



UFOP

Universidade Federal
de Ouro Preto

**Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas**

**Sistema de Informação para controle
de materiais e doações aos bombeiros
voluntários**

Silvandro Sergio Martins Oliveira

**TRABALHO DE
CONCLUSÃO DE CURSO**

**ORIENTAÇÃO:
Prof. Dr. Euler Horta Marinho**

**Dezembro, 2018
João Monlevade–MG**

Silvandro Sergio Martins Oliveira

**Sistema de Informação para controle de
materiais e doações aos bombeiros voluntários**

Orientador: Prof. Dr. Euler Horta Marinho

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Universidade Federal de Ouro Preto

João Monlevade

Dezembro de 2018

O482s Oliveira, Silvano Sergio Martins.
Sistema de Informação para controle de materiais e doações aos bombeiros voluntários [manuscrito] / Silvano Sergio Martins Oliveira. - 2018.

91f.: il.: color; graf; tabs.

Orientador: Prof. Dr. Euler Horta Marinho.

Monografia (Graduação). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Departamento de Computação e Sistemas de Informação.

1. Software de aplicação - desenvolvimento. 2. Aplicativos móveis. 3. Aplicações web. 4. Sistemas de Informação. I. Marinho, Euler Horta. II. Universidade Federal de Ouro Preto. III. Título.

CDU: 004.775

Catálogo: ficha.sisbin@ufop.edu.br

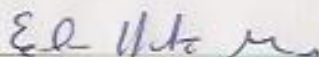
Curso de Sistemas de Informação

FOLHA DE APROVAÇÃO DA BANCA EXAMINADORA

Sistema de Informação para Controle de Materiais
e Doações aos Bombeiros Voluntários

Silvandro Sergio Martins Oliveira

Monografia apresentada ao Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto como requisito parcial da disciplina CSI499 – Trabalho de Conclusão de Curso II do curso de Bacharelado em Sistemas de Informação e aprovada pela Banca Examinadora abaixo assinada:



Prof. Dr. Euler Horta Marinho
DECSI - UFOP
Professor Orientador



Prof. M^a. Daniela Rodrigues Dias
Faculdade Doctum – João Monlevade
Professora Convidada



Prof. Dr. Diego Zuquim Guimarães Garcia
DECSI - UFOP
Professor Convidado

João Monlevade, 03 de Dezembro de 2018

Este trabalho é dedicado a minha mãe, por sempre acreditar em mim e me apoiar, a minha avó, que de onde ela estiver agora sei que estará orgulhosa de mim e a todos os meus amigos.

Agradecimentos

Agradeço primeiramente a Deus, por ter me dado essa oportunidade, e me proporcionado coragem e força para poder chegar até aqui.

A minha mãe Célia, guerreira, que sempre batalhou para fazer o melhor por mim, me apoiando e incentivando, e me dando forças para poder seguir. A minha avó Maria, por todo ensinamento e valores passados por ela, e por sempre acreditar em mim. A toda minha família por acreditar no meu potencial e me apoiar.

A minha namorada Isabela, pelo amor, companheirismo e motivação que sempre me proporcionou, e apesar dos momentos difíceis, juntos sempre fomos mais.

Aos meus eternos amigos, os Hermanos, pelo companheirismo e todo apoio que me deram nos momentos mais difíceis.

A equipe da Informática Desenvolvimento da Bio Extratus, onde eu pude obter um grande desenvolvimento pessoal e profissional.

Ao Bráulio da corporação de São Domingos do Prata, por acreditar e contribuir com esse projeto, sempre estando lado a lado.

Ao Euler, pelos ensinamentos em sala de aula, por acreditar em mim e poder me auxiliar no desenvolvimento deste projeto.

Aos professores e professoras que tive durante a vivência na graduação, pelo ensino de qualidade que me proporcionaram.

A todos que direta ou indiretamente, fizeram parte dessa minha trajetória, e que nunca deixaram eu desistir, afinal, "a persistência é o caminho do êxito".

“If you think you can do a thing or think you can’t do a thing, you’re right.”

— Henry Ford (1863 – 1947)

Resumo

Com o aumento do volume de dados gerados pelas organizações e o avanço tecnológico, as organizações necessitam que as informações sejam estruturadas, apoiando assim os processos e pessoas. Os sistemas de informações gerenciais auxiliam as organizações, no qual são reunidas as informações essenciais para desempenho de tarefas e tomada de decisão, e em caso de organizações do terceiro setor, ainda pode haver um grande impacto social com a utilização da tecnologia da informação a seu favor. Este trabalho propõe o desenvolvimento de novas funcionalidades para o projeto SisGera, no qual serão desenvolvidas com o propósito de apoiar as atividades administrativas das corporações, e para tal, será realizado um estudo em abordagens utilizadas no desenvolvimento de aplicações *Web*, bem como a utilização de ferramentas voltadas ao desenvolvimento *Web* para implementação e validação das novas funcionalidades. No desenvolvimento do sistema, foi possível realizar o estudo de abordagens e ferramentas utilizadas na atualidade, bem como a utilização de técnicas para o projeto de tal software. Com a implantação do sistema, será possível obter um ganho nas atividades administrativas das corporações, otimizando assim os processos de tais corporações, e trazendo um impacto social positivo para a população que utiliza os serviços prestados pelas corporações.

Palavras-chaves: SIG. Sistemas *Web*. *IFML*.

Abstract

With the increased volume of data generated by organizations and technological progress, such as organizations that information fits, thus supporting processes and people. Management information systems, such as organizations, do not qualify as essential to task and decision making. Private sector organizations can be socially impacted by the use of information technology. The work was created for new functionalities of the SisGera project to support the administrative activities of the companies. A study was carried out on Web application development practices and a tool for Web development for the implementation and validation of the new functionalities of the SisGera project. In the development of the system, it was possible to study the approaches and tools used today, as well as to use techniques for software design. With the implementation of the system, it was possible to obtain a gain in the administrative activities of the corporations, thus optimizing the processes of such corporations, and bringing a positive social impact to the use of the services provided by the corporations.

Key-words: MIS. Web Systems. IFML.

Lista de ilustrações

Figura 1 – Linha do tempo do projeto SisGera	17
Figura 2 – Evolução do Sistema	21
Figura 3 – Diagrama ER para representação de um banco de dados de um banco.	24
Figura 4 – Principais fases do projeto de banco de dados	25
Figura 5 – Diagrama de Classes	26
Figura 6 – Teste de Unidade	28
Figura 7 – Painel de Controle do <i>AdminLTE</i>	35
Figura 8 – Diagrama ER da funcionalidade de controle de bens patrimoniais	39
Figura 9 – Diagrama <i>IFML</i> da funcionalidade de histórico do usuário	40
Figura 10 – Diagrama de classe da funcionalidade de controle de doações	41
Figura 11 – Exemplos de testes criados para a funcionalidade de controle de estoque.	43
Figura 12 – Caso de teste criado para a funcionalidade de geração do B.O. pelo usuário externo no SisGera.	45
Figura 13 – Tela Gerar Ocorrência.	46
Figura 14 – B.O. simplificado com a chave de autenticidade gerado por um usuário externo.	47
Figura 15 – Tela Validar Ocorrência.	48
Figura 16 – Tela Gerenciar Bens Patrimoniais.	48
Figura 17 – Tela Gerenciar Doações.	49
Figura 18 – Tela Gerenciar Produtos.	50
Figura 19 – Tela Controle de Estoque.	51
Figura 20 – Tela Permissões de Acesso.	51
Figura 21 – Tela Controle de Históricos do Usuário.	52
Figura 22 – Tela Fale Conosco.	53
Figura 23 – Funcionalidade de gerenciamento de patrimônios no ANY3	54
Figura 24 – Funcionalidade de gestão de doações do HYB	55
Figura 25 – Diagrama ER da funcionalidade de geração e validação do B.O. pelo usuário externo.	66
Figura 26 – Diagrama ER da funcionalidade de controle de bens patrimoniais.	67
Figura 27 – Diagrama ER da funcionalidade de controle de doações.	68
Figura 28 – Diagrama ER das funcionalidades de cadastro de produtos e controle de estoque.	69
Figura 29 – Diagrama ER da funcionalidade de permissões de acesso.	70
Figura 30 – Diagrama ER da funcionalidade de histórico do usuário.	71
Figura 31 – Esboço completo das tabelas utilizadas nas novas funcionalidades do SisGera.	72

Figura 32 – Diagrama <i>IFML</i> das funcionalidades de geração e validação do B.O. pelo usuário externo.	73
Figura 33 – Diagrama <i>IFML</i> da funcionalidade de controle de bens patrimoniais.	74
Figura 34 – Diagrama <i>IFML</i> da funcionalidade de gestão de doações.	75
Figura 35 – Diagrama <i>IFML</i> da funcionalidade de cadastro de produto.	76
Figura 36 – Diagrama <i>IFML</i> da funcionalidade de controle de estoque.	77
Figura 37 – Diagrama <i>IFML</i> da funcionalidade de permissões de acesso.	78
Figura 38 – Diagrama <i>IFML</i> da funcionalidade de histórico do usuário.	78
Figura 39 – Diagrama <i>IFML</i> da funcionalidade Fale Conosco.	79
Figura 40 – Diagrama de classes funcionalidade de histórico de requisição do B.O. pelo usuário externo.	80
Figura 41 – Diagrama de classes da funcionalidade de controle de bens patrominiais.	81
Figura 42 – Diagrama de classes da funcionalidade de controle de doações.	82
Figura 43 – Diagrama de classes da funcionalidade de cadastro de produtos.	83
Figura 44 – Diagrama de classes da funcionalidade de controle de estoque.	84
Figura 45 – Diagrama de classes da funcionalidade de permissões de acesso.	84
Figura 46 – Diagrama de classes da funcionalidade de histórico do usuário.	85

Lista de tabelas

Tabela 1 – Notações para modelagem IFML	32
Tabela 2 – Histórias de usuário do SisGera	38
Tabela 3 – Comparativo de funcionalidades entre o SisGera e os sistemas correlatos.	56

Lista de abreviaturas e siglas

Ajax *Assynchronous Javascript and XML*

CGI *Common Gateway Interface*

CSS *Cascading Style Sheets*

DER Diagrama Entidade-Relacionamento

ER Entidade-Relacionamento

GNU *GNU's Not Unix*

GPL *General Public Licence*

HTML *Hypertext Markup Language*

HTTP *Hypertext Transfer Protocol*

IFML *Interaction Flow Modeling Language*

MVC *Model-View-Controller*

ODMG *Object Database Management Group*

OMG *Object Management Group*

PHP *PHP Hypertext Preprocessor*

PIM *Platform-Independent Model*

SGBD Sistema de Gerenciamento de Banco de Dados

SIG Sistemas de Informações Gerenciais

SisGera Sistema de Gerenciamento e Registro de Atividades

SGBD Sistema de Gerenciamento de Banco de Dados

SQL *Structured Query Language*

WebML *Web Modeling Language*

WWW *World Wide Web*

Sumário

1	INTRODUÇÃO	16
1.1	Problema	16
1.2	Objetivos	17
1.2.1	Objetivo Geral	17
1.2.2	Objetivos Específicos	18
1.3	Justificativa	18
1.4	Funcionalidades	19
1.5	Organização do Trabalho	19
2	REVISÃO BIBLIOGRÁFICA	20
2.1	Sistemas de Informações Gerenciais (SIG)	20
2.2	Processo de Desenvolvimento de <i>Software</i>	21
2.2.1	Engenharia de Requisitos	22
2.2.1.1	Histórias de Usuário	22
2.2.2	Projeto de <i>Software</i>	23
2.2.2.1	Entidade-Relacionamento (ER)	24
2.2.2.2	Diagrama de Classes	25
2.2.3	Teste de <i>Software</i>	27
2.2.3.1	Teste de Unidade	27
2.2.3.2	Teste Funcional	28
2.3	Desenvolvimento de Sistemas <i>Web</i>	28
2.3.1	<i>Web Modeling Language</i> (WebML)	29
2.3.2	<i>Interaction Flow Modeling Language</i> (IFML)	30
2.3.3	Notações da <i>IFML</i>	31
3	DESENVOLVIMENTO E RESULTADOS	33
3.1	Escolha tecnológica	33
3.1.1	<i>MySQL</i>	33
3.1.2	<i>Bootstrap</i>	34
3.1.2.1	AdminLTE	34
3.1.3	<i>PHP Hypertext Preprocessor</i> (PHP)	35
3.1.4	<i>Laravel</i>	36
3.2	Requisitos	37
3.2.1	Histórias de usuário	37
3.3	Projeto	38
3.3.1	Entidade-Relacionamento (ER)	38

3.3.2	<i>Interaction Flow Modeling Language (IFML)</i>	39
3.3.3	Diagrama de Classes	40
3.4	Testes	41
3.4.1	Teste de Unidade	41
3.4.2	Teste Funcional	43
3.5	Apresentação das funcionalidades	45
3.5.1	Geração e validação do boletim de ocorrência usuário externo	46
3.5.2	Controle de bens patrimoniais	48
3.5.3	Controle de doações	49
3.5.4	Cadastro de produtos	49
3.5.5	Controle de estoque	50
3.5.6	Controle de permissões do usuário	51
3.5.7	Controle de históricos do usuário	52
3.5.8	Fale Conosco	52
3.6	Sistemas Similares	53
3.6.1	Economato	53
3.6.2	ANY3	53
3.6.3	HYB	54
3.7	Considerações finais	55
4	CONCLUSÃO	57
4.1	Trabalhos futuros	57
	REFERÊNCIAS	59
	APÊNDICES	61
	APÊNDICE A – LEVANTAMENTO DE REQUISITOS	62
A.1	Geração do B.O.	62
A.2	Controle Patrimonial	62
A.3	Doações	63
A.4	Produtos	63
A.5	Estoque	64
A.6	Usuários	65
A.7	Fale Conosco	65
	APÊNDICE B – PROJETO	66
B.1	Entidade-Relacionamento	66
B.2	Diagramas <i>IFML</i>	73

B.3	Diagramas de Classe	80
------------	--------------------------------------	-----------

	ANEXOS	86
--	---------------	-----------

1 Introdução

No trabalho de [Arantes \(2018\)](#), foi realizado o desenvolvimento do projeto do Sistema de Gerenciamento e Registro de Atividades ([SisGera](#)), no qual consistia em um sistema para controle de ocorrências das corporações de bombeiros voluntários de São Domingos do Prata e Barão de Cocais. Porém, ao ser desenvolvido o sistema, novos requisitos e necessidades surgiram, motivando assim o desenvolvimento de uma nova versão do projeto SisGera, agregando a ele funcionalidades para gestão de atividades administrativas das corporações, otimizando assim os processos internos das corporações, bem como auxiliando na tomada de decisão.

1.1 Problema

Fundada em 28 de março de 2012, a Associação de Bombeiros Voluntários de São Domingos do Prata é uma entidade voluntária que presta serviços de resgate e combate a incêndio, 24 horas por dia, 7 dias por semana, e está sediada na cidade de São Domingos do Prata/MG. De maneira mais específica, eles prestam atendimento a vítimas de trauma, busca e salvamento, ajudam em acidentes com produtos perigosos, e na prevenção e combate a incêndio.

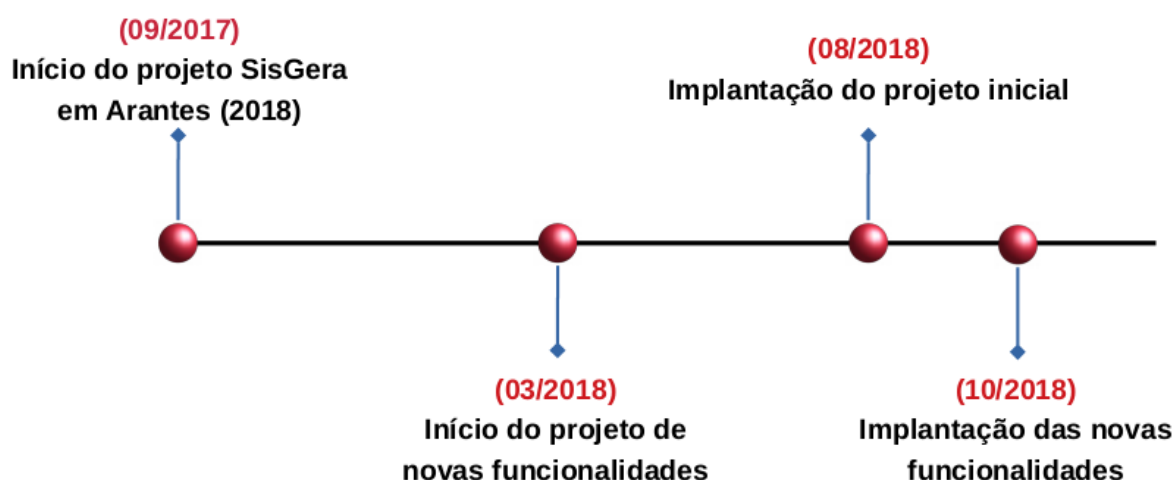
A corporação de São Domingos do Prata conta com 26 membros na equipe, sendo que eles não possuem nenhum tipo de remuneração, e são profissionais de diferentes áreas e cidades. Para a realização das atividades, contam com doações de órgãos públicos e privados, e pessoas físicas. O raio de atendimento contempla as rodovias 381 e 262, nas proximidades de São Domingos do Prata, Dionísio, São José do Goiabal, MGC 120, LMG 820 e estradas da área rural nas intermediações. Atendem em média 28 ocorrências mensais.

O Grupo de Resgate Voluntário de Emergência é uma entidade voluntária que também presta serviços de resgate, vítimas de trauma e combate a incêndio, e está sediada na cidade de Barão de Cocais/MG, prestando serviço nas cidades de Barão de Cocais, Santa Bárbara, Catas Altas e no trecho entre a BR 381 até o trevo de Boa Vista, e atualmente conta com 48 membros na equipe.

Atualmente, as corporações realizam operações administrativas de forma manual, pois como prestam serviço voluntário, não contam com recursos para adesão de um sistema de informação comercial. Sendo assim, durante o desenvolvimento do projeto SisGera no trabalho de [Arantes \(2018\)](#), foram levantadas novas necessidades das corporações, sendo tais necessidades voltadas para as atividades administrativas, motivando assim o desenvolvimento deste trabalho. Na seção [3.6](#) do Capítulo [3](#), são apresentados sistemas

disponíveis no mercado com funcionalidades similares, porém tais sistemas não atendem aos grupos voluntários, além de serem pagos, sendo assim o sistema desenvolvido neste trabalho além de ser uma solução personalizada para as corporações, proporcionará agilidade, assertividade, segurança, controle dos dados e geração de informações para auxílio na tomada de decisão das corporações. A [Figura 1](#) demonstra uma linha do tempo desde o início do projeto SisGera em [Arantes \(2018\)](#), até o desenvolvimento deste trabalho.

Figura 1 – Linha do tempo do projeto SisGera



Fonte: Elaborado pelo autor (2018)

1.2 Objetivos

Nesta seção, serão descritos o objetivo geral do trabalho e os objetivos específicos em que serão apresentadas as práticas desempenhadas para chegar ao objetivo geral deste trabalho.

1.2.1 Objetivo Geral

O presente trabalho tem como objeto a continuidade do projeto SisGera, criado inicialmente no trabalho de [Arantes \(2018\)](#), no qual serão criadas novas funcionalidades para o controle de atividades administrativas e estoque das corporações, sendo que tais funcionalidades abordam áreas de Gestão de Pessoas, Controle Patrimonial, Controle de Doações, Impressão do Boletim de Ocorrência e Controle de Estoque. Atualmente, todas essas operações são controladas manualmente pelas corporações, e a expectativa é que tais funcionalidades possam aumentar a agilidade e assertividade nas operações.

1.2.2 Objetivos Específicos

Este trabalho tem como objetivos específicos os seguintes:

- Revisar a literatura acerca de Sistemas de Informações Gerenciais, tecnologia *Web*, Engenharia de *Software* e sistemas similares.
- Levantar os requisitos junto aos *stakeholders* do projeto.
- Gerar artefatos a partir da modelagem da aplicação.
- Desenvolver novas funcionalidades no projeto SisGera para apoio de atividades administrativas e controle de estoque das corporações.
- Criar testes automatizados para validação da aplicação desenvolvida.
- Implantar o sistema para utilização nas corporações.

1.3 Justificativa

Como as corporações de Barão de Cocais e São Domingos do Prata prestam serviços voluntários à comunidade, elas não possuem qualquer tipo de remuneração, e com isso não possuem recursos para adquirir um sistema de informação comercial conforme algum dos exemplificados na seção 3.6 do Capítulo 2. Sendo assim, houve a oportunidade do desenvolvimento de novas funcionalidades no projeto SisGera que contemplam as atividades administrativas das corporações e que possibilitam agilizar, organizar e minimizar erros operacionais nas atividades desempenhadas, além de prover um sistema personalizado, no qual se encaixa na realidade das corporações, e gratuito.

Embora o projeto SisGera tenha sido criado para atender inicialmente as corporações de São Domingos do Prata/MG e Barão de Cocais/MG, a ferramenta disponibiliza o cadastro de novas corporações, não se limitando apenas as duas mencionadas, e com isso as funcionalidades oferecidas pela ferramenta podem ter grande impacto social, uma vez que além de disponibilizar funcionalidades para controle das operações administrativas, também disponibilizará a geração do boletim de ocorrência pelo usuário externo, trazendo assim comodidade para a população.

Além de promover um grande impacto social positivo, o desenvolvimento deste trabalho possibilita também o estudo de técnicas e abordagens utilizadas no desenvolvimento e projeto de aplicações web.

1.4 Funcionalidades

Para dar continuidade ao projeto do SisGera criado por [Arantes \(2018\)](#), as seguintes funcionalidades foram desenvolvidas nesse trabalho:

- Geração e validação do B.O. pelo usuário externo;
- Controle patrimonial;
- Controle de doações;
- Cadastro de produtos;
- Controle de estoque;
- Controle de permissões do usuário;
- Controle de histórico do usuário;
- Fale conosco

As funcionalidades em questão serão detalhadas e demonstradas na seção [3.5](#) do Capítulo [3](#).

1.5 Organização do Trabalho

O restante deste trabalho é organizado como segue. O Capítulo [2](#) apresenta uma revisão da bibliografia, no qual são mencionados temas relacionados ao desenvolvimento deste projeto e sistemas correlatos disponíveis no mercado. O Capítulo [3](#) descreve as escolhas tecnológicas e as abordagens utilizadas no processo de desenvolvimento das novas funcionalidades do SisGera, desde o levantamento de requisitos, desenvolvimento, até a validação das funcionalidades. O Capítulo [4](#) relata as considerações finais do projeto e propostas para eventuais trabalhos futuros.

2 Revisão bibliográfica

Este capítulo apresenta uma revisão da literatura, bem como a apresentação de sistemas comerciais que oferecem funcionalidades similares.

2.1 Sistemas de Informações Gerenciais (SIG)

Antes de se falar sobre os Sistemas de Informações Gerenciais (SIG), é importante entender primeiramente o significado de um Sistema de Informação. Segundo [Laudon e Laudon \(2011\)](#), um Sistema de Informação tecnicamente pode ser definido como um conjunto de elementos relacionados entre si que coletam ou recuperam, processam, armazenam e distribuem informações para o apoio na tomada de decisões, a coordenação e o controle de uma organização. Tais sistemas também auxiliam gerentes e trabalhadores na análise de problemas, visualização de assuntos complexos e na criação de novos produtos, sendo que contém informações sobre pessoas, locais e itens significativos para a organização ou o ambiente que a cerca.

Atualmente, as organizações necessitam de informações estruturadas para auxílio na tomada de decisão, uma vez que com o crescimento da tecnologia de informação, um grande volume de informação desnecessária é gerado. Os Sistemas de Informações Gerenciais tem como objetivo transformar dados gerados pela organização em informação de forma organizada e estruturada, auxiliando na tomada de decisão da empresa, podendo assim otimizar os processos. O SIG trabalha com a obtenção dos dados, armazenamento, recuperação e processamento dos dados, resultando em informação e conhecimento.

Em [Plantullo e Hoffman \(2012\)](#), os SIGs são definidos como um conjunto de procedimentos, talentos e competências humanas essenciais, acrescidos de recursos tecnológicos que proporcionam condições de garantir uma atualização do seu sistema de informações. Ainda segundo [Plantullo e Hoffman \(2012\)](#), o SIG é mais voltado para o funcionamento das operações, porém existe SIG que é voltado para objetivos estratégicos.

As organizações podem ser divididas em níveis, tendo o nível estratégico, no qual é composto pela alta administração, e são tomadas as principais decisões da empresa, determinando assim os objetivos a longo prazo; o nível tático, no qual é articulado o nível estratégico e o operacional da organização, sendo composto por chefes e gerentes de cada setor, e transforma os objetivos definidos no nível estratégico em ações para o nível operacional, e o nível operacional, sendo composto por profissionais que administram e executam as tarefas essenciais no dia a dia da organização. Segundo [Laudon e Laudon \(2011\)](#), os SIGs estão no nível tático, no qual proporcionam relatórios sobre o desempenho

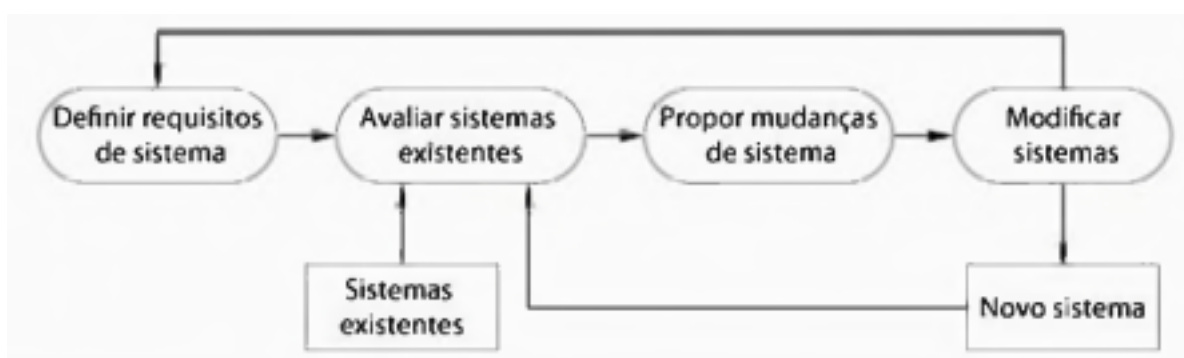
atual das organizações, permitindo assim monitorar, controlar e fazer previsões sobre o desempenho futuro da organização.

2.2 Processo de Desenvolvimento de *Software*

Sommerville (2011, p. 24) descreve quatro como as atividades básicas do processo de desenvolvimento de *software*, sendo elas:

- Especificação de *software*: processo de compreensão e definição das funcionalidades e restrições do sistema;
- Projeto e implementação de *software*: o projeto de *software* consiste na descrição da estrutura do *software* a ser desenvolvido, dos modelos, estruturas de dados, interface entre componentes e algoritmos usados. A implementação de *software* é o processo de conversão da especificação do sistema em um sistema executável;
- Validação de *software*: tem por objetivo mostrar que um *software* está coeso quanto as suas especificações, satisfazendo as especificações do cliente do sistema;
- Evolução do *software*: processo de manutenção do *software*, no qual o processo de desenvolvimento do *software* é pensado como um processo evolutivo, sendo constantemente alterado durante o período de vida em função de alteração dos requisitos e necessidades do cliente. O processo de evolução pode ser exemplificado na Figura 2.

Figura 2 – Evolução do Sistema



Fonte: Sommerville (2011, p. 29)

2.2.1 Engenharia de Requisitos

Para entender a engenharia de requisitos, primeiramente é necessário entender o significado de requisito, sendo que segundo [Sommerville \(2011, p. 57\)](#), os requisitos são as descrições do que o sistema deve fazer, especificando os serviços oferecidos e considerando restrições em seu funcionamento. Ainda conforme descrito por [Sommerville \(2011, p. 57\)](#), tais requisitos refletem as necessidades do cliente para um sistema.

Conforme descrito em [Pressman \(2011, p. 127\)](#), a engenharia de requisitos consiste no amplo espectro de tarefas e técnicas que levam ao entendimento dos requisitos, fornecendo o mecanismo apropriado para entender o que o cliente deseja, conforme suas necessidades, avaliando a viabilidade, negociando uma solução razoável, sem ambiguidades, validando a especificação e gerenciando necessidades à medida que são transformadas em um sistema.

[Sommerville \(2011, p. 57\)](#) descreve a engenharia de requisitos como o processo de analisar, documentar e verificar os serviços oferecidos pelo sistema e as restrições que ele deve considerar.

2.2.1.1 Histórias de Usuário

Conforme descrito em [Beck e Fowler \(2000\)](#), as histórias de usuário descrevem uma parte da funcionalidade que é valiosa ao usuário. Em [Cohn \(2004\)](#), as histórias de usuário são divididas em três aspectos:

- Uma descrição por escrito da história em um cartão;
- Conversas sobre a história com o objetivo de um detalhamento maior da mesma;
- Testes para a documentação e confirmação de detalhes, podendo ser usados também para determinar quando uma história é concluída.

No trabalho de [Longo e Silva \(2014\)](#), as histórias de usuário são citadas como uma declaração informal de um requisito de usuário, substituindo assim um grande documento de requisitos.

[Cohn \(2004\)](#) cita que para a criação de boas histórias de usuários, deve-se concentrar em seis aspectos, sendo os seguintes:

- Independente: deve-se ter cuidado para evitar dependências entre histórias, pois tais dependências podem trazer problemas de priorização e planejamento das histórias;
- Negociável: as histórias são descrições curtas da funcionalidade, sendo que os detalhes devem ser negociados entre cliente e equipe de desenvolvimento;

- Valiosa para os donos ou usuários: as histórias criadas nem sempre são valiosas aos usuários, existem situações no qual as histórias são criadas para atender alguma necessidade específica dos donos do *software* e irrelevantes aos usuários;
- Estimável: é importante que os desenvolvedores possam estimar o tamanho de uma história ou a quantidade de tempo que levará para se codificar tal história;
- Pequena: o tamanho da história é de grande importância pois se não tiverem um tamanho ideal, não poderão ser usadas no planejamento;
- Testável: as histórias devem ser escritas para serem testáveis, sendo que passar pelos testes significa que a história foi desenvolvida com sucesso.

Ainda segundo [Cohn \(2004\)](#), são descritas algumas técnicas para levantar as histórias de usuários, sendo as seguintes:

- Entrevistas: uma das abordagens mais utilizadas. Porém para o seu sucesso, uma das chaves é a seleção dos entrevistados, sendo importante entrevistar diferentes níveis de usuário, e na entrevista não perguntar apenas o que o usuário precisa, pois os usuários costumam não saber expressar suas verdadeiras necessidades e podem ocorrer situações no qual é desenvolvido o que o usuário pede mas não o que ele precisa;
- Questionários: é uma técnica eficaz para coletar informações de histórias que já se tem, sendo bastante úteis quando se necessita de respostas de um grande número de usuários para questões específicas;
- Observação: no qual se é observado a forma em que o usuário interage com o sistema, podendo assim obter informações para melhorar a produtividade e a experiência do usuário com o sistema;
- *Workshop* de redação de histórias - é uma reunião que inclui desenvolvedores, usuários, clientes do produto e outras partes que podem contribuir escrevendo histórias. Um *workshop* de redação de histórias conduzido de forma adequada pode ser uma maneira rápida de se escrever um grande número de histórias.

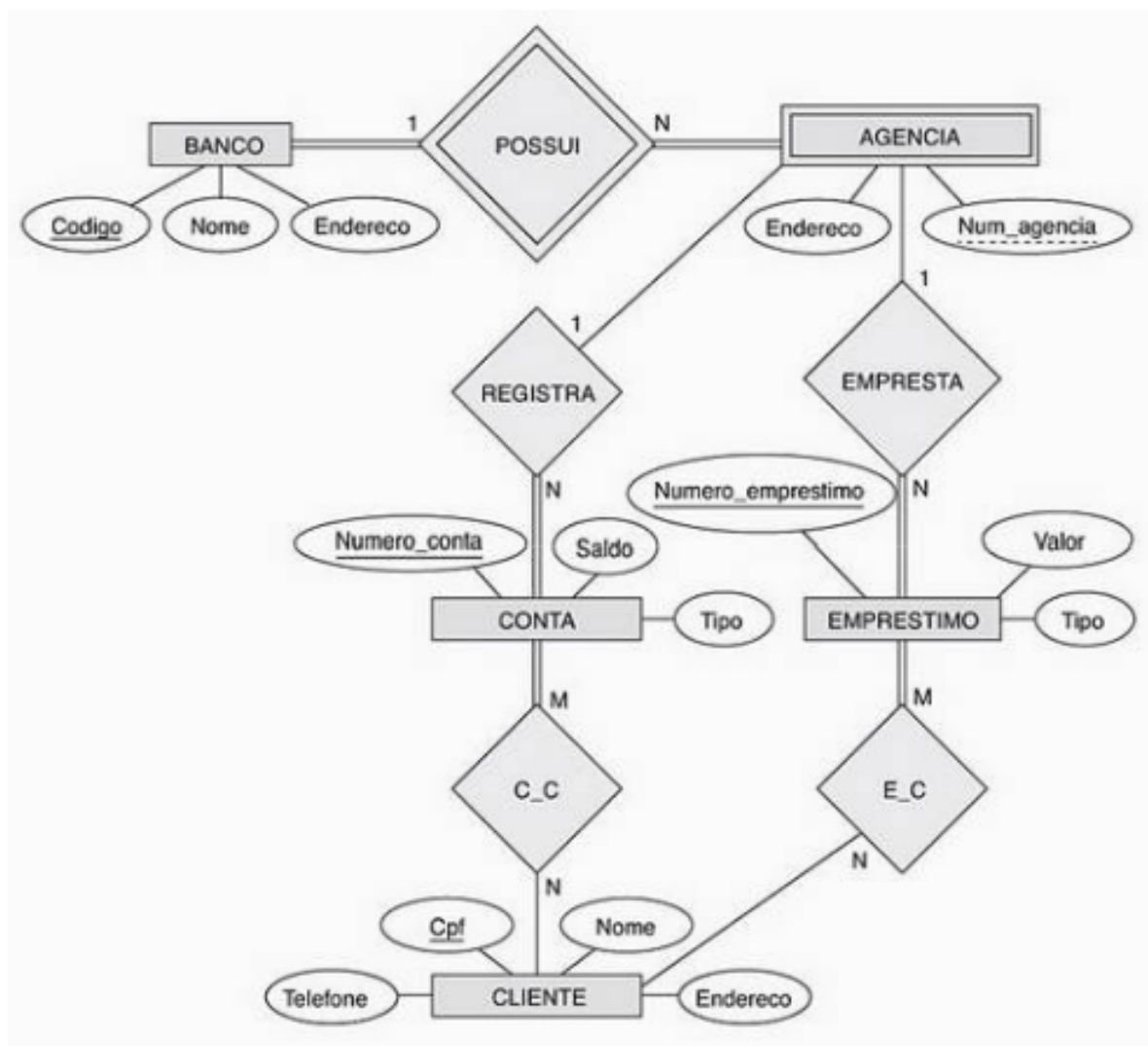
2.2.2 Projeto de *Software*

Esta seção irá realizar uma revisão da literatura acerca de abordagens utilizadas na fase de projeto de *software*, no qual são gerados artefatos do *software* a ser construído.

2.2.2.1 Entidade-Relacionamento (ER)

O modelo Entidade-Relacionamento (ER) é um modelo conceitual de alto nível que conta com um conjunto de objetos e o relacionamento entre tais objetos. Tal modelo representa a estrutura lógica do banco de dados da aplicação. Para representação gráfica, são utilizados os Diagrama Entidade-Relacionamento (DER), exemplificado na Figura 3.

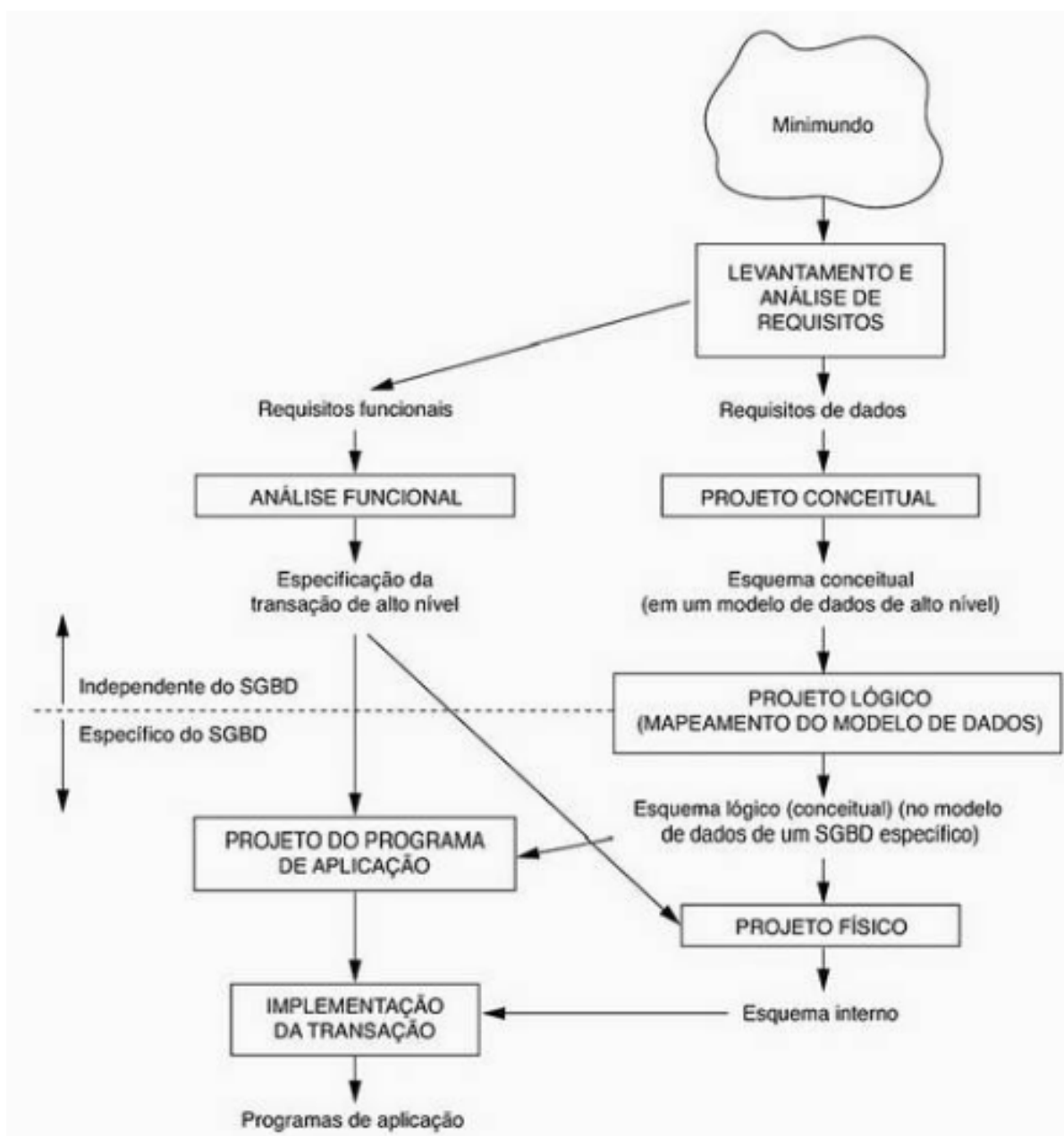
Figura 3 – Diagrama ER para representação de um banco de dados de um banco.



Fonte: Elmasri e Navathe (2010, p. 157)

Segundo Elmasri e Navathe (2010), o modelo Entidade-Relacionamento (ER) descreve os dados como entidades, relacionamentos e atributos. É um esquema conceitual de alto nível que representa o banco de dados, sendo uma descrição concisa dos requisitos de dados dos usuários e inclui detalhes dos tipos de entidades, relacionamentos e as restrições. O modelo ER é criado na etapa do projeto conceitual, uma das etapas do projeto de banco de dados conforme Figura 4.

Figura 4 – Principais fases do projeto de banco de dados



Fonte: [Elmasri e Navathe \(2010, p. 133\)](#)

2.2.2.2 Diagrama de Classes

Conforme descrito por [Sommerville \(2011\)](#), os diagramas de classe são utilizados para desenvolver um modelo de um sistema orientado a objetos demonstrando as classes e as suas associações, sendo possível expressá-los em diferentes níveis de detalhamento.

Em [Booch, Rumbaugh e Jacobson \(2006\)](#), os diagramas de classes são definidos como um conjunto de classes, interfaces e colaborações e seus relacionamentos, sendo

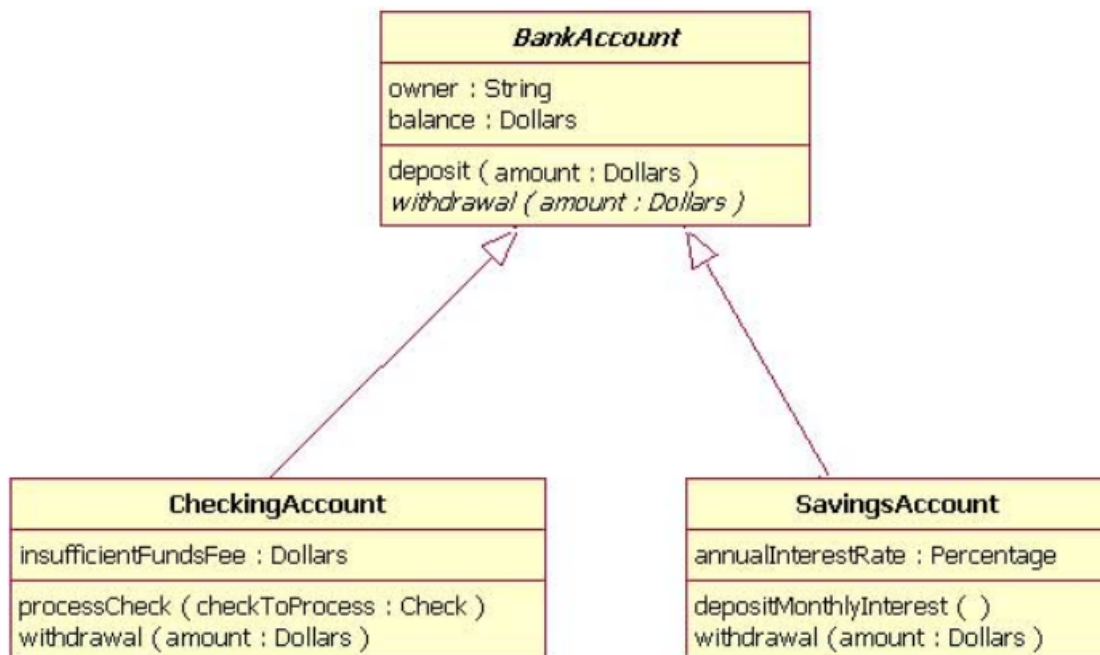
graficamente representados por uma coleção de vértices e arcos. Os diagramas de classes basicamente são utilizados para prover a modelagem da visão estática do sistema.

Ainda conforme descrito por [Booch, Rumbaugh e Jacobson \(2006\)](#), ao se usar os diagramas de classes para modelar a visão estática do sistema, tais diagramas podem ser usados em três formas:

- Fazer modelagem do vocabulário do sistema: decidir sobre quais abstrações fazem parte do sistema e quais estão fora dos limites do sistema;
- Fazer modelagem de colaborações simples: conjunto de classes, interfaces e outros elementos que trabalham em conjunto para prover um comportamento cooperativo;
- Fazer modelagem do esquema lógico de um banco de dados: modelagem de esquemas de banco de dados relacional ou orientado a objetos.

[Bell \(2016\)](#) cita que o objetivo dos diagramas de classes é mostrar os tipos que estão sendo modelados no sistema, sendo que esses tipos podem conter: uma classe, uma interface, um tipo de dados ou um componente. A [Figura 5](#) apresenta um exemplo de diagrama de classes.

Figura 5 – Diagrama de Classes



Fonte: [Bell \(2016, p. 6\)](#)

2.2.3 Teste de *Software*

O enorme avanço tecnológico impactou nas mais variadas áreas, inclusive no desenvolvimento de *software Web*, porém, embora tenha surgido diversas tecnologias de modo a facilitar e otimizar o desenvolvimento, a tarefa ainda assim não é algo simples, já que pode haver um aumento de complexidade a partir da dimensão e características do *software* criado. Atualmente, os sistemas *Web* são desenvolvidos para os mais diversos segmentos, e com isso houve a necessidade de se garantir a qualidade e confiabilidade de tais sistemas, uma vez que diversas aplicações *Web* podem acarretar em prejuízos enormes para a organização.

Conforme descrito em [Delamaro, Maldonado e Jino \(2007\)](#), é necessário o *software* passar pelo processo de "Validação, Verificação e Teste", que são uma série de atividades que tem como finalidade garantir que o *software* esteja em conformidade com o que foi especificado. Para entender a diferença entre validação e verificação, as seguintes perguntas podem ser feitas conforme descrito em [Koscianski e Soares \(2007\)](#):

- Validação: estamos construindo o produto certo?
- Verificação: estamos construindo certo o produto?

[Sommerville \(2011\)](#) divide o processo de testes em dois objetivos distintos, sendo:

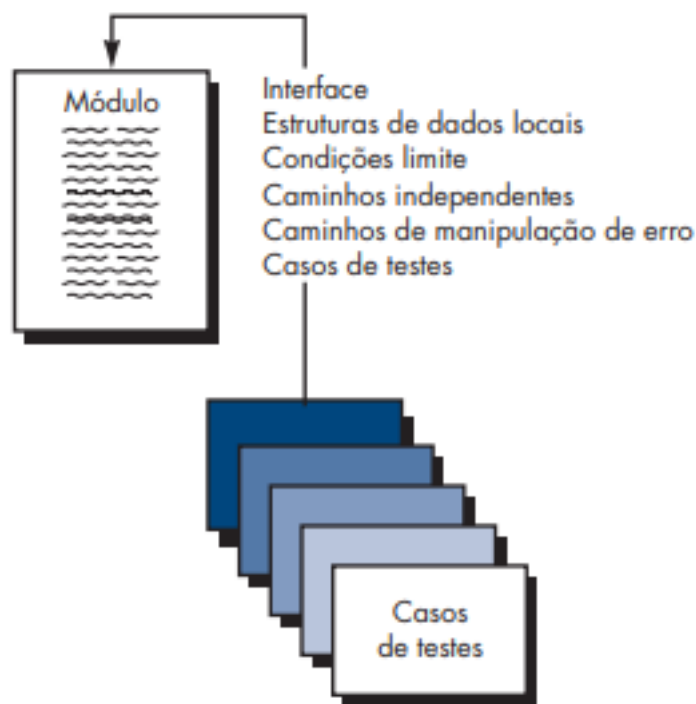
- Demonstrar que o *software* atende aos requisitos;
- Descobrir situações em que o *software* se comporta de forma diferente ao especificado.

Em [Koscianski e Soares \(2007\)](#), os testes são classificados de forma mais genérica em teste de caixa-preta e teste de caixa-branca. O teste de caixa-preta, ou também conhecido como teste funcional, analisa o comportamento da aplicação, não se preocupando com sua construção interna e se concentrando nos requisitos funcionais do *software*. O teste de caixa-branca, de forma contrária, analisa a estrutura como um todo, permitindo que sejam testadas partes específicas de um componente, e permitindo a verificação de forma mais precisa do comportamento do *software*.

2.2.3.1 Teste de Unidade

O teste de unidade é um tipo de teste caixa-branca no qual são testados os componentes do programa, como métodos e classes de objeto conforme descrito em [Sommerville \(2011\)](#). Em [Pressman \(2011\)](#), é dito que o foco do teste de unidade é a verificação da menor unidade de projeto de *software*. Ao se testar as classes de objeto, devem ser projetados testes que forneçam uma cobertura de todas as características do objeto. A [Figura 6](#) exhibe o teste de unidade esquematicamente.

Figura 6 – Teste de Unidade



Fonte: [Pressman \(2011, p. 408\)](#)

2.2.3.2 Teste Funcional

O teste funcional é um tipo de teste caixa-preta no qual são projetados casos de teste, fornecendo entradas e avaliando se as saídas geradas estão em conformidade com os objetivos especificados, conforme descrito em [Delamaro, Maldonado e Jino \(2007\)](#).

2.3 Desenvolvimento de Sistemas *Web*

No início dos anos 90, surge a *World Wide Web (WWW)*, que buscava disponibilizar conteúdo em um formato padrão, simples, porém poderoso, e tal conteúdo consistia em informações oferecidas a partir de hipertexto em *Hypertext Markup Language (HTML)*.

Conforme descrito em [Delamaro, Maldonado e Jino \(2007, p. 209\)](#), a arquitetura utilizada nas aplicações *Web* era a cliente-servidor de duas camadas, sendo um cliente *Web browser* visitando *Web sites* resididos em diferentes servidores que enviam arquivos *Hypertext Markup Language (HTML)* para o cliente. Porém, com o grande crescimento das aplicações *Web*, novas arquiteturas e tecnologias surgiram nos últimos anos.

[Marinho e Resende \(2011\)](#) descrevem que na evolução das aplicações *Web* surgiram as chamadas Aplicações Ricas para Internet, sendo possibilitado por tecnologias de de-

envolvimento como *Asynchronous Javascript and XML* ([Ajax](#)) e permitindo uma maior interação com o usuário, de forma mais avançada e sofisticada.

2.3.1 *Web Modeling Language* ([WebML](#))

Com o grande avanço da tecnologia *Web* nos últimos anos, as aplicações tem se tornado mais complexas, e com isso surgiu a demanda de métodos adequados para o projeto de aplicações *Web*. Conforme descrito em [Ceri, Fraternali e Bongio \(2000\)](#), com a crescente modernização das aplicações *Web*, as empresas que desenvolvem tais aplicações necessitam profundamente de métodos de projeto, formalismos, linguagens e ferramentas, o que poderia ser realizado de maneira eficaz, cobrindo todos os aspectos do processo de design. Foi em resposta a tal necessidade que o projeto W3I3 produziu a linguagem de modelagem *Web* denominada *WebML*, no qual aborda a especificação em alto nível de aplicativos *Web* multiplataformas.

Ainda segundo [Ceri, Fraternali e Bongio \(2000\)](#), o *WebML* permite que os designs especifiquem os principais recursos de um site em alto nível, sem a necessidade de se preocupar com detalhes da arquitetura da aplicação, e a modelagem *WebML* descreve as aplicações *Web* em quatro modelos principais, sendo os seguintes:

- Modelo estrutural: expressa o conteúdo de dados da aplicação em termos estruturais. Embora a *WebML* não tenha proposto uma linguagem padrão para este modelo, ela oferece suporte para outras notações como modelo ER, modelo *Object Database Management Group* ([ODMG](#)) e diagramas de classe da *UML*.
- Modelo de hipertexto: descreve os hipertextos exibidos ao usuário na navegação, sendo que cada hipertexto diferente define uma visão no sistema.
- Modelo de apresentação: expressa o layout e a aparência gráfica das páginas, independentemente de qual é o dispositivo de saída e a linguagem de execução, por meio de uma sintaxe *XML* abstrata.
- Modelo de personalização: é utilizada para armazenar conteúdo específico ou individual de um usuário ou um grupo, como sugestões de compras, lista de favoritos e preferências para personalização gráfica da aplicação.

Em [Ferreira \(2013\)](#), é citado a *WebML* como uma linguagem de notação visual simples, no qual conta com gráficos e especificações, englobando assim um projeto completo e que podem ser auxiliados por ferramentas de design visual. Ainda em [Ferreira \(2013\)](#), são descritos como principais objetivos da modelagem *WebML*:

- Descrição em alto nível da estrutura da aplicação, para a realização de consultas, atualizações ou manutenções.

- Disponibilização de diferentes visões do mesmo conteúdo.
- Separação do conteúdo de informação, compondo em páginas, navegação e apresentação.
- Armazenamento de informações do processo de projeto, podendo as mesmas serem utilizadas na geração dinâmica de páginas *Web*.
- Especificação de políticas de personalização.
- Especificação de operações de manipulação de dados para atualização do *site*.

2.3.2 *Interaction Flow Modeling Language (IFML)*

Com o grande crescimento de sistemas *Web*, houve a necessidade de criação de um novo padrão de modelagem que fosse mais adequado a tal tipo de sistema, uma vez que o padrão proposto pela *WebML* já não era suficientemente completo para a modelagem das novas formas no qual os usuários se interagem com o sistema. Foi então criada a *Interaction Flow Modeling Language (IFML)*, uma linguagem de modelagem que tem como proposta expressar o comportamento de conteúdo, interação com o usuário e controle de *front-end* de aplicações.

Casarin (2014) cita que a origem da *IFML* foi por conta de um grupo de pesquisadores tinha a ideia de se criar uma linguagem baseando na *WebML*. O projeto foi iniciado em 2011, no qual foram convidados para participar do projeto os pesquisadores Stefano Butti, Marcos Brambilla, Piero Fraternali, Emanuele Molteni, Matteo Sassi e Matteo Silva.

Conforme especificação da *IFML* na *Object Management Group (OMG)*¹, o objetivo da *IFML* é fornecer aos desenvolvedores, engenheiros e arquitetos de *software* ferramentas para definição de modelos de fluxo de interação que descrevem as principais dimensões de um *front-end* de aplicativo:

- A parte de exibição do aplicativo, feita de contêineres de exibição e exibição de componentes;
- Os objetos que incorporam o estado do aplicativo e as referências às ações de lógica de negócio que podem ser executadas;
- A ligação de componentes de visualização a objetos e eventos de dados;
- A lógica de controle que determina as ações a serem executadas após uma ocorrência de evento;

¹ Disponível em: <<https://www.omg.org>>

- A distribuição de controle, dados e lógica de negócios nas diferentes camadas da arquitetura.

Conforme descrito em Casarin (2014), os recursos tecnológicos, principalmente voltados para a *Web*, tem aumentado consideravelmente, trazendo sistemas cada vez mais personalizados e complexos. Casarin (2014) cita que segundo a *OMG*, o surgimento de diferentes dispositivos, plataformas e canais de comunicação não é acompanhado pelo surgimento de uma abordagem adequada para a criação de um *Platform-Independent Model* (*PIM*) que expresse as decisões de design de interação de forma independente de plataforma de implementação. Ainda conforme descrito em Casarin (2014), a *IFML* trás vários benefícios para o processo de desenvolvimento de aplicações *front-ends*, pois está particularmente atenta para modelagem de usabilidade e inteligibilidade, sendo abordados os fatores que tornam a *PIM* de fácil aprendizado, uso e aberto a extensibilidade.

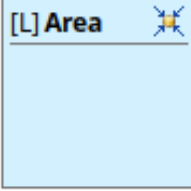
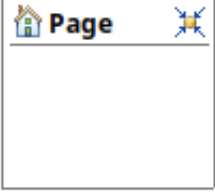
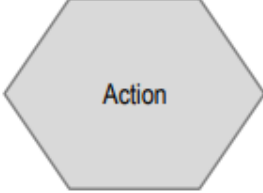
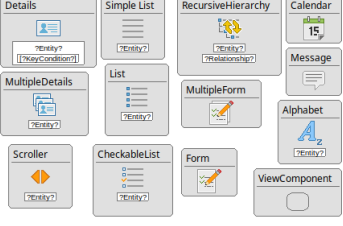
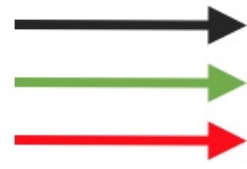
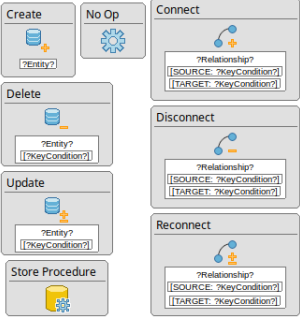
De acordo com Berre e Brambilla (2014), os seguintes aspectos são cobertos pela *IFML*:

- Múltiplas visualizações para um mesmo aplicativo
- Aplicativos móveis e multiplataforma.
- Visualização e entrada de dados, e produção de eventos
- Componentes independentes, *widgets* e apresentações
- Fluxo de interação
- Contexto do usuário
- Modularização do modelo
- Validação de entrada do usuário

2.3.3 Notações da *IFML*

Para a construção dos diagramas *IFML* voltados para sistemas *Web*, pode ser utilizada a ferramenta *WebRatio Web Platform*, ferramenta disponibilizada pela empresa *WebRatio*, a empresa que mantém a *IFML*. Conforme descrito em Arantes (2018), certas notações gráficas disponibilizadas pela ferramenta se diferem dos padrões originais da *IFML*. A Tabela 1 descreve algumas notações disponibilizadas pela ferramenta e que foram utilizadas na modelagem *front-end* efetuada neste trabalho, sendo tal modelagem apresentação na seção 3.3, do Capítulo 3.

Tabela 1 – Notações para modelagem *IFML*.

Componente	Significado	Notação
<i>Area</i>	<i>View Container</i> composto de <i>Pages</i> e <i>Areas</i> aninhadas para publicação ou gerenciamento de conteúdo homogêneo.	
<i>Page</i>	<i>View Container</i> que mostra componentes de exibição para o usuário.	
<i>Action</i>	Uma parte da lógica de negócios acionada por um evento, podendo ser do lado do servidor ou do cliente.	
<i>View Components</i>	Elemento da interface que exibe conteúdo ou serve para a entrada de dados.	
<i>Flows</i>	Fluxos de navegação, no qual indicam que quando o usuário pressiona um elemento da lista, o componente visado pelo fluxo é executado.	
<i>Operations</i>	Representação de operações sobre o modelo de dados.	

Fonte: Adaptado de WebRatio (2018).

3 Desenvolvimento e Resultados

Este capítulo apresenta as abordagens e tecnologias utilizadas para a especificação, projeto e desenvolvimento das funcionalidades, no qual foram utilizadas as mesmas metodologias do trabalho de Arantes (2018), no qual foi desenvolvida a versão inicial do projeto SisGera, e este trabalho é uma continuidade de tal projeto.

3.1 Escolha tecnológica

Esta seção apresenta as tecnologias escolhidas para o desenvolvimento das novas funcionalidades do projeto SisGera.

3.1.1 MySQL

O *MySQL* é um Sistema de Gerenciamento de Banco de Dados (SGBD) relacional de código aberto, desenvolvido pela empresa sueca *MySQL AB* em 1995¹. O sistema trabalha com a linguagem *Structured Query Language* (SQL), que atualmente é a linguagem mais popular para gerir conteúdo armazenado em um banco de dados.

MySQL foi desenvolvido em C e C++, e isso faz com que ele seja compatível com a maioria dos sistemas operacionais oferecidos atualmente no mercado. Ele é desenvolvido e distribuído pelas licenças *GNU's Not Unix (GNU)/General Public Licence (GPL)*. Podemos descrever as seguintes características do *MySQL*²:

- **Portabilidade:** como o *MySQL* foi desenvolvido em C e C++, é fácil a portabilidade entre diferentes sistemas, plataformas e compiladores, e também possui módulos de interface para múltiplas linguagens como *PHP*, *Delphi*, *Java*, *Python*, *ASP*, *Ruby*, entre outras.
- **Formas de armazenamento:** conta com diversos tipos de tabela para armazenamento de dados, no qual cada tipo pode priorizar características como velocidade, volume de dados, entre outras.
- **Velocidade:** conta com diferentes *engines* de tabelas como *MyISAM* e *InnoDB*, utilização de caches em consultas, indexação *BTREE*, algoritmos de busca, dentre outros recursos.

¹ Disponível em: <<https://goo.gl/K8hWGt>>

² Disponível em: <<https://goo.gl/8ferpB>>

- **Capacidade:** em tabelas com *engine InnoDB*, no qual o armazenamento pode ser realizado em um ou vários arquivos separados, é possível armazenar volumes de dados equivalente a *TB (TeraBytes)* de tamanho. O *MySQL* suporta execuções de scripts *Structured Query Language (SQL)* com até 61 milhões de tabelas *joins*, além de ter a capacidade de fazer o processamento de milhões de transações por minuto.

3.1.2 *Bootstrap*

O *Bootstrap* é um *framework* de código aberto para desenvolvimento *front-end* em aplicações *Web*, utilizando *Hypertext Markup Language (HTML)*, *Cascading Style Sheets (CSS)* e *JavaScript*³. O *framework* foi desenvolvido em 2011 por Jacob Thorton e Mark Otto, engenheiros do *Twitter*, que tinham como objetivo otimizar o desenvolvimento do *Twitter* através da adoção de uma estrutura única, e pelo grande potencial da ferramenta, eles a disponibilizaram no *Github* como um *software* livre⁴. O *Bootstrap* conta com um poderoso sistema de *grid*, possibilitando a criação de páginas responsivas e diversos componentes personalizados que podem ser utilizados em aplicações *Web*.

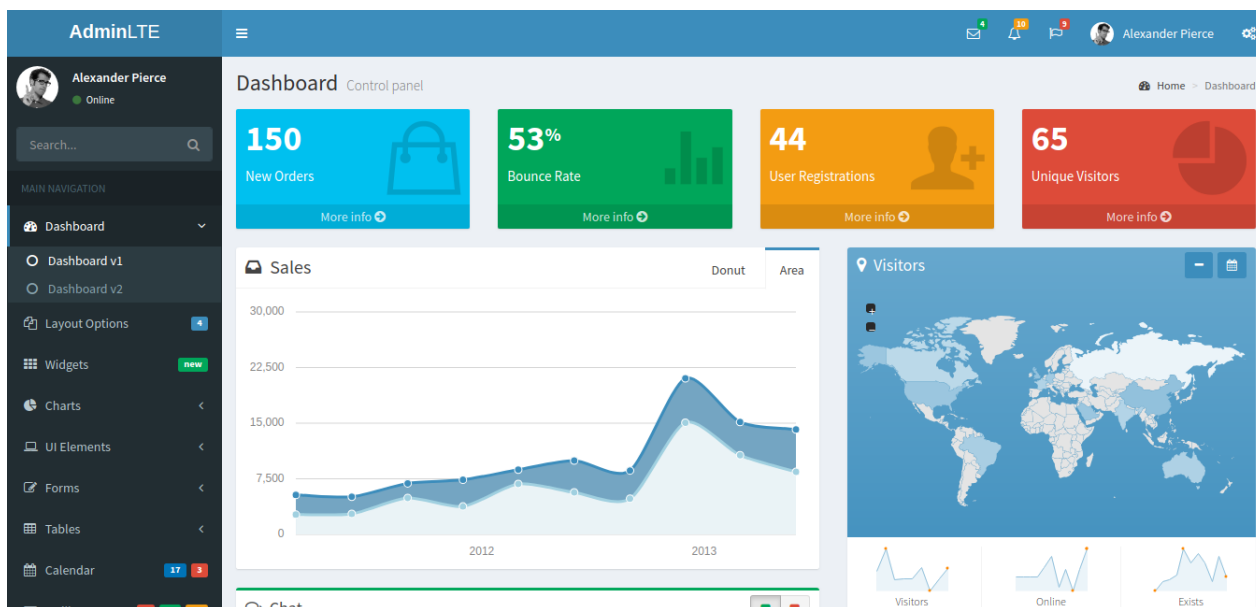
3.1.2.1 AdminLTE

O *AdminLTE* é um *template* para aplicações *Web* de código aberto que auxilia na construção de painéis de administração e controle⁵. Foi desenvolvido por Abdullah Almsaeed, desenvolvedor *Web* e pesquisador da *University of Tennessee* em Knoxville, EUA. O *template* é responsivo e baseado no *Bootstrap 3*, utilizando assim todos os componentes do *Bootstrap* em seu design. O *AdminLTE* é fácil de se utilizar e customizar, por conta de ser baseado em um design modular, e tem um design consistente por conta de redesenhar muito *plugins* usados com frequência. A [Figura 7](#) é um exemplo do painel de controle do *AdminLTE*.

³ Disponível em: <<https://getbootstrap.com>>

⁴ Disponível em: <<https://goo.gl/ubjzvz>>

⁵ Disponível em: <<https://goo.gl/bUx62J>>

Figura 7 – Painel de Controle do *AdminLTE*

Fonte: [AdminLTE](#) (2018)

3.1.3 *PHP Hypertext Preprocessor (PHP)*

O *PHP Hypertext Preprocessor (PHP)* é uma linguagem interpretada de código aberto que é usada para desenvolvimento de aplicações *Web*. Criado em 1994 por Rasmus Lerdof, inicialmente era apenas um *Common Gateway Interface (CGI)* escrito em C no qual o Rasmus tinha como objetivo facilitar o controle e acompanhamento de visitas em seu currículo *on-line*⁶. A linguagem conta com diversos benefícios, dentre os quais podem ser destacados os seguintes:

- Curva de aprendizado: na construção do *PHP* foram utilizados elementos provenientes das linguagens *Perl*, *C* e *Java*, fazendo com que programadores que já possuem conhecimento em tais linguagens tenham maior familiaridade com o *PHP*, tornando assim sua curva de aprendizado mais baixa comparada a outras linguagens de programação⁷.
- Robustez: sua versão atual trouxe melhorias de performance, no qual é possível executar várias funções sem alterar a velocidade do servidor⁸.

⁶ Disponível em: <<https://goo.gl/qvHxMN>>

⁷ Disponível em: <<https://goo.gl/k7YYBi>>

⁸ Disponível em: <<https://goo.gl/tuUAm7>>

- Código aberto: por se tratar de uma linguagem de código aberto, são feitas melhorias constantes por qualquer programador, onde podem ser corrigidos *bugs* ou o acréscimo de novas funções.
- Multiplataforma: conta com versões que rodam em diversos sistemas operacionais, como distribuições do *Linux* e *Windows*, ou até mesmo *Solaris*, *IRIX*, *AIX*, *RISC OS*, *Novell Netware*, *AS/400*, *OS/2*, *Mac OS* e *FreeBSD*.
- Compatibilidade com bancos de dados: é compatível com inúmeros bancos de dados, como *Oracle*, *Sybase*, *PostgreSQL*, *InterBase*, *MySQL*, *SQLite* e *MSSQL*, não se limitando apenas a estes.
- Comunidade: conta com uma das comunidades mais ativas dentre as linguagens mais populares. Também possui uma série de bibliotecas e pacotes disponibilizados para uso.
- Hosting: é possível encontrar hospedagem para sistemas *PHP* em praticamente todas as empresas que oferecem serviços de hospedagem, sendo que muitas dessas empresas inclusive possuem o *PHP* instalado de forma nativa em seus servidores.

3.1.4 *Laravel*

O *Laravel* é um *framework PHP* de código aberto para desenvolvimento *Web* no qual utiliza o padrão *Model-View-Controller (MVC)*. Atualmente é um dos *frameworks* mais populares, tendo grande aceitação no mercado, e conta com uma equipe de desenvolvedores competentes e uma grande comunidade. Os seguintes pontos podem ser destacados para justificar a utilização do *Laravel* como *framework*⁹:

- *Composer*: o *Laravel* utiliza o *Composer* como gerenciador de dependências, permitindo assim gerenciar pacotes de terceiros de forma fácil na aplicação.
- Documentação: a documentação disponibilizada pelo *Laravel* é bastante completa e simplista, possibilitando o aprofundamento nos recursos oferecidos pelo *framework*.
- Sistema de rotas: o *Laravel* conta com um poderoso sistema de rotas, sendo possível registrar rotas que respondem a qualquer verbo *Hypertext Transfer Protocol (HTTP)* e permite trabalhar com parâmetro de rotas, restrição de parâmetros com expressões regulares, agrupamento de rotas, além de outros recursos mais avançados.
- *Blade* - Sistema de *templates*: o *Laravel* conta com o *Blade*, o compilador de *templates* do *framework*, no qual ele se destaca de outros *templates* por conta de sua flexibilidade, por não restringir o uso de *PHP* puro misturado a sintaxe do *template*. E tem como

⁹ Disponível em: <<https://goo.gl/CrU1QG>>

principais benefícios a herança e seções, permitindo assim trabalhar com o conceito *master page* de forma fácil.

- *Eloquent* e *QueryBuilder*: o *Laravel* conta com o *Eloquent* como *ORM* padrão do *framework*. O *Eloquent* aplica o padrão de projeto *Active Record*, no qual cada tabela do banco de dados é representada por uma classe *model* que mapeia tal tabela, podendo assim realizar operações na aplicação sem utilizar *SQL* puro. E caso não se queira utilizar o nível de abstração oferecido pelo *Eloquent*, é possível utilizar o *QueryBuilder* como construtor de *queries*, que trata as *queries* num nível de abstração menor, e com ele é possível realizar a maioria das operações do banco de dados.
- *Artisan Console*: o *Laravel* conta com o *Artisan* que é uma interface de linha de comando que oferece diversos comandos que auxiliam no desenvolvimento da aplicação, e também é possível realizar a criação dos próprios comandos.

3.2 Requisitos

Esta seção apresenta a estratégia adotada no trabalho para especificação dos requisitos, levantados a partir de reuniões com os *stakeholders* do projeto.

3.2.1 Histórias de usuário

A especificação dos requisitos das novas funcionalidades do projeto SisGera foi realizada utilizando a abordagem de histórias de usuário. Para levantamento das histórias de usuário, foi utilizada a técnica de entrevista descrita em [Cohn \(2004\)](#) e para a representação das histórias foi utilizada a mesma estrutura citada em [Longo e Silva \(2014\)](#), sendo a seguinte:

- **Como um...** (Quem) (Papel, ator)
- **eu quero...** (O que) (Funcionalidade a ser desenvolvida)
- **de modo que...** (Por que) (Benefício a ser obtido)

A [Tabela 2](#) exemplifica duas histórias de usuário das novas funcionalidades do SisGera e no [Apêndice A](#) são apresentadas todas as histórias de usuário geradas para as novas funcionalