensão clara da saúde do projeto, identificando quais áreas necessitam de atenção especial.

De acordo com o PMI (2013, p. 88): “O monitoramento inclui a coleta, medição e distribuição das informações de desempenho e a avaliação das medições e tendências para efetuar melhorias no processo”.

Assim, para análise do monitoramento do projeto *SafeWalk*, a planilha contendo o relatório de desempenho para cada fase do projeto encontra-se disponível no Apêndice C.

4.2.3.2 Controle de Configuração

O controle de configuração do projeto visa também revisar todas as solicitações de mudanças, aprovar as mudanças e gerenciá-las dentro do prazo de entrega. Ele é responsável por garantir a integridade dos artefatos produzidos, ou seja, que nenhuma mudança possa refletir de forma negativa no cronograma do projeto ou atrapalhar a entrega prevista.

Desta maneira, neste projeto, cada integrante da equipe é designado a executar uma ou mais atividades. Assim, um repositório foi criado no *GitHub* para o controle e *backup* das versões do projeto. Esse repositório é compartilhado com todos os membros da equipe, onde, todos possuem permissões para fazer as alterações que forem necessárias, porém, somente após comunicar os membros da equipe.

Além disso, a ferramenta *Google Docs* é utilizada na execução de atividades de maior importância, onde todos os membros da equipe participam. Essa ferramenta permite que todos editem em paralelo o mesmo documento.

Ademais, um segundo repositório foi criado no *Google Drive* para o controle das versões dos artefatos. Esse repositório é compartilhado com todos os integrantes da equipe e todos possuem permissão para alterar os artefatos à medida que for necessário ao longo do desenvolvimento do projeto.

Visto posto, o repositório criado no *Google Drive* possui 4 pastas principais designadas a cada membro da equipe para uma melhor organização. As tarefas enviadas pela equipe são revisadas pelo gerente do projeto e, posteriormente, adicionadas ao documento

principal. Além das pastas citadas, existe a pasta “artigos” que armazena os artigos técnico-científicos consultados durante a pesquisa e desenvolvimento do tema. Existe também a pasta “apoio”, que possui a função de armazenar itens diversos como *e-books*, tutoriais e documentos de consulta que auxiliam a equipe na execução do projeto.

A Figura 10 mostra o repositório criado para armazenar a documentação do projeto *SafeWalk*.

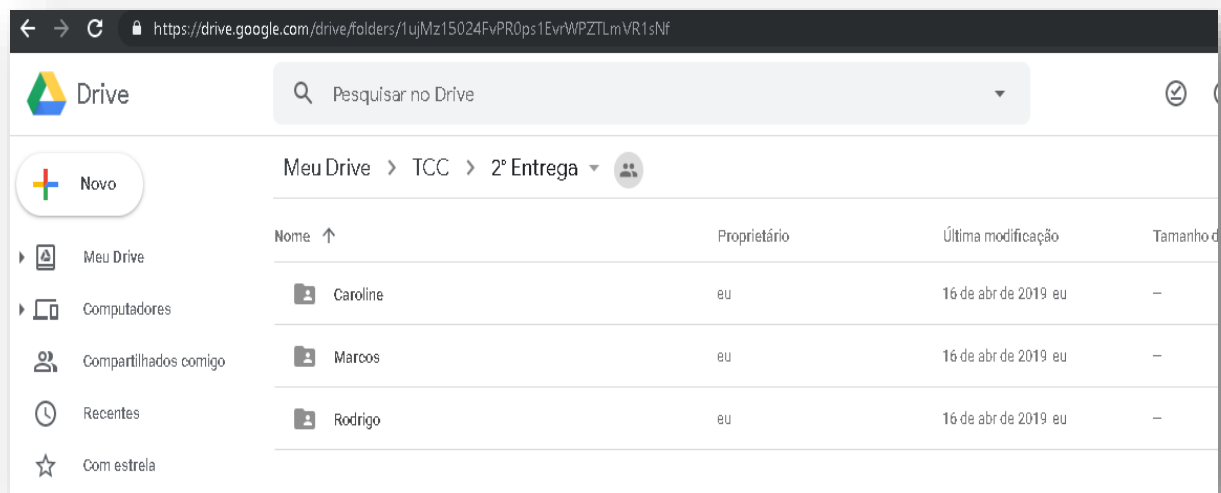


FIGURA 10 - Estrutura do repositório para a documentação
FONTE: Google Drive (2019)

Outrora, os arquivos de codificação do projeto ficam armazenados em diretórios no *GitHub*, organizados para facilitar o desenvolvimento utilizando o padrão *Model View Controller* (MVC). O diretório *model* possui toda a implementação das classes de negócio do projeto e o diretório *web* possui a codificação da interação com o usuário e todo o controle e direcionamento das funcionalidades.

4.2.3.3 Controle das Mudanças no Escopo

Durante a realização do projeto algum requisito proposto pode ter sua especificação alterada, bem como o planejamento pode ser alterado devido à aplicação de mudanças no projeto. Assim sendo, é necessário analisar os impactos e realizar um controle para que as mudanças sejam realizadas ou não.

As mudanças surgem, porque planejamentos realizados no início do projeto tendem a ter um nível elevado de incertezas, que posteriormente sofrem adaptações após o processo de entendimento. Se isso ocorrer, é necessário iniciar uma solicitação de mudanças.

Desta maneira, as mudanças podem ser solicitadas por qualquer parte interessada, devendo a mudança ser documentada e introduzida no gerenciamento de mudanças. As solicitações de mudanças, para serem aprovadas ou rejeitadas, devem passar por um processo de análise.

Ademais, para o controle de mudanças no projeto *SafeWalk*, formou-se o Comitê de Controle de Mudança, que é constituído por todos os membros da equipe. A solicitação de mudança é feita por *Whatsapp*, pois, dessa forma se mantém o registro do pedido. Em caso de solicitação de mudança, a alteração é passada por todos os membros do Comitê, sendo feita uma avaliação do custo dessa mudança, ou seja, o impacto que pode causar no projeto e o tempo demandado aos desenvolvedores para executá-la.

Em caso de a mudança ser aceita, a mesma é passada para um ou mais desenvolvedores da equipe. Em caso de a mudança ser aceita, mas o impacto no projeto for grande, é agendada uma reunião com a orientadora Eunice Gomes de Siqueira e, mediante sua opinião, a mudança será aprovada ou não. Não será feita a mudança caso a mesma não seja aceita pelo Comitê ou se ela não for aconselhada pela orientadora a ser realizada.

As propostas de mudanças apresentadas para o projeto *SafeWalk* encontram-se dispostas no Quadro 3.

Continua.

Solicitação de mudança	Descrição	Status	Responsáveis pela aprovação	Impacto da mudança
Remoção do <i>framework</i> Angular 6 do <i>front-end</i> .	Devido à complexidade e dificuldades encontradas durante a codificação com o <i>framework</i> Angular, somada à inexperience da equipe, foi solicitada a remoção deste framework. Esta	Mudança aprovada.	Comitê de mudanças.	Baixo

	remoção se faz necessária para diminuir os riscos de não finalizar o projeto dentro do prazo estipulado. Em seu lugar, será utilizada a tecnologia <i>Java Server Pages</i> (JSP), que já é dominada pela equipe, diminuindo assim os riscos de não cumprir os prazos.			
Criação do requisito Manter Cadastro de Áreas Inseguras	Para uma melhor utilidade do sistema, é necessário separar o cadastro de relatos do cadastro de áreas inseguras. Pois, ao criar um relato o usuário irá informar apenas o que ocorreu naquele local. Já o cadastro de áreas inseguras é mapeado pelo poder público.	Mudança aprovada.	Comitê de mudanças.	Médio
Remoção do <i>framework</i> Spring Boot do <i>back-end</i> .	Devido à complexidade e dificuldades encontradas durante a codificação com o Spring Boot, somada à inexperiência da equipe, foi solicitada a remoção desse <i>framework</i> . Esta remoção se faz necessária para diminuir os riscos de não finalizar o projeto dentro do prazo estipulado. Em seu lugar, será utilizado o padrão MVC, que já é dominado	Mudança aprovada.	Comitê de mudanças.	Baixo

	pela equipe, diminuindo assim os riscos de não cumprir os prazos.			
--	---	--	--	--

QUADRO 3 - Propostas de mudanças para o projeto *SafeWalk*

FONTE: elaborado pelos autores (2019)

4.2.4 Gestão da Qualidade

A gestão da qualidade é aplicada ao longo do tempo de todo desenvolvimento do projeto, tanto para o software quanto para a documentação. Esta seção tem como objetivo a apresentação detalhada de toda atividade de gestão da qualidade aplicada ao projeto. A figura 11 descreve cada processo de gestão da qualidade da seguinte forma:

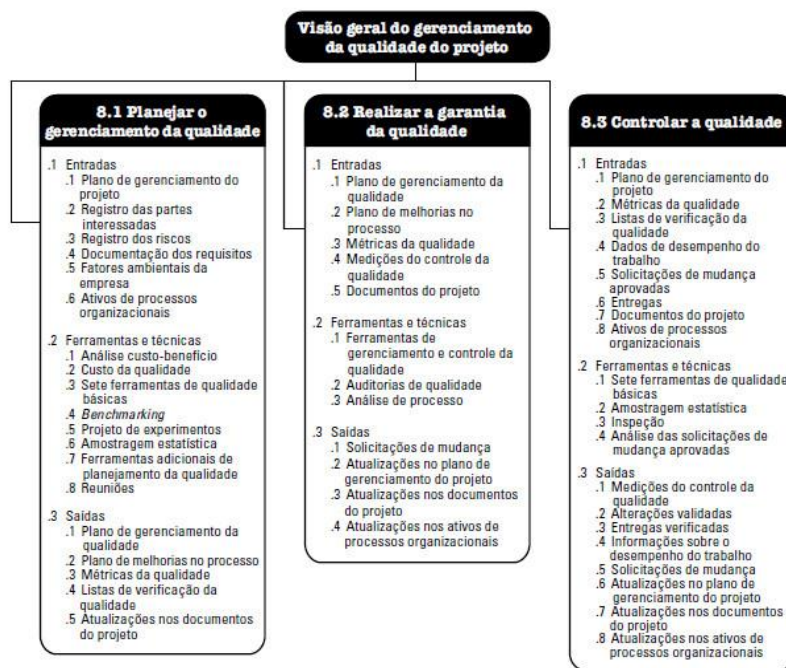


FIGURA 11 - Visão geral do gerenciamento da qualidade do projeto – PMBOK – 5ª. ed.

FONTE: PMI (2013, p. 227)

Durante todo o planejamento do processo de qualidade, são estipulados padrões a serem seguidos e revisões. As revisões verificam as adequações às convenções de programação *Java Code Conventions* para o projeto SafeWalk. Já para a elaboração do documento, é utilizado o documento com as diretrizes para elaboração de trabalhos científicos: padrão da Associação Brasileira de Normas Técnicas (ABNT) e adaptação às

normas institucionais da FAI - Centro de Ensino Superior em Gestão, Tecnologia e Educação. Em relação a nomenclatura dos documentos é adotado o padrão *SafeWalk* seguindo por *Underline* e posteriormente a sua versão. Um exemplo deste padrão é mostrado a seguir: SafeWalk_01.00. Para o controle da qualidade, alguns procedimentos necessários são tomados, sendo eles:

- a) antes de qualquer entrega de documentação ao final da fase, esta é revisada por todos os membros da equipe;
- b) a partir da segunda fase, os tópicos redigidos na documentação da fase anterior são revisados e corrigidos a partir das falhas apontadas pelos professores;
- c) a nomenclatura de documentos deve ser seguida segundo especificação criada na gerência da integração.

Assim, a fim de que a qualidade seja mantida, fica sob a responsabilidade do gerente de projeto a gerência da qualidade, que deve gerir as reuniões e obter retorno das atividades designadas para todos os membros. É obrigação de cada membro manter a qualidade dos artefatos designados a ele. Desta maneira, no Apêndice D encontra-se a lista com as verificações de qualidade realizadas no projeto.

4.2.5 Gestão dos Riscos

O risco do projeto é um evento ou condição incerta que, se ocorrer, afetará positiva ou negativamente um ou mais objetivos do projeto, escopo, cronograma, custo ou qualidade. Um risco pode ter uma ou mais causas e, se ocorrer, pode ter um ou mais impactos. A Figura 12 descreve cada processo de gestão de risco da seguinte forma:

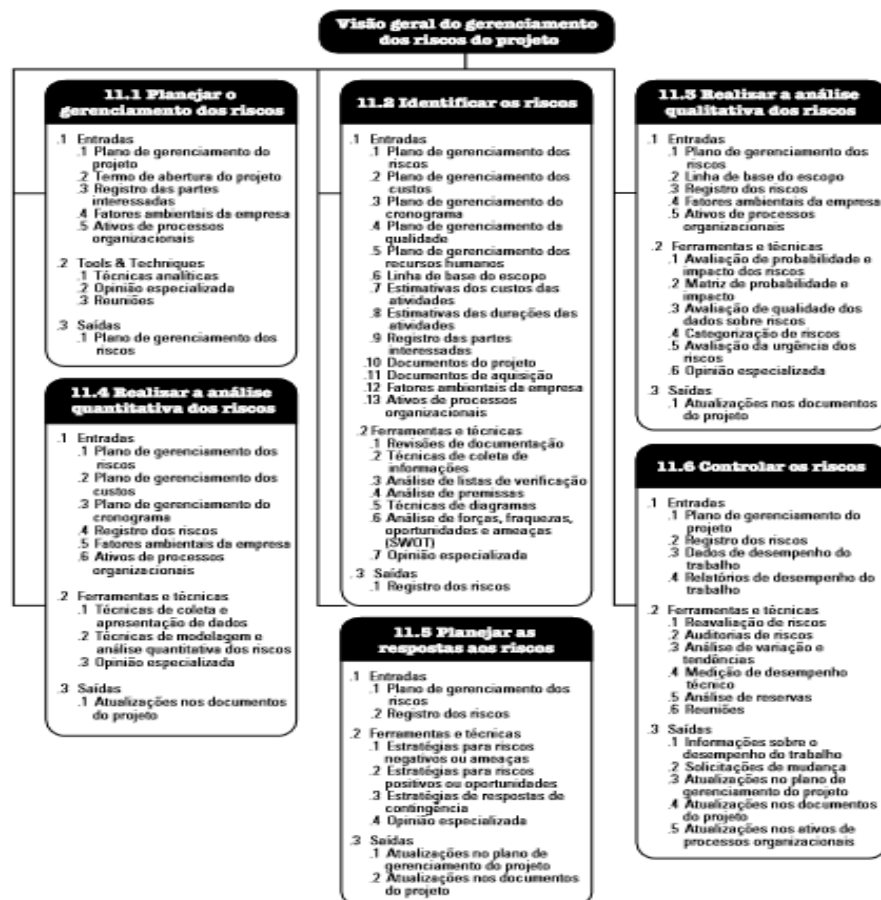


FIGURA 12 - Visão geral do gerenciamento dos riscos do projeto – PMBOK – 5ª. ed.
 FONTE: PMI (2013, p. 309)

Ainda de acordo com o PMI (2013), existem 4 estratégias para tratamento dos riscos de ameaças, sendo elas:

- eliminar;
- transferir;
- mitigar;
- aceitar.

Desta maneira, o Apêndice I contém a planilha de acompanhamento dos riscos de ameaças do projeto. Os riscos identificados são qualificados em relação à probabilidade de ocorrência, estratégia para tratamento e são definidos os planos de redução e contingência para cada risco.

5 ESPECIFICAÇÃO E ANÁLISE DOS REQUISITOS

Neste capítulo são abordados os requisitos funcionais e não funcionais do sistema *SafeWalk*, bem como as visões funcional e de dados do sistema de software. Para estabelecer a priorização dos requisitos funcionais os mesmos serão classificados em três níveis: essencial, importante e desejável. Uma vez definida, a priorização, ficara mais fácil resolver conflitos, planejar entregas e decidir sobre quais requisitos implementar. A decisão de priorizar as necessidades de um projeto deve ser tomada a partir de uma parceria entre as partes interessadas.

5.1 REQUISITOS DO SISTEMA DE SOFTWARE

Segundo Sommerville (2011), os requisitos de sistema de software são divididos em requisitos funcionais e não funcionais. Os requisitos funcionais especificam como será o comportamento do sistema em determinadas situações e o seu comportamento para entradas de dados específicas. Em alguns casos, o requisito também define o que o sistema não irá fazer. Já os requisitos não funcionais são as limitações e restrições do sistema, como as de tempo, no processo de desenvolvimento ou impostas por normas e regulamentos organizacionais. Desta maneira, este capítulo mostra os requisitos do sistema, contendo os requisitos funcionais e os requisitos não funcionais.

5.1.1 Requisitos Funcionais

A descrição de requisitos é uma parte importante para todo o desenvolvimento, já que ela explicita as necessidades que devem ser atendidas pelo sistema. Assim, os requisitos de um sistema são descrições de um serviço fornecido pelo sistema e as suas restrições operacionais. Esses requisitos refletem as necessidades dos clientes de um sistema que ajuda a resolver algum problema, por exemplo, controlar um dispositivo, enviar um pedido ou encontrar informações (SOMMERVILLE, 2011, p.79).

Além disso, para um melhor entendimento, os requisitos funcionais do sistema serão classificados de acordo com a ordem de prioridade: essencial, importante e desejável. Um requisito essencial é fundamental para o sistema e quando não são implementados impedem a implantação ou a conclusão do projeto. Já um requisito importante deve ser parte do escopo, mas não bloqueia o sistema a entrar em produção. Por fim, um requisito priorizado como

desejável não é indispensável para o sistema estar completo. No entanto, sem ele o sistema deve funcionar de maneira satisfatória, atendendo completamente o seu objetivo.

Nesta seção, os requisitos funcionais (RF) estão agrupados por módulos, a saber:

- a) Módulo de Acesso;
- b) Módulo para o Cidadão;
- c) Módulo do Administrador;
- d) Módulo para o Poder Público.

5.1.1.1 Módulo de Acesso

Neste módulo estão os requisitos referentes ao modo como todos os usuários do sistema farão o acesso à aplicação.

RF01 - Autenticar

Este requisito refere-se ao modo como um usuário previamente cadastrado na aplicação irá realizar a sua identificação. O usuário deverá fornecer (1) e-mail e (2) senha previamente cadastrados, para ter acesso às demais funcionalidades do sistema. Caso ele ainda não esteja cadastrado, haverá a possibilidade de se cadastrar na aplicação (RF04). Em caso de perda da senha, ela poderá ser recuperada pelo usuário após a verificação da sua situação no sistema. Se a situação do usuário estiver como ativa, basta informar o seu e-mail previamente cadastrado (RF02).

Prioridade: essencial.

RF02 - Recuperar Senha

Este requisito refere-se ao modo como um usuário previamente cadastrado poderá recuperar uma senha esquecida, para conseguir ter acesso à aplicação. Ao solicitar uma recuperação de senha, o usuário deverá informar (1) e-mail de autenticação e se estiver cadastrado e com situação ativa no sistema, uma nova senha será enviada para seu e-mail.

Prioridade: essencial.

RF03 - Alterar Senha

Este requisito refere-se ao modo de como um usuário previamente cadastrado poderá alterar sua senha quando achar necessário. Para realizar a alteração de sua senha, o usuário precisará informar a (1) senha atual, (2) a nova senha e (3) confirmar a nova senha, que precisará estar de acordo com as regras de composição de senha. A troca da senha será feita no momento em que o usuário confirmar a sua intenção e mediante a entrada correta da senha corrente.

Prioridade: essencial.

RF04 - Manter Cadastro de Usuário

Este requisito refere-se ao cadastro de usuários, que será realizado por eles próprios na aplicação, com exceção do usuário do tipo “administrador”. O sistema terá um administrador inicial que será responsável por criar outras contas do tipo “administrador”. Para realizar o cadastro de um usuário, deverão ser informados: (1) nome completo, (2) CPF, (3) sexo, (4) data de nascimento, (5) endereço composto por: (logradouro, número e bairro), (6) município, (7) UF, (8) CEP, (9) telefone fixo, (10) telefone celular, (11) e-mail, (12) senha, (13) situação (verifica se o usuário está ativo ou não na aplicação) e (14) data de realização do cadastro. Após a consistência dos dados ser realizada uma mensagem de notificação informando que o cadastro foi realizado é enviada. Caso algum erro seja encontrado, uma nova tentativa será necessária.

Prioridade: essencial.

RF05 - Aceitar Termo de Uso

Este requisito refere-se à aceitação do termo de uso do sistema. Assim, para a finalização do cadastro é necessário que o usuário leia e aceite o termo de uso do sistema.

Prioridade: essencial.

5.1.1.2 Módulo para o Cidadão

Este módulo apresenta os requisitos necessários para o cidadão interagir com o sistema como um todo.

RF06 - Visualizar Mapa

Este requisito refere-se à capacidade do cidadão visualizar a sua região em um mapa geográfico tridimensional. O usuário deverá ativar a geolocalização do dispositivo pelo qual a aplicação é acessada. Para isso deve informar: endereços de (1) origem e (2) destino. Caso o sistema de geolocalização esteja desativado, uma mensagem será mostrada solicitando a sua ativação.

Prioridade: essencial.

RF07 - Manter Cadastro de Relatos

Este requisito refere-se ao modo como um usuário poderá cadastrar, visualizar, alterar e excluir um relato. Para realizar esse cadastro, deverão ser informados: (1) título para o relato, (2) data de criação, (3) autor, (4) descrição e (5) categoria dividida em: (iluminação, assaltos, dentre outras).

Prioridade: essencial.

RF08 - Avaliar Relato

Este requisito refere-se ao modo como um usuário poderá avaliar um relato cadastrado na aplicação. Para avaliar um relato é necessário informar: (1) classificar o relato em útil ou não útil, (2) comentar o relato e (3) a data da avaliação.

Prioridade: essencial.

RF09 - Mostrar e Editar Perfil de Usuário

Este requisito refere-se à visualização e edição do perfil do usuário. Para tal, o sistema deve fornecer para cada tipo de usuário, acesso à sua respectiva conta. Desta maneira, o mesmo pode visualizar os campos preenchidos durante o cadastro e caso seja necessário fazer as devidas alterações.

Prioridade: essencial.

RF10 - Remover Conta

Este requisito refere-se à possibilidade de um usuário remover a sua conta do sistema por completo.

Prioridade: essencial.

5.1.1.3 Módulo do Administrador

Este módulo possui as funcionalidades que permitirão a supervisão do funcionamento da aplicação, buscando um ambiente confiável para todos os usuários.

RF11 - Gerenciar Relatos

Para um melhor gerenciamento da aplicação, um administrador poderá visualizar todos os relatos cadastrados.

Prioridade: importante.

RF12 - Gerenciar Usuários

Para um melhor gerenciamento da aplicação, um administrador poderá visualizar todos os cadastros de usuários. Ele poderá também aplicar advertências aos usuários e comunicando-os os motivos. Em casos de reincidência, o administrador poderá tornar uma conta de usuário inativa.

Prioridade: importante.

5.1.1.4 Módulo para o Poder Público

Este módulo apresenta os requisitos necessários para o poder público (como Polícia Militar e Secretaria de Segurança Pública) interagir com o sistema.

RF13 - Manter Cadastro de Áreas Inseguras

Este requisito refere-se ao modo como um usuário poderá cadastrar, visualizar, alterar e excluir uma área insegura do mapa. Para realizar esse cadastro, deverão ser informados: (1) latitude (2) longitude e (3) raio.

Prioridade: essencial.

5.1.2 Requisitos não Funcionais

Os requisitos não funcionais são aqueles que não estão diretamente relacionados com as funcionalidades específicas oferecidas pelo sistema a seus usuários. Eles podem estar relacionados às propriedades emergentes do sistema, como confiabilidade, tempo de resposta e ocupação de espaço em disco (SOMMERVILLE, 2011, p. 60). Assim, nesta seção, os requisitos não funcionais do sistema são apresentados.

5.1.2.1 Requisitos de Produto

Os requisitos de produto referem-se a como o produto deve comportar-se, ou seja, a sua velocidade de execução, confiabilidade, dentre outros.

a) Requisitos de Eficiência

RNF01 - Disponibilidade

Para um desempenho satisfatório, a aplicação deverá estar disponível ao usuário 24 horas por dia em 7 dias por semana, salvo as paradas planejadas para manutenção. A disponibilidade requerida será de 99,98% do tempo ao longo do ano.

b) Requisitos de Confiabilidade**RNF02 - Confiabilidade dos dados**

O sistema deverá aplicar verificações nos valores de dados fornecidos pelos usuários, de modo a evitar dados inconsistentes e manter a integridade dos dados mantidos pelo Sistema Gerenciador de Banco de Dados (SGBD).

c) Requisitos de Portabilidade**RNF03 - Navegadores Web**

A aplicação *SafeWalk* deverá ser manipulada por meio dos seguintes navegadores: Google Chrome versão 68.0; Firefox 61.0 ou Microsoft Edge 4.4.

d) Requisitos de Usabilidade**RNF04 - Interface Web responsiva**

A aplicação deverá adotar um design responsivo, desta forma as interfaces de usuário devem ser projetadas para serem exibidas com qualidade e comportamento adequado nos mais variados tamanhos de telas dos dispositivos de acesso previamente planejados.

