CENTRO UNIVERSITÁRIO DE BRUSQUE – UNIFEBE
DIEGO OSCAR IMHOF

SISTEMA ALERTA VERMELHO DE BRUSQUE

BRUSQUE
2017

CENTRO UNIVERSITÁRIO DE BRUSQUE – UNIFEBE
DIEGO OSCAR IMHOF

SISTEMA ALERTA VERMELHO DE BRUSQUE

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Sistemas de Informação no Centro Universitário de Brusque – UNIFEBE.

Orientador: Prof. Pedro Sidnei Zanchett, M.e

BRUSQUE
2017

DIEGO OSCAR IMHOF

SISTEMA ALERTA VERMELHO DE BRUSQUE

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel no curso de Sistemas de Informação do Centro Universitário de Brusque, sendo aprovado com a nota _____, pela seguinte banca examinadora, em 13 de dezembro de 2017:

_____

Prof. Pedro Sidnei Zanchett, M.e

_____

Prof. Paulo Henrique Bluning, Esp.

_____

Prof. (a) Tatiana Pereira Filgueiras, M.e

Dedico este trabalho aos meus pais por todo apoio e incentivo ao longo dos anos.

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus que me sustentou e capacitou em toda essa trajetória.

Aos meus pais Luiz Oscar Imhof e Isaura Marlene Imhof, que me apoiaram e incentivaram durante todo este período.

Aos meus amigos que me deram todos os apoios possíveis.

Ao meu orientador, professor Pedro Sidnei Zanchett que acreditou e colaborou na realização deste trabalho.

Aos avaliadores professor Paulo Henrique Bluning e a professora Tatiana Pereira Filgueiras por terem colaborado com sugestões de melhorias.

RESUMO

IMHOF, Diego Oscar. **Sistema Alerta Vermelho de Brusque.** Brusque, 2017. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação) – Centro Universitário de Brusque, Brusque, 2017.

Com o avanço da tecnologia soluções automatizadas surgem a todo momento, dentre estas aplicações pode-se citar as que têm como objetivo o auxílio no processo de tomada de decisão em situações de emergências oferecendo maior agilidade de atendimento em situações de desastres. E isso não é diferente para o Corpo de Bombeiros que poderá se beneficiar com a tecnologia a fim de garantir rapidez e segurança em seus processos. Este projeto tem como objetivo desenvolver um sistema de gerenciamento para o programa "Operação Alerta Vermelho" a ser utilizado pelo Corpo de Bombeiros de Brusque – SC e sua população. O sistema "Alerta Vermelho" consiste em mapear os bairros da cidade com maiores riscos de segurança para a população afim de auxiliar o Corpo de Bombeiros por meio dos dados obtidos na análise e na tomada de decisão. Este sistema será divido em três módulos: um módulo em nível de gestão para o Corpo de Bombeiros, outro módulo do aluno para realizar as pesquisas junto ao público-alvo e um módulo para acesso aos resultados das pesquisas e dicas de segurança para a comunidade. O sistema "Alerta Vermelho" será disponibilizado nas plataformas *web* e *mobile*. Para desenvolver estas soluções foram utilizadas metodologias de desenvolvimento *Scrum* no levantamento de requisitos de acordo com a necessidade do Corpo de Bombeiros e recursos de controle de versões garantindo integridade dessas informações.

Palavras-chave: Operação Alerta Vermelho. Corpo de Bombeiros. Sistemas de Informação.

ABSTRACT

With the advancement of technology automated solutions emerge every time, among these applications can be cited those which are intended to aid in the decision-making process in emergency situations thus providing greater agility of service in disaster situations. And this is no different to the Fire Department which may be benefited from the technology to ensure speed and safety in its processes. The purpose of this project is to develop a system of management for the program "Operation Red Alert" to be used by the Fire Department in Brusque - SC and its population. The system "Red Alert" is to map the neighborhoods of the city with high security risks for the population, in order to assist the Fire Department, by means of the data obtained through analysis and in decision making. This system will be divided into three modules: a module in the management level for the Fire Department, a module of the student to perform the searches by the audience and a module for access to research results and safety tips for the population. The system "Red Alert" will be available on the web and mobile platforms. To develop these solutions are used methodologies of development in Scrum requirements according to the need of the Fire Department and resources of version control ensuring integrity this information.

Keywords*: Operation Red Alert. Fire department. System information.*

LISTA DE FIGURAS

Figura 1: A edificação está em área de risco. 21
Figura 2: Camadas da engenharia de software. 28
Figura 3: Fluxo do processo scrum. 31
Figura 4: Diagrama de caso de uso do Corpo de Bombeiros. 47
Figura 5: Diagrama de classes do Corpo de Bombeiros. 67
Figura 6: Diagrama de caso de uso do aluno. 70
Figura 7: Diagrama de classes do aluno. 73
Figura 8: Digrama de caso de uso da população. 77
Figura 9: Diagrama de classes da população. 82
Figura 10: Diagrama de componentes. 83
Figura 11: Modelo entidade relacionamento. 84
Figura 12: Tela Administrador. 86
Figura 13: Tela Cadastro de Pesquisa. 87
Figura 14: Página Inicial. 88
Figura 15: Menu Alerta Vermelho. 90
Figura 16: Início do aplicativo. 91
Figura 17: Consulta de telefone. 91
Figura 18: Dicas de segurança. 91
Figura 19: Relatório de Bairro. 91
Figura 20: Relatório Comparativo. 91
Figura 21: Sobre. 91
Figura 22: Menu Alerta Vermelho Aluno. 92
Figura 23: Login. 93
Figura 24: Cadastro de pesquisa. 93
Figura 25: Listagem de pesquisa. 93
Figura 26: Minha conta. 93

LISTA DE TABELAS

Tabela 1: Requisito cadastro de bairro. 35
Tabela 2: Requisito listagem de bairro. 36
Tabela 3: Requisito alteração e exclusão de bairro. 36
Tabela 4: Requisito cadastro de ano. 37
Tabela 5: Requisito listagem de ano. 37
Tabela 6: Requisito alteração e exclusão de ano. 38
Tabela 7: Requisito cadastro de pergunta. 38
Tabela 8: Requisito listagem de pergunta. 39
Tabela 9: Requisito alteração e exclusão de pergunta. 39
Tabela 10: Requisito cadastro de opção da pergunta. 40
Tabela 11: Requisito listagem de opção da pergunta. 40
Tabela 12: Requisito alteração e exclusão de opção da pergunta. 41
Tabela 13: Requisito listagem de pesquisa. 41
Tabela 14: Requisito alteração e exclusão de pesquisa. 42
Tabela 15: Requisito relatório de bairro. 42
Tabela 16: Requisito relatório comparativo. 42
Tabela 17: Requisito realizar login. 43
Tabela 18: Requisito cadastro de administrador. 43
Tabela 19: Requisito listagem de administrador. 44
Tabela 20: Requisito alteração e exclusão de administrador. 44
Tabela 21: Requisito cadastro de aluno. 45
Tabela 22: Requisito listagem de aluno. 45
Tabela 23: Requisito alteração, ativação e inativação de aluno. 46
Tabela 24: Cadastro de bairro. 48
Tabela 25: Listagem de bairro. 48
Tabela 26: Eliminar bairro. 49
Tabela 27: Editar bairro. 49
Tabela 28: Cadastro de ano. 50
Tabela 29: Listagem de ano. 50
Tabela 30: Eliminar ano. 51
Tabela 31: Editar ano. 52
Tabela 32: Cadastro de pergunta. 52
Tabela 33: Listagem de pergunta. 53
Tabela 34: Eliminar pergunta. 53
Tabela 35: Editar pergunta. 54
Tabela 36: Cadastro da opção da pergunta. 54
Tabela 37: Listagem da opção pergunta. 55
Tabela 38: Eliminar opção da pergunta. 56
Tabela 39: Editar opção da pergunta. 56
Tabela 40: Listagem de pesquisa. 57
Tabela 41: Eliminar pesquisa. 57
Tabela 42: Editar pesquisa. 58
Tabela 43: Relatório de bairro. 59
Tabela 44: Relatório comparativo. 59
Tabela 45: Login. 60
Tabela 46: Cadastro de administrador. 60
Tabela 47: Listagem de administrador. 61

Tabela 48: Eliminar administrador. 62
Tabela 49: Editar administrador. 62
Tabela 50: Cadastro de aluno. 63
Tabela 51: Listagem de aluno. 64
Tabela 52: Ativar e inativar aluno. 65
Tabela 53: Editar aluno. 65
Tabela 54: Requisito cadastro de pesquisa. 68
Tabela 55: Requisito listagem de pesquisa. 68
Tabela 56: Requisito realizar login. 69
Tabela 57: Cadastro de pesquisa. 70
Tabela 58: Listagem de pesquisa. 71
Tabela 59: Login 72
Tabela 60: Requisito início. 74
Tabela 61: Requisito relatório de bairro. 74
Tabela 62: Requisito relatório comparativo. 75
Tabela 63: Requisito telefones. 75
Tabela 64: Requisito dicas de segurança. 76
Tabela 65: Requisito sobre. 76
Tabela 66: Início 77
Tabela 67: Relatório de bairro. 78
Tabela 68: Relatório comparativo. 79
Tabela 69: Telefones 80
Tabela 70: Dicas de segurança 80
Tabela 71: Sobre 81

LISTA DE ABREVIATURAS

ANDROID:	Sistema Operacional;
BI:	Inteligência de negócios (*Business Intelligence*);
BRANCHING:	Ponteiro móvel que a ponta para o último *commit*;
COMMIT:	Utilizado para salvar alterações;
CSS:	Linguagem de folhas de estilo (*Cascading Style Sheets*);
HTML:	Linguagem de Marcação de Hipertexto (*HyperText Markup Language*);
IDE:	Ambiente de Desenvolvimento Integrado (*Integrated Development Environment*);
JAVASCRIPT:	Linguagem de programação;
MVC:	Modelo-visão-controlador (*model-view-controller*);
PHP:	Processador de Hipertexto (*Hypertext Processor*) é uma linguagem de programação para desenvolvimento *web*;
SAAS:	Software como serviço (*software as a service*);
TECNOLOGIA *MOBILE*:	Refere-se ás tecnologias voltada para dispositivos moveis;
TECNOLOGIA *WEB*:	Refere-se às tecnologias voltadas ao desenvolvimento de sites, portais, sistemas e aplicações *web*. Têm-se como exemplo o *HTML*, *CSS*, *JavaScript*, *PHP*, etc.

SUMÁRIO

1 INTRODUÇÃO 13
1.1 OBJETIVOS 15
1.1.1 Objetivo Geral 15
1.1.2 Objetivos Específicos 15
1.2 METODOLOGIA 16
1.2.1 Classificação da Pesquisa 16
1.2.1.1 Com Base nos Objetivos 16
1.2.1.2 Com Base nos Procedimentos Técnicos 16
1.2.1.3 Com Base na Natureza da Pesquisa 17
1.2.1.4 Com Base na Abordagem do Problema 17
1.3 ESTRUTURA DO TRABALHO 17
2 FUNDAMENTAÇÃO TEÓRICA 18
2.1 SISTEMA DE INFORMAÇÃO 18
2.1.1 Sistema de Tomada de Decisão 18
2.1.2 Sistema de Informação Gerencial 19
2.2 CORPO DE BOMBEIROS 19
2.3 PROGRAMA OPERAÇÃO ALERTA VERMELHO 20
2.4 TECNOLOGIAS DO PROJETO 21
2.4.1 PHP 22
2.4.2 HTML 22
2.4.3 JavaScript 22
2.4.4 CSS 23
2.4.5 Bootstrap 23
2.4.6 MySQL 23
2.4.7 Model View Controller 24
2.5 *FRAMEWORK* DE DESENVOLVIMENTO 25
2.5.1 Zend *Framework* 25
2.6 ANDROID *STUDIO* 26
2.7 CONTROLE DE VERSÃO DE *SOFTWARE* 26
2.8 PROCESSO DE DESENVOLVIMENTO DE PROJETO DE *SOFTWARE* 27
2.8.1 Engenharia de *Software* 27
2.8.2 Processo e Metodologia de Desenvolvimento de *Software* 28
2.8.3 Metodologias Ágeis 30
2.8.4 Qualidade de *Software* 32
3 PROJETO 33
3.1 VISÃO GERAL DO PROJETO DE *SOFTWARE* 34
3.1.1 Módulo do Corpo de Bombeiros 35
3.1.1.1 Levantamento de requisitos do Módulo do Corpo de Bombeiros 35
3.1.1.2 Diagrama de Caso de uso do Módulo do Corpo de Bombeiros 46
3.1.1.3 Diagrama de Classes do Módulo do Corpo de Bombeiros 67
3.1.2 Módulo do Aluno 68
3.1.2.1 Levantamento de requisitos do Módulo do Aluno 68
3.1.2.2 Diagrama de Caso de uso do Módulo do Aluno 69
3.1.2.3 Diagrama de classes do Módulo do Aluno 73
3.1.3 Módulo da População 74
3.1.3.1 Levantamento de requisitos do Módulo da População 74

3.1.3.2 Diagrama de Caso de Uso do Módulo da População 77
3.1.3.3 Diagrama de classes do Módulo da População 82
3.2 DIAGRAMA DE COMPONENTES 83
3.3 MODELO ENTIDADE RELACIONAMENTO 84
4 DESENVOLVIMENTO 85
4.1 DESENVOLVIMENTO DO SISTEMA *WEB* 85
4.1.1 Estrutura do Sistema 85
4.1.1.1 Módulo do Corpo de Bombeiros 85
4.1.1.2 Módulo do Aluno 87
4.1.1.3 Módulo da População 88
4.2 DESENVOLVIMENTO DO APLICATIVO MÓVEL 89
4.2.1 Estrutura do Sistema 89
4.2.1.1 Alerta Vermelho 89
4.2.1.2 Alerta Vermelho Aluno 92
4.3 PUBLICAÇÃO E DISPONIBILIZAÇÃO DO SISTEMA 94
4.3.1 Estatísticas de Acesso e Avaliação do Sistema 95
5 CONSIDERAÇÕES FINAIS 96
6 TRABALHOS FUTUROS 97
REFERÊNCIAS BIBLIOGRÁFICAS 98
APÊNDICES 103

1 INTRODUÇÃO

O programa de extensão Operação Alerta Vermelho é uma parceria entre a Unifebe e o Corpo de Bombeiros, que tem por objetivo identificar os bairros com maiores riscos de acidentes domiciliares e passar instruções de segurança para a população de Brusque. Neste programa os acadêmicos se dirigem até o quartel do Corpo de Bombeiros e recebem as instruções necessárias para passa-las a população. Após esse treinamento os acadêmicos acompanhados dos bombeiros se deslocam a um bairro pré-definido pelo Corpo de Bombeiros e realizam a pesquisa de campo junto à população.

Atualmente o programa Operação Alerta Vermelho mantém um conjunto de itens mapeados os quais visam identificar, por exemplo se a pessoa conhece o número de emergência do Corpo de Bombeiros, se sabe utilizar extintor de incêndios, sabe agir diante de uma situação de emergência, verifica a instalação do gás de cozinha e a validade da mangueira e da válvula de segurança, é conferido se o sistema elétrico possui disjuntor e também é analisado se a casa está em situação de risco como deslizamento e enchente entre outros itens.

Durante a pesquisa o pesquisador utiliza um formulário impresso do qual após cada resposta das perguntas feitas durante as entrevistas são preenchidas manualmente. No último dia da pesquisa os acadêmicos realizam a última etapa da pesquisa na própria unidade do Corpo de Bombeiros para transcrever estas informações no Excel a fim de apresentar seus resultados graficamente com o somatório dos dados coletados.

Baseado nesses gráficos o Corpo de Bombeiros utiliza-os para realizar tomadas de decisões durante algum evento de emergência na cidade de Brusque que possa ocorrer por desastre naturais como enchentes, descargas elétricas e desmoronamentos.

Todas as informações coletadas são de extrema importância para o Corpo de Bombeiros, porém como essas informações são todas preenchidas em um formulário de papel torna-se um grande risco em rasurar ou até mesmo perder essas informações.

Outra dificuldade encontrada é ter organizado as informações históricas, pois como é utilizado planilhas em Excel se torna algo muito trabalhoso e demorado de pesquisar as datas e históricos de cada bairro. Sendo que essas planilhas acabam dificultando o trabalho dos bombeiros levando muito tempo para a tomada de decisão.

Quando acontece um evento de desastre natural, se torna difícil consultar cada gráfico e comparar cada pesquisa que foi realizada para ver qual é a região que precisa de mais atenção do Corpo de Bombeiros.

Percebendo que o Corpo de Bombeiros teria grandes benefícios com a utilização da tecnologia, podendo assim por meio de um sistema automatizar seus processos e facilitando a consulta a informação e agilizando no processo de tomada de decisão. Foi discutido juntamente com o orientador na Unifebe e também o responsável do Corpo de Bombeiros para criar este projeto e desenvolve-lo no trabalho de conclusão de curso.

Esta proposta de pesquisa irá disponibilizar para a comunidade e ao Corpo de Bombeiros de Brusque resultados do programa Alerta Vermelho, trabalho realizado pelos acadêmicos que prestam serviços através da bolsa de estudos do projeto social da Unifebe.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Desenvolver um sistema via plataformas *web* e *mobile* para o programa “Operação Alerta Vermelho” do Corpo de Bombeiros de Brusque, automatizando as pesquisas referente ao levantamento de prevenções, riscos, acidentes domiciliares para auxiliar no apoio a tomada de decisão.

1.1.2 Objetivos Específicos

- Pesquisar sobre referências bibliográficas na área de aplicações voltadas a alertas e apoio ao Corpo de Bombeiros;
- Definir juntamente com o Corpo de Bombeiros de Brusque os requisitos do programa Operação Alerta Vermelho;
- Fazer levantamentos das tecnologias das plataformas *web* e *mobile* utilizadas para o desenvolvimento do sistema Alerta Vermelho e definir sua metodologia de desenvolvimento e a especificação técnica do projeto;
- Implementar o sistema conforme o escopo, tecnologias e metodologia *Scrum* identificadas para desenvolver o projeto;
- Validar junto ao Corpo de Bombeiros a qualidade do sistema desenvolvido;
- Disponibilizar para o Corpo de Bombeiros e a população acesso público ao sistema Alerta Vermelho por meio das plataformas *web* e *mobile*;

1.2 METODOLOGIA

Serão utilizados o tipo de pesquisa descritiva e o método de pesquisa bibliográfica.

Conforme SILVA; SCHAPPO (2002), o tipo de pesquisa descritiva:

> "Visa à observação, o registro, análise e correlacionar fenômenos ou fatos em um contexto na busca da frequência com que eles ocorrem. Tem como objetivo primordial a descrição das características de determinada população ou fenômeno ou então, o estabelecimento de relações entre as variáveis que ocorrem. Este é um tipo de pesquisa que se fundamenta em estudos e que concentra-se na análise, na descrição de características ou propriedades, ou ainda, das relações entre essas propriedades e determinados fatos/fenômenos relacionados à certa realidade".

Conforme GIL (2010, p.29), sobre o método de pesquisa bibliográfica:

> "A pesquisa bibliográfica é elaborada com base em material já publicado. Tradicionalmente, esta modalidade de pesquisa inclui material impresso, como livros, revistas, jornais, teses, dissertações e anais de eventos científicos. Todavia, em virtude da disseminação de novos formatos de informação, estas pesquisas passaram a incluir outros tipos de fontes, como discos, fitas magnéticas, CDs, bem como o material disponibilizado pela Internet".

1.2.1 Classificação da Pesquisa

Com base na classificação proposta por (SILVA, 2005), este trabalho de conclusão de curso enquadra-se da seguinte forma:

1.2.1.1 Com Base nos Objetivos

Exploratória e Explicativa, pois com o objetivo de aprimorar ideias, foi realizada uma pesquisa bibliográfica e análise de exemplos, apresentando ao final características explicativas.

1.2.1.2 Com Base nos Procedimentos Técnicos

Bibliográfica e Experimental, pois foi elaborada com base em material já existente (artigos e livros) e avaliada experimentalmente através de simulações.

1.2.1.3 Com Base na Natureza da Pesquisa

Aplicada. A pesquisa gera conhecimentos com finalidades imediatas, por exemplo, indica os fatores de riscos baseados nos dados coletados.

1.2.1.4 Com Base na Abordagem do Problema

Quantitativa. Utilizaram-se os parâmetros estatísticos das simulações para análise e qualificação do modelo proposto.

1.3 ESTRUTURA DO TRABALHO

Este trabalho de conclusão de curso será desenvolvido em três partes distintas:

A primeira parte, se baseará no levantamento de referências bibliográficas em livros, artigos, materiais na internet e informações do Corpo do Bombeiros de Brusque. A fundamentação teórica, irá abranger os seguintes assuntos: Sistema de Informação, Corpo de Bombeiros, Programa Operação Alerta Vermelho, Tecnologias do Projeto, Processo de Desenvolvimento de Projeto de *Software*, *Framework* de Desenvolvimento, Android *Studio* e Controle de Versão de *Software*.

Na segunda parte será feito o levantamento dos requisitos necessários, e após isso serão criados os diagramas necessários para um desenvolvimento sem falhas, de acordo com a necessidade dos usuários e sem atrasos. Os diagramas serão criados no *software* Astah Professional, que é um *software* utilizado para modelagem de diagramas.

E por fim, a terceira parte será realizado o desenvolvimento dos códigos fontes do sistema. O desenvolvimento do aplicativo Alerta Vermelho será realizado na IDE Android *Studio*, utilizando a linguagem de programação Android, a aplicação vai ter uma *web view* onde estará direcionando para o *web* site desenvolvido. O desenvolvimento do *web site* será realizado no ambiente de desenvolvimento integrado NetBeans IDE com as linguagens de programação PHP e JavaScript, junto com as linguagens de formatação HTML e de estilo CSS. Além disso, serão utilizados os *frameworks* Bootstrap e Zend, para auxiliar no desenvolvimento. O banco de dados que será utilizado é o MySQL, pois ele tem uma boa integração com a linguagem a ser utilizada. E por fim os testes serão realizados juntamente com o Corpo de Bombeiros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capitulo irá abordar os assuntos referente a Sistema de Informação, Sistema de Tomada de Decisão, Sistema de Informação Gerencial, Corpo de Bombeiros, Programa Operação Alerta Vermelho, Tecnologias do Projeto, PHP, HTML, JavaScript, CSS, Bootstrap, MySQL, *Model View Controller*, *Framework* de Desenvolvimento, Zend *Framework*, Android *Studio*, Controle de Versão de *Software*, Processo de Desenvolvimento de Projeto de *Software*, Engenharia de *Software*, Processo e Metodologia de *Software* e Qualidade de *Software*.

Assuntos estes tratados neste trabalho de conclusão de curso e que são de extrema importância para o desenvolvimento do sistema Alerta Vermelho de Brusque.

2.1 SISTEMA DE INFORMAÇÃO

Sistema de Informação é a expressão utilizada, para descrever um sistema. É um conjunto de elementos que interagem entre si, é composto por pessoas, máquina e processos. Seu objetivo é armazenar, tratar e fornecer informações de tal modo a apoiar as funções ou processos de uma organização.

Segundo Freitas et al. (1997, p. 77), os "sistemas de informações são mecanismos cuja função é coletar, guardar e distribuir informações para suportar as funções gerenciais e operacionais das organizações".

2.1.1 Sistema de Tomada de Decisão

Um sistema de tomada de decisão hoje em dia é comum dentro das organizações, nas grandes empresas ele se torna essencial. Ele permite ao gestor o controle da instituição, podendo assim analisar as informações e históricos de forma fácil e pratica.

“Os sistemas de apoio à decisão são sistemas de informação interativos que assistem o decisor a aceder a problemas mal estruturados oferecendo modelos analíticos e acesso a bases de dados" (ZWASS, 1992).

O uso do *software* de tomada de decisão, como por exemplo, o BI (*business intelligence*), se tornou um grande aliado para o gestor. O BI permite coletar, organizar, analisar e disseminar o conteúdo para que ele possa servir prontamente aos objetivos da empresa.

O auxílio de um *software* de tomada de decisão se faz importante, pois por meio dele o gestor consegue tomar decisões antecipadas e assertivas.

Para Oliveira (2001, p. 37), a informação é o produto da análise dos dados existentes na empresa, devidamente registrados, classificados, organizados, relacionados e interpretados em um determinado contexto, para transmitir conhecimento e permitir a tomada de decisão de forma otimizada. A informação representa a consolidação de poder na empresa, desde o momento de posse dos dados básicos que são transformados em informação, até a possibilidade de otimizar conhecimentos técnicos, domínios de políticas e possibilidade de maior especialização e consequente respeito profissional ao executivo considerado.

2.1.2 Sistema de Informação Gerencial

Sistemas de Informação Gerencial (SIG), são sistemas ou processos que fornecem as informações necessárias para gerenciar com eficácia as organizações. Os SIGs possuem uma multiplicidade de produtos de informação, que são apresentadas através de relatórios, que ajudam no processo para a tomada de decisão.

O Sistemas de Informações Gerenciais (SIG) "é o conjunto de tecnologias que disponibiliza os meios necessários à operação do processo decisório em qualquer organização por meio do tratamento dos dados disponíveis" (CRUZ, 2008, p.56).

Um sistema de tomada de decisão permiti ao gestor ter mais segurança no controle da instituição, tendo maior agilidade ao acesso da informação e podendo assim se antecipar diante de problemas que venham surgir.

As informações para a tomada de decisão podem ser textuais (relatórios descritivos), por planilhas ou de modo gráfico. Este último caso é o preferido pelos administradores, pois oferece mais informações em menor espaço.

2.2 CORPO DE BOMBEIROS

O Corpo de Bombeiros da cidade de Brusque, é a 3ª companhia de bombeiros militar de Santa Catarina, foi fundado em 9 de novembro de 1982, a instituição faz parte do Corpo de Bombeiros Militar de Santa Catarina (CBMSC).

Segundo esta referência o, "órgão da administração direta do Governo do Estado de Santa Catarina, é uma instituição prestadora de serviços públicos na área da segurança pública, tendo

como jurisdição o território catarinense. Instituição estatal de direito público, têm objetivos definidos em leis que orientam e se constituem na sua razão de ser. Constitucionalmente estruturada como Força Auxiliar e Reserva do Exército Brasileiro é organizada – a exemplo das Forças Armadas - com base na disciplina e hierarquia e composta por militares estaduais" (CBMSC, 2016).

2.3 PROGRAMA OPERAÇÃO ALERTA VERMELHO

O Alerta Vermelho é um programa de extensão da Unifebe em parceria com o Corpo de Bombeiros de Brusque, que tem por finalidade possibilitar a participação por parte dos acadêmicos, em ações desenvolvidas junto à comunidade, com foco na transmissão de conhecimentos básicos para a prevenção de incêndios em edificações unifamiliares e acidentes domésticos. Com o objetivo de transmitir de forma prática e eficiente, conhecimentos básicos necessários para prevenir a ocorrência de incêndios e acidentes em residências unifamiliares, bem como atuar no enfrentamento a ocorrências dessa natureza.

O programa Alerta Vermelho é realizado em cinco encontros no sábado no período da tarde. No primeiro encontro é transmitido aos acadêmicos conhecimentos básicos de segurança referente a incêndios e acidentes em residências unifamiliares, para que os mesmos estejam aptos a responder qualquer dúvida que possa surgir durante a pesquisa realizada na comunidade. Durante os próximos encontros os acadêmicos já saem a campo para realizar a pesquisa juntamente acompanhado do Corpo de Bombeiros.

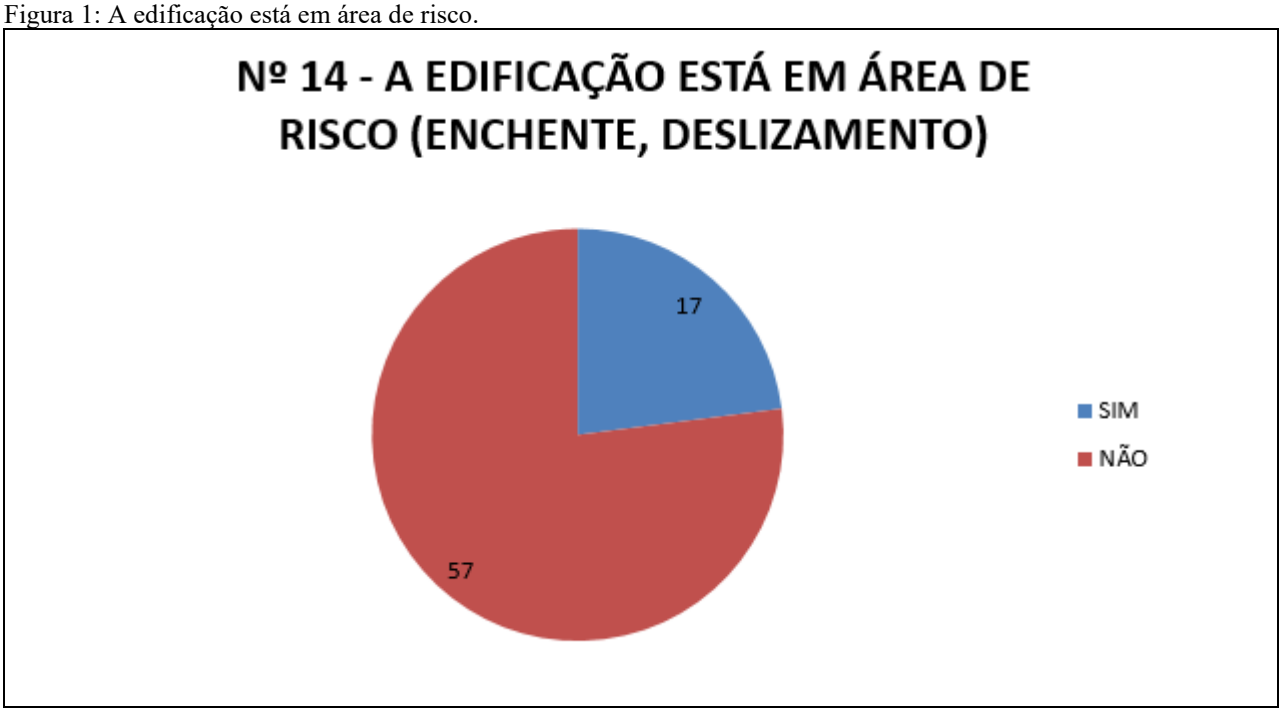
A pesquisa é realizada em um bairro escolhido pelo Corpo de Bombeiros, os acadêmicos são separados em duplas e se dirigem as residências com um questionário impresso e preenchem o questionário de acordo com a resposta do entrevistado.

O questionário contém perguntas de segurança como se o entrevistado conhece o número de emergência do Corpo de Bombeiros, se sabe utilizar extintor de incêndios, sabe agir diante de uma situação de emergência, o acadêmico verifica a instalação do gás de cozinha, a validade da mangueira e da válvula de segurança, é conferido se o sistema elétrico possui disjuntor e também é analisado se a casa está em situação de risco como deslizamento e enchente.

No último encontro os acadêmicos ficam no quartel e fazem os somatórios das respostas de cada questão com o intuito de realizar um relatório por meio de gráficos feito através do Excel.

A Figura 1 representa o somatório de respostas da pergunta "a edificação está em área de risco" realizada na pesquisa de campo no primeiro semestre de 2016.

Figura 1: A edificação está em área de risco.



Fonte: Corpo de Bombeiros de Brusque (2016).

Para cada pergunta realizada no questionário é feito um gráfico, finalizado este gráfico é entregue um relatório para o Corpo de Bombeiros e outro para Unifebe.

Esse relatório por gráfico é muito importante para o Corpo de Bombeiros, pois ele auxilia na tomada de decisão durante algum evento de desastre natural que possa ocorre na cidade de Brusque.

2.4 TECNOLOGIAS DO PROJETO

Neste tópico será abordado as tecnologias utilizadas para desenvolver o sistema Alerta Vermelho. O sistema será baseado nas plataformas *web* e *mobile*. Na plataforma *web*, será oferecido para a população o modelo *SaaS* (*software as a service – software* como serviço), desta forma qualquer usuário, a partir de um dispositivo que possua acesso à *internet* e um navegador compatível com HTML5, poderá usufruir do sistema, como exemplos temos *desktops*, *notebooks*, *smartphones*, *tablet's*, entre outros dispositivos. Para a plataforma *mobile*, será oferecido um aplicativo para o sistema operacional *Android*, por ter o maior número de usuários e baixo custo em manutenção.

"Para distribuir informação de maneia global, é necessário haver uma linguagem que seja estendida universalmente por diversos meios de acesso. O HTML se propõe a ser esta linguagem" (FERREIRA; EIS, 2010).

Para o Corpo de Bombeiros de Brusque o serviço será oferecido nas plataformas *web e mobile* um sistema de informação gerencial (SIG), para que possam realizar o cadastro da pesquisa, gerar relatórios de bairros e relatórios comparativos, desta formar poderá gerenciar a informação de forma fácil e prática.

"Os dados são um recurso organizacional essencial que precisa ser administrado como outros importantes ativos das empresas. A maioria das organizações não conseguiria sobreviver ou ter sucesso sem dados de qualidade sobre suas operações internas e seu ambiente externo" (O'BRIEN 2004, p. 133).

2.4.1 PHP

"O PHP (um acrônimo recursivo para *PHP: Hypertext Preprocessor*) é uma linguagem de *script open source* de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento *web* e que pode ser embutida dentro do HTML" (PHP, 2017).

Segundo o manual do PHP (2017), "o PHP como é conhecido hoje, é na verdade o sucessor para um produto chamado PHP/FI. Criado em 1994 por Rasmus Lerdof, a primeira encarnação do PHP foi um simples conjunto de binários *Common Gateway Interface* (CGI) escrito em linguagem de programação C".

2.4.2 HTML

O HTML - *HyperText Markup Lang*uage, que significa Linguagem de Marcação de Hipertexto, é uma linguagem de marcação utilizada para desenvolvimento de sites.

"O HTML foi criado em 1991, por Tim Berners-Lee, no CERN (European Council for Nuclear Research) na Suíça. Inicialmente o HTML foi projetado para interligar instituições de pesquisa próximas, e compartilhar documentos com facilidade. Em 1992, foi liberada a biblioteca de desenvolvimento WWW (World Wide Web), uma rede de alcance mundial, que junto com o HTML proporcionou o uso em escala mundial da WEB" (PACIEVITCH, 2017).

2.4.3 JavaScript

JavaScript é uma linguagem de programação *web* que tem como finalidade adicionar interatividade e comportamentos na página criado pelo HTML. Segundo Maurício Samy Silva (2010, p.23), "JavaScript foi criada pela Netscape em parceria com Sun Microsystems, com a finalidade de fornecer um meio de adicionar interatividade a uma página web".

Conforme Silva (2010, p.23), “JavaScript é uma linguagem desenvolvida para rodar no lado do cliente, isto é, a interpretação e o funcionamento da linguagem dependem de funcionalidades hospedadas no navegador do usuário. Isso é possível porque existe um interpretador JavaScript hospedado no navegador”.

2.4.4 CSS

O CSS - *Cascading Style Sheets*, é uma linguagem de folha de estilos em cascata, é utilizada para personalizar o *layout* de páginas *web*. Segundo o W3schools (2017), “CSS é uma linguagem que descreve o estilo de um documento HTML e descreve como elementos HTML deve ser exibido”.

“Cascading Style Sheets (CSS) é uma "folha de estilo" composta por “camadas” e utilizada para definir a apresentação (aparência) em páginas da internet que adotam para o seu desenvolvimento linguagens de marcação (como XML, HTML e XHTML). O CSS define como serão exibidos os elementos contidos no código de uma página da internet e sua maior vantagem é efetuar a separação entre o formato e o conteúdo de um documento” (PEREIRA, 2017).

2.4.5 Bootstrap

Bootstrap é um *framework front-end* que facilita no desenvolvimento de *web sites* responsivos. Segundo Erian do Nascimento Estraich (2017), o Bootstrap é uma “coleção de vários elementos e funções personalizáveis para projetos da web, empacotados previamente em uma única ferramenta. Ao projetar um site com o Bootstrap, os desenvolvedores podem escolher quais elementos querem usar. E, o mais importante, podem ter a certeza de que os elementos escolhidos não conflitarão entre si”.

“O Bootstrap foi uma ferramenta desenvolvida originalmente por Mark Otto e Jacob Thorton, então engenheiros do Twitter, como uma tentativa de incentivar o uso de uma única estrutura pela equipe de engenharia do Twitter” (ESTRAICH, 2017).

2.4.6 MySQL

“O MySQL é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL (Structured Query Language ou Linguagem de Consulta Estruturada) como interface” (TEIXEIRA, 2017).

“Banco de dados é uma coleção de dados, organizados em tabelas, referentes a um assunto ou propósito específico, com o objetivo de organizar os dados de modo a tornar a vida dos usuários do negócio em questão mais prática, precisa, rápida e confiável” (MILANI, 2010).

Optou-se em utiliza-lo, por questões de custo, e que também não deixa a desejar quando se utiliza seu motor de banco de dados InnoDB.

“Muitas das maiores organizações e mundiais de mais rápido crescimento, incluindo Facebook, Google, Adobe, Alcatel Lucent e Zappos contam com o MySQL para poupar tempo e dinheiro alimentar seus sites de alto volume, sistemas críticos de negócios e *software* empacotado” (MYSQL, 2016).

2.4.7 Model View Controller

Segundo Ademir Cristiano Gabardo (2012), “MVC é um padrão de design de projetos de *software* que separa as camadas de lógica e negócio da camada de apresentação”.

“Os models ou modelos são os componentes da camada de abstração de dados. Cada model é uma classe que herda as propriedades da superclasse model do framework e que, dessa forma, tem as características e comportamentos da superclasse. O mais comum na utilização prática dos models é que a eles sejam atribuídas as tarefas de gravar e recuperar dados do banco de dados, muito embora haja outras maneiras para executar essa tarefa” (GABARDO ,2012).

“As views são a principal parte da camada de apresentação. Elas recebem dados dos controllers e não deverão se comunicar diretamente com os models ou com bancos de dados, nem deverão enviar dados diretamente para models ou bancos de dados. Views basicamente são arquivos .html ou .php, contendo estruturas HTML, XHTMLou HTML5 e outros formatos de saída, que apresentarão os dados gerados pela aplicação para o usuário” (GABARDO, 2012).

“Os controllers são a camada responsável pela comunicação entre a camada de dados e abstração (models) e a camada de apresentação (views). Os controllers também são responsáveis por carregar os models e views, receber informações e validá-las antes de encaminhar ao model etc. Um controller é uma classe que herda as propriedades da classe ascendente, que é o controller do framework” (GABARDO, 2012).

2.5 *FRAMEWORK* DE DESENVOLVIMENTO

Um *framework* de desenvolvimento é um conjunto de componentes que tem por finalidade facilitar e agilizar o processo de desenvolvimento de *software*.

"Um *framework* descreve a arquitetura de um sistema orientado a objetos, os tipos de objetos e as interações entre os mesmos. Ele pode ser vislumbrado como o esqueleto – *template* – de uma aplicação que pode ser customizado pelo programador e aplicado a um conjunto de aplicações de um mesmo domínio. Com *frameworks* não se busca apenas reutilizar simples componentes de *software*, mas subsistemas, aumentando assim o grau de reutilização e contribuindo para uma melhor qualidade do produto – *software*" (LEITE, 2016).

"[...] em outras palavras, um *framework* de desenvolvimento é uma base de onde se pode desenvolver algo maior ou mais específico. É uma coleção de códigos-fonte, classes, funções, técnicas e metodologias que facilitam o desenvolvimento de novos *softwares*" (MINETTO, 2007).

2.5.1 Zend *Framework*

O *Zend Framework* é um *framework* de código aberto para aplicações PHP, é escrito em PHP 5 e utiliza os recursos de orientação a objetos fornecidos, além de prezar a utilização de padrões de projetos, visando à construção de componentes altamente reutilizáveis.

Para Minetto (2007) a orientação à objetos foi atingida com sucesso a partir lançamento da versão 5 do PHP. A partir desse "amadurecimento" o PHP começou a chamar a atenção de programadores acostumados a utilizar o paradigma de programação orientada a objetos, além de favorecer a criação de aplicações mais robustas e de melhores componentes de *software*.

O *Zend Framework* utiliza padrões de projetos voltado ao desenvolvimento de aplicações no modelo MVC (*Model View Controller*). É um dos *frameworks* mais utilizados pela comunidade PHP, o principal patrocinador do projeto Zend *Framework* é a Zend Technologies, maior desenvolvedora de ferramentas para PHP do mundo e grande contribuidora da linguagem, este *framework* também conta com o apoio de grandes empresas como a Oracle e a IBM.

2.6 ANDROID *STUDIO*

Para o desenvolvimento *mobile* optou-se em utilizar o Android *Studio*, é uma IDE desenvolvida pela Google para criação de aplicativos moveis para a plataforma Android.

O Android *Studio* atualmente é a IDE preferida dos desenvolvedores para o desenvolvimento de aplicativos moveis, pois é homologado exclusivamente para o sistema operacional Android, e também por ser gratuita e baseada na IDE IntelliJ IDEA que por outro lado é paga.

"Uma vantagem de desenvolver aplicativos Android é a franqueza (ou grau de abertura) da plataforma" (DEITEL, 2013).

2.7 CONTROLE DE VERSÃO DE *SOFTWARE*

O controle de versão de *software* é importante para o desenvolvedor de *software*, pois por meio dele é possível rastrear todas as alterações realizadas, e caso futuramente ocorra algum problema com o código, o desenvolvedor poderá retroceder a parte do código com problema.

"O controle de versão é um sistema que registra as mudanças feitas em um arquivo ou um conjunto de arquivos ao longo do tempo de forma que você possa recuperar versões específicas. [...] você pode usá-lo com praticamente qualquer tipo de arquivo em um computador" (GIT, 2016).

Desta forma o uso de um sistema de controle de versionamento auxilia o desenvolvedor a saber quais alterações foram realizadas no programa em desenvolvimento. Muitas empresas hoje em dia utiliza um sistema de versionamento como o Facebook, Google, Twitter entre outras.

Para este trabalho optou-se em utilizar o sistema de controle de versão Git. Git é um "sistema de controle de versão de *software* distribuído, diferente de outros sistemas de controle versão que utilizam um servidor para centralizar o repositório de dados, o Git permite que cada usuário tenha um repositório em seu próprio equipamento e funciona mesmo *off-line"*(GIT, 2016).

O envio de arquivos acontece em duas fases, o primeiro é por *Commit*, fase em que as alterações são armazenadas apenas localmente. E o segundo por *Push*, fase em que as alterações são enviadas para um servidor que concentra todos os conjuntos de alterações (chamados de *changesets*) que podem ser recombinados entre si livremente.

O Git é compatível com os principais ambientes de desenvolvimento como Eclipse, Netbeans, Android *Studio* e Visual *Studio*.

"Desde sua concepção em 2005, o GIT evoluiu e amadureceu a ponto de ser um sistema fácil de usar e ainda assim mantém essas qualidades iniciais. É incrivelmente rápido, bastante eficiente

com grandes projetos e possui um sistema impressionante de *branching* para desenvolvimento não-linear" (GIT, 2016).

2.8 PROCESSO DE DESENVOLVIMENTO DE PROJETO DE *SOFTWARE*

Para que um *software* seja desenvolvido de forma consistente, é preciso aliar boas práticas da engenharia de *software* com um robusto e eficiente processo de desenvolvimento. Diferentes tipos de sistemas necessitam de diferentes processos de desenvolvimento. Por exemplo, um *software* de tempo real de uma aeronave deve ser completamente especificado antes do início do desenvolvimento, enquanto que um sistema de comércio eletrônico a especificação e o desenvolvimento do software podem ser conduzidos paralelamente. O uso de um processo de *software* inadequado pode reduzir a qualidade ou a utilidade do produto de *software* a ser desenvolvido e/ou aumentar os custos de desenvolvimento. Este fato leva as organizações que produzem *software* a usar processos de desenvolvimento que sejam eficientes e que atendam plenamente suas necessidades (SOMMERVILLE, 2007).

2.8.1 Engenharia de *Software*

Segundo Ian Sommerville (2011), "a Engenharia de *Software* é uma disciplina da engenharia de sistemas que se ocupa de todos os aspectos da produção de *software*, desde os estágios iniciais de levantamento e especificação de requisitos até a implantação e manutenção, ou seja, que entrou em operação. É um conjunto de atividades, parcialmente ou totalmente ordenadas, com a finalidade de obter um produto de *software* de qualidade e cumprir corretamente os contratos de desenvolvimento".

"É considerado um dos principais mecanismos para obter um *software* de qualidade e cumprir corretamente os contratos de desenvolvimento" (SOMMERVILLE, 2011).

Segundo Roger Pressman (2006), a Engenharia de *Software* poderá ser mais bem entendida como uma tecnologia em camadas ou níveis, conforme pode ser vista na Figura 2.

Figura 2: Camadas da engenharia de software.

Fonte: Adaptado de Pressman (2011).

Na base da Figura, formando a camada foco na qualidade, dá-se ênfase à preocupação de qualquer disciplina de engenharia, que é a qualidade. A qualidade na Engenharia de *Software* é baseada nos conceitos de gerenciamento de qualidade total (TQM – *Total Quality Management)* para a melhoria contínua dos processos para obter sucesso em longo prazo através da satisfação dos clientes.

A camada de processo permite integrar as camadas de métodos e de ferramentas para que se possa desenvolver um *software* nos prazos acordados e de maneira adequada. Um processo permite que se planeje e se controle projeto de *software*.

A camada de métodos provê as abordagens e as atividades necessárias para a construção de um *software*. Os métodos abrangem um conjunto amplo de tarefas que incluem análise de requisitos, projeto, implementação, testes e manutenção. Os métodos de Engenharia de *Software* são baseados em um conjunto de princípios que governam cada área de tecnologia e incluem atividades de modelagem e técnicas descritivas.

A camada de ferramentas provê apoio automatizado ou semi automatizado para processos e métodos. As ferramentas da área de Engenharia de *Software* são conhecidas como CASE (Engenharia de *Software* Apoiada por Computador, do termo em inglês *Computer-Aided Software Engeneering*).

2.8.2 Processo e Metodologia de Desenvolvimento de *Software*

A existência do processo de *software* é extremamente importante pois oferece estabilidade, padronização e controle na execução das atividades tornando os projetos mais fáceis de serem realizados e garantindo maior qualidade do produto de *software* a ser entregue ao cliente.

“A elaboração de *software* de computador é um processo de aprendizado, e o resultado, é a incorporação de conhecimentos coletados, destilados e organizados à medida que o processo é

conduzido. Processo é o alicerce da engenharia de *software*. É ele que permite o desenvolvimento racional e oportuno de *softwares* de computador" (PRESSMAN, 2006).

Um processo de *software* é composto por métodos (aquilo que diz o que, em uma determinada tarefa), por ferramentas (que dão suporte automatizado aos métodos) e procedimentos (que fazem o elo de ligação entre os métodos e as ferramentas). Uma organização que possui um processo de engenharia de *software* deverá levar muito a sério estes três princípios de processo, a fim de que seus projetos de *software* sejam de sucesso, ou seja, equipes produtivas e *softwares* bem feitos.

A melhoria do processo de *software* pode ser considerada hoje uma das grandes prioridades para as organizações que trabalham com *software*. Isto se deve à exigência do mercado por produtos com maior qualidade, que sejam entregues mais rapidamente e com menor custo de desenvolvimento.

Estudos apontam que ao tentarem melhorar seus processos, as empresas estão em busca de: entender as características dos processos existentes e os fatores que afetam a sua capacidade; planejar, justificar e implementar ações que modificarão os processos, tornando-os mais coerentes com as necessidades de negócios e; avaliar os impactos e benefícios ganhos, comparando-os com os custos advindos das mudanças realizadas (OLIVEIRA, 2016).

Segundo Fernandes (1999), metodologia de sistemas se define como um conjunto de normas, procedimentos, técnicas e ferramentas de análise que definem o padrão desejado por uma empresa para o desenvolvimento de projetos de sistemas.

Fernandes (1999) nos diz que, para que uma metodologia de desenvolvimento de sistemas seja consistente, oferecendo maior produtividade e qualidade, deverá atender a alguns requisitos fundamentais:

- **Padronização**: executar as atividades de maneira idêntica, fazendo com que haja aperfeiçoamento do processo;
- **Flexibilidade**: é a capacidade de se adaptar às mudanças;
- **Documentação**: manter informações sobre o produto e garantir rapidez diante das mudanças;
- **Modularização**: consiste em dividir um conjunto de atividades em vários conjuntos menores, objetivando melhor visualização e acompanhamento por parte de todos os interessados no resultado final;

- **Planejamento**: forma madura de administrar o tempo, é programar o futuro em relação às metas e aos objetivos a serem alcançados.

2.8.3 Metodologias Ágeis

O uso de uma metodologia ágil durante o processo de desenvolvimento de *software* se faz muito importante, pois por meio dele é possível garantir maior produtividade e eficiência durante todo o processo.

Segundo André Koscianski (2007), "as metodologias ágeis são adequadas para situações em que a mudança de requisitos é frequente. Para ser realmente considerada ágil, a metodologia deve aceitar a mudança em vez de tentar prever o futuro".

"As Metodologias Ágeis de Desenvolvimento de *Software* são indicadas como sendo uma opção às abordagens tradicionais para desenvolver *softwares*: produzem pouca documentação, é recomendado documentar somente o que será útil. Em essência, as Metodologias Ágeis foram desenvolvidas com o objetivo de vencer as fraquezas percebidas e reais da Engenharia de *Software*" (PRESSMAN, 2010).

"O uso de metodologia, mesmo que ainda não fortemente sedimentada, no desenvolvimento de *software* é de extrema importância, para que o sistema construído atenda às necessidades dos interessados, com um mínimo de qualidade" (LEITE, 2006).

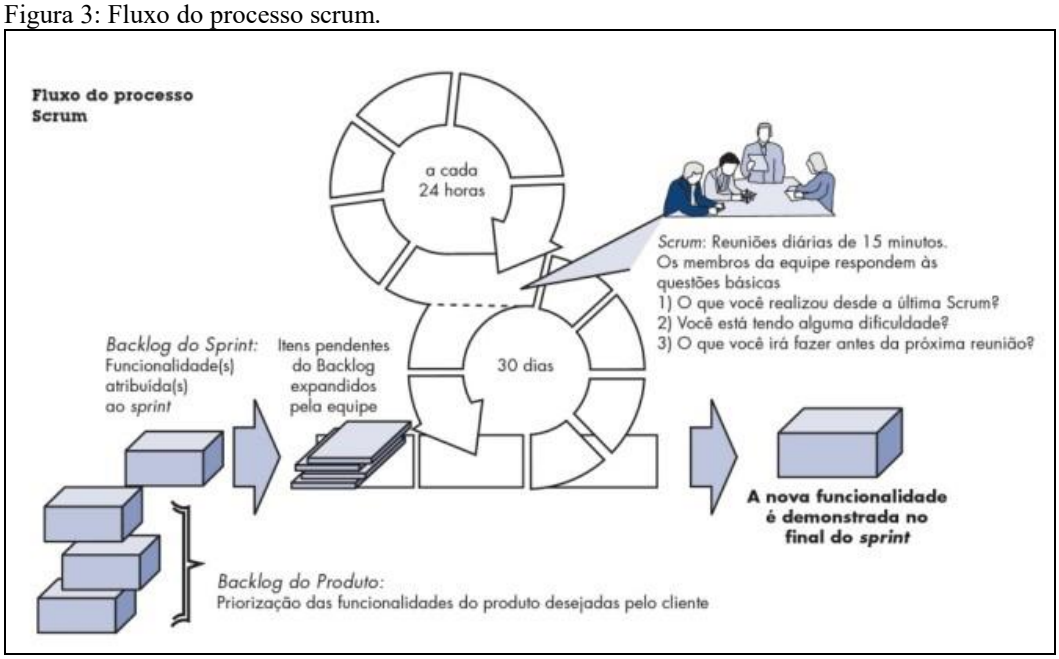
Entre todos os métodos ágeis podem-se citar como exemplo o *Scrum, Extreme Programming, Adaptive Software Development, Dynamic Systems Development* e *Crystal.*

Por ser uma metodologia ágil que possui bastante flexibilidade e controle durante todo processo do desenvolvimento, onde é possível entregar resultados de maneira mais rápida e com menor custo e que não é necessária uma grande equipe de desenvolvedores para se utiliza-la, optou-se em utilizar o *Scrum*.

O nome *Scrum* provém de uma jogada que ocorre durante uma partida de rugby, onde um grupo de jogadores faz uma formação em torno da bola e seus companheiros de equipe trabalham juntos para avançar com a bola em direção ao fundo do campo. O *Scrum* é uma metodologia de gerencia de equipes de desenvolvimento ágil de *software* concebido por Jeff Sutherland e sua equipe de desenvolvimento no início dos anos 90 e mais recentemente foram realizados desenvolvimentos adicionais nos métodos gráficos *Scrum* por Ken Schwaber e Mike Beedle (PRESSMAN, 2011).

Segundo Pressman (2011), “o *Scrum* enfatiza o uso de um conjunto de padrões de processos de *software* que provam ser eficazes para projetos com prazos de entrega apertados, requisitos mutáveis e críticos de negócio”. O fluxo geral do processo *Scrum* é ilustrado na Figura 3.

Figura 3: Fluxo do processo scrum.



Fonte: Adaptado de Pressman (2011).

Pressman (2011), nos apresenta as etapas do *Scrum*:

> “Registro pendente de trabalhos (Backlog) – uma lista com prioridades dos requisitos ou funcionalidades do projeto que fornecem valor comercial ao cliente. Os itens podem ser adicionados a esse registro em qualquer momento.
>
> Urgências (corridas de curta distância) sprints – consistem de unidades de trabalho solicitadas para atingir um requisito estabelecido no registro de trabalho (backlog) e que precisa ser ajustado dentro de um prazo já fechado” (p. 95).

No *Scrum*, os projetos são divididos em ciclos (tipicamente mensais) chamados de *Sprints*. O Sprint representa um *Time Box* dentro do qual um conjunto de atividades deve ser executado. Metodologias ágeis de desenvolvimento de software são iterativas, ou seja, o trabalho é dividido em iterações, que são chamadas de *Sprints* no caso do *Scrum* (DESENVOLVIMENTO ÁGIL, 2016).

Pressman (2011), nos apresenta ainda essas próximas etapas do *Scrum*:

> “Alterações (por exemplo, itens do registro de trabalho – *backlog work itens*) não são introduzidas durante execução de urgências (*sprint*). Portanto, o sprint permite que os membros de uma equipe trabalhem em um ambiente de curto prazo, porém estável.
>
> Reuniões *Scrum* – são reuniões curtas (tipicamente 15 minutos), realizadas diariamente pela equipe *Scrum*. São feitas três perguntas-chave e respondidas por todos os membros da equipe:

> O que você realizou desde a última reunião de equipe?
>
> Quais obstáculos está encontrando?
>
> O que planeja realizar até a próxima reunião da equipe?
>
> *Demos* – entrega do incremento de software ao cliente para que a funcionalidade implementada possa ser demonstrada e avaliada pelo cliente. É importante notar que a demo pode não ter toda a funcionalidade planejada, mas sim funções que possam ser entregues no prazo estipulado" (p. 95-96).

2.8.4 Qualidade de *Software*

Qualidade de *software* está relacionada a entregar ao cliente o produto final que satisfaça suas expectativas, dentro daquilo que foi acordado inicialmente por meio dos requisitos do projeto. Nesse contexto, qualidade de software que objetiva garantir essa qualidade pela definição de processos de desenvolvimento (ENGHOLM JR., 2010).

Na gestão da qualidade de *software* existem diversas atividades voltadas à garantia da qualidade e ao controle de qualidade de *software*. A primeira é para a definição padronizada das atividades voltadas a prevenção de defeitos e problemas, que podem surgir nos produtos de trabalho. Área que define padrões, metodologias, técnicas e ferramentas de apoio ao desenvolvimento tendo como entrada o plano de qualidade de *software* e os resultados de medições de qualidade. A segunda é voltada para o monitoramento de resultados específicos do projeto, ou seja, a detecção de defeitos, executadas através do uso de técnicas que incluem revisões por pares, teste e análise de tendências, entre outras (VASCONCELOS et al., 2006).

3 PROJETO

A proposta deste projeto vem de encontro com a necessidade do Corpo de Bombeiros em automatizar o processo de execução do programa Operação Alerta Vermelho, onde tem como função identificar quais são os tipos de incidências e riscos domiciliares. Este programa tem parceria com a Unifebe onde os acadêmicos realizam entrevista na comunidade alimentando uma base de dados para que o Corpo de Bombeiro possa ser informado e auxiliar na tomada de decisão.

Com o objetivo de se atingir à entrega completa do sistema, é necessário o cumprimento das seguintes etapas:

1. Levantamento de requisitos ao começo do projeto e ao final de cada s*print* (divisão de ciclos segundo à metodologia *Scrum*);
2. Definição da documentação técnica (casos de uso e diagramas);
3. Codificação do que foi especificado;
4. Validação e manutenção do *software.*

Através da execução destas etapas, espera-se implementar todos os requisitos solicitados pelos usuários no sistema. Ao final da etapa 4, caso o projeto ainda não esteja concluído, reinicia-se o ciclo das etapas quantas vezes for preciso até completar o projeto.

A documentação de um *software*, é algo muito importante, durante o processo de desenvolvimento, é fundamental a utilização de técnicas de modelagem de *software*. Mediante ao uso de digramas é possível ter um maior entendimento do *software* como um todo.

"A utilização de diversos diagramas permite que falhas possam ser descobertas nos diagramas anteriores, diminuindo a possibilidade da ocorrência de erros durante a fase de desenvolvimento do *software*" (UML 2, 2014).

Para desenvolvimento do *front-end* do *software*, optou-se pela adoção de *template open source* produzido em *HTML5*, *CSS* (com *framework Bootstrap*) e *JavaScript* (com *framework Jquery*), pelo fato de possuir uma biblioteca grande de componentes como gráficos, calendários, campos estilizados entre outros, e também permitir desta forma focar nas regras de negócio do sistema. Também será utilizado a biblioteca Google Chats Tools, para apresentar aos usuários uma

visão melhor das pesquisas de cada bairro, por meio dela é possível realizar gráficos intuitivos e dinâmicos.

No *back-end*, ou o sistema em si, será utilizado a linguagem de programação PHP com *Zend Framework*, que é um *framework* robusto, gratuito e com padrão MVC.

A plataforma *mobile*, será desenvolvida utilizando o Android *Studio*. Será utilizado o componente *web view* para exibir o conteúdo do sistema.

Para controle das versões do *software* e seu código fonte, será utilizado o sistema de versão GIT, o qual não exige um servidor para centralizar o repositório.

Destaca-se que a escolha pelas tecnologias (PHP, CSS, HTML5, MYSQL, *BootStrap*, Zend *Framework*, Android *Studio*), para este projeto são todas *open-source* de fácil atualização sem gerar custo por sua utilização, visto que está proposta é atender uma instituição pública.

3.1 VISÃO GERAL DO PROJETO DE *SOFTWARE*

O sistema como um todo abrangerá as plataformas *web* e *mobile*. A plataforma *web* está projetada para executar os módulos: (1) Administrativo: para o uso do Corpo de Bombeiros. (2) Aluno: para o uso do aluno que realizará a entrevista. (3) População: para uso da população a acesso e pesquisa das informações. A plataforma *mobile* está projeta para executar os módulos: (1) Aluno: para o uso do aluno que realizará a entrevista. (2) População: para uso da população a acesso e pesquisa das informações.

Este projeto foi desenvolvido para uso do programa Operação Alerta Vermelho, onde o Corpo de Bombeiros terá uma versão *web* para cadastro das informações que farão parte das entrevistas afim de mapear todos os tipos incidências e alertas que correm nas residências, bem como usuário administrador irá realizar o cadastro dos alunos.

Já os módulos Aluno e População terão duas versões uma *web* e outra *mobile*. O usuário aluno acessará um questionário para realizar as entrevistas junto a população, assim gerando todos os dados necessários do programa Operação Alerta Vermelho. E a versão da população é para que todos tenham acesso dos resultados do projetos e dicas de seguranças. Destaco como principal benefício a automatização do programa Operação Alerta Vermelho garantindo maior segurança sobre seus resultados e a análise de seus dados históricos que auxiliarão no apoio a tomada de decisão. Tanto internamente pelo Corpo de Bombeiros como maior visibilidade a população sobre está situação, visualizando por exemplo como se encontra as prevenções nos bairros e residências.

3.1.1 Módulo do Corpo de Bombeiros

No decorrer será especificado toda a estrutura do Módulo do Corpo de Bombeiros. Este módulo utilizará apenas a plataforma *web*, visto que será de acesso restrito. A seguir será apresentado os requisitos, casos de uso e digrama de classes.

3.1.1.1 Levantamento de requisitos do Módulo do Corpo de Bombeiros

O levantamento de requisitos é fundamental para atender a todas as exigências dos usuários do sistema, garantem que nenhuma funcionalidade solicitada deixará de ser implementada. Durante o processo de desenvolvimento, torna-se necessário a documentação e consulta dos requisitos.

A seguir serão apresentadas as tabelas de levantamento de requisitos pertinentes ao sistema *web*, será documentado requisitos funcionais e não funcionais.

A Tabela 1 apresenta como deverá ser feito o cadastro de bairro.

Tabela 1: Requisito cadastro de bairro.

Identificação		**F1**	
Título		Cadastro de bairro	
Descrição		O sistema deve permitir o cadastro de bairro.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF1.1	Dados Obrigatórios	O cadastro deve conter o nome e a URL do google *maps* do bairro.	Especificação
NF1.2	Código do Bairro	O sistema deve gerar um código automaticamente.	Especificação

Fonte: O Autor (2017).

A Tabela 2 apresenta como deverá ser feito a listagem de bairro.

Tabela 2: Requisito listagem de bairro.

Identificação		**F2**	
Título		Listagem de bairro	
Descrição		O sistema deve permitir a listagem de bairro.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF2.1	Ausência de registros	Em caso de ausência de registro exibir mensagem avisando o usuário.	Especificação
NF2.2	Botões de editar e excluir	Na listagem deve aparecer os botões para editar o cadastro e eliminar o mesmo.	Especificação
NF2.3	Paginação	Para esta listagem deverá haver paginação, com 30 registros por página.	Especificação

Fonte: O Autor (2017).

A Tabela 3 apresenta como deverá ser feito a alteração e exclusão de bairro.

Tabela 3: Requisito alteração e exclusão de bairro.

Identificação		**F3**	
Título		Alteração e exclusão de bairro	
Descrição		O sistema deve permitir alterar e excluir um bairro.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF3.1	Aviso de confirmação	Exibir aviso confirmando uma edição ou exclusão.	Especificação

Fonte: O Autor (2017).

A Tabela 4 apresenta como deverá ser feito o cadastro de ano.

Tabela 4: Requisito cadastro de ano.

Identificação		**F4**	
Título		Cadastro de ano	
Descrição		O sistema deve permitir o cadastro de ano.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF4.1	Dados Obrigatórios	O cadastro deve conter o número do ano precedido de um ponto e o número do semestre.	Especificação
NF4.2	Código do Bairro	O sistema deve gerar um código automaticamente.	Especificação

Fonte: O Autor (2017).

A Tabela 5 apresenta como deverá ser feito a listagem de ano.

Tabela 5: Requisito listagem de ano.

Identificação		**F5**	
Título		Listagem de ano	
Descrição		O sistema deve permitir a listagem de ano.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF5.1	Ausência de registros	Em caso de ausência de registro exibir mensagem avisando o usuário.	Especificação
NF5.2	Botões de editar e excluir	Na listagem deve aparecer os botões para editar o cadastro e eliminar o mesmo.	Especificação
NF5.3	Paginação	Para esta listagem deverá haver paginação, com 30 registros por	Especificação

		página.	

Fonte: O Autor (2017).

A Tabela 6 apresenta como deverá ser feito a alteração e exclusão de ano.

Tabela 6: Requisito alteração e exclusão de ano.

Identificação		**F6**	
Título		Alteração e exclusão de ano	
Descrição		O sistema deve permitir alterar e excluir um ano.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF6.1	Aviso de confirmação	Exibir aviso confirmando uma edição ou exclusão.	Especificação

Fonte: O Autor (2017).

A Tabela 7 apresenta como deverá ser feito o cadastro de pergunta.

Tabela 7: Requisito cadastro de pergunta.

Identificação		**F7**	
Título		Cadastro de pergunta	
Descrição		O sistema deve permitir o cadastro da pergunta.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF7.1	Dados Obrigatórios	O cadastro deve conter a descrição da pergunta.	Especificação
NF7.2	Código da Pergunta	O sistema deve gerar um código automaticamente.	Especificação

Fonte: O Autor (2017).

A Tabela 8 apresenta como deverá ser feito a listagem de pergunta.

Tabela 8: Requisito listagem de pergunta.

Identificação		**F8**	
Título		Listagem de pergunta	
Descrição		O sistema deve permitir a listagem de pergunta.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF8.1	Ausência de registros	Em caso de ausência de registro exibir mensagem avisando o usuário.	Especificação
NF8.2	Botões de editar e excluir	Na listagem deve aparecer os botões para editar o cadastro e eliminar o mesmo.	Especificação
NF8.3	Paginação	Para esta listagem deverá haver paginação, com 30 registros por página.	Especificação

Fonte: O Autor (2017).

A Tabela 9 apresenta como deverá ser feito a alteração e exclusão de pergunta.

Tabela 9: Requisito alteração e exclusão de pergunta.

Identificação		**F9**	
Título		Alteração e exclusão de pergunta	
Descrição		O sistema deve permitir alterar e excluir uma pergunta.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF9.1	Aviso de confirmação	Exibir aviso confirmando uma edição ou exclusão.	Especificação

Fonte: O Autor (2017).

A Tabela 10 apresenta como deverá ser feito o cadastro de opção da pergunta.

Tabela 10: Requisito cadastro de opção da pergunta.

Identificação		**F10**	
Título		Cadastro de opção da pergunta	
Descrição		O sistema deve permitir o cadastro de opção da pergunta.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF10.1	Dados Obrigatórios	O cadastro deve conter a descrição da opção da pergunta.	Especificação
NF10.2	Código da Opção da Pergunta	O sistema deve gerar um código automaticamente.	Especificação
NF10.3	Botão Adicionar Opção	No cadastro deve exibir o botão adicionar opção.	Especificação

Fonte: O Autor (2017).

A Tabela 11 apresenta como deverá ser feito a listagem de opção da pergunta.

Tabela 11: Requisito listagem de opção da pergunta.

Identificação		**F11**	
Título		Listagem de opção da pergunta	
Descrição		O sistema deve permitir a listagem de opção da pergunta.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF11.1	Ausência de registros	Em caso de ausência de registro exibir mensagem avisando o usuário.	Especificação
NF11.2	Botão de editar	Na listagem deve aparecer o botão para editar o cadastro.	Especificação
NF11.3	Paginação	Para esta listagem deverá haver paginação, com 30 registros por	Especificação

		página.	

Fonte: O Autor (2017).

A Tabela 12 apresenta como deverá ser feito a alteração e exclusão de opção da pergunta.

Tabela 12: Requisito alteração e exclusão de opção da pergunta.

Identificação		**F12**	
Título		Alteração e exclusão de Opção da Pergunta	
Descrição		O sistema deve permitir alterar e excluir uma opção da pergunta.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF12.1	Aviso de confirmação	Exibir aviso confirmando uma edição ou exclusão.	Especificação

Fonte: O Autor (2017).

A Tabela 13 apresenta como deverá ser feito a listagem de pesquisa.

Tabela 13: Requisito listagem de pesquisa.

Identificação		**F13**	
Título		Listagem de pesquisa	
Descrição		O sistema deve permitir a listagem de pesquisa.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF13.1	Ausência de registros	Em caso de ausência de registro exibir mensagem avisando o usuário.	Especificação
NF13.2	Botões de visualizar, editar e excluir	Na listagem deve aparecer o botão para visualizar, editar e excluir a pesquisa.	Especificação
NF13.3	Paginação	Para esta listagem deverá haver paginação, com 30 registros por	Especificação

		página.	

Fonte: O Autor (2017).

A Tabela 14 apresenta como deverá ser feito a alteração e exclusão de pesquisa.

Tabela 14: Requisito alteração e exclusão de pesquisa.

Identificação		**F14**	
Título		Alteração e exclusão de pesquisa	
Descrição		O sistema deve permitir alterar e excluir uma pesquisa.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF14.1	Aviso de confirmação	Exibir aviso confirmando uma edição ou exclusão.	Especificação

Fonte: O Autor (2017).

A Tabela 15 apresenta como deverá ser feito o relatório de bairro.

Tabela 15: Requisito relatório de bairro.

Identificação		**F15**	
Título		Relatório de bairro	
Descrição		O sistema deve permitir gerar relatório dos bairros pesquisados.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF15.1	Relatório de Bairro	Relatório que exibe a quantidade de resposta de cada pergunta feita na pesquisa do bairro pesquisado.	Especificação

Fonte: O Autor (2017).

A Tabela 16 apresenta como deverá ser feito o relatório comparativo.

Tabela 16: Requisito relatório comparativo.

Identificação	**F16**
Título	Relatório comparativo
Descrição	O sistema deve permitir gerar relatório

		comparativo.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF16.1	Relatório de Comparativo	Relatório compara a pergunta selecionada com outros bairros.	Especificação

Fonte: O Autor (2017).

A Tabela 17 apresenta como deverá ocorrer o realizar *login* dos usuários no sistema.

Tabela 17: Requisito realizar login.

Identificação		**F17**	
Título		Realizar *login*	
Descrição		Será a função do sistema que dará acesso a funcionalidades administrativas.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF17.1	Aviso de confirmação	Exibir mensagem se o *login* foi realizado com sucesso.	Especificação
NF17.2	Redirecionar	Redirecionar para a página inicial do sistema.	Especificação

Fonte: O Autor (2017).

A Tabela 18 apresenta como deverá ser feito o cadastro de administrador.

Tabela 18: Requisito cadastro de administrador.

Identificação		**F18**	
Título		Cadastro de administrador	
Descrição		O sistema deverá permitir cadastrar um usuário administrador.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF18.1	Dados obrigatórios	O cadastro deve conter nome, *e-mail*, senha e confirmação de	Especificação

		senha.	
NF18.2	Validar senha	O sistema deve realizar a validação da senha.	Especificação

Fonte: O Autor (2017).

A Tabela 19 apresenta como deverá ser feito a listagem de administrador.

Tabela 19: Requisito listagem de administrador.

Identificação		**F19**	
Título		Listagem de administrador	
Descrição		O sistema deve permitir a listagem dos usuários.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF19.1	Ausência de registros	Em caso de ausência de registro exibir mensagem avisando o usuário.	Especificação
NF19.2	Botões de editar e excluir	Na listagem deve aparecer os botões para editar o cadastro e eliminar o mesmo.	Especificação
NF19.3	Paginação	Para esta listagem deverá haver paginação, com 30 registros por página.	Especificação

Fonte: O Autor (2017).

A Tabela 20 apresenta como deverá ser feito a alteração e exclusão de administrador.

Tabela 20: Requisito alteração e exclusão de administrador.

Identificação		**F20**	
Título		Alteração e exclusão de administrador	
Descrição		O sistema deve permitir alterar e excluir um administrador.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria

NF20.1	Aviso de confirmação	Exibir aviso confirmando uma edição ou exclusão.	Especificação
NF20.2	Exclusão	O usuário não poderá excluir ele mesmo.	Especificação

Fonte: O Autor (2017).

A Tabela 21 apresenta como deverá ser feito o cadastro de aluno.

Tabela 21: Requisito cadastro de aluno.

Identificação		**F21**	
Título		Cadastro de aluno	
Descrição		O sistema deverá permitir cadastrar usuários do tipo aluno.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF21.1	Dados obrigatórios	O cadastro deve conter nome, curso, código de aluno da Unifebe, *e-mail*, senha e confirmação de senha.	Especificação
NF21.2	Validar senha	O sistema deve realizar a validação da senha.	Especificação
NF21.3	Expiração de Aluno	O sistema deverá expirar os alunos do semestre anterior quando for cadastrado um novo ano com seu respectivo semestre.	Especificação

Fonte: O Autor (2017).

A Tabela 22 apresenta como deverá ser feito a listagem de aluno.

Tabela 22: Requisito listagem de aluno.

Identificação	**F22**
Título	Listagem de aluno
Descrição	O sistema deve permitir a listagem de alunos.
Oculto	Não

Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF22.1	Ausência de registros	Em caso de ausência de registro exibir mensagem avisando o usuário.	Especificação
NF22.2	Botões de editar e ativar/inativar	Na listagem deve aparecer os botões para editar o cadastro e ativar/inativar o mesmo.	Especificação
NF22.3	Paginação	Para esta listagem deverá haver paginação, com 30 registros por página.	Especificação

Fonte: O Autor (2017).

A Tabela 23 apresenta como deverá ser feito a alteração, ativação e inativação de aluno.

Tabela 23: Requisito alteração, ativação e inativação de aluno.

Identificação		**F23**	
Título		Alteração e ativação/inativação de aluno	
Descrição		O sistema deve permitir alterar e ativar/inativar um aluno.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF23.1	Aviso de confirmação	Exibir aviso confirmando uma edição ou ativação/inativação.	Especificação

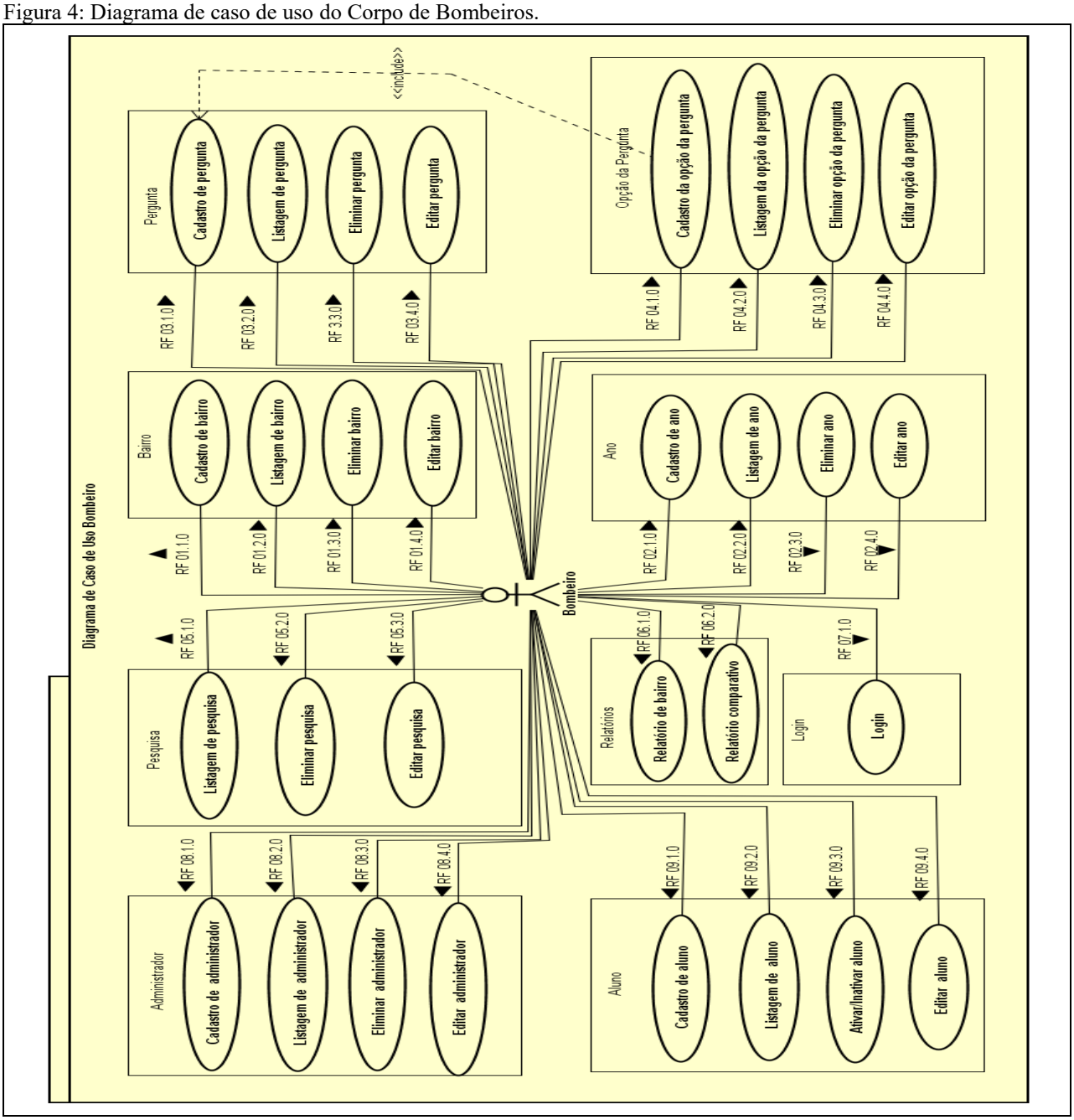
Fonte: O Autor (2017).

3.1.1.2 Diagrama de Caso de uso do Módulo do Corpo de Bombeiros

Caso de uso proporciona uma visão abrangente das funcionalidades que o sistema contemplará, deixando-o mais claro para o analista e o cliente.

Na Figura 4 é apresentado os casos de uso do sistema em forma de diagrama, o qual auxilia no entendimento dos mesmos. Todo caso de uso é identificado por um código composto por três casas. Após a Figura 4, são apresentadas as tabelas com descrições dos casos de uso, cada uma correspondendo a um código e caso de uso no diagrama da Figura 4.

Figura 4: Diagrama de caso de uso do Corpo de Bombeiros.



Fonte: O Autor (2017).

Tabela 24 apresenta como é feito o cadastro de bairro.

Tabela 24: Cadastro de bairro.

Identificação	01.1.0
Título	Cadastro de bairro
Descrição do caso de uso	Permitir o cadastro de um bairro.
Atores e casos de uso	Administrador
Requisitos correlacionados	F1
Fluxo básico	
Descrição	O usuário administrador, ao cadastrar um novo bairro, deverá informar o nome e a URL do google *maps* do mesmo.
Regras de negócio	• Todos os campos de preenchimento obrigatório;
Pré-condições	• Usuário autenticado.
Pós condições	• Emitir mensagem confirmando o cadastro. • Redirecionar para o caso de uso 01.2.0
Fluxo alternativo	
• Não se aplica	
Fluxo de exceção	
• Caso um campo esteja em branco, o usuário deverá ser alertado.	

Fonte: O Autor (2017).

A Tabela 25 apresenta como é feito a listagem de bairro.

Tabela 25: Listagem de bairro.

Identificação	01.2.0
Título	Listagem de bairro
Descrição do caso de uso	Exibe todos os bairros cadastrados no sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F2
Fluxo básico	
Descrição	O usuário administrador, poderá visualizar os bairros cadastrados no sistema.
Regras de negócio	• Ao listar, deverá existir a opção de editar e eliminar o mesmo; • Deverá haver paginação para a listagem de bairro, limitando em 30 por página;
Pré-condições	• Usuário autenticado;
Pós condições	• Não se aplica

Fluxo alternativo
• Não se aplica
Fluxo de exceção
• Caso não possua dados registrados, exibir a mensagem “Não tem dados para exibir! ”.

Fonte: O Autor (2017).

A Tabela 26 apresenta como é feito a exclusão de um bairro.

Tabela 26: Eliminar bairro.

Identificação	01.3.0
Título	Eliminar bairro
Descrição do caso de uso	Permite remover um bairro do sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F3
Fluxo básico	
Descrição	O usuário administrador, deverá selecionar um bairro que esteja cadastrado no sistema, e escolher a opção eliminar.
Regras de negócio	• Não se aplica.
Pré-condições	• Usuário autenticado;
Pós condições	• Eliminar a linha da tabela de listagem. • Emitir uma mensagem confirmando a exclusão do bairro.
Fluxo alternativo	
• Não se aplica.	
Fluxo de exceção	
• Não se aplica.	

Fonte: O Autor (2017).

A Tabela 27 apresenta como é feito a alteração no cadastro de bairro.

Tabela 27: Editar bairro.

Identificação	01.4.0
Título	Editar bairro
Descrição do caso de uso	Permite ao usuário administrador, editar informações dos bairros cadastrados no sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F3
Fluxo básico	
Descrição	O usuário administrador, deverá selecionar um bairro que esteja

	cadastrado no sistema.
Regras de negócio	• Todos os campos de preenchimento obrigatório.
Pré-condições	• Usuário autenticado.
Pós condições	• Exibir a mensagem "Alteração concluída" ao usuário, quando concluir a alteração.
Fluxo alternativo	
• Poderá cancelar a operação clicando no botão cancelar.	
Fluxo de exceção	
• Caso um campo esteja em branco, o usuário deverá ser alertado.	

Fonte: O Autor (2017).

A Tabela 28 apresenta como é feito o cadastro de ano.

Tabela 28: Cadastro de ano.

Identificação	02.1.0
Título	Cadastro de ano
Descrição do caso de uso	Permitir o cadastro de um ano.
Atores e casos de uso	Administrador
Requisitos correlacionados	F4
Fluxo básico	
Descrição	O usuário administrador, ao cadastrar um ano, deverá informar o ano precedido de um ponto e adicionar o número 1 ou 2 representando o semestre.
Regras de negócio	• Todos os campos de preenchimento obrigatório;
Pré-condições	• Usuário autenticado.
Pós condições	• Emitir mensagem confirmando o cadastro. • Redirecionar para o caso de uso 02.2.0
Fluxo alternativo	
Não se aplica	
Fluxo de exceção	
• Caso um campo esteja em branco, o usuário deverá ser alertado.	

Fonte: O Autor (2017).

A Tabela 29 apresenta como é feito a listagem de ano.

Tabela 29: Listagem de ano.

Identificação	02.2.0
Título	Listagem de ano
Descrição do caso de uso	Exibe todos os anos cadastrados no

	sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F5
Fluxo básico	
Descrição	O usuário administrador, poderá visualizar os anos cadastrados no sistema.
Regras de negócio	• Ao listar, deverá existir a opção de editar e eliminar o mesmo; • Deverá haver paginação para a listagem de ano, limitando em 30 por página;
Pré-condições	• Usuário autenticado;
Pós condições	• Não se aplica
Fluxo alternativo	
Não se aplica	
Fluxo de exceção	
• Caso não possua dados registrados, exibir a mensagem “Não tem dados para exibir! ”.	

Fonte: O Autor (2017).

A Tabela 30 apresenta como é feito a exclusão de um ano.

Tabela 30: Eliminar ano.

Identificação	02.3.0
Título	Eliminar ano
Descrição do caso de uso	Permite remover um ano do sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F6
Fluxo básico	
Descrição	O usuário administrador, deverá selecionar um ano que esteja cadastrado no sistema, e escolher a opção eliminar.
Regras de negócio	• Não se aplica.
Pré-condições	• Usuário autenticado;
Pós condições	• Eliminar a linha da tabela de listagem. • Emitir uma mensagem confirmando a exclusão do ano.
Fluxo alternativo	
• Não se aplica.	
Fluxo de exceção	
• Não se aplica.	

Fonte: O Autor (2017).

A Tabela 31 apresenta como é feito a alteração no cadastro de bairro.

Tabela 31: Editar ano.

Identificação	02.4.0
Título	Editar ano
Descrição do caso de uso	Permite ao usuário administrador, editar informações dos anos cadastrados no sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F6
Fluxo básico	
Descrição	O usuário administrador, deverá selecionar um ano que esteja cadastrado no sistema.
Regras de negócio	• Todos os campos de preenchimento obrigatório.
Pré-condições	• Usuário autenticado.
Pós condições	• Exibir a mensagem "Alteração concluída" ao usuário, quando concluir a alteração.
Fluxo alternativo	
• Poderá cancelar a operação clicando no botão cancelar.	
Fluxo de exceção	
• Caso um campo esteja em branco, o usuário deverá ser alertado.	

Fonte: O Autor (2017).

A Tabela 32 apresenta como é feito o cadastro de pergunta.

Tabela 32: Cadastro de pergunta.

Identificação	03.1.0
Título	Cadastro de pergunta
Descrição do caso de uso	Permitir o cadastro de uma pergunta.
Atores e casos de uso	Administrador
Requisitos correlacionados	F7
Fluxo básico	
Descrição	O usuário administrador, ao cadastrar uma pergunta, deverá descrever a pergunta desejada.
Regras de negócio	• Todos os campos de preenchimento obrigatório;
Pré-condições	• Usuário autenticado.
Pós condições	• Emitir mensagem confirmando o cadastro. • Redirecionar para o caso de uso 03.2.0
Fluxo alternativo	

• Não se aplica.
Fluxo de exceção
• Caso um campo esteja em branco, o usuário deverá ser alertado.

Fonte: O Autor (2017).

A Tabela 33 apresenta como é feito a listagem de pergunta.

Tabela 33: Listagem de pergunta.

Identificação	03.2.0
Título	Listagem de pergunta
Descrição do caso de uso	Exibe todas as perguntas cadastradas no sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F8
Fluxo básico	
Descrição	O usuário administrador, poderá visualizar as perguntas cadastradas no sistema.
Regras de negócio	• Ao listar, deverá existir a opção de editar e eliminar o mesmo; • Deverá haver paginação para a listagem de ano, limitando em 30 por página;
Pré-condições	• Usuário autenticado;
Pós condições	Não se aplica
Fluxo alternativo	
• Não se aplica.	
Fluxo de exceção	
• Caso não possua dados registrados, exibir a mensagem “Não tem dados para exibir! ”.	

Fonte: O Autor (2017).

A Tabela 34 apresenta como é feito a exclusão de uma pergunta.

Tabela 34: Eliminar pergunta.

Identificação	03.3.0
Título	Eliminar pergunta
Descrição do caso de uso	Permite remover uma pergunta do sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F9
Fluxo básico	
Descrição	O usuário administrador, deverá selecionar uma pergunta que esteja cadastrado no sistema, e escolher a opção eliminar.
Regras de negócio	• Não se aplica.

Pré-condições	• Usuário autenticado;
Pós condições	• Eliminar a linha da tabela de listagem. • Emitir uma mensagem confirmando a exclusão do ano.
Fluxo alternativo	
• Não se aplica.	
Fluxo de exceção	
• Não se aplica.	

Fonte: O Autor (2017).

A Tabela 35 apresenta como é feito a alteração no cadastro de pergunta.

Tabela 35: Editar pergunta.

Identificação	03.4.0
Título	Editar pergunta
Descrição do caso de uso	Permite ao usuário administrador, editar informações das perguntas cadastradas no sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F9
Fluxo básico	
Descrição	O usuário administrador, deverá selecionar uma pergunta que esteja cadastrada no sistema.
Regras de negócio	• Todos os campos de preenchimento obrigatório.
Pré-condições	• Usuário autenticado.
Pós condições	• Exibir a mensagem “Alteração concluída” ao usuário, quando concluir a alteração.
Fluxo alternativo	
• Poderá cancelar a operação clicando no botão cancelar.	
Fluxo de exceção	
• Caso um campo esteja em branco, o usuário deverá ser alertado.	

Fonte: O Autor (2017).

A Tabela 36 apresenta como é feito o cadastro da opção da pergunta.

Tabela 36: Cadastro da opção da pergunta.

Identificação	04.1.0
Título	Cadastro da opção da pergunta
Descrição do caso de uso	Permitir o cadastro da opção da pergunta.
Atores e casos de uso	Administrador
Requisitos correlacionados	F10

Fluxo básico	
Descrição	O usuário administrador, ao clicar em adicionar uma opção da pergunta, deverá selecionar uma pergunta que deseja adicionar as opções , e em seguida clicar no botão adicionar opção.
Regras de negócio	• Todos os campos de preenchimento obrigatório;
Pré-condições	• Usuário autenticado.
Pós condições	• Emitir mensagem confirmando o cadastro. • Redirecionar para o caso de uso 04.2.0
Fluxo alternativo	
• Não se aplica.	
Fluxo de exceção	
• Caso um campo esteja em branco, o usuário deverá ser alertado.	

Fonte: O Autor (2017).

A Tabela 37 apresenta como é feito a listagem da opção da pergunta.

Tabela 37: Listagem da opção pergunta.

Identificação	04.2.0
Título	Listagem da opção da pergunta
Descrição do caso de uso	Exibe todas as perguntas cadastradas no sistema que já tenham pelo menos uma opção da pergunta cadastrada.
Atores e casos de uso	Administrador
Requisitos correlacionados	F11
Fluxo básico	
Descrição	O usuário administrador, poderá visualizar as perguntas e as opções das perguntas cadastradas no sistema.
Regras de negócio	• Ao listar, deverá existir a opção de editar e eliminar o mesmo; • Deverá haver paginação para a listagem de ano, limitando em 30 por página;
Pré-condições	• Usuário autenticado;
Pós condições	• Não se aplica
Fluxo alternativo	
• Não se aplica.	
Fluxo de exceção	
• Caso não possua dados registrados, exibir a mensagem “Não tem dados para	

exibir! ”.

Fonte: O Autor (2017).

A Tabela 38 apresenta como é feito a exclusão de uma opção da pergunta.

Tabela 38: Eliminar opção da pergunta.

Identificação	04.3.0
Título	Eliminar opção da pergunta
Descrição do caso de uso	Permite remover uma opção da pergunta do sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F12
Fluxo básico	
Descrição	O usuário administrador, deverá selecionar uma pergunta que esteja cadastrado no sistema, e escolher uma opção da pergunta que deseja eliminar.
Regras de negócio	• Não se aplica.
Pré-condições	• Usuário autenticado;
Pós condições	• Eliminar a linha da tabela de listagem. • Emitir uma mensagem confirmando a exclusão do ano.
Fluxo alternativo	
• Não se aplica.	
Fluxo de exceção	
• Não se aplica.	

Fonte: O Autor (2017).

A Tabela 39 apresenta como é feito a alteração no cadastro da opção da pergunta.

Tabela 39: Editar opção da pergunta.

Identificação	04.4.0
Título	Editar opção da pergunta
Descrição do caso de uso	Permite ao usuário administrador, editar informações das opções das perguntas cadastradas no sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F12
Fluxo básico	
Descrição	O usuário administrador, deverá selecionar uma pergunta que esteja cadastrada no sistema e editar a opção da pergunta da mesma.
Regras de negócio	• Todos os campos de

	preenchimento obrigatório.
Pré-condições	• Usuário autenticado.
Pós condições	• Exibir a mensagem "Alteração concluída" ao usuário, quando concluir a alteração.
Fluxo alternativo	
• Poderá cancelar a operação clicando no botão cancelar.	
Fluxo de exceção	
• Caso um campo esteja em branco, o usuário deverá ser alertado.	

Fonte: O Autor (2017).

A Tabela 40 apresenta como é feito a listagem de pesquisa.

Tabela 40: Listagem de pesquisa.

Identificação	05.1.0
Título	Listagem de pesquisa.
Descrição do caso de uso	Exibe as pesquisas cadastradas no sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F13
Fluxo básico	
Descrição	O usuário administrador, poderá visualizar as pesquisas cadastradas.
Regras de negócio	• Ao listar, deverá existir a opção de editar e eliminar o mesmo; • Na listagem de cada pesquisa deverá ter código, bairro, ano, data e usuário. • Deverá haver paginação para a listagem de ano, limitando em 30 por página;
Pré-condições	• Usuário autenticado;
Pós condições	• Não se aplica
Fluxo alternativo	
• Não se aplica	
Fluxo de exceção	
• Caso não possua dados registrados, exibir a mensagem "Não tem dados para exibir! ".	

Fonte: O Autor (2017).

A Tabela 41 apresenta como é feito a exclusão de uma pesquisa.

Tabela 41: Eliminar pesquisa.

Identificação	05.2.0
Título	Eliminar pesquisa
Descrição do caso de uso	Permite remover uma pesquisa do

	sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F14
Fluxo básico	
Descrição	O usuário administrador, deverá selecionar uma pesquisa que esteja cadastrada no sistema, e escolher a opção eliminar.
Regras de negócio	• Não se aplica.
Pré-condições	• Usuário autenticado;
Pós condições	• Eliminar a linha da tabela de listagem. • Emitir uma mensagem confirmando a exclusão do ano.
Fluxo alternativo	
• Não se aplica.	
Fluxo de exceção	
• Não se aplica.	

Fonte: O Autor (2017).

A Tabela 42 apresenta como é feito a alteração de uma pesquisa.

Tabela 42: Editar pesquisa.

Identificação	05.3.0
Título	Editar pesquisa
Descrição do caso de uso	Permite ao usuário administrador, editar informações da pesquisa cadastrada no sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F14
Fluxo básico	
Descrição	O usuário administrador, deverá selecionar uma pesquisa que esteja cadastrada no sistema.
Regras de negócio	• Todos os campos de preenchimento obrigatório.
Pré-condições	• Usuário autenticado.
Pós condições	• Exibir a mensagem “Alteração concluída” ao usuário, quando concluir a alteração.
Fluxo alternativo	
• Poderá cancelar a operação clicando no botão cancelar.	
Fluxo de exceção	

• Caso um campo esteja em branco, o usuário deverá ser alertado.

Fonte: O Autor (2017).

A Tabela 43 apresenta como é feito o relatório de bairro.

Tabela 43: Relatório de bairro.

Identificação	06.1.0
Título	Relatório de Bairro
Descrição do caso de uso	Permite ao usuário administrador, realizar um relatório de um determinado bairro que foi pesquisado.
Atores e casos de uso	Administrador
Requisitos correlacionados	F15
Fluxo básico	
Descrição	O usuário administrador, deverá selecionar o bairro e ano.
Regras de negócio	• Todos os campos de preenchimento obrigatório.
Pré-condições	• Usuário autenticado.
Pós condições	• Exibir o relatório ao usuário, quando clicar em 'Gerar Relatório'.
Fluxo alternativo	
• Poderá cancelar a operação clicando no botão cancelar.	
Fluxo de exceção	
• Caso um campo esteja em branco, o usuário deverá ser alertado.	

Fonte: O Autor (2017).

A Tabela 44 apresenta como é feito o relatório de comparativo.

Tabela 44: Relatório comparativo.

Identificação	06.2.0
Título	Relatório de comparativo
Descrição do caso de uso	Permite ao usuário administrador, selecionar uma pergunta e comparar a quantidade de respostas com todos os bairros.
Atores e casos de uso	Administrador
Requisitos correlacionados	F16
Fluxo básico	
Descrição	O usuário administrador, deverá selecionar uma pergunta.
Regras de negócio	• Todos os campos de preenchimento obrigatório.
Pré-condições	• Usuário autenticado.
Pós condições	• Exibir o relatório ao usuário, quando clicar em 'Gerar

	Relatório’.
Fluxo alternativo	
• Poderá cancelar a operação clicando no botão cancelar.	
Fluxo de exceção	
• Caso um campo esteja em branco, o usuário deverá ser alertado.	

Fonte: O Autor (2017).

A Tabela 45 apresenta como é realizado o *login* no sistema.

Tabela 45: Login.

Identificação	07.1.0
Título	*Login*
Descrição do caso de uso	Permiti aos usuários administradores acessarem funções administrativas.
Atores e casos de uso	Usuário administrador
Requisitos correlacionados	F17
Fluxo básico	
Descrição	Qualquer usuário que não esteja autenticado no sistema, será solicitado a realizar o *login*.
Regras de negócio	• Todos os campos serão de preenchimento obrigatório; • Exibir confirmação se o *login* foi realizado com sucesso ou não.
Pré-condições	• Não se aplica
Pós condições	• Usuário autenticado. • Redireciona o usuário se o *login* estiver correto.
Fluxo alternativo	
• Não se aplica	
Fluxo de exceção	
• Não se aplica.	

Fonte: O Autor (2017).

A Tabela 46 apresenta como é feito o cadastro de administrador.

Tabela 46: Cadastro de administrador.

Identificação	08.1.0
Título	Cadastro de administrador
Descrição do caso de uso	O cadastro de administrador permiti ao administrador criar outros usuários administradores para utilizar o sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F18
Fluxo básico	
Descrição	O usuário administrador ao realizar outro cadastro, deverá indicar um login, o qual

	será validado pelo sistema, afim de garantir que não haja duplicidade, informará uma senha e também um endereço de e-mail.
Regras de negócio	• A senha deverá conter no mínimo 6 dígitos; • Não poderá haver duplicidade de login; • O usuário não pode deletar a si mesmo;
Pré-condições	• Usuário autenticado.
Pós condições	• Ao finalizar o cadastro, caso for informado o e-mail do novo usuário, o mesmo receberá sua senha na caixa de entrada do e-mail. • Informa ao usuário que o cadastro foi realizado com sucesso;
Fluxo alternativo	
• Poderá cancelar a operação clicando no botão cancelar.	
Fluxo de exceção	
• Caso o e-mail seja inválido, exibirá um alerta, informando e-mail incorreto. • *Login* duplicado, deverá ser emitido alerta pelo sistema. • Caso a senha tenha menos de 6 dígitos, o sistema exibirá um alerta.	

Fonte: O Autor (2017).

A Tabela 47 apresenta como é feito a listagem de cadastro de administrador.

Tabela 47: Listagem de administrador.

Identificação	08.2.0
Título	Listagem de administrador
Descrição do caso de uso	Exibe todos os administradores cadastrados no sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F19
Fluxo básico	
Descrição	O usuário administrador, poderá visualizar todos os usuários cadastrados no sistema.
Regras de negócio	• Ao listar, para cada usuário, deverá existir a opção de editar e eliminar o mesmo; • Deverá haver paginação para a listagem de usuários, limitando em 30 por página.
Pré-condições	• Usuário autenticado.
Pós condições	• Listagem dos dados.
Fluxo alternativo	

Não se aplica
Fluxo de exceção
• Caso não possua dados registrados, exibir a mensagem “Não tem dados para exibir! ”.

Fonte: O Autor (2017).

A Tabela 48 apresenta como é feito a exclusão de administrador.

Tabela 48: Eliminar administrador.

Identificação	08.3.0
Título	Eliminar administrador
Descrição do caso de uso	Permiti remover um usuário administrador do sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F20
Fluxo básico	
Descrição	O usuário administrador, deverá selecionar um usuário que esteja cadastrado no sistema, e escolher a opção eliminar.
Regras de negócio	• Ao ser eliminar, o usuário deverá ser marcado com status 0 no banco de dados, o mesmo não poderá ter seu registro excluído da base de dados;
Pré-condições	• Usuário autenticado; • Exibir um aviso perguntando se tem certeza que deseja excluir o mesmo; • Usuário deverá estar previamente cadastrado.
Pós condições	• Eliminar a linha da tabela de listagem; • Exibir mensagem confirmando a exclusão.
Fluxo alternativo	
• Não se aplica	
Fluxo de exceção	
• Não se aplica.	

Fonte: O Autor (2017).

A Tabela 49 apresenta como é feito a alteração no cadastro de administrador.

Tabela 49: Editar administrador.

Identificação	08.4.0
Título	Editar administrador
Descrição do caso de uso	Permite ao usuário administrador, editar informações dos usuários do sistema.
Atores e casos de uso	Administrador

Requisitos correlacionados	F20
Fluxo básico	
Descrição	O usuário administrador, deverá selecionar um usuário que esteja cadastrado no sistema, poderá alterar as informações de login.
Regras de negócio	• A senha deverá conter no mínimo 6 dígitos; • Não poderá haver duplicidade de login; • O usuário não pode deletar a si mesmo.
Pré-condições	• Usuário autenticado. • Usuário deverá estar previamente cadastrado. • A senha não deverá ser impressa no campo, mas se for informado uma nova senha, a mesma deverá ser validada.
Pós condições	• Ao finalizar o cadastro, caso for informado o e-mail do novo usuário, o mesmo receberá sua senha na caixa de entrada do e-mail.
Fluxo alternativo	
• Poderá cancelar a operação clicando no botão cancelar.	
Fluxo de exceção	
• Caso o e-mail seja inválido, exibirá um alerta, informando e-mail incorreto; • Login duplicado, deverá ser emitido alerta pelo sistema; • Caso a senha tenha menos de 6 dígitos, o sistema exibirá um alerta.	

Fonte: O Autor (2017).

A Tabela 50 apresenta como é feito o cadastro de aluno.

Tabela 50: Cadastro de aluno.

Identificação	09.1.0
Título	Cadastro de aluno
Descrição do caso de uso	O cadastro de usuário aluno permiti ao administrador criar outros usuários do tipo aluno para utilizar o sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F21
Fluxo básico	
Descrição	O administrador ao realizar o cadastro de aluno, deverá indicar nome, curso, código de aluno da Unifebe, *e-mail*. O qual será validado pelo sistema, afim de garantir que não haja duplicidade, informará uma senha.

Regras de negócio	• A senha deverá conter no mínimo 6 dígitos; • Não poderá haver duplicidade de login; • O usuário não pode deletar a si mesmo;
Pré-condições	• Usuário autenticado.
Pós condições	• Ao finalizar o cadastro, caso for informado o e-mail do novo usuário, o mesmo receberá sua senha na caixa de entrada do e-mail. • Informa ao usuário que o cadastro foi realizado com sucesso;
Fluxo alternativo	
• Poderá cancelar a operação clicando no botão cancelar.	
Fluxo de exceção	
• Caso o e-mail seja inválido, exibirá um alerta, informando e-mail incorreto. • *Login* duplicado, deverá ser emitido alerta pelo sistema. • Caso a senha tenha menos de 6 dígitos, o sistema exibirá um alerta.	

Fonte: O Autor (2017).

A Tabela 51 apresenta como é feito a listagem de cadastros de aluno.

Tabela 51: Listagem de aluno.

Identificação	09.2.0
Título	Listagem de aluno
Descrição do caso de uso	Exibe todos os alunos cadastrados no sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F22
Fluxo básico	
Descrição	O administrador, poderá visualizar todos os alunos cadastrados no sistema.
Regras de negócio	• Ao listar, para cada usuário, deverá existir a opção de editar e eliminar o mesmo; • Deverá haver paginação para a listagem de usuários, limitando em 30 por página.
Pré-condições	• Usuário autenticado.
Pós condições	• Listagem dos dados.
Fluxo alternativo	
• Não se aplica	
Fluxo de exceção	
• Caso não possua dados registrados, exibir a mensagem “Não tem dados para exibir! ”.	

Fonte: O Autor (2017).

A Tabela 52 apresenta como é feito a ativação ou inativação de aluno.

Tabela 52: Ativar e inativar aluno.

Identificação	09.3.0
Título	Ativar /Inativar aluno
Descrição do caso de uso	Permiti ativar ou inativar um aluno do sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F23
Fluxo básico	
Descrição	O administrador, deverá selecionar um aluno que esteja cadastrado no sistema, e escolher a opção ativar/ inativar.
Regras de negócio	• Ao ser ativado, o usuário deverá ser marcado com status 1 no banco de dados. • Ao ser inativo, o usuário deverá ser marcado com status 2 no banco de dados;
Pré-condições	• Usuário autenticado; • Exibir um aviso perguntando se tem certeza que deseja ativar/inativar o mesmo; • Aluno deverá estar previamente cadastrado.
Pós condições	• Efetuar as alterações no banco de dados;
Fluxo alternativo	
• Não se aplica	
Fluxo de exceção	
• Não se aplica.	

Fonte: O Autor (2017).

A Tabela 53 apresenta como é feito a alteração no cadastro de aluno.

Tabela 53: Editar aluno.

Identificação	09.4.0
Título	Editar aluno
Descrição do caso de uso	Permite ao usuário administrador, editar informações dos usuários do sistema.
Atores e casos de uso	Administrador
Requisitos correlacionados	F23
Fluxo básico	
Descrição	O administrador, deverá selecionar um aluno que esteja cadastrado no sistema, o mesmo poderá alterar informações como código do aluno da Unifebe, nome, e-mail, curso, status e senha do aluno.

Regras de negócio	• A senha deverá conter no mínimo 6 dígitos; • Não poderá haver duplicidade de login;
Pré-condições	• Usuário autenticado. • Usuário deverá estar previamente cadastrado. • A senha não deverá ser impressa no campo, mas se for informado uma nova senha, a mesma deverá ser validada.
Pós condições	• Ao finalizar o cadastro, caso for informado o e-mail do novo usuário, o mesmo receberá sua senha na caixa de entrada do e-mail.
Fluxo alternativo	
• Poderá cancelar a operação clicando no botão cancelar.	
Fluxo de exceção	
• Caso o e-mail seja inválido, exibirá um alerta, informando e-mail incorreto; • Login duplicado, deverá ser emitido alerta pelo sistema; • Caso a senha tenha menos de 6 dígitos, o sistema exibirá um alerta.	

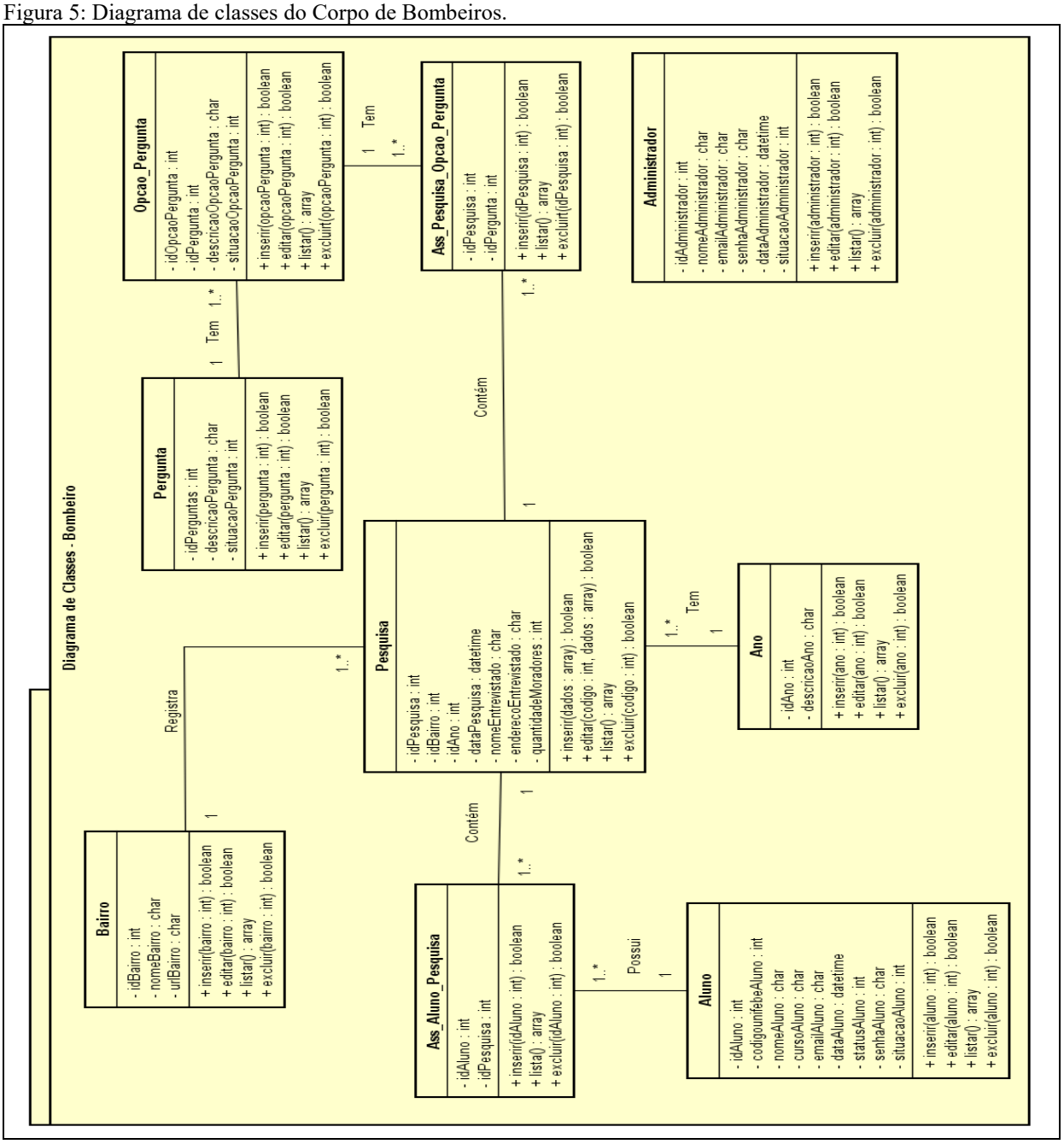
Fonte: O Autor (2017).

3.1.1.3 Diagrama de Classes do Módulo do Corpo de Bombeiros

"O diagrama de classes define a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos que cada classe tem, além de estabelecer como as classes se relacionam e trocam informações entre si" (GUEDES, 2009).

Na Figura 5 é apresentado o diagrama de classes com suas relações.

Figura 5: Diagrama de classes do Corpo de Bombeiros.



Fonte: O Autor (2017).

3.1.2 Módulo do Aluno

No decorrer será especificado o sistema *web* e o aplicativo *mobile* para o Modulo do Aluno, apresentando seus requisitos, casos de uso e digrama de classes.

3.1.2.1 Levantamento de requisitos do Módulo do Aluno

A Tabela 54 apresenta como deverá ser feito o cadastro de pesquisa.

Tabela 54: Requisito cadastro de pesquisa.

Identificação		**F1**	
Título		Cadastro de pesquisa	
Descrição		O sistema deve permitir o cadastro da pesquisa.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF1.1	Dados Obrigatórios	O cadastro deve conter o nome do bairro, o ano da pesquisa em formato de semestre, a data atual, nome do entrevistado, número da casa do entrevistado, quantidade de pessoas que moram na casa do entrevistado, nome da rua, descrição da pergunta e as opções de cada pergunta.	Especificação
NF1.2	Código da Pergunta	O sistema deve gerar um código automaticamente.	Especificação

Fonte: O Autor (2017).

Tabela 55 apresenta como deverá ser feito a listagem de pesquisa.

Tabela 55: Requisito listagem de pesquisa.

Identificação	**F2**
Título	Listagem de pesquisa
Descrição	O sistema deve permitir a listagem de pesquisa.
Oculto	Não

Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF2.1	Ausência de registros	Em caso de ausência de registro exibir mensagem avisando o usuário.	Especificação
NF2.2	Botão de visualizar	Na listagem deve aparecer o botão para visualizar a pesquisa.	Especificação
NF2.3	Paginação	Para esta listagem deverá haver paginação, com 30 registros por página.	Especificação

Fonte: O Autor (2017).

A Tabela 56 apresenta como deverá ocorrer o realizar *login* dos usuários no sistema.

Tabela 56: Requisito realizar login.

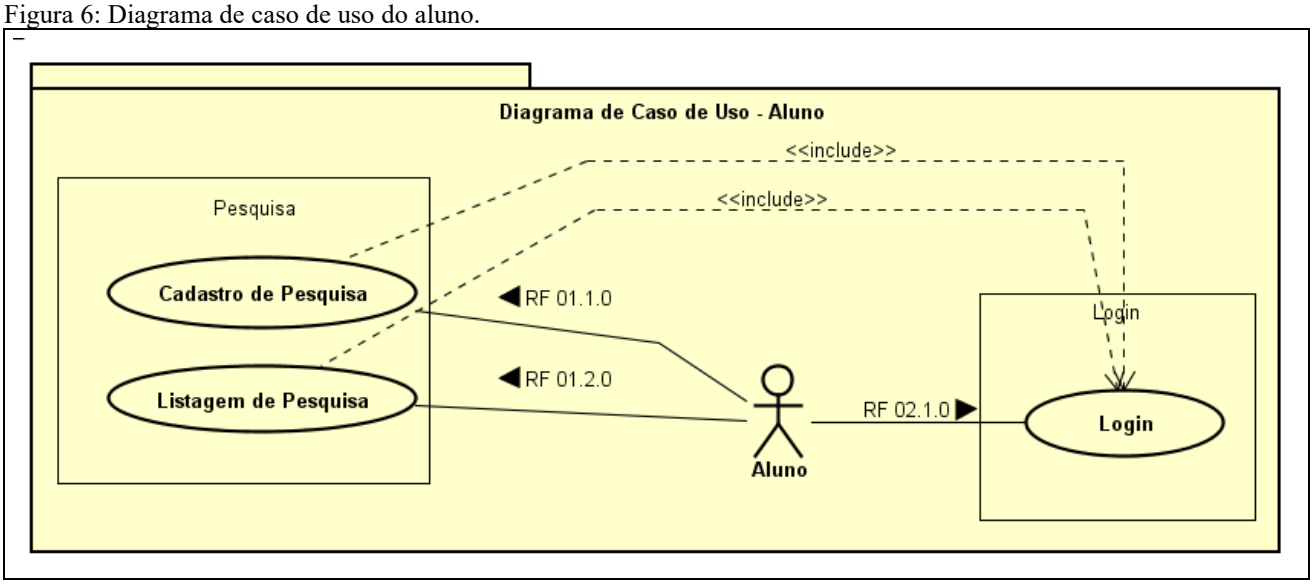
Identificação		**F3**	
Título		Realizar *login*	
Descrição		Será a função que dará acesso a funcionalidades do sistema para o aluno.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF3.1	Aviso de confirmação	Exibir mensagem se o *login* foi realizado com sucesso.	Especificação
NF3.2	Redirecionar	Redirecionar para a página inicial do sistema.	Especificação

Fonte: O Autor (2017).

3.1.2.2 Diagrama de Caso de uso do Módulo do Aluno

Na Figura 6 é apresentado os casos de uso do sistema *web* para o aluno em forma de diagrama, o qual auxilia no entendimento dos mesmos. Todo caso de uso é identificado por um código composto por três casas. Após a Figura 6, são apresentadas as tabelas com descrições dos casos de uso, cada uma correspondendo a um código e caso de uso no diagrama da Figura 6.

Figura 6: Diagrama de caso de uso do aluno.



Fonte: O Autor (2017).

A Tabela 57 apresenta como é feito o cadastro de pesquisa.

Tabela 57: Cadastro de pesquisa.

Identificação	01.1.0
Título	Cadastro de pesquisa
Descrição do caso de uso	Permitir o cadastro de uma pesquisa.
Atores e casos de uso	Aluno
Requisitos correlacionados	F1
Fluxo básico	
Descrição	O usuário aluno cadastra uma pesquisa no sistema.
Regras de negócio	• Todos os campos de preenchimento obrigatório;
Pré-condições	• Usuário autenticado.
Pós condições	• Emitir mensagem confirmando o cadastro. • Redirecionar para o caso de uso 01.2.0
Fluxo alternativo	
• Não se aplica	
Fluxo de exceção	

• Caso um campo esteja em branco, o usuário deverá ser alertado.

Fonte: O Autor (2017).

A Tabela 58 apresenta como é feito a listagem de pesquisa.

Tabela 58: Listagem de pesquisa.

Identificação	01.2.0
Título	Listagem de pesquisa.
Descrição do caso de uso	Exibe as pesquisas cadastradas no sistema.
Atores e casos de uso	Aluno
Requisitos correlacionados	F2
Fluxo básico	
Descrição	O usuário aluno, poderá visualizar as pesquisas cadastradas.
Regras de negócio	• Ao listar, deverá existir a opção de editar e eliminar o mesmo; • Na listagem de cada pesquisa deverá ter código, bairro, ano, data e usuário. • Deverá haver paginação para a listagem de ano, limitando em 30 por página;
Pré-condições	• Usuário autenticado;
Pós condições	• Não se aplica
Fluxo alternativo	
Não se aplica	
Fluxo de exceção	
Caso não possua dados registrados, exibir a mensagem “Não tem dados para exibir! ”.	

Fonte: O Autor (2017).

A Tabela 59 apresenta como é realizado o *login* no sistema.

Tabela 59: Login

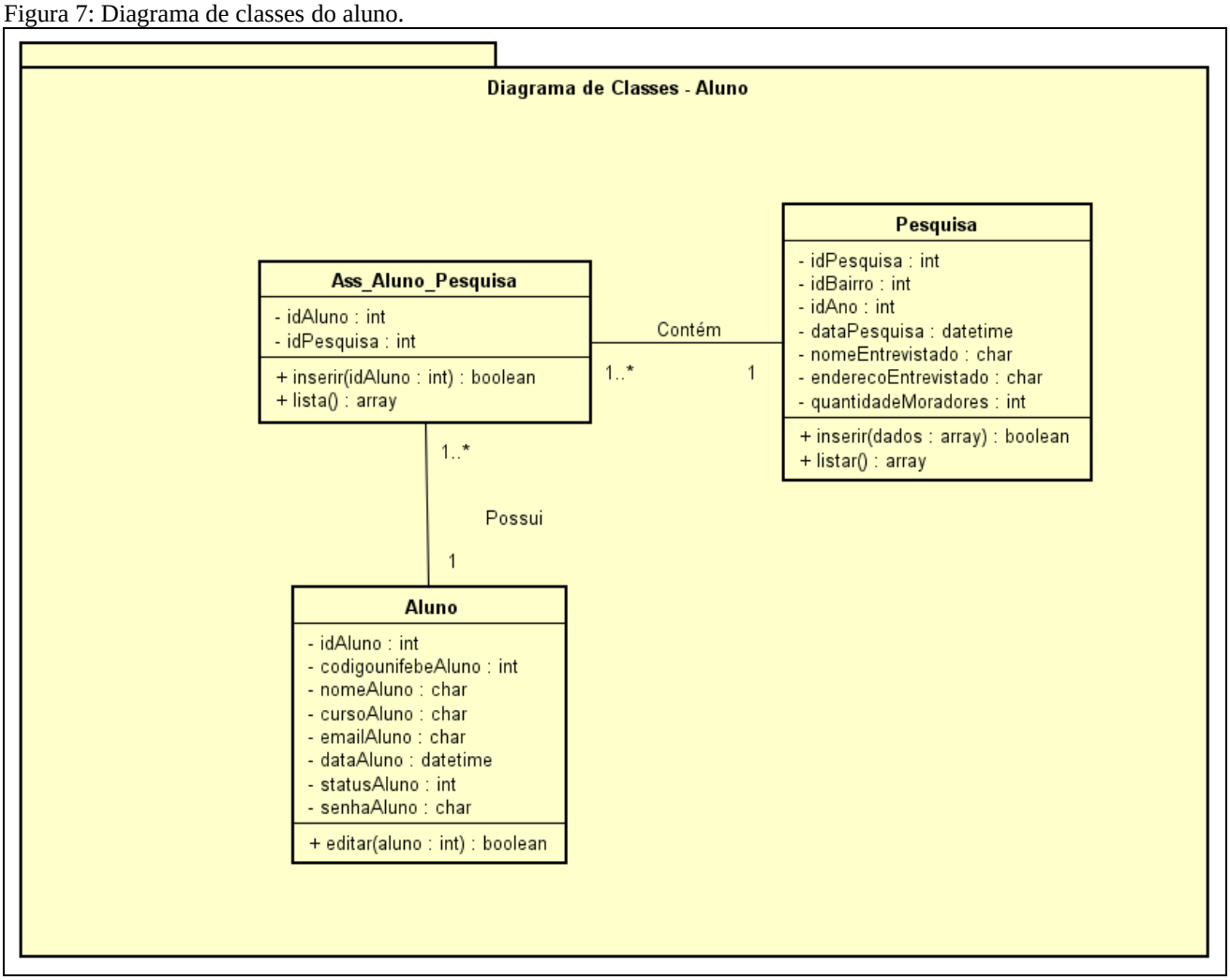
Identificação	02.1.0
Título	*Login*
Descrição do caso de uso	Permiti aos usuários alunos acessarem funções do sistema.
Atores e casos de uso	Usuário aluno
Requisitos correlacionados	F3
Fluxo básico	
Descrição	Qualquer usuário que não esteja autenticado no sistema, será solicitado a realizar o *login*.
Regras de negócio	• Todos os campos serão de preenchimento obrigatório; • Exibir confirmação se o *login* foi realizado com sucesso ou não.
Pré-condições	• Não se aplica
Pós condições	• Usuário autenticado. • Redireciona o usuário se o *login* estiver correto.
Fluxo alternativo	
• Não se aplica	
Fluxo de exceção	
• Não se aplica.	

Fonte: O Autor (2017).

3.1.2.3 Diagrama de classes do Módulo do Aluno

Na Figura 7 é apresentado o digrama de classes do aluno com suas relações.

Figura 7: Diagrama de classes do aluno.



Fonte: O Autor (2017).

3.1.3 Módulo da População

No decorrer será especificado o sistema *web* e aplicativo *mobile* para a população apresentando seus requisitos, casos de uso e digrama de classes.

3.1.3.1 Levantamento de requisitos do Módulo da População

A Tabela 60 apresenta como deverá ser exibido a tela principal do sistema.

Tabela 60: Requisito início.

Identificação		**F1**	
Título		Início	
Descrição		O Aplicativo deverá apresentar informações sobre o que é o programa Alerta Vermelho.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF1.1	Informações para apresentar.	Deverá apresentar uma descrição do programa Alerta Vermelho, como sua importância e utilidade para população.	Especificação

Fonte: O Autor (2017).

A Tabela 61 apresenta como deverá ser feito o relatório de bairro.

Tabela 61: Requisito relatório de bairro.

Identificação	**F2**	
Título	Relatório de bairro	
Descrição	O sistema deve permitir gerar relatório dos bairros pesquisados.	
Oculto	Não	
Requisitos Não Funcionais		
Nome	Restrição	Categoria

NF2.1	Relatório de Bairro	Relatório que exibe a quantidade de respostas de cada pergunta feita na pesquisa.	Especificação

Fonte: O Autor (2017).

A Tabela 62 apresenta como deverá ser feito o relatório comparativo.

Tabela 62: Requisito relatório comparativo.

Identificação		**F3**	
Título		Relatório comparativo	
Descrição		O sistema deve permitir gerar relatório comparativo.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF3.1	Relatório de Comparativo	Relatório compara a pergunta selecionada com outros bairros.	Especificação

Fonte: O Autor (2017).

A Tabela 63 apresentar como deverá funcionar a exibição dos telefones.

Tabela 63: Requisito telefones.

Identificação		**F4**	
Título		Telefones	
Descrição		O sistema deverá conter telefones de utilidade pública.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF4.1	Informações para apresentar.	Deverá apresentar telefones, logo do contato, endereço e site.	Especificação

Fonte: O Autor (2017).

A Tabela 64 apresentar como deverá funcionar a exibição das dicas de segurança.

Tabela 64: Requisito dicas de segurança.

Identificação		**F5**	
Título		Dicas de segurança	
Descrição		O sistema deverá incluir dicas de segurança para a população.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF5.1	Informações para apresentar.	Deverá apresentar discas de segurança.	Especificação

Fonte: O Autor (2017).

A Tabela 65 apresenta como deverá ser exibido o sobre do aplicativo.

Tabela 65: Requisito sobre.

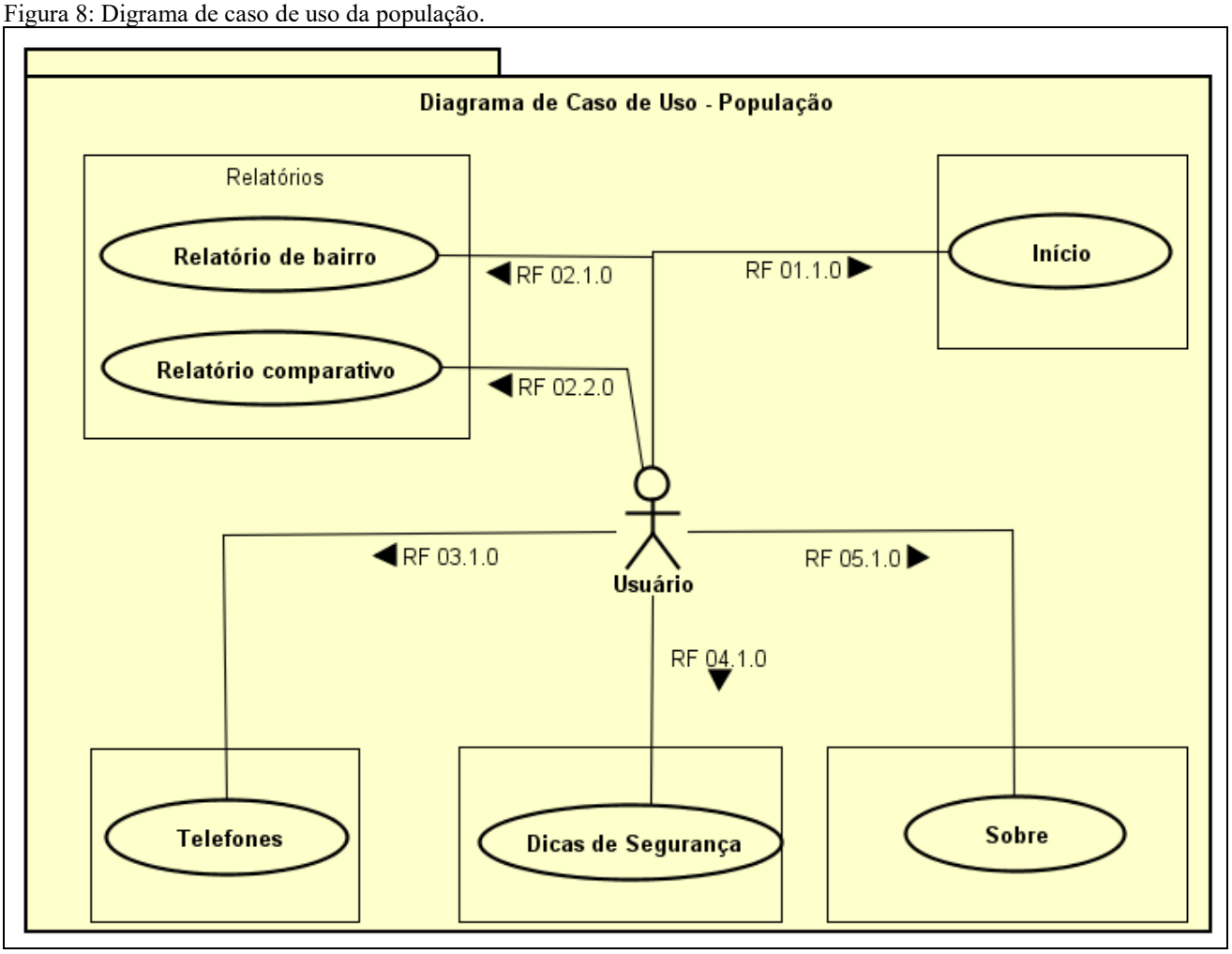
Identificação		**F6**	
Título		Sobre	
Descrição		O Aplicativo deverá apresentar informações sobre o autor.	
Oculto		Não	
Requisitos Não Funcionais			
Nome		Restrição	Categoria
NF6.1	Informações para apresentar.	Deverá apresentar os créditos do aplicativo.	Especificação

Fonte: O Autor (2017).

3.1.3.2 Diagrama de Caso de Uso do Módulo da População

Na Figura 8 é apresentado os casos de uso do sistema *web* e aplicativo móvel em forma de diagrama, o qual auxilia no entendimento dos mesmos. Todo caso de uso é identificado por um código composto por três casas. Após a Figura 8, são apresentadas as tabelas com descrições dos casos de uso, cada uma correspondendo a um código e caso de uso no diagrama da Figura 8.

Figura 8: Digrama de caso de uso da população.



Fonte: O Autor (2017).

A Tabela 66 apresenta como é feito o início.

Tabela 66: Início

Identificação	01.1.0
Título	Início
Descrição do caso de uso	Permiti acesso a página principal do *site*, onde está terá informações do programa Operação Alerta Vermelho.

Atores e casos de uso	Usuário
Requisitos correlacionados	F1
Fluxo básico	
Descrição	O usuário acessa a página início.
Regras de negócio	• Ter acesso à Internet.
Pré-condições	• Não se aplica.
Pós condições	• Exibir a página acessada.
Fluxo alternativo	
• Não se aplica	
Fluxo de exceção	
• Não se aplica	

Fonte: O Autor (2017).

A Tabela 67 apresenta como é feito o relatório de bairro.

Tabela 67: Relatório de bairro.

Identificação	02.1.0
Título	Relatório de bairro
Descrição do caso de uso	Permite ao usuário, gerar um relatório de um determinado bairro que foi pesquisado.
Atores e casos de uso	Usuário
Requisitos correlacionados	F2
Fluxo básico	
Descrição	O usuário, deverá selecionar o bairro e ano.
Regras de negócio	• Todos os campos de preenchimento obrigatório.
Pré-condições	• Não se aplica.
Pós condições	• Exibir o relatório ao usuário, quando clicar em ‘Gerar Relatório’.
Fluxo alternativo	

• Não se aplica
Fluxo de exceção
• Se o bairro selecionado ainda não foi pesquisado deverá exibir um alerta informando o usuário.

Fonte: O Autor (2017).

A Tabela 68 apresenta como é feito o relatório comparativo.

Tabela 68: Relatório comparativo.

Identificação	02.1.0
Título	Relatório comparativo
Descrição do caso de uso	Permite ao usuário, selecionar uma pergunta e comparar a quantidade de respostas com todos os bairros.
Atores e casos de uso	Usuário
Requisitos correlacionados	F3
Fluxo básico	
Descrição	O usuário, deverá selecionar uma pergunta.
Regras de negócio	• Todos os campos de preenchimento obrigatório.
Pré-condições	• Não se aplica.
Pós condições	• Exibir o relatório ao usuário, quando clicar em ‘Gerar Relatório’.
Fluxo alternativo	
• Não se aplica	
Fluxo de exceção	
• Não se aplica	

Fonte: O Autor (2017).

A Tabela 69 apresenta como é feito os telefones.

Tabela 69: Telefones

Identificação	03.1.0
Título	Telefones
Descrição do caso de uso	Lista todos os telefones úteis cadastrados.
Atores e casos de uso	Usuário
Requisitos correlacionados	F4
Fluxo básico	
Descrição	Ao clicar na opção “Telefones”, o sistema listará todos os telefones cadastrados.
Regras de negócio	• Lista os telefones; • Ao clicar em um telefone deverá exibi-lo.
Pré-condições	• Não se aplica.
Pós condições	• Lista todos os telefones para o usuário.
Fluxo alternativo	
• Não se aplica	
Fluxo de exceção	
• Não se aplica	

Fonte: O Autor (2017).

A Tabela 70 apresenta como é feito as dicas de segurança.

Tabela 70: Dicas de segurança

Identificação	04.1.0
Título	Dicas de Segurança
Descrição do caso de uso	Lista todas as dicas de segurança cadastradas.
Atores e casos de uso	Usuário
Requisitos correlacionados	F5
Fluxo básico	
Descrição	Ao clicar na opção “Dicas de

	Segurança”, o sistema listará todas as dicas de segurança cadastradas.
Regras de negócio	• Lista as dicas de segurança; • Ao clicar em um telefone deverá exibi-lo.
Pré-condições	• Não se aplica.
Pós condições	• Lista as dicas de segurança para o usuário.
Fluxo alternativo	
• Não se aplica	
Fluxo de exceção	
• Não se aplica	

Fonte: O Autor (2017).

A Tabela 71 apresenta como são exibidos os créditos do sistema.

Tabela 71: Sobre

Identificação	05.1.0
Título	Sobre
Descrição do caso de uso	Exibe os créditos do sistema.
Atores e casos de uso	Usuário
Requisitos correlacionados	F6
Fluxo básico	
Descrição	Ao clicar na opção “Sobre”, o sistema listará os créditos do sistema.
Regras de negócio	• Lista informação do autor do sistema. • Lista as instituições que apoiaram o projeto.
Pré-condições	• Não se aplica.
Pós condições	• Lista os créditos do sistema para o usuário.
Fluxo alternativo	
• Não se aplica	

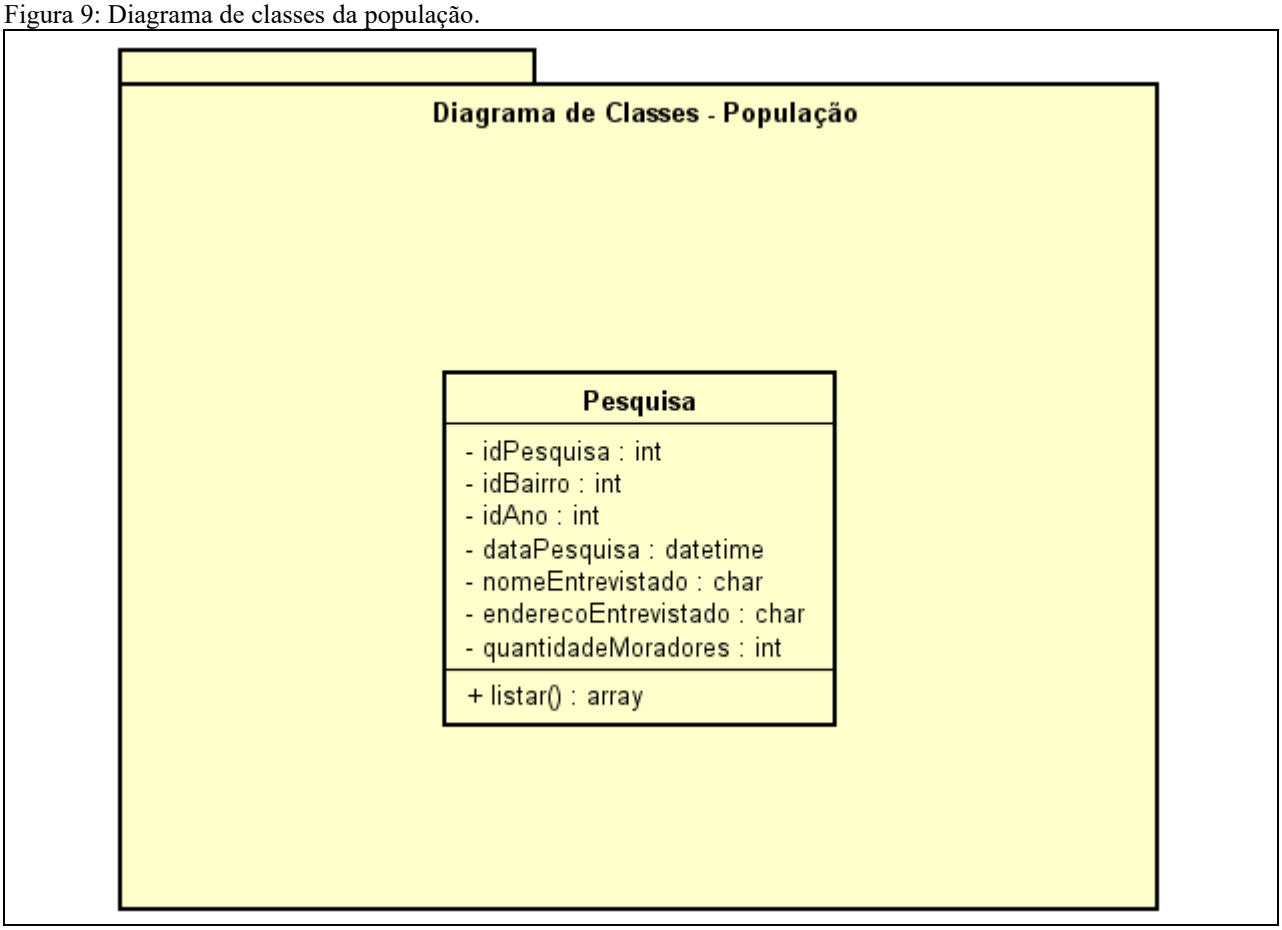
Fluxo de exceção
• Não se aplica

Fonte: O Autor (2017).

3.1.3.3 Diagrama de classes do Módulo da População

Na Figura 9 é apresentado o digrama de classes da população.

Figura 9: Diagrama de classes da população.

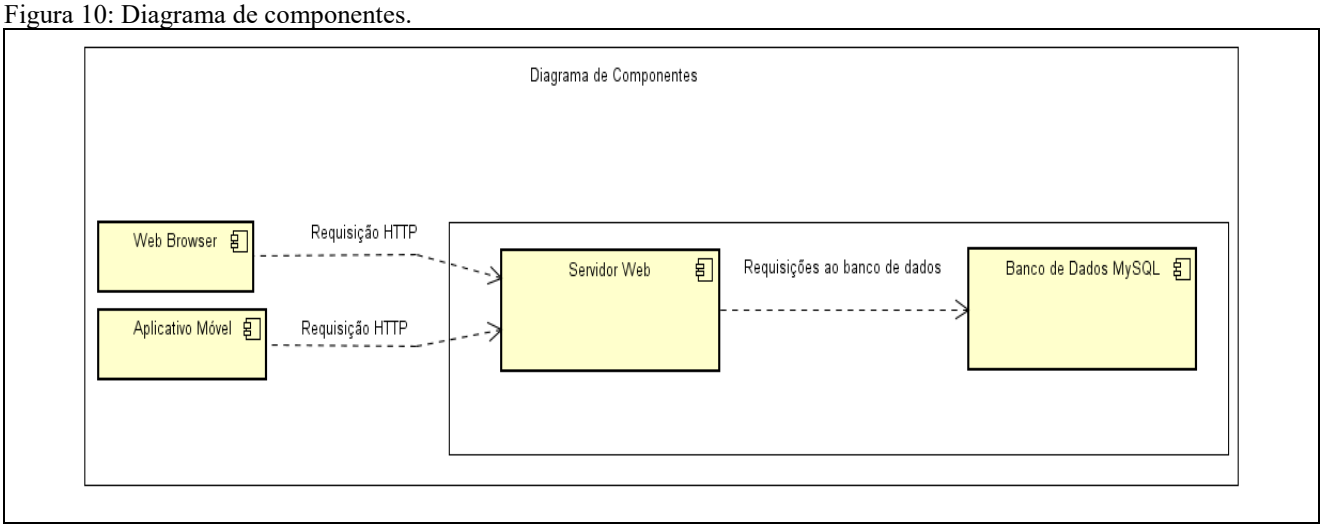


Fonte: O Autor (2017).

3.2 DIAGRAMA DE COMPONENTES

Na Figura 10 é apresentado o digrama de componentes do sistema *web* e do aplicativo móvel que fazem uma requisição para o servidor *web* e em seguida esse servidor *web* realiza uma requisição para o servidor de banco de dados MYSQL.

Figura 10: Diagrama de componentes.



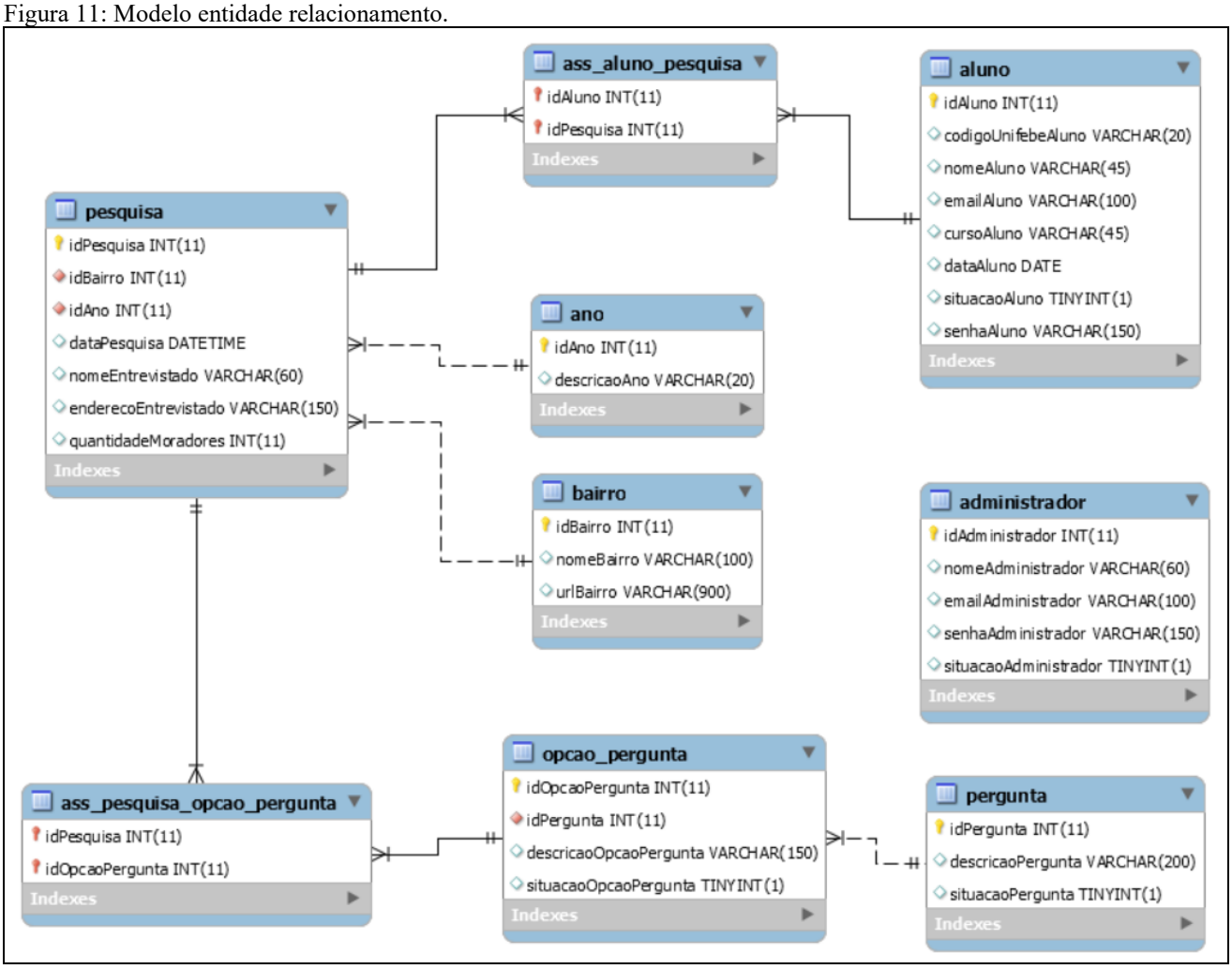
Fonte: O Autor (2017).

3.3 MODELO ENTIDADE RELACIONAMENTO

"O Modelo Entidade Relacionamento é um modelo conceitual utilizado na Engenharia de Software para descrever os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos)" (DEVMEDIA, 2017).

Na Figura 11 é representando o Modelo de Entidade Relacionamento.

Figura 11: Modelo entidade relacionamento.



Fonte: O Autor (2017).

4 DESENVOLVIMENTO

4.1 DESENVOLVIMENTO DO SISTEMA *WEB*

Para o desenvolvimento do sistema *web* foi utilizado a arquitetura MVC, já implementada pelo Zend *Framework*, versão 1.12.17. A versão do PHP utilizada durante o desenvolvimento, foi a 5.5.31, e o banco de dados escolhido foi o MySQL, na versão 5.7.12.

As interfaces de usuários, foram criadas com base em componentes baseados no *framework* Bootstrap, ou seja, o sistema poderá ser utilizado em diversos dispositivos. Para facilitar o desenvolvimento das telas e do *layout* foi utilizado o *template* administrativo SB Admin 2, disponibilizado de forma gratuita pela Start Bootstrap.

4.1.1 Estrutura do Sistema

Devido ao fato da aplicação ter sido concebida a partir de um *framework* de baixo acoplamento, ela foi dívida em alguns módulos que trabalham de forma independente, apenas necessitando das mesmas camadas de modelo. A aplicação ficou definida nos seguintes módulo:

- Módulo do Corpo de Bombeiros;
- Módulo do Aluno;
- Módulo da População.

4.1.1.1 Módulo do Corpo de Bombeiros

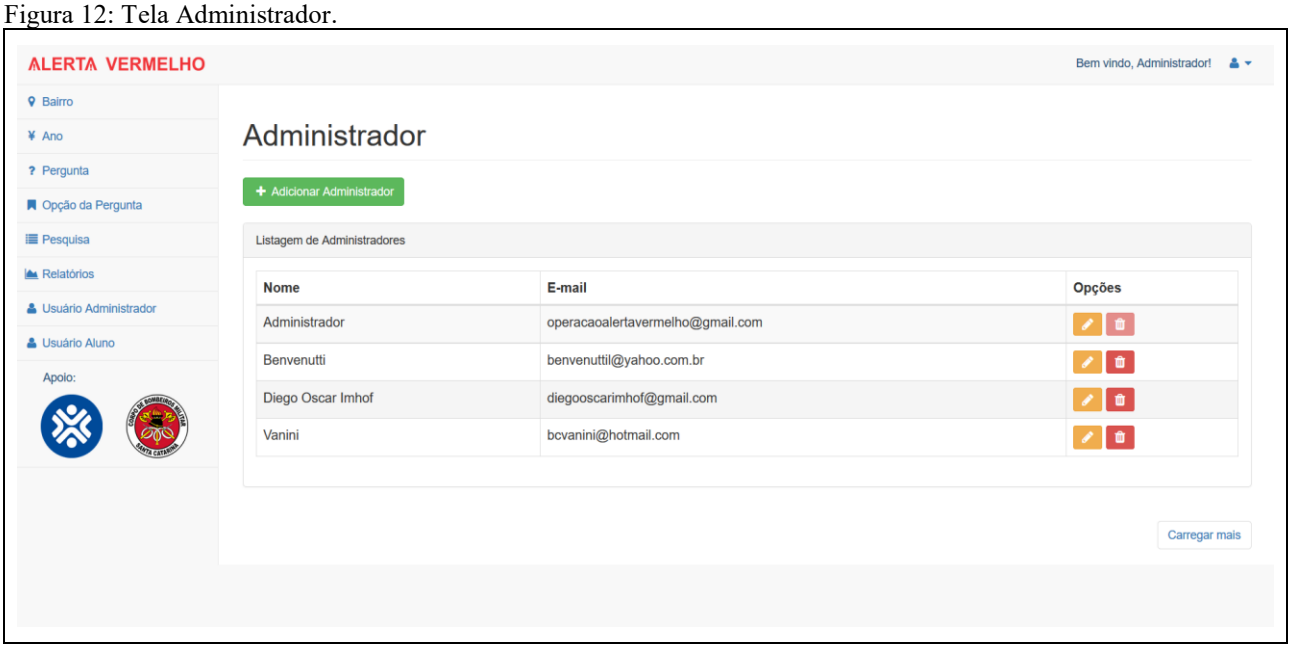
Módulo de acesso restrito somente para o Corpo de Bombeiros, é neste módulo que ocorre o gerenciamento do programa Operação Alerta Vermelho, por meio dele é possível incluir, editar e excluir itens do sistema, como também gerar relatórios. Neste módulo o usuário administrador terá acesso aos seguintes itens que fazem parte da pesquisa:

- Bairro;
- Ano;
- Pergunta;

- Opção da Pergunta;
- Pesquisa;
- Relatórios;
- Usuário Administrador;
- Usuário Aluno.

Na Figura 12 é possível visualizar a tela de administrador, como também o menu principal do sistema a esquerda.

Figura 12: Tela Administrador.



Fonte: O autor (2017).

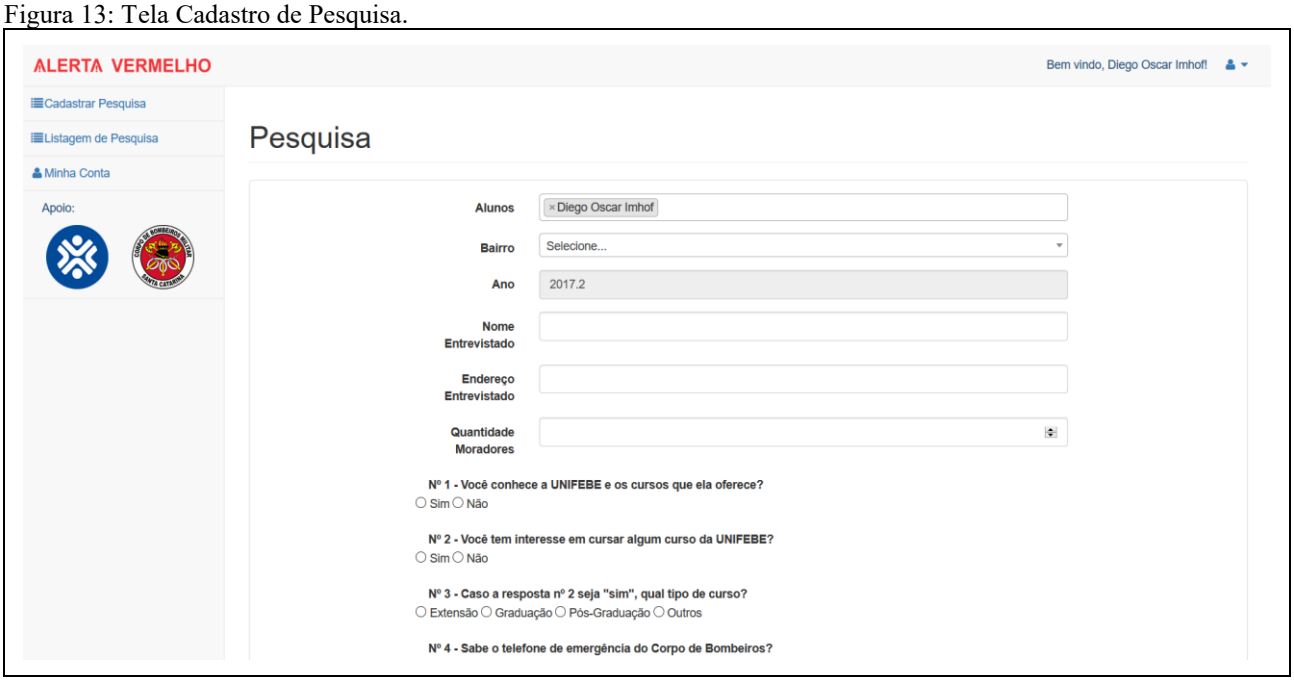
4.1.1.2 Módulo do Aluno

Módulo responsável por realizar as pesquisas. É neste módulo que o aluno pode realizar as pesquisas nos bairros junto à população. O aluno terá acesso aos seguintes itens:

- Cadastrar Pesquisa;
- Listagem de Pesquisa;
- Minha Conta.

Na Figura 13 é possível visualizar a tela cadastro de pesquisa, como também todo o menu do sistema a esquerda.

Figura 13: Tela Cadastro de Pesquisa.



Fonte: O autor (2017).

4.1.1.3 Módulo da População

Módulo de acesso ao público. Esse módulo está disponível para a comunidade em geral para que o mesmo tenha conhecimento sobre o programa Operação Alerta Vermelho. O usuário terá acesso aos seguintes itens:

- Início;
- Telefones;
- Dicas de Segurança;
- Relatórios;
- Sobre.

Na Figura 14 é possível visualizar a página inicial como também os outros menus do sistema.

Figura 14: Página Inicial.

Fonte: O autor (2017).

4.2 DESENVOLVIMENTO DO APLICATIVO MÓVEL

Para o desenvolvimento do aplicativo móvel, foi utilizado a IDE Android *Studio*, versão 2.2.1 para desenvolver os aplicativos na plataforma Android.

4.2.1 Estrutura do Sistema

O aplicativo móvel, foi implementado utilizando o componente *web view*, este componente permite incorporar todo o sistema *web* dentro do aplicativo móvel.

Foram desenvolvidos os seguintes aplicativos móveis em Android:

- Alerta Vermelho;
- Alerta Vermelho Aluno.

4.2.1.1 Alerta Vermelho

O aplicativo Alerta Vermelho é módulo da população que foi incorporado por meio do componente *web view* do Android *Studio*.

A seguir serão apresentando as telas do sistema Alerta Vermelho.

Na Figura 15, é apresentado o menu do aplicativo, abaixo será apresentado um breve resumo da função de cada item do menu do aplicativo:

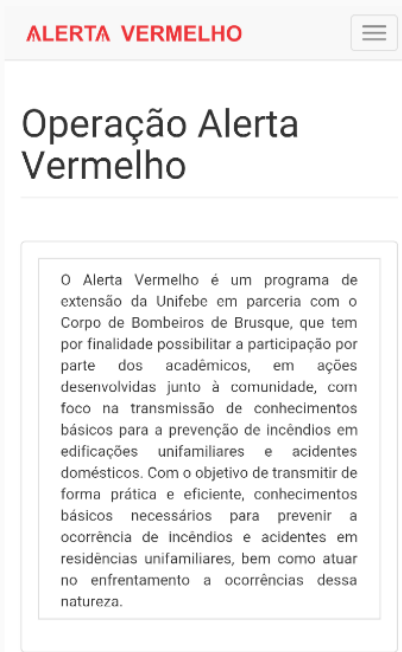
- Início: apresenta informações do programa Operação Alerta Vermelho;
- Telefones: para o usuário consultar telefones úteis.
- Dicas de Segurança: esta funcionalidade, permite que o usuário consulte dicas de segurança doméstica;
- Relatórios: permite ao usuário consultar o relatório de bairro e o relatório comparativo;
- Sobre: apresenta créditos do aplicativo.

Figura 15: Menu Alerta Vermelho.

Fonte: O autor (2017).

Nas Figuras 16, 17 e 18 é apresentado o início do aplicativo, a consulta de um telefone e a tela de dicas de segurança respectivamente.

Figura 16: Início do aplicativo.



Fonte: O Autor (2017).

Figura 17: Consulta de telefone.

Fonte: O Autor (2017).

Figura 18: Dicas de segurança.

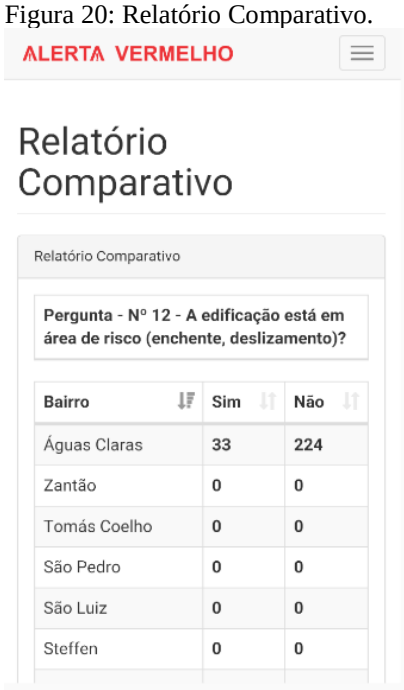
Fonte: O Autor (2017).

As figuras 19, 20, 21 apresentam a consulta do relatório de bairro, relatório comparativo e a tela sobre respectivamente.

Figura 19: Relatório de Bairro.

Fonte: O Autor (2017).

Figura 20: Relatório Comparativo.



Bairro	Sim	Não
Águas Claras	33	224
Zantão	0	0
Tomás Coelho	0	0
São Pedro	0	0
São Luiz	0	0
Steffen	0	0

Fonte: O Autor (2017).

Figura 21: Sobre.

Fonte: O Autor (2017).

4.2.1.2 Alerta Vermelho Aluno

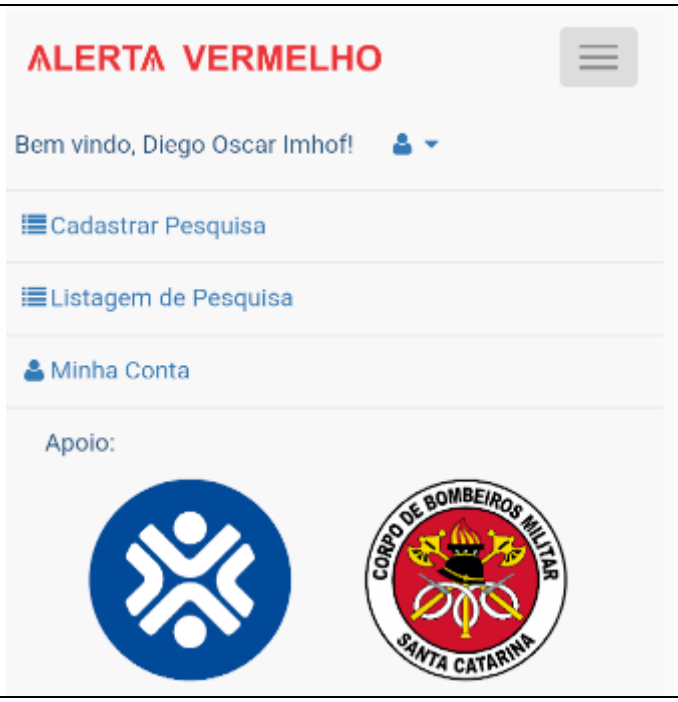
O aplicativo Alerta Vermelho Aluno é o módulo do aluno que foi incorporado por meio do componente *web view* do Android *Studio*.

A seguir serão apresentando as telas do sistema Alerta Vermelho Aluno.

Na figura 22, é apresentado o menu do aplicativo, abaixo será apresentado um breve resumo da função de cada item do menu do aplicativo:

- Cadastrar Pesquisa: permite ao aluno cadastrar uma pesquisa;
- Listagem de Pesquisa: permite ao aluno visualizar as pesquisas cadastradas.
- Minha Conta: permite ao aluno gerenciar sua conta.

Figura 22: Menu Alerta Vermelho Aluno.



Fonte: O autor (2017).

Nas figuras 23, 24 e 25 é apresentado o *login* no aplicativo, a tela de cadastro de pesquisa, tela de listagem de pesquisa e a tela minha conta respectivamente.

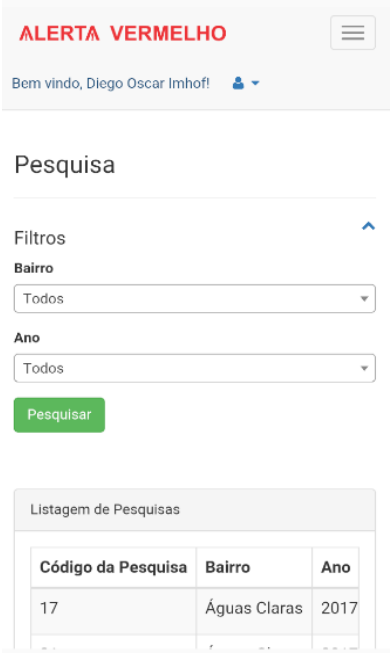
Figura 23: Login.

Fonte: O Autor (2017).

Figura 24: Cadastro de pesquisa.

Fonte: O Autor (2017).

Figura 25: Listagem de pesquisa.



Fonte: O Autor (2017).

A figura 26 apresenta a tela minha conta.

Figura 26: Minha conta.

Fonte: O Autor (2017).

4.3 PUBLICAÇÃO E DISPONIBILIZAÇÃO DO SISTEMA

O sistema *web* foi lançado oficialmente no dia 14 de setembro de 2017, o aplicativo móvel "Alerta Vermelho" foi lançado oficialmente no dia 21 de setembro de 2017 e o aplicativo "Alerta Vermelho Aluno" foi lançado dia 03 de outubro de 2017. Antes destas datas já haviam algumas versões destes *softwares* disponibilizadas ao público. O sistema "Alerta Vermelho" foi disponibilizado para o Corpo de Bombeiros de Brusque no seguinte link: <http://alertavermelho.unifebe.edu.br/admin>. O módulo do aluno está disponibilizado no seguinte link: <http://alertavermelho.unifebe.edu.br/aluno>. E módulo da população está disponível no seguinte link: <http://alertavermelho.unifebe.edu.br/>.

Para utilizar o sistema *web*, basta que o usuário acesse o link indicado a partir de qualquer navegador compatível com HTML 5, independente se o acesso for a partir de um *desktop* ou dispositivo móvel.

O aplicativo móvel, foi compilado e lançado para a plataforma Android sob o nome de "Alerta Vermelho" e "Alerta Vermelho Aluno". Os aplicativos estão disponíveis na Google Play, é preciso que o usuário disponha de um dispositivo com o Android 4.0 instalado ou superior. Abaixo segue os links para *download* dos aplicativos:

- Alerta Vermelho:
  - <https://play.google.com/store/apps/details?id=br.edu.unifebe.alertavermelho.alertavermelhopopulacao>;
  - URL Curto: <https://goo.gl/hn5df2>.
- Alerta Vermelho Aluno:
  - <https://play.google.com/store/apps/details?id=br.edu.unifebe.alertavermelho.alertavermelhoaluno>;
  - URL Curto: <https://goo.gl/icHm8h>.

4.3.1 Estatísticas de Acesso e Avaliação do Sistema

O sistema Alerta Vermelho foi bem recebido pelo Corpo de Bombeiros de Brusque e pelos acadêmicos bolsistas do programa Operação Alerta Vermelho. Conforme o depoimento do Corpo do Bombeiros (Apêndice A), os bombeiros demonstraram grande satisfação pela automatização do programa Operação Alerta Vermelho. E para os alunos pode-se constatar como *feedback* da maioria dos participantes que além de automatizar este processo o aplicativo demonstrou muito eficiente, prático e de fácil manuseio. O sistema foi utilizado durante o segundo semestre de 2017 pelos bombeiros e os acadêmicos, foram realizadas no total 257 pesquisas que foram realizadas no bairro Águas Claras.

5 CONSIDERAÇÕES FINAIS

Este trabalho de conclusão de curso, a partir de um amplo estudo na área de aplicações voltadas a alertas e apoio ao Corpo de Bombeiros vem apresentar como resultado final o desenvolvimento do sistema “Alerta Vermelho” nas plataformas *web* e *mobile*. O sistema foi divido em três módulo, um módulo para o Corpo de Bombeiros para realizar o gerenciamento do programa Operação Alerta Vermelho a fim de auxilia-los no processo de tomada de decisão. Um módulo para o aluno realizar pesquisa junto ao público-alvo, e um módulo para a comunidade em geral para a consulta das pesquisas realizadas e dicas de segurança.

Vale destacar que estes três módulos foram elaborados e validados juntamente com o Corpo de Bombeiros de Brusque e os alunos bolsistas da Unifebe participantes do programa de extensão universitária Operação Alerta Vermelho, etapas estas de extrema importância para sua correta funcionalidade e automatização dos seus processos resultando no apoio a tomada de decisão.

Optou-se o desenvolvimento nas plataformas *web* e *mobile*, pelo motivo que o sistema *web* poderá ser acessível de qualquer navegador que tenha acesso à *internet* e o sistema *mobile* poderá ser acessado em tempo real em dispositivo móvel com a plataforma Android que também tenha acesso a *internet* durante as pesquisas em campo.

Durante o desenvolvimento foi realizada a especificação completa do sistema. O levantamento de requisitos e os diagramas permitiram uma visão mais ampla e detalhada, facilitando assim o desenvolvimento do sistema.

Enfim, pode-se perceber que o sistema Alerta Vermelho trouxe grandes benefícios para o Corpo de Bombeiros de Brusque e para a população. Ressalta-se que este programa é uma parceria entre o Corpo de Bombeiros e a Unifebe onde se gerenciam questões importantes a serem pesquisadas nos bairros da cidade e apresentar visões e resultados por meio de gráficos. Já para a população o sistema conta com dicas de segurança residencial, telefones de utilidade pública e o acesso aos relatórios dos bairros pesquisados. Essas informações, auxiliarão para evitar acidentes domiciliares, também como agir diante da ocorrência desses acidentes. Já os relatórios dos bairros poderão ser utilizados pela população para conscientização e na tomada de decisão.

A partir do momento que o projeto de desenvolvimento foi concluído o sistema *web* e *mobile* passaram a ser utilizados pelo Corpo de Bombeiros e os acadêmicos da Unifebe durante o segundo semestre de 2017, onde, neste período obteve-se resultados expressivamente positivos, além de facilitar e agilizar o processo foi possível garantir e mapear de forma segura os bairros pesquisados na cidade de Brusque para posterior avaliação e reutilização.

6 TRABALHOS FUTUROS

Para este trabalho de conclusão de curso, futuramente poderão ser agregadas novas funcionalidades como desenvolver o aplicativo para a plataforma iOS da Apple. Para o módulo administrativo do Corpo de Bombeiros, pode-se criar uma nova tela para cadastrar dicas de segurança, e criar uma outra tela para adicionar fotos do programa Operação Alerta Vermelho, ambas as telas serão visualizadas no módulo da população.

Para o módulo do aluno aperfeiçoar o sistema para que funcione com o cadastro da pesquisa off-line, e quando o aluno estiver conectado a uma rede de internet esses dados sincronizem automaticamente com o sistema.

Compartilhar estas funcionalidades do programa Alerta Vermelho com o aplicativo Alerta Brusque como forma de serviços entre o Corpo de Bombeiros e a Defesa Civil de Brusque.

REFERÊNCIAS BIBLIOGRÁFICAS

CBMSC. **A instituição**. Disponível em: <https://portal.cbm.sc.gov.br/index.php/institucional/o-cbmsc>. Acessado 08: out. 2016.

CRUZ, Tadeu. **Sistemas de informações gerenciais: tecnologia da informação e a empresa do século XXI**, 3.ed. São Paulo: Atlas, 2008.

DEITEL, Paul. **Android para programadores: uma abordagem baseada em aplicativos**. Porto Alegre: Bookman, 2013.

DESENVOLVIMENTO ÁGIL. **Scrum**. Disponível em: < http://www.desenvolvimentoagil.com.br/scrum/>. Acessado em: 20 nov. 2016.

ENGHOLM JR, Hélio. **Engenharia de software na prática**. 1ª Ed. São Paulo. Editora Novatec, 2010.

ESTRAICH, Erian do Nascimento. **Bootstrap**. Disponível em: < http://servicesweb.com.br/artigos/bootstrap/ >. Acessado em: 03 set. 2017.

FEITOSA, Jean Martins. **Sistema de Informação Gerencial para o Corpo de Bombeiros de Balsas – MA: Uma Proposta de Implantação.** Disponível em: <http://www.sirc.unifra.br/artigos2010/11.pdf>. Acessado em: 27 set. 2016.

FEITOSA, Jean Martins. **Sistema de informação gerencial – geográfico para o corpo de bombeiros de balsas – MA**. Disponível em: <http://si.unibalsas.com.br/wp-content/uploads/2010/10/TCC_Jean-v.pdf>. Acessado em: 27 set. 2016.

FERNANDES, D. B. **Metodologia dinâmica para o desenvolvimento de sistemas versáteis**. São Paulo: Érica, 1999.

FERREIRA E., EIS D. **HTML5 – Curso W3C Escritório Brasil**. Disponível em: < http://www.w3c.br/pub/Cursos/CursoHTML5/html5-web.pdf>. Acessado em: 17 ago. 2017.

FLORENTINO, Pablo Vieira. **Sistemas de informações.** Disponível em: <http://www.ifba.edu.br/professores/pablovf/repositorio/siTiposSi.pdf>. Acessado em: 04 out. 2016.

FREITAS, H.; BECKER, J. L.; KLADIS, C. **Informação para a decisão**. Porto Alegre: Ortiz, 1997.

GABARDO, Ademir Cristiano. **PHP e MVC com Codelgniter**. São Paulo. Editora Novatec 2012.

GERALDO, Fernando. **Implementação em MVC utilizando o Zend Framework**. Disponível em: <http://www.devmedia.com.br/implementacao-em-mvc-utilizando-o-zend-framework/19072>. Acessado em: 16 out. 2016.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa.** 4. ed. São Paulo: Atlas, 2008.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 5ª ed. São Paulo: Atlas, 2010.

GIT. **Primeiros passos - Sobre Controle de Versão**. Disponível em: https://git-scm.com/book/pt-br/v1/Primeiros-passos-Sobre-Controle-de-Vers%C3%A3o>. Acessado em: 16 out. 2016.

GUEDES, Gilleanes T.A; **UML 2 Guia Prático**. 2ª Edição. São Paulo. Editora Novatec, 2009.

GUEDES, Gilleanes T.A; **UML 2 Guia Prático**. 2ª Edição. São Paulo. Editora Novatec, 2014.

GUIA PMBOK. **Um Guia do Conhecimento em Gerenciamento de Projetos.** Quinta edição. Project Management Institute. Pennsylvania - USA ,2013.

GUIMARÃES, Natalia. **Business Intelligence: o que é e qual a importância para sua empresa.** Disponível em: < http://www.newsmonitor.com.br/blog/business-intelligence-o-que-%C3%A9-e-qual-a-import%C3%A2ncia>. Acessado em: 15 out. 2016.

KOSCIANSKI, André. **Qualidade de Software**. 2ª Ed. São Paulo. Editora Novatec, 2007.

LEAL, Tiago. **Sistemas de informações gerenciais e sua importância para tomada de decisões.** Disponível em: http://www.administradores.com.br/artigos/carreira/sig-e-sua-importancia-para-tomada-de-decisoes/26869/. Acessado em: 09 out. 2016.

LEITE F. A. “Frameworks e Padrões de Projeto”; **Dev Media**. Disponível em: < http://www.devmedia.com.br/frameworks-e-padroes-de-projeto/1111>. Acessado em: 16 out. 2016.

LEITE F. A. “Metodologia de desenvolvimento de Software”; **Dev Media**. Disponível em: < http://www.devmedia.com.br/metodologia-de-desenvolvimento-de-software/1903>. Acessado em: 16 out. 2016.

MARTINEZ, Marina. **Sistema de Informação Gerencial.** Disponível em: <http://www.infoescola.com/administracao_/sistema-de-informacao-gerencial/>. Acessado em: 06 out. 2016.

MENDEZ, Cigano Morrison. **Diferenças entre Git, SVN e CVS**. Disponível em: http://pt.stackoverflow.com/questions/8315/diferen%C3%A7as-entre-git-svn-e-cvs. Acessado em: 19 out. 2016.

MILANI, André. **Construindo aplicações *Web* com PHP e MySQL.** São Paulo: Novatec, 2010.

MINETTO E. L. **Frameworks para desenvolvimento PHP**. São Paulo: Novatec Editora, 2007.

MYSQL. **MySQL.** Disponível em: <http://gsw.net.br/mysql/>, acessado em 09 out. 2016.

OLIVEIRA, Djalma de Pinho Rebouças de. **Sistemas de informações gerenciais: estratégicas, táticas, operacionais**.7. ed. São Paulo: Atlas, 2001.

OLIVEIRA, Rodrigo. **Conceitos introdutórios sobre melhoria e avaliação de processos de software**. Disponível em: <http://www.devmedia.com.br/conceitos-introdutorios-sobre-melhoria-e-avaliacao-de-processos-de-software/10577>. Acessado em: 26 out. 2016.

O’BRIEN, J. A. **Sistemas de informação e as decisões gerenciais na era da internet: gerenciamento de dados.** Tradução Célio Knipel Moreira e Cid Knipel Moreira. 2. ed. São Paulo: Saraiva, 2004

PACIEVITCH, Yuri. **HTML**. Disponível em: <http://www.infoescola.com/informatica/html/>. Acessado em: 21 ago. 2017.

PEREIRA, Ana Paula. **O que é CSS?**. Disponível em: < https://www.tecmundo.com.br/programacao/2705-o-que-e-css-.htm >. Acessado em: 03 set. 2017.

PHP. **O que é PHP?**. Disponível em:<https://secure.php.net/manual/pt_BR/intro-whatis.php>, acessado em 21 ago. 2017.

PHP. **História do PHP**. Disponível em:<https://secure.php.net/manual/pt_BR/history.php.php>, acessado em 02 set. 2017.

PRESSMAN, Roger S. **Engenharia de software**. 3ª. ed. São Paulo: McGraw-Hill, 2006.

PRESSMAN, Roger. S. **Engenharia de Software**. 6ª edição McGraw-Hill, 2010.

PRESSMAN, Roger S. **Engenharia de software**. 7ª. ed. Porto Alegre: AMGH, 2011.

RODRIGUES, Joel. **Modelo Entidade Relacionamento (MER) e Diagrama Entidade-Relacionamento (DER).** Disponível em: <http://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332>. Acessado em: 14 mai. 2017.

ROMANATO, Allan. **Introdução ao Android Studio**. Disponível em: http://www.devmedia.com.br/introducao-ao-android-studio/34003. Acessado em: 16 out. 2016.

SILVA, E. L.; MENEZES, E. M. **Metodologia da pesquisa e elaboração de dissertação**. Florianópolis: Universidade Federal de Santa Catarina, 2005.

SILVA, Marise Borba da; SCHAPPO, Vera Lúcia**. Introdução a pesquisa em educação**. Florianópolis: UDESC, 2002.

SILVA, Maurício Samy. **JavaScript: guia do programador.** São Paulo: Novatec, 2010.

SOMMERVILLE, I. **Engenharia de Software**. 8ª edição. Pearson Addison-Wesley, 2007.

SOMMERVILLE, I. **Engenharia de Software**. 9ª edição. Pearson Addison-Wesley, 2011.

TEIXEIRA, José Ricardo. **Introdução ao MYSQL**. Disponível em: https://www.devmedia.com.br/introducao-ao-mysql/27799. Acessado em: 05 jul. 2017.

VASCONCELOS A. M. L. de et al. **Introdução à engenharia de software e à qualidade de software**. UFLA/FAEPE, 2006. 57 p. (Curso de Pós-graduação “Lato Sensu” (Especialização) à Distância – Melhoria de Processo de Software.

W3SCHOOLS. **CSS Tutorial**. Disponível em: <https://www.w3schools.com/css/>. Acessado em: 03 set. 2017.

ZWASS, V. **Management Information Systems**. Dubuque: Wm. C. Brown Publishers, 1992.

APÊNDICES

APÊNDICE A – DEPOIMENTO DO CORPO DE BOMBEIROS

O sistema implantado pelo acadêmico Diego para o projeto Alerta Vermelho, nos permitiu com muito sucesso a trazer as informações com mais rapidez, eficiência e com nível de confiança da pesquisa realizada pelos acadêmicos e não esquecendo na facilidade de uso, facilidade da manutenção e da interação social entre acadêmicos, instituições (Bombeiros x UNIFEBE) e de toda comunidade brusquense e aos arredores.

Portanto, podemos dizer com certeza que este sistema implantado por este acadêmico preocupado com o bem-estar de toda a comunidade vai nos ajudar no planejamento, na organização e minimizar seja ela de uma catástrofe de impacto natural ou provocada pelo homem com mais eficiência e rapidez.

Parabéns pelo seu empenho e dedicação de estarmos juntos nestes anos no projeto Alerta Vermelho.

Obrigado !!!

_____

B.C Benvenutti

_____

B.C Vanini