RNF05 - Interface de usuário simplificada

Para uma melhor adaptação do usuário, a aplicação deverá ter uma interface de usuário de fácil entendimento, para que ele aprenda a usá-la rapidamente e tenha uma experiência agradável durante a navegação.

5.1.2.2 Requisitos Organizacionais

Os requisitos organizacionais definem como será o processo de desenvolvimento do sistema, especificando as linguagens de programação, o ambiente de desenvolvimento ou as normas de processo a serem seguidas.

a) Requisitos de Entrega**RNF06 - Datas de entrega**

Ao longo do projeto deverão ser realizadas a entrega de 4 incrementos do sistema, conforme já definidos no Quadro 1.

b) Requisitos de Codificação

RNF07 – Frameworks de desenvolvimento

A aplicação deverá ser implementada utilizando a biblioteca *Java Server Pages Standard Tag Library* (JSTL), com *Bootstrap* para o componente cliente (*frontend*) e o *framework Spring MVC* para o componente servidora (*backend*).

RNF08 - Ambiente de desenvolvimento

A aplicação deverá ser implementada utilizando as IDEs Netbeans 8.2 para o desenvolvimento em Java 8 e o PostgreSQL 9.6.1 para o Sistema Gerenciador de Banco de Dados (SGDB).

c) Requisitos de Padrões

RNF09 - Padrões de documentação

O documento técnico do sistema deverá seguir as normas para trabalhos científicos da ABNT e as Diretrizes para a Elaboração de Trabalhos Acadêmicos da FAI.

RNF10 - Padrões de modelagem e convenções para codificação

A linguagem de modelagem será a UML e a programação do código fonte seguirá as convenções propostas pela *Java Code Conventions* para a linguagem Java.

5.1.2.3 Requisitos Externos

A fim de garantir que o sistema opere de acordo com a legislação vigente, os requisitos externos abrangem tópicos decorrentes de fatores externos ao sistema e ao processo de desenvolvimento.

a) Requisitos de Interoperabilidade**RN11 - Interoperabilidade**

O sistema deverá interagir com o SGDB PostgreSQL 9.6.1; com o *container* de aplicação Apache Tomcat 9; com o servidor *Simple Mail Transfer Protocol* (SMTP), para o envio de e-mails e com o provedor de mapas *Google Maps*. O sistema operacional escolhido deve ter compatibilidade com o Apache Tomcat 9 (por exemplo, Windows 10).

b) Requisitos de Privacidade e Segurança**RNF12 - Protocolo de comunicação segura**

O protocolo *Hypertext Transfer Protocol Secure* (HTTPS) deverá ser utilizado na comunicação entre os navegadores compatíveis e o Apache Tomcat 9.

RNF13 - Autenticação

O acesso aos módulos restritos do sistema deverá ser permitido apenas a usuários que estejam autenticados e ativos, fornecendo seu e-mail e senha.

RNF 14 - Termo de uso

Para que o acesso ao sistema seja permitido, todos os usuários no momento do cadastro devem aceitar o termo de uso apresentado. Somente após a autorização do termo de uso o acesso será liberado.

RNF 15 - Criptografia dos dados

Para que o usuário possa acessar o sistema de uma forma segura, os dados de acesso serão criptografados no momento em que forem inseridos no banco de dados.

RNF16 - Regras para formação de senhas

Deverão ser aceitas senhas que atendam à regra de ter no mínimo 8 caracteres, sendo compostos de pelo menos um caractere especial, um número e uma letra em seus 8 dígitos ou mais.

5.2 ANÁLISE DOS REQUISITOS

Para representar a análise dos requisitos funcionais, utiliza-se o modelo de casos de uso, que é composto pelo Diagrama de Casos de Uso e pela descrição dos fluxos de eventos de cada um deles.

5.2.1 Modelos de Caso de Uso

De acordo com Pressman (2013, p.137):

Essencialmente, um caso de uso conta uma história estilizada sobre um usuário final (desempenhando um de uma série de papéis possíveis) interage com o sistema sob um conjunto de circunstâncias específicas. A história poderia ser um texto narrativo, uma descrição geral das tarefas ou interações, uma descrição baseada em gabaritos ou uma representação esquemática. Independentemente de sua forma, um caso de uso representa o software ou o sistema do ponto de vista do usuário final.

Ainda segundo Pressman (2013), o primeiro passo para construir um caso de uso é definir o conjunto de atores envolvidos no sistema. Um ator é qualquer elemento que se comunica com o sistema e está externo à fronteira dele. Todo ator possui uma ou mais metas ao utilizar o sistema. Desta maneira, esta seção apresenta o diagrama de cada caso de uso referente ao sistema *SafeWalk* e suas respectivas descrições. Os diagramas dos casos de uso encontram-se no Apêndice E. As descrições dos casos de uso estão no Apêndice F deste documento.

5.2.2 Modelo Conceitual dos Dados

A modelagem de dados é essencial durante o desenvolvimento de qualquer sistema computacional, tendo como objetivo descrever de maneira conceitual os dados utilizados pelo sistema. Para a representação do modelo conceitual dos dados, é utilizado modelo entidade-relacionamento, com o objetivo de especificar os dados utilizados pelo sistema. Desta maneira, o Diagrama Entidade-Relacionamento encontra-se disponível no Apêndice G deste documento.

5.2.3 Modelo Inicial de Interface Usuário

O Modelo Inicial da Interface de Usuário encontra-se disponível no Apêndice H deste documento.

5.3 MÉTRICAS PARA ESTIMATIVA DE ESFORÇO

Antes de começar um projeto de desenvolvimento de software é importante mensurar o seu tamanho para que dele se possa estimar o tempo que será gasto em sua construção e realizar uma melhor distribuição dos recursos disponíveis. Para isso, no *SafeWalk* são utilizadas as métricas de Pontos de Casos de Uso (PCU) e de Pontos de Função (PF) para estimar o tamanho e esforço de desenvolvimento.

Segundo Andrade e Oliveira (2005, p.16), Pontos de Casos de Uso estimam o tamanho de sistemas de software orientados a objetos com base no modelo de casos de uso gerado no início do projeto. O tamanho do projeto é estimado com um cálculo baseado na complexidade dos atores e suas interações com o sistema. Nesse cálculo, segundo Andrade e Oliveira (2005, p.16) primeiro é calculado o PCU não ajustados, para isso, soma-se o peso dos atores e o peso dos casos de uso. Em seguida é calculado o PCU ajustado que é obtido da multiplicação entre os PCU não ajustados pelo fator de complexidade técnica (características internas do sistema) e pelo fator de complexidade ambiental (são as características do ambiente em que será desenvolvido o projeto).

Já segundo Bomfim e Andrade (2015, p. 6), o Ponto de Função é uma unidade com o objetivo de medir o tamanho de um sistema de software, não importando a tecnologia a ser utilizada no projeto. Esta unidade busca medir o que o software faz e não como foi construído, estando diretamente relacionado aos requisitos que o software deseja abordar. Devido às suas características, ela pode ser usada em diversos ambientes de desenvolvimento durante todas as suas fases. Essa técnica auxilia na avaliação, planejamento, gestão e controle da produção de software por fornecer uma medida objetiva e comparativa.

As estimativas de esforços encontradas para o *SafeWalk* utilizando as métricas de Ponto de Casos de Uso e Pontos de Função são apresentadas no Quadro 4.

Métrica	Fase 2	Fase 3 (Reestimativa)	Fase 4 (Reestimativa)
PCU – Schneider e Winters	2590 Hh	2924Hh	2924Hh
PCU – Karner	2590 Hh	2924Hh	2924Hh
PF – Pontos por Função	-	2240Hh	2240Hh

QUADRO 4 - Estimativas de Esforços

A planilha com os cálculos realizados para calcular os Pontos de Caso de Uso e Pontos de Função encontram-se no Apêndice S deste documento.

6 ARQUITETURA E PROJETO DO SISTEMA DE SOFTWARE

A arquitetura e projeto têm como objetivo definir a estrutura e comportamento dos componentes internos do sistema.

Segundo Sommerville (2011, p.103), a estrutura geral de um sistema e a compreensão de como ele deve ser organizado é uma preocupação das atividades de projeto de sistemas (ou *design*). É nesta etapa que ocorre a identificação dos principais componentes estruturais e os relacionamentos entre eles, sendo considerado o elo crítico entre a engenharia de requisitos e as atividades de *design*. O resultado desta etapa é um modelo de arquitetura que demonstra como o sistema está organizado em um conjunto de componentes e de sua comunicação.

Desta maneira, neste capítulo são apresentadas as visões estrutural, comportamental, de dados e física e o projeto da interação humano-computador e de um sistema distribuído.

6.1 VISÃO ESTRUTURAL

Nesta seção encontram-se os diagramas estruturais utilizados na modelagem e documentação do *SafeWalk*, descrevendo o conjunto de pacotes e classes necessárias para a construção desse sistema.

6.1.1 Diagrama de Pacotes

O diagrama de pacotes descreve os pacotes do sistema divididos em agrupamentos lógicos mostrando as dependências entre eles.

No contexto da UML:

Um pacote é um mecanismo de propósito geral para organizar elementos em grupos. Os pacotes ajudam a organizar os elementos em modelos, de maneira que você seja capaz de compreendê-los com maior facilidade. Os pacotes também permitem controlar o acesso a seus conteúdos, de modo que você possa controlar as costuras existentes na arquitetura do sistema. (BOOCH; RUMBAUGH; JACOBSON, 2006, p.169).

Assim, o diagrama de pacotes do *SafeWalk* encontra-se disponível no Apêndice J deste documento.

6.1.2 Diagrama de Classes

Diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos.

De acordo com Sommerville (2011, p. 90):

Os diagramas de classe são usados no desenvolvimento de um modelo de sistema orientado a objetos para mostrar as classes de um sistema e as associações entre essas classes. Em poucas palavras, uma classe de objeto pode ser pensada como uma definição geral de um tipo de objeto do sistema. Uma associação é um link entre classes que indica algum relacionamento entre essas classes. Consequentemente, cada classe pode precisar de algum conhecimento sobre sua classe associada.

Assim, o diagrama de classes do pacote *Entity* do projeto *SafeWalk* encontra-se no Apêndice K deste documento.

6.1.3 Diagrama de Objetos

Um diagrama de objetos representa uma instância específica de um diagrama de classes em um determinado momento.

De acordo com Booch, Rumbaugh e Jacobson (2006, p.187):

Os diagramas de objetos fazem a modelagem de instâncias de itens contidos em diagramas de classes. Um diagrama de objetos mostra um conjunto de objetos e seus relacionamentos em determinado ponto no tempo. Você usa os diagramas de objetos para fazer a modelagem da visão estática do projeto ou do processo de um sistema. Isso envolverá a modelagem de retrato do sistema em determinado momento e a representação de um conjunto de objetos, seus estados e relacionamentos.

Os diagramas de objetos para os casos de uso “Manter Cadastro de Relatos” e “Autenticar” encontram-se no Apêndice L deste documento.

6.2 VISÃO COMPORTAMENTAL

A visão comportamental demonstra como o sistema atuará a partir de eventos ou estímulos externos. Por meio dela, é possível apresentar a relação entre os objetos das classes

e as mensagens trocadas entre eles. Nesta seção encontra-se o projeto de interações e diagrama de atividades.

6.2.1 Projeto das Interações Entre Objetos

Esta subseção apresenta os aspectos dinâmicos decorrentes da colaboração entre objetos do sistema, ou seja, o comportamento do sistema desenvolvido.

6.2.1.1 Diagrama de Sequência

Os diagramas de sequência representam a sequência das trocas de mensagens entre os objetos de um sistema, em determinado cenário.

Segundo Pressman (2013), diagramas de sequência indicam como os eventos provocam transições de objeto para objeto, uma representação de como os eventos provocam o fluxo de um objeto para o outro em função do tempo.

Desta maneira, os diagramas de sequência dos casos de uso “Manter Cadastro de Áreas Inseguras” e “Manter Cadastro de Relatos” deste projeto encontram-se no Apêndice O.

6.2.1.2 Diagrama de Visão Geral da Interação

Os diagramas de visão geral de interação fornecem um alto nível de abstração em um modelo. Segundo Booch, Rumbaugh e Jacobson (2006, p. 258), um diagrama de visão geral da interação tem por objetivo estruturar as atividades, criando atividades subordinadas, que não se restringem ao diagrama de atividades, pode ser usado também pelo diagrama de sequência.

Desta maneira, um diagrama de visão geral da interação do projeto *SafeWalk* encontra-se no Apêndice P.

6.2.2 Diagrama de Atividades

Um diagrama de atividade é essencialmente um gráfico de fluxo, mostrando o fluxo de controle de uma atividade para outra e serão empregados para fazer a modelagem de aspectos dinâmicos do sistema.

O diagrama de atividades representa o fluxo de tarefas que podem ser executadas pelo sistema ou por um ator. Pode-se definir uma atividade como sendo uma tarefa que necessita ser realizada por uma pessoa ou computador. Em perspectiva de implementação, uma atividade pode representar um método sobre uma classe (FURLAN, 1998).

Desta maneira, um diagrama de atividades do caso de uso “Manter Cadastro de Usuários” do projeto *SafeWalk* encontra-se no Apêndice Q deste documento.

6.3 VISÃO DOS DADOS

Esta seção apresenta os itens relacionados à visão física do projeto, dividindo-se em Modelo Lógico e o respectivo dicionário de dados.

6.3.1 Modelo Lógico

Por meio do Diagrama Entidade-Relacionamento (DER), desenvolvido na ferramenta *Software Ideas Modeler*, foi elaborada a modelagem lógica dos dados, com o objetivo de fornecer uma visão precisa de como os dados serão armazenados em seu banco de dados relacional.

Assim, o modelo lógico encontra-se no Apêndice M deste documento.

6.3.2 Dicionário de Dados do Modelo Lógico

O dicionário de dados consiste numa lista organizada de todos os elementos de dados que são pertinentes a um sistema.

Segundo Sommerville (2011), um dicionário de dados é uma lista de nomes incluídos nos modelos do sistema, onde são descritas suas entradas, saídas e composição de dados.

Assim, o dicionário de dados do modelo lógico encontra-se no Apêndice R.

6.4 VISÃO FÍSICA

A visão física ou de implementação, segundo Booch, Rumbaugh e Jacobson (2007, p.33), abrange os componentes e os arquivos para a montagem e fornecimento do sistema físico. Essa visão envolve o gerenciamento da configuração das versões do sistema, composto por arquivos e componentes independentes que podem ser reunidos de diferentes formas para a produção de um sistema.

6.4.1 Diagrama de Componentes

O diagrama de componentes é utilizado para demonstrar como estão organizados os componentes e suas dependências. Desta maneira, segundo Booch, Rumbaugh e Jacobson (2007, p.387), para a modelagem da visão estática de implementação de um sistema, deve ser utilizado o diagrama de componentes.

Uma vez que, os diagramas de componentes são importantes para visualizar, especificar, documentar sistemas baseados em componentes e elaborar executáveis por meio de engenharia de produção e reversa. Assim, o diagrama de componentes do projeto *SafeWalk* encontra-se disponível no Apêndice U.

6.4.2 Frameworks Adotados

Framework é um conjunto de técnicas, ferramentas ou conceitos pré-definidos usados para resolver um problema de um projeto em um domínio específico.

Desta maneira, os *frameworks* utilizados no projeto *SafeWalk* são:

- a) **Bootstrap:** é um *framework* para facilitar o desenvolvimento de páginas *Web*. Este *framework* utiliza HTML, CSS e *Javascript* para criar páginas *Web* com interfaces agradáveis e responsivas, melhorando a experiência do usuário;

- b) **JSTL:** um acrônimo para *Java Server Pages Standard Template Library*, sendo um componente da plataforma *Java Web* que é utilizado para adicionar uma biblioteca de *tags*. Estas *tags* são utilizadas para executar algumas tarefas comuns como a execução condicional e criar *loops*. Este componente é utilizado para melhorar a forma de adicionar lógica de programação em uma JSP, pois o uso de *tags* simplifica a forma de criar ou editar uma página;
- c) **Spring MVC:** O Spring MVC é um *framework* que ajuda no desenvolvimento de aplicações *Web*. Nessa arquitetura, a camada *Controller* processa as requisições e as interconecta com o segmento *Model*, onde ficam armazenadas as regras essenciais. Já na parte denominada *View* estão concentradas as interações com os usuários;
- d) **API Google Maps:** é um serviço disponibilizado pela *Google* de forma gratuita para que qualquer pessoa possa utilizar em seu site. Esta API é acionada via trechos de código em *Javascript* e permite manipular mapas, utilizando funções disponibilizadas pela *Google*, tais como traçar rotas ou encontrar uma localidade.

6.4.3 Design Patterns Aplicados

Design patterns ou padrões de projeto devem ser compreendidos como uma solução geral para um problema que ocorre com frequência dentro de um determinado contexto no projeto de software. Os *designs Patterns* usados no projeto *SafeWalk* podem ser compreendidos a partir do Quadro 5.

Propósito	Padrões utilizados	Classes e/ou métodos aplicados
Criacional	<i>Singleton</i>	ConnectionManager.java
Outros padrões	MVC <i>Data Access Object (DAO)</i> <i>Create Read Update Delete (CRUD)</i> <i>Criteria</i>	SpringMVC

QUADRO 5 - *Design Patterns* no projeto *SafeWalk*

- a) **MVC:** é um padrão arquitetural que possui como premissa básica fornecer uma divisão trilateral, isto é, *controller*, *view* e *model*. Os *controllers* são responsáveis por fazer a conexão entre o usuário e as regras de negócio contidas no *model*, assim, é por meio deles que as solicitações e requisições das *views* são transmitidas ao *model* e retornadas à *view*. O *model* é responsável por conter todas as regras de negócio da aplicação e, além disto, as classes de persistência com o sistema gerenciador de banco de dados. A *view* é a camada na qual o usuário interage diretamente com o sistema;
- b) **DAO:** é um padrão utilizado que representa o pacote que é composto pelas classes de persistência com o banco de dados. Estas classes são compostas de métodos de leitura, gravação, alterações, exclusão e inserção de elementos nas tabelas do Banco de Dados;
- c) **CRUD:** é um padrão que está relacionado com o pacote que comporta as classes que fazem as requisições necessárias ao DAO. Além disto, as classes que compõe este pacote são responsáveis por abrir conexão com o banco de dados e, posteriormente, fechá-la;
- d) **criteria:** é um padrão que corresponde a um conjunto de classes que possuem como principal finalidade abrigar todos os critérios de busca no banco de dados;
- e) **Singleton:** é uma abordagem que permite que apenas uma instância de cada classe seja criada por vez e seja acessada por um único ponto global. Sendo assim, isto pode resultar em uma aplicação com um maior desempenho. No sistema *SafeWalk* é utilizado o padrão *Singleton* na classe que faz a configuração da conexão com o banco de dados.

6.4.4 Convenções e Guias para Codificação

As convenções de codificação definem um conjunto de regras que deve ser seguido durante a escrita do código fonte da aplicação. Essas regras têm por objetivo formatar o código, bem como auxiliar na interação da equipe de desenvolvimento. Seguindo essas regras, pode-se fazer com que o código se torne mais manutenível. Assim, as convenções de codificação aplicadas no projeto *SafeWalk* são moldadas com base nas *Java Code Conventions*. As convenções utilizadas são respectivamente:

- a) **Pacotes:** todos os pacotes são nomeados com letras minúsculas e devem seguir uma determinada hierarquia, sendo separados por ponto. Exemplo: “safewalk.model”;

- b) **Classes e interfaces:** são nomeadas utilizando um ou mais substantivo sem a utilização de caracteres especiais. O primeiro caractere de cada palavra deve ser em letra maiúscula. Exemplo: “UsuarioService.class”;
- c) **Métodos e atributos:** podem ser nomeados com uma ou mais palavras, sendo que a primeira letra da primeira palavra deve ser em minúscula e as demais palavras começam com a primeira letra em maiúscula. Os métodos normalmente devem ser compostos por verbos ou frases verbais. Exemplo: “public usuario readById()”;
- d) **Variáveis:** normalmente são utilizados substantivos para nomeação das variáveis. Pode-se utilizar mais de uma palavra na formação do nome da variável. Sua nomeação a primeira letra em maiúscula. Exemplo: “private dataNasc”;
- e) **Constantes:** toda constante deve ser declarada em letras maiúsculas. O nome de uma constante pode utilizar uma ou mais palavras para sua constituição, sendo que em caso de nomes compostos, as mesmas são separadas pelo caractere underline (_).

6.5 PROJETO DO SISTEMA DISTRIBUÍDO

Um sistema distribuído é um conjunto de computadores independentes e interligados por uma rede de conexão, executando um software distribuído por ela. Os processos deste sistema são executados de forma concorrente, interagem para alcançar um objetivo em comum e a sua coordenação é feita por troca de mensagens entre os componentes.

Segundo Coulouris, Dollimore e Kindberg (2007, p.16), um sistema distribuído pode ser definido como um sistema em que seus componentes são interligados por uma rede de computadores, podendo ser hardware ou software, separados por qualquer distância e que enviam mensagens entre si para se comunicarem e coordenarem suas ações.

Nesta seção é mostrado como é o tratamento dos desafios de projeto de um sistema distribuído, sendo eles: heterogeneidade, escalabilidade, abertura, segurança, manuseio de falhas, concorrência e transparência, além das tecnologias e arquiteturas de distribuição do sistema.

Para a representação dos componentes e seus respectivos papéis na composição do sistema, utiliza-se o diagrama arquitetural mostrado no final desta seção.

6.5.1 Procedimentos para Tratamentos dos Desafios

No desenvolvimento de um sistema distribuído, as barreiras encontradas pela equipe de desenvolvimento são consideradas desafios do projeto e elas devem ser superadas para que o sistema a ser implementado possua a qualidade esperada pelas partes interessadas.

6.5.1.1 Heterogeneidade

A heterogeneidade é o acesso a serviços e aplicativos por meio de um conjunto heterogêneo de computadores pela Internet, sendo ela o principal exemplo de um sistema heterogêneo, por possuir muitos tipos de redes com suas diferenças mascaradas por protocolos presentes em todos os computadores conectados a ela. Uma das formas de superar o desafio da heterogeneidade é o uso de uma camada intermediária entre o sistema operacional e a aplicação, conhecida como *middleware* (COULOURIS; DOLLIMORE; KINDBERG, 2007).

Ainda, segundo Coulouris, Dollimore e Kindberg (2007, p.28), um *middleware*, além de resolver problemas de heterogeneidade, fornece um modelo computacional usado pelos programadores de serviços e aplicativos distribuídos, sendo eles a invocação remota de objetos, a notificação remota de eventos, o acesso remoto a banco de dados e o processamento de transação distribuído. A implementação desses modelos computacionais permite que uma mensagem passe por uma rede heterogênea, permitindo assim que um componente possa realizar um pedido ou receber uma resposta.

Assim, para superar o desafio da heterogeneidade, o sistema *SafeWalk* pode ser executado em diversos sistemas operacionais que possuam uma versão compatível da *Java Virtual Machine* (JVM) e utiliza o protocolo em nível de aplicação (HTTPS) para comunicação dos dados entre os componentes.

6.5.1.2 Escalabilidade

A escalabilidade é a característica de um sistema distribuído de manter o mesmo comportamento independentemente do número de usuários que o acessam. “Um sistema é descrito como escalável se permanece eficiente quando há um aumento significativo no número de recursos e no número de usuários” (COULOURIS; DOLLIMORE; KINDBERG, 2007, p.25).

No entanto, o desafio da escalabilidade não é tratado no projeto do *SafeWalk*. Mas, caso surja a necessidade, poderão ser utilizadas as propriedades de escalabilidade fornecidas pelo *container* da aplicação, o *Apache Tomcat* e a aplicação hospedada em provedores que oferecem serviços em nuvem.

6.5.1.3 Abertura

Um sistema distribuído é considerado aberto quando permite a sua expansão de diversas maneiras, permitindo que novos serviços sejam adicionados para diferentes tipos de usuários.

Assim, de acordo com Coulouris, Dollimore, Kindberg (2007, p. 29) quando a documentação e as principais interfaces de software dos componentes de um sistema distribuído estão disponíveis para outros desenvolvedores, ele ganha a característica de aberto, sendo um processo similar aos realizados por organizações de padronização. No entanto, o desafio de abertura não é tratado pelo Sistema *SafeWalk* pois ele não exportará dados para aplicações de terceiros. Contudo, aplicam-se nele, padrões de projeto e convenções de código conhecidos pela comunidade de desenvolvedores Java.

6.5.1.4 Segurança

A segurança em sistemas distribuídos é um aspecto importante no projeto e implementação deste tipo de sistema. Uma vez que os vários componentes de hardware e software precisam se comunicar e colaborar entre si, muitas vezes, de modo remoto, passando por redes de comunicação públicas.

No *SafeWalk*, com o objetivo de garantir a segurança do sistema, os usuários precisam se autenticar para acessá-lo. Também são adotadas regras para acessar o banco de dados visando garantir sua integridade. Dessa maneira, para garantir a segurança dos dados trocados pelos componentes do *SafeWalk* é utilizado o HTTPS, que é um protocolo de segurança para criptografar as mensagens trocadas entre os componentes do sistema, criando assim uma navegação segura em cima de uma rede pública.

6.5.1.5 Manuseio de Falhas

Os elementos de um sistema distribuído estão sujeitos às falhas (como qualquer outra entidade computacional). Quando tais falhas forem identificadas, elas devem ser reportadas ao usuário e tratadas, a fim de manter o sistema coerente e confiável.

“As falhas em um sistema distribuído são parciais - isto é, alguns componentes falham, enquanto outros continuam funcionando. Portanto, o tratamento de falhas é particularmente difícil” (COULOURIS; DOLLIMORE; KINDBERG, 2006, p. 33).

Todos os sistemas são passíveis a falhas. Sendo assim é essencial tratá-las de maneira adequada. Desta maneira, o manuseio de falhas pode ser compreendido como a maestria em detectar, mascarar, tolerar e recuperar falhas durante a execução do sistema.

O *SafeWalk* trata parte do desafio de manuseio de falhas apenas em nível do código implementado na aplicação.

6.5.1.6 Concorrência

Em um sistema distribuído, a concorrência ocorre quando dois componentes de um mesmo sistema estão sendo executados, a fim de terem acesso simultâneo. Assim, para que um objeto mantenha coerência em um ambiente concorrente, suas operações devem ser sincronizadas de tal maneira que seus dados permaneçam consistentes. Isso pode ser obtido pelo emprego de técnicas padrão, como semáforos, que são disponíveis na maioria dos sistemas operacionais (COULOURIS; DOLLIMORE; KINDBERG, 2007, p. 34).

Desta maneira, para que o *SafeWalk* consiga gerenciar a concorrência gerada pelos seus componentes, será preciso controlar o número de conexões ao sistema e o seu banco de dados. Para realizar este controle, é utilizado o *Apache Tomcat*, que é responsável por controlar o acesso simultâneo de usuários ao sistema junto com o pool de conexões e *Java Database Connectivity* (JDBC), que é responsável por controlar o acesso ao banco de dados mantido pelo PostgreSQL.

6.5.1.7 Transparência

De acordo Coulouris, Dollimore, Kindberg (2007, p. 34), a transparência é a ocultação da separação que existe entre os componentes de um sistema distribuído, para que ele seja

percebido tanto pelo usuário quanto pelo desenvolvedor, como um todo, ou seja, um único componente ao invés de vários componentes independentes.

Assim sendo, o sistema *SafeWalk* é uma aplicação *Web*, por isso ele utiliza alguns processos sem o conhecimento do usuário para realizar operações de acesso ao servidor da aplicação, ao banco de dados e ao navegador, garantindo um nível de transparência na aplicação.

6.5.1.8 Tecnologias e Arquiteturas de Distribuição

A estrutura do sistema *SafeWalk* se baseia na arquitetura distribuída cliente/servidor. Os serviços, cargas e tarefas são distribuídos entre os servidores, enquanto os clientes realizam requisições dos serviços por meio dos protocolos *Ethernet*, *Transmission Control Protocol*, (TCP/IP), *Internet Protocol* (IP) e HTTPS.

O processo de comunicação entre os componentes do sistema *SafeWalk* está representado na Figura 13. Como exposto, existem quatro processos, sendo P1, P2 e P3 processos do servidor, responsáveis por atender as requisições do processo cliente, representado por P4.

Assim, o processo P4 é um processo cliente, executado no navegador, fazendo requisições ao servidor. As requisições feitas pelo processo cliente são atendidas pelo *Apache Tomcat*, um *container* de *servlets*, representado pelo P2. Já o processo P3 é a JVM, o processo responsável por interpretar o código em Java e converter para códigos binários de máquina. Caso para atender a requisição do cliente, houver a necessidade de acessar o banco de dados, esta requisição é também atendida pelo SGBD representado pelo processo P1.

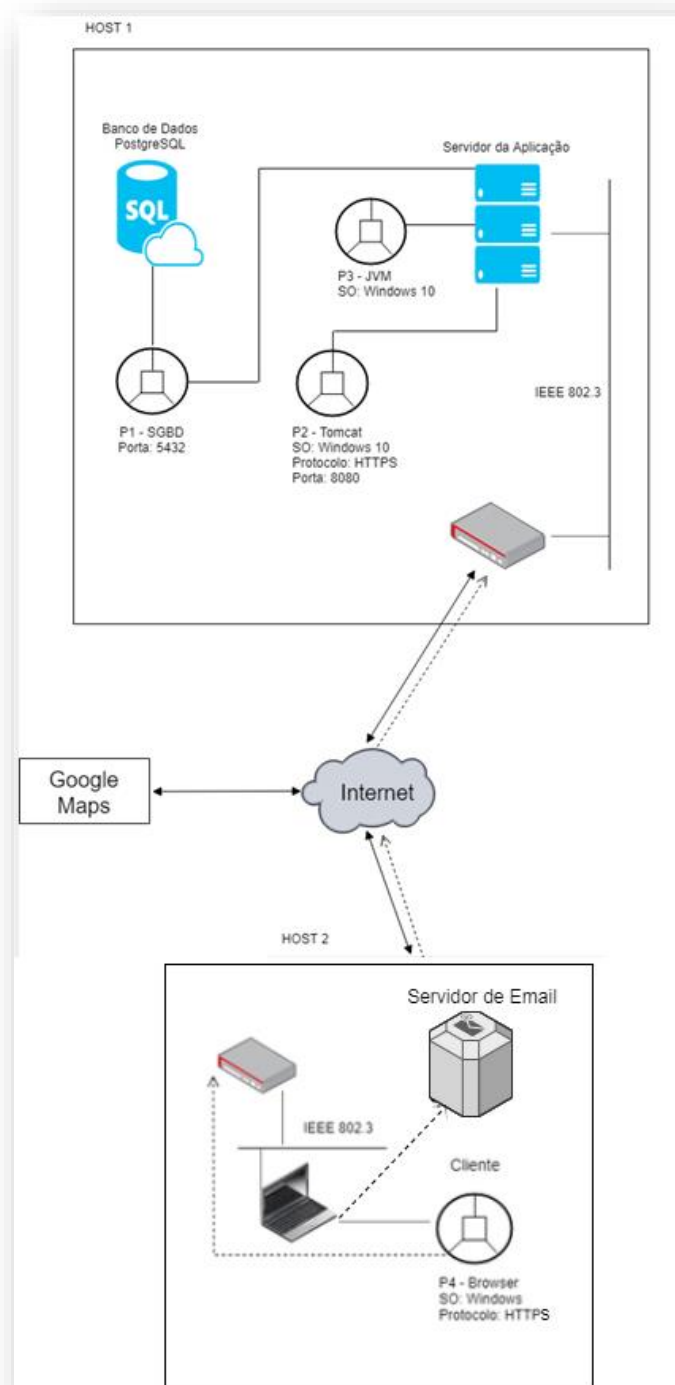


FIGURA 13 - Processo de comunicação entre os componentes do sistema *SafeWalk*

6.6 PROJETO DA INTERAÇÃO HUMANO-COMPUTADOR

O projeto de interação humano-computador busca levantar os dados para a construção da interface de usuário que intermediará a comunicação entre o ser humano e computador. Esses dados são coletados por meio de entrevistas, pesquisas e estudos das características dos

usuários. Com base nas atividades realizadas, é projetada a interface de usuário que deve ter usabilidade e acessibilidade.

Segundo Pressman (2013, p. 287), “seguinto-se um conjunto de princípios de projeto de interfaces, o projeto identifica um layout de tela que forma a base para um protótipo de interface do usuário”. Como consequência, o projeto de interfaces acrescenta uma medida qualitativa da facilidade e eficiência com a qual um ser humano consegue explorar as funções e recursos oferecidos pelo sistema.

Assim, com base no perfil dos usuários, o sistema deve oferecer uma interface que seja compreensível e agradável, criando uma boa experiência de uso.

6.6.1 Perfil do Usuário

Esta subseção apresenta informações a respeito do perfil dos possíveis usuários do sistema. Dados como idade, sexo, experiência com a utilização de outros sistemas, experiência quanto ao domínio do problema, frequência de uso do sistema e preferências gerais foram analisados de acordo com o público alvo da Seção 3.4, para que a construção do sistema pudesse atender a todos os perfis. Os perfis analisados foram:

- a) **Administrador:** idade acima de 18 anos e terá acesso ao sistema para realizar configurações diversas. Esses usuários possuem vasta experiência com a utilização e gerência de sistemas. Quanto às preferências gerais de utilização, a navegação pelo sistema deve acontecer de forma bem organizada;
- b) **Cidadão:** idade acima de 16 anos. Considera-se que esses usuários possuem experiência mediana com a utilização de sistemas. Quanto às preferências gerais de utilização, a navegação pelo sistema deve ser simples e intuitiva, pois este grupo apresenta características distintas, tanto pessoas que não possuem tanta familiaridade com a utilização de sistemas, quanto pessoas que possuem muita familiaridade;
- c) **Poder Público:** idade acima de 18 anos e que realiza funções administrativas. Pressupõe-se que esses usuários possuam experiência mediana com a utilização de sistemas. Seu conhecimento a respeito do problema tratado é vasto, além de estarem inseridos neste contexto no dia a dia, conhecem todas as políticas envolvidas nos processos. Quanto às preferências gerais de utilização, a navegação pelo sistema deve

ser simples e intuitiva, além de bem organizada, para que possa gerenciar todas as etapas.

Desta maneira, o roteiro com as entrevistas realizadas com os possíveis usuários do projeto *SafeWalk* encontra-se disponível no Apêndice N.

6.6.2 Aspecto Visual da Interface do Usuário

É evidente que as pessoas interagem com um sistema por meio daquilo que elas veem. Segundo Benyon (2011, p. 212), é preciso “elaborar um design de forma que as pessoas vejam e percebam coisas em uma tela. Os botões devem ser grandes o suficiente para serem vistos e precisam ser identificados de forma compreensível para as pessoas”.

Desta maneira, esta seção apresenta os aspectos visuais (cores, ícones e fontes tipográficas) encontrados na interface de usuário do sistema *SafeWalk*.

6.6.2.1 Cores

Para realizar o projeto da IU devem-se considerar as cores que serão utilizadas na aplicação, de modo que a combinação delas ocorra de forma harmoniosa, sem sobrecarregar a percepção visual do usuário.

A cor é manipulada pelo ser humano desde as primeiras manifestações da atividade humana sendo utilizada como canal de comunicação e de projeção dos conhecimentos e sentimentos, representando sempre uma ferramenta que transmite ideias, promove a atenção e demonstra emoções (BARROS, 2006 apud KULPA; PINHEIRO; SILVA, 2011, p. 124).

Visto que as cores nos meios de comunicação visual influenciam nas emoções de quem as observa, as cores da interface do *SafeWalk*, baseiam-se tons de azul, laranja e branco. Ademais, a interface utiliza uma combinação de cores (botões, ícones, *dashboards*) que visam facilitar a compreensão e uso do sistema e, ao mesmo tempo, deixá-lo com uma aparência agradável.

6.6.2.2 Ícones

De acordo com Pressman (2013), uma das regras de ouro para uma interface com o usuário excelente é diminuir a carga de memória do usuário, isto é, proporcionar uma interação entre o usuário e o software que não exaure a memória do usuário e faça com que haja uma menor probabilidade de ocorrência de erros.

Uma das medidas para auxiliar na redução da carga de memória do usuário é fornecer metáforas do mundo real que o auxiliem em relação à familiarização dos componentes do sistema. Diante disto, os ícones foram criados de forma que representassem elementos reais, são alguns deles são mostrados na Figura 14.



FIGURA 14 - Exemplo de ícones utilizados no sistema

As fontes de um sistema devem ser de fácil entendimento para os usuários do sistema. A partir disto, o *SafeWalk* utiliza a fonte de texto Poppins em predominância, com o intuito de promover um maior conforto na experiência de navegação do usuário.

6.6.3 Heurísticas de Usabilidade

A análise heurística é uma técnica de avaliação de usabilidade amplamente difundida. Para que ela seja realizada utilizam-se “bons princípios” propostos por especialistas em usabilidade. Nielsen (1994) apresenta 10 heurísticas de usabilidade. São elas:

- a) Visibilidade do estado atual do sistema;
- b) Correspondência entre o sistema e o mundo real;
- c) Liberdade de controle para o usuário;
- d) Consistência e padrões;
- e) Prevenção de erros;

- f) Reconhecimento em vez de memorização;
- g) Flexibilidade e eficiência de uso;
- h) Estética e design minimalista;
- i) Ajudar os usuários a identificarem, diagnosticarem e recuperarem-se de erros;
- j) Ajuda e documentação.

Essas heurísticas são utilizadas como um método de averiguação da praticidade da interface de usuário. Com isso, o *SafeWalk* adota medidas para a melhorar a experiência de seus usuários.

6.6.3.1 Visibilidade do estado atual no sistema

Esta heurística aborda o fato de que o sistema deve manter os usuários informados sobre o que está acontecendo naquele determinado contexto. Além disso, o usuário deve receber as informações requeridas a todo momento, com consistência e de forma que possa proporcionar a melhor sensação de uso possível.

A Figura 15 demonstra um exemplo no qual o sistema informa ao usuário seu estado atual no sistema. Para demonstrar isso, usa-se de exemplo a página atual do sistema.

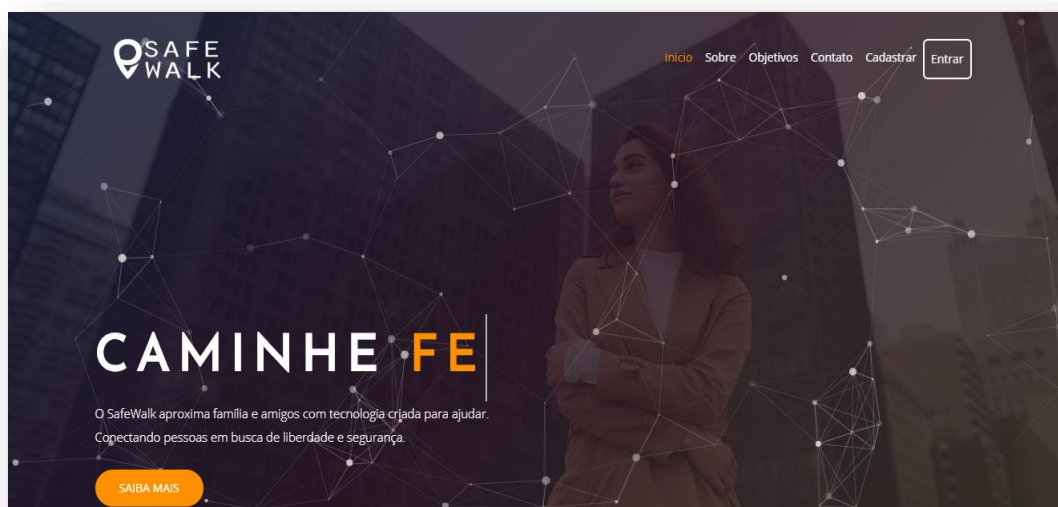


FIGURA 15 - Página inicial do *SafeWalk*

6.6.3.2 Correspondência entre o sistema e o mundo real

A correspondência entre o sistema e o mundo real é abordado pela linguagem presente na interface de usuário. A linguagem deve ser familiar aos usuários, seguindo convenções do mundo real, apresentando as informações em uma ordem natural e lógica.

A Figura 16 demonstra alguns exemplos de ícones do mundo real presentes na barra de navegação do sistema, facilitando assim o entendimento do usuário.

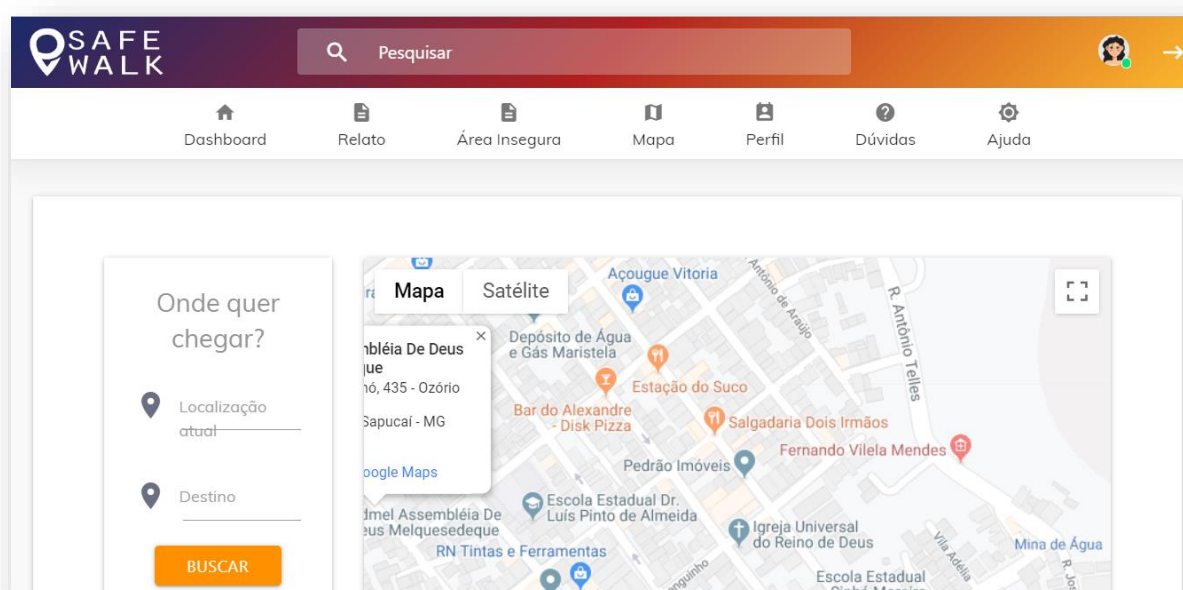
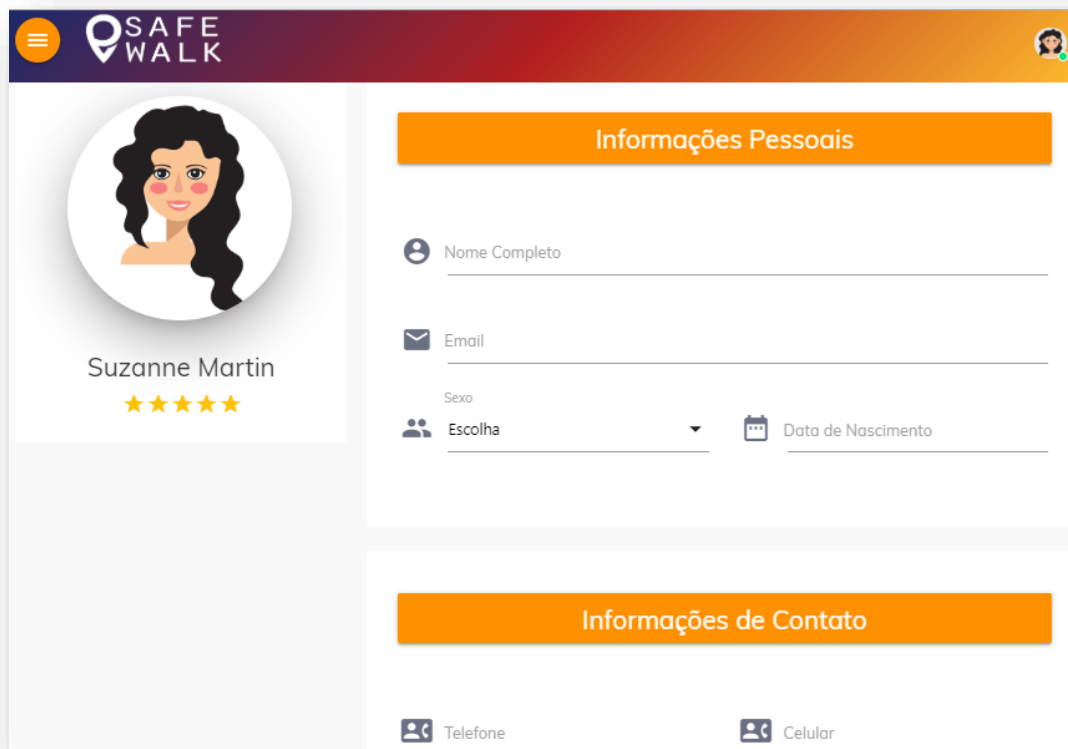


FIGURA 16 - Exemplo de ícones

6.6.3.3 Liberdade de controle para o usuário

A liberdade de controle do usuário parte do princípio que um usuário deve ter autonomia para usar o sistema de forma livre, isto é, que o mesmo possa realizar ou reverter as suas ações a qualquer momento.

A Figura 17 exemplifica uma das autonomias oferecidas ao usuário pelo sistema. O usuário tem a autonomia de editar o seu perfil a qualquer instante, da mesma forma que ele poderá reverter as mudanças que realizou.



The screenshot shows a user profile interface for 'SAFE WALK'. On the left, there is a circular profile picture of a woman with dark hair, labeled 'Suzanne Martin' with a five-star rating below it. The main area is divided into two sections: 'Informações Pessoais' (Personal Information) and 'Informações de Contato' (Contact Information). The 'Informações Pessoais' section includes fields for 'Nome Completo' (Full Name), 'Email', 'Sexo' (Gender) with a dropdown menu labeled 'Escolha', and 'Data de Nascimento' (Date of Birth). The 'Informações de Contato' section includes fields for 'Telefone' (Phone) and 'Celular' (Cellular).

FIGURA 17 - Exemplo de alteração de dados

6.6.3.4 Consistência e padrões

A consistência e padrões é uma heurística de suma importância, pois ao atendê-la, diminui-se a carga de memória exigida do usuário. Uma mesma ação ou comando deve apresentar o mesmo efeito quando comparado a de outros sistemas existentes, tornando compatível com outros sistemas disponíveis ou conhecidos pelos usuários.

A Figura 18 exemplifica alguns ícones também encontrados em outros sistemas *Web* e utilizados no *SafeWalk*.



FIGURA 18 - Exemplo de ícones conhecidos

6.6.3.5 Prevenção de erros

Melhor do que possuir boas mensagens de erros é o sistema ser preparado para que possa impedir a ocorrência deles, prevenindo as possíveis condições que possam gerar algum erro. Além disso, devem-se mostrar as mensagens de forma clara, facilitando o entendimento do usuário.

O *SafeWalk* emite alertas de erros a seus usuários. Esses alertas ajudam-no a entender o erro e encontrar uma possível solução, sem que tenha que recorrer ao suporte do sistema.

6.6.3.6 Reconhecimento em vez de memorização

Não se deve exigir do usuário a lembrança permanente de dados e controles. E, para isso, é necessário tornar os objetos, ações e opções visíveis familiares ao usuário, de forma que ele consiga fazer a associação daquilo com a sua função esperada e não precise ficar memorizando o que cada item faz. Portanto, as ações e instruções devem ser intuitivas, gerando uma melhor navegação possível.

6.6.3.7 Flexibilidade e eficiência de uso

Este princípio demonstra a necessidade de o usuário personalizar as suas ações frequentes de formas alternativas oferecendo atalhos e a possibilidade de customização da interface de usuário.

6.6.3.8 Estética e *design* minimalista

Esta heurística adota o princípio que as informações exibidas aos usuários não devam ser excessivas. É preciso simplificar a apresentação das informações e os controles, visando facilitar o entendimento do sistema e o seu manuseio. A navegação pelo sistema deve ser de forma simples e eficiente para o manuseio do usuário.

6.6.3.9 Ajudar os usuários a identificarem, diagnosticarem e recuperarem-se de erros

Em busca de facilitar o entendimento do usuário, esta heurística expressa que as mensagens de erros devem ser feitas de forma simples, sem números de erros ou detalhes técnicos desnecessários, indicando precisamente qual é o problema e deixando clara a solução para o mesmo.

Seguindo esta heurística, o *SafeWalk* procura mostrar mensagens de erros de fácil entendimento para o usuário. Com isso, o usuário compreende o seu erro e dessa forma consegue se recuperar da falha em questão.

6.6.3.10 Ajuda e documentação

A documentação do sistema deve ser apresentada de forma detalhada, com mecanismos que permitam que determinada informação seja facilmente encontrada. Além disso, deve-se focar nas ações que o usuário efetuará de forma simples e com linguagem próxima do seu público alvo.

O *SafeWalk* oferece manuais e outros recursos para ajudar o usuário na sua operação eficiente e eficaz.

6.6.4 Análise de Complexidade Algorítmica

A complexidade de um algoritmo segundo Toscani e Veloso (2009, p. 14) reflete um esforço computacional necessário para executar um algoritmo. O tempo de execução ou a quantidade de memória requerida para realizar um esforço computacional são os parâmetros de medida para saber a quantidade de trabalho necessário para realizá-la.

Entre as funções que modelam a complexidade de um algoritmo, as mais conhecidas são: constante, logarítmica, log quadrático, linear, $n \log n$, quadrática, cúbica e exponencial.

O trecho de código da Figura 19 tem por finalidade buscar um município cadastrado na base de dados pela sua chave primária (id). O bloco *try ... catch* testa se foi possível encontrar um município para o id fornecido. Esta operação será executada apenas uma vez e a função *readbyid* depende de n , sendo considerada $O(n)$.


```

public Municipio readById(Long id) throws Exception {
    Connection conn = ConnectionManager.getInstance().getConnection();
    Municipio municipio = new Municipio();
    try {
        MunicipioDAO dao = new MunicipioDAO();
        municipio = dao.readById(conn, id);
        conn.commit();
        conn.close();
    } catch (SQLException e) {
        conn.rollback();
        conn.close();
        throw e;
    }
    return municipio;
}

```

FIGURA 19 - Exemplo de classe de complexidade $O(1)$

O trecho de código apresentado na Figura 20 apresenta uma lista de áreas inseguras com seus dados organizados dentro de uma tabela. O *c:forEach* é uma *tag* utilizada para iterar uma lista de dados contidos nela na forma de uma tabela na página web. Esse trecho de código apresenta classe linear $O(n)$, pois seu tempo de execução depende do número de elementos na lista, aumentando linearmente de acordo com o aumento de n .

```

<div class="col s12" style="padding: 0;overflow-x: auto">
<table class="table table-hover" id="data-table-simple" class="display">
  <thead>
    <...9 linhas />
  </thead>
  <tbody id="body-relatos-list">
    <c:forEach items="${List}" var="areaInsegura">
      <tr>
        <td hidden style="text-align:center">${areaInsegura.id}</td>
        <...8 linhas />
        <...3 linhas />
        <td hidden style="text-align:center">${areaInsegura.latitude}</td>
        <td hidden style="text-align:center">${areaInsegura.longitude}</td>
        <td style="max-width: 300px; text-align:center">${areaInsegura.descricao}</td>
        <...10 linhas />
      </tr>
    </c:forEach>
  </tbody>
</table>
</div>

```

FIGURA 20 - Exemplo de classe de complexidade $O(n)$

7 PLANO DE TESTES

Este capítulo apresenta o planejamento das atividades de teste de validação do sistema *SafeWalk*. A realização do plano de teste se faz necessária para assegurar a qualidade do software, bem como verificar a existência de defeitos, erros e falhas que necessitem de correção.

Esta seção se segmenta em quatro subseções, sendo elas finalidade, escopo, especificação de casos de testes e resultado dos testes.

7.1 FINALIDADE

Os testes apresentados têm por objetivo verificar o correto funcionamento das funcionalidades do sistema *SafeWalk* de acordo com os requisitos funcionais e não funcionais levantados que se encontram nas Seções 5.1.1 e 5.1.2 deste documento.

7.2 ESCOPO

Nesta seção são apresentados os documentos relevantes para a construção do plano de testes do sistema *SafeWalk*, bem como os itens e o ambiente a ser utilizado para a realização dos testes.

7.2.1 Referências a Documentos Relevantes

Os documentos relevantes para a construção e realização do plano de testes para o projeto *SafeWalk* são apresentados no Quadro 6.

Tipo de Material	Referência
Requisitos funcionais e não funcionais	Seção 5.1 deste documento

QUADRO 6 - Documentos relevantes para testes.

7.2.2 Ambiente para a Realização dos Testes

Os Quadros 7 e 8 mostram os equipamentos e softwares utilizados nos testes.

Equipamento	Marca/Modelo/Configuração	Finalidade
Desktop	Processador Core i5 4690, 8GB de RAM, 1 HD de 1TB.	Executar e testar o sistema

QUADRO 7 – Equipamentos a serem utilizados para a realização dos testes

Software/Versão	Fabricante	Finalidade
NetBeans 8.1	Oracle	Gerenciador de serviços.
Windows 10	Microsoft	Plataforma de Sistema Operacional.
Google Maps	Google	Acessar a API do Google Maps.
PostgreSQL	<i>The PostgreSQL Global Development Group</i>	Acessar o banco de dados.
Navegador Google Chrome 78.0	Google	Acessar o sistema localmente.

QUADRO 8 - Softwares a serem utilizados para a realização dos testes.

7.3 ESPECIFICAÇÃO DOS CASOS DE TESTE

Nesta seção são abordados os itens a serem testados, a rastreabilidade entre os requisitos e os casos de teste e, por fim, a descrição dos casos de teste.

7.3.1 Item a Testar

O Quadro 9 apresenta os itens do *SafeWalk* que serão testados.

Identificação	Descrição
1	[RF02] - Alterar senha
2	[RF09] - Mostrar e editar perfil do usuário
3	[RF10] - Remover conta
4	[RF13] - Manter Cadastro de áreas inseguras

QUADRO 9 - Identificação dos itens a serem testados

7.3.2 Rastreabilidade entre Requisitos e Casos de Teste

O Quadro 10 apresenta a rastreabilidade entre os requisitos e os casos testes do sistema *SafeWalk*.

Identificação dos Requisitos	Caso(s) de teste(s) aplicável(eis)
[RF02] - Alterar senha	CT01 - Alterar Senha
[RF09] - Mostrar e editar perfil do usuário	CT02 - Mostrar e editar perfil do usuário
[RF10] - Remover conta	CT03 - Remover Conta
[RF13] - Manter Cadastro de áreas inseguras	CT04 - Editar área insegura CT05 - Excluir área insegura CT06 - Inserir área insegura

QUADRO 10 - Requisitos e casos de teste

7.3.3 Descrição dos Casos de Testes

A descrição dos casos de testes previstos neste plano encontram-se no Apêndice V.

7.4 RESULTADOS DOS TESTES

Nesta seção são apresentados os resultados obtidos com a realização dos testes elaborados.

7.4.1 Histórico de Realização

O histórico de realização dos testes encontra-se no Apêndice V deste documento.

7.4.2 Resultados

O plano de testes desenvolvido para o projeto *SafeWalk* engloba parte das funcionalidades da aplicação, como cadastro e alteração dos usuários. O resultado dos testes foi satisfatório uma vez que a aplicação apresentou resultados de acordo com o que foi levantado como requisitos inicialmente.

8 PLANO DE IMPLANTAÇÃO

Neste capítulo são apresentadas a metodologia da implantação, os treinamentos que devem ser realizados, o cronograma, os documentos de apoio à implantação e a visão de implantação do sistema *SafeWalk*.

8.1 METODOLOGIA

Esta seção aborda a metodologia de implantação adotada para o sistema *SafeWalk*, descrevendo o processo de implantação. Nesta seção também consta a matriz de responsabilidade que define as funções designadas para esta etapa.

8.1.1 Descrição da Metodologia

Para que a implantação do sistema *SafeWalk* ocorra de forma efetiva e satisfatória, os seguintes passos devem ser executados, tanto para o usuário do tipo “administrador” quanto para o usuário do tipo “poder público”:

- a) designar o responsável pela implantação;
- b) realizar a configuração do servidor de Internet;
- c) criar uma instância do banco de dados PostgreSQL no servidor do banco de dados;
- d) executar os *scripts* de criação e povoamento do banco de dados;
- e) adicionar o arquivo *Web application ARchive* (WAR) do *SafeWalk* no servidor;
- f) obter uma licença da API Google Maps;
- g) realizar as configurações necessárias.

8.1.2 Matriz de Responsabilidades

As responsabilidades de cada membro do projeto com relação a implantação do sistema são apresentadas na matriz de responsabilidade ilustrada no Quadro 11.

Continua.

Atividades	Responsáveis
Planejamento	

Definição da equipe de implantação	Ângela
Levantamento de recursos necessários de hardware	Caroline
Levantamento de recursos necessários de software	Caroline
Programação dos treinamentos	Marcos
Preparação dos testes de aceitação (apenas pelo Poder Público)	Rodrigo
Execução	
Configuração da infraestrutura de Tecnologia da Informação	Marcos
Instalação	Rodrigo
Conversão e migração de base de dados	Rodrigo
Treinamento	Caroline
Avaliação	
Acompanhamento pós-implantação	Ângela
Reunião final da implantação	Ângela

QUADRO 11 - Matriz de Responsabilidades

8.2 TREINAMENTO PREVISTO

Para o treinamento os usuários do poder público, uma visão geral do funcionamento do sistema será apresentada, além de uma qualificação para o usuário do tipo “administrador”.

A qualificação é necessária para que esse tipo de usuário consiga operar o sistema e realizar as suas atividades corretamente e de forma eficiente. Já para os usuários do tipo cidadão receberam um treinamento básico, poderão conhecer como é o sistema bem como seu funcionamento. Os treinamentos previstos para a implantação do *SafeWalk* encontram-se no Quadro 12.

Continua.

Treinamento	Conteúdo	Grupo de Usuários
Configuração do ambiente de instalação do sistema.	Instalação dos sistemas de software necessários.	Responsáveis do setor de TI.

Conclusão.

Demonstração das funcionalidades do sistema para o usuário “administrador”.	Supervisão e controle do sistema.	Administrador do sistema.
Demonstração das funcionalidades do sistema para o usuário “poder público”.	Utilização e controle do sistema.	Poder Público.
Demonstração das funcionalidades do sistema para o usuário “cidadão”.	Utilização do sistema.	Cidadão.

QUADRO 12 – Treinamentos Previstos

8.3 CRONOGRAMA DE IMPLANTAÇÃO

No Quadro 13 são ilustradas as tarefas realizadas, a respectiva duração bem como o período de execução dos treinamentos.

Tarefas	Duração	Período
Instalação do servidor	20 horas	A definir.
Configuração do banco de dados.	5 horas	A definir.
Configurações necessárias de integração.	6 horas	A definir.
Treinamento dos usuários do tipo “Administrador”.	10 horas	A definir.
Treinamento dos usuários do tipo “Poder Público”.	10 horas	A definir.
Treinamento dos usuários do tipo “Cidadão”.	10 horas	A definir
Tempo estimado total:	61 horas	

QUADRO 13 - Tarefas previstas na implantação

8.4 DOCUMENTOS DE APOIO À IMPLANTAÇÃO

Os documentos que podem ser utilizados para apoiar o processo de implantação do sistema e seu uso posterior são listados no Quadro 14.

Documento	Referência
Manual de Instalação e configuração do Apache Tomcat.	Vide Anexo A.
Manual de Instalação e configuração da JRE.	Vide Anexo A.
Manual de Instalação e configuração do PostgreSQL.	Vide Anexo B.
Manual do Usuário do <i>SafeWalk</i>	Vide Apêndice Y.

QUADRO 14 - Documentos de apoio à implantação

8.5 VISÃO DA IMPLANTAÇÃO

O diagrama de implantação é um modelo que expressa a organização da arquitetura física do hardware na qual o sistema será implantado e executado. Esse diagrama mostra instâncias de máquinas (computadores pessoais, servidores, entre outros.) onde os componentes do sistema serão executados e quais protocolos eles usarão para se comunicar entre si (GUEDES, 2007).

Assim, para uma melhor compreensão do plano de implantação, um exemplo de um diagrama de implantação do sistema *SafeWalk* encontra-se no Apêndice X.

9 CONCLUSÃO

Com o desenvolvimento do projeto *SafeWalk* foi possível obter muitos aprendizados, tanto na área tecnológica, quanto na área de segurança pública, sendo de grande valia para a equipe todas as fases do projeto, desde o planejamento até a sua fase de conclusão.

Para a construção do projeto foram necessárias pesquisas em diversos artigos científicos a fim de obter o embasamento teórico deste projeto, para com isso justificar sua aplicabilidade e viabilidade. Também foram necessários estudos sobre as tecnologias de informação utilizadas no projeto, tais como API do Google Maps e Bootstrap. Desta maneira, a equipe agregou importantes conhecimentos, aprendizados e experiências.

Ademais, os objetivos estabelecidos no início do projeto foram alcançados de forma satisfatória. O desenvolvimento do sistema foi muito gratificante para todos, em especial, os conhecimentos que foram adquiridos durante o ano de 2019 auxiliaram na trajetória dos integrantes no mercado de trabalho, possibilitando um bom entendimento de diversos assuntos.

Contudo, apesar dos objetivos terem sido alcançados, o sistema possui algumas limitações, por exemplo, ainda não é possível traçar uma rota segura, e sim, apenas consultar as rotas com relatos e/ou áreas inseguras marcadas. Também é preciso ampliar a segurança dos dados do sistema.

Pretende-se aprimorar o sistema, acrescentando futuramente uma versão para plataforma *mobile*, com a qual os usuários do tipo cidadão poderão acessar por seus dispositivos móveis o *SafeWalk*, além de algumas correções para que um grande número de usuários possa ser melhor atendido.

Tendo em vista a satisfação dos *stakeholders*, o projeto *SafeWalk* trouxe novas percepções para a equipe sobre determinados pontos, importantes também para a vida pessoal dos integrantes, como o desenvolvimento do interesse na participação da segurança pública, salientando a preocupação com a transparência de acesso às informações.

Assim sendo, o projeto concedeu aos integrantes um excelente desempenho na realização do trabalho em equipe, de modo que cada um pôde produzir os artefatos de forma independente para, posteriormente, agregá-los, formando assim uma equipe autogerenciável.

A equipe obteve vasto conhecimento e prática em todas as fases do projeto e pretende aplicá-los em situações vivenciadas na vida pessoal e profissional.

Por fim, o projeto *SafeWalk* foi uma experiência ímpar para a equipe, pois, além de agregar valores para o crescimento profissional dos integrantes, trouxe novas percepções sobre aspectos relacionados à responsabilidade social, visto que o projeto é voltado para este fim e contribui para o interesse pela segurança pública do Brasil.

REFERÊNCIAS

- ANDRADE, Edméia Leonor Pereira de; OLIVEIRA, Káthia Marçal de. Aplicação de Pontos de Função e Pontos de Caso de Uso de Forma Combinada no Processo de Gestão de Estimativa de Tamanho de Projetos de Software Orientados a Objetos. **Informática Pública**, vol. 7 (1), p.13-30, 2005.
- ARAÚJO, Maria de Fátima. Gênero e violência contra a mulher: o perigoso jogo de poder e dominação. **Psicologia para América Latina**, México, n. 14, out. 2008. Disponível em: < http://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S1870-350X2008000300012 />. Acesso em: março 2019.
- BENYON, David. **Interação humano-computador**. 2. ed. São Paulo: Pearson Prentice Hall, 2011.
- BOMFIM, Márcia Regina Guiotti; ANDRADE, José Romildo. **Guia de Contagem de Pontos de Função do Ministério do Planejamento, Orçamento e Gestão (MP)**. Brasil: 2015. Versão 1.0.
- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: Guia do Usuário**. 2.ed. Rio de Janeiro: Elsevier, 2006. 474p.
- BSAFE. **Our Story**. 2019. Disponível em: <<https://getbsafe.com/our-story/>>. Acesso em: abril 2019.
- BRASIL. Lei nº. 11.340, de 7 de agosto de 2006. **Diário Oficial [da] União**, Brasília, DF, 8 Agosto 2006. Disponível em: < http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2006/lei/111340.htm >.
- CARVALHO, Fábio Câmara Araújo de. **Gestão de Projetos**. 1. ed. São Paulo: Pearson Education do Brasil, 2014.
- CALHEIROS, Renan. **A falta da segurança**, 2014. Disponível em: < <https://www.brasil247.com/pt/247/artigos/138344/A-falta-que-a-seguran%C3%A7a-faz.htm> >. Acesso em: abril 2019.
- CAVALCANTE, Érika Claudine Rodrigues. **Violência contra a mulher**, 2016. Disponível em:< <https://erikacrcavalcante.jusbrasil.com.br/artigos/251026383/violencia-contra-mulher/> >. Acesso em: março 2019.
- CIRCLE OF 6. **Our Story**. 2019. Disponível em: <<https://www.circleof6app.com/>>. Acesso em: abril 2019.
- COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim. **Sistemas Distribuídos Conceitos e Projetos**. 4. ed. Porto Alegre: Bookman, 2007.
- DANTAS, Kamila. **Violência urbana no Brasil**. 2019. Disponível em: < <https://www.imagineie.com.br/temas/violencia-urbana-no-brasil/> >. Acesso em: março 2019.
- GOMES, Caio César Santos. **A face da violência urbana**. 2016. Disponível em: <

<https://meuartigo.brasilecola.uol.com.br/atualidades/a-face-violencia-urbana-questoes-atuais.htm/>>. Acesso em: março 2019.

FÓRUM BRASILEIRO DE SEGURANÇA PÚBLICA. **Anuário Brasileiro de Segurança Pública 2017**, ano 11, São Paulo, 2017.

FURLAN, José Davi. **Modelagem de Objetos através da UML: Análise e Desenho Orientados a Objeto**. São Paulo: Makron Books, 1998. 329 p.

GUEDES, Brena Kécia Sales; GOMES, Flâmela Kevilla Silva. Violência Contra a mulher. **Faculdade Cearense em Revista**, Ceará, v. 7, 2014.

GUEDES, G. T. (2007). UML UMA ABORDAGEM PRÁTICA. Novatec.

INSTITUTO DE PESQUISA ECONÔMICA APLICADA. **Atlas da violência 2018**. Brasília: Instituto de Pesquisa Econômica Aplicada, Fórum Brasileiro de Segurança Pública, 2018.

KULPA, Cínthia Costa; PINHEIRO, Eluza Toledo; SILVA, Régio Pierre da. A Influência das Cores na Usabilidade de Interfaces Através do Design Centrado no Comportamento Cultural Do Usuário. **Perspectivas em Gestão & Conhecimento**, João Pessoa, v. 1, p. 119-136, 2011.

LEXICO. **Morbus Social**. 2019. Disponível em: <<https://www.lexico.pt/morbus/>>. Acesso em: abr. 2019.

MACHADO, Cynthia Braz; DAHER, Donizete Vago; TEIXEIRA, Enéas Rangel, ACIOLI, Sonia. Violência urbana e repercussão nas práticas de cuidado no território da saúde da família. **Revista Enfermagem Uerj**, Rio de Janeiro, v. 24, n. 5, 2016. Disponível em: <<https://www.e-publicacoes.uerj.br/index.php/enfermagemuerj/article/view/25458>>. Acesso em: março. 2019.

MALALAI. **Sobre Nós**. Belo Horizonte: 2017. Disponível em: <<http://malalai.com.br/site/>>. Acesso em: abril 2019.

MONTE, Jéssica. **A violência urbana e suas formas de prevenção**. 2006. Disponível em: <<https://www.direitonet.com.br/artigos/exibir/3762/A-violencia-urbana-e-suas-formas-de-prevencao>>. Acesso em: março. 2019.

NIELSEN, J. **Ten Usability Heuristics**. USA: [S.n], 1994. Disponível em: <<https://www.nngroup.com/articles/ten-usability-heuristics/>>. Acesso em: set. 2019.

OLIVEIRA, André Bernardo de; CHIARI, Renê. **Fundamentos em Gerenciamento de Projetos Baseado no PMBOK**. 5. ed. Editora Communit, 2014.

POLÍCIA MILITAR DO ESTADO DE SÃO PAULO. **Manual de autoproteção do cidadão**. São Paulo: [S. n.], 2018. Disponível em: <<https://www.ssp.sp.gov.br/servicos/manual-seguranca.aspx>>. Acesso em: abril 2019.

PRESSMAN, Roger S. **Engenharia de Software: uma abordagem profissional**. 7. ed. São Paulo: AMGH Editora, 2013.

PROJECT MANAGEMENT INSTITUTE (PMI). **Um guia do conhecimento em gerenciamento de projetos (Guia PMBOK)**. 5. ed. EUA: Project Management Institute, 2013. 595 p.

SILVA, Ana Cláudia Gonçalves da Silva; OLIVEIRA FILHO, Josélio Soares; SANTOS, Karla Patrícia Ferreira; BARRÊTO, Jaqueline Roque; BEZERRA, Cíntia Almeida; ALMEIDA, Sandra Aparecida. **Violência contra mulher**: uma realidade imprópria, 2010. Disponível em:< <http://www.facene.com.br/wp-content/uploads/2010/11/Viol%C3%Aancia-contra-a-mulher.pdf>>. Acesso em: março 2019.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Education do Brasil, 2011.

TOSCANI, Laira Vieira; VELOSO, Paulo A. S. **Complexidade de Algoritmos**: Série Livros Didáticos Informática UFRGS. v.13. Porto Alegre: Bookman, 2009. Volume 13.

APÊNDICE A - ESTRUTURA ANALÍTICA DE PROJETO

A EAP encontra-se no formato PNG e está disponível na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE B - CRONOGRAMA

O cronograma encontra-se no formato MPP e está disponível na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE C - RELATÓRIO DE DESEMPENHO

O relatório de desempenho encontra-se no formato XLS e está disponível na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE D - LISTA DE VERIFICAÇÃO DA QUALIDADE

A lista de verificação da qualidade encontra-se disponível na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE E - DIAGRAMA DE CASOS DE USO

O diagrama de casos de uso encontra-se no formato PNG e está disponível na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE F - DESCRIÇÃO DOS CASOS DE USO

A descrição dos casos de uso encontra-se disponível na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE G - DIAGRAMA ENTIDADE E RELACIONAMENTO

O diagrama entidade e relacionamento encontra-se disponível na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE H - MODELO DE INTERFACE DO USUÁRIO

O modelo de interface do usuário encontra-se disponível na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE I - ACOMPANHAMENTO DOS RISCOS

A planilha de acompanhamento dos riscos encontra-se no formato XLS e está disponível na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE J - DIAGRAMA DE PACOTES

Os diagramas de pacotes estão no formato PNG e estão disponíveis na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE K - DIAGRAMA DE CLASSES

O diagrama de classes encontra-se no formato PNG e está disponível na pasta “APÊNDICE” do que acompanha este documento.

APÊNDICE L - DIAGRAMA DE OBJETOS

Os diagramas de objetos encontram-se no formato PNG e estão disponíveis na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE M - MODELO LÓGICO DOS DADOS

O modelo operacional dos dados encontra-se no formato PNG e está disponível na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE N - ENTREVISTA DO USUÁRIO

As entrevistas com possíveis usuários encontram-se disponível na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE O - DIAGRAMAS DE SEQUÊNCIA

Os diagramas de sequência encontram-se no formato PNG e estão disponíveis na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE P - DIAGRAMA DE VISÃO GERAL DE INTERAÇÃO

Os diagramas de interação geral encontram-se no formato PNG e estão disponíveis na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE Q - DIAGRAMAS DE ATIVIDADES

O diagrama de atividades encontra-se no formato PNG e está disponível na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE R - DICIONÁRIO DE DADOS DO MODELO LÓGICO

O dicionário de dados encontra-se disponível na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE S - PONTOS POR CASOS DE USO E PONTOS POR FUNÇÃO

A planilha com os cálculos das métricas Pontos por Caso de Uso e Pontos por Função encontram-se disponíveis na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE T - SCRIPTS DDL E DML

Os scripts DDL e DML encontram-se disponível na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE U - DIAGRAMA DE COMPONENTES

O diagrama de componentes encontra-se no formato PNG e está disponível na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE V - CASOS DE TESTE E HISTÓRICO DE REALIZAÇÃO

Os casos de teste e o histórico de realização de teste do projeto encontram-se disponível na pasta APÊNDICE do arquivo anexado a este documento.

APÊNDICE W - DIAGRAMA DE IMPLANTAÇÃO

O diagrama de implantação encontra-se no formato PNG e está disponível na pasta “APÊNDICE” que acompanha este documento.

APÊNDICE X - MANUAL DO USUÁRIO

O manual do usuário encontra-se no formato PDF e está disponível na pasta “APÊNDICE” que acompanha este documento.

ANEXO A – MANUAL DE INSTALAÇÃO APACHE TOMCAT E JRE

O manual de instalação do *Apache Tomcat* e do JRE estão disponíveis em formato PDF e encontram-se na pasta “Anexos” que acompanha este documento.

ANEXO B - MANUAL DE INSTALAÇÃO POSTGRESQL

O manual de instalação do PostgreSQL está disponível em formato PDF e encontra-se na pasta “Anexos” que acompanha este documento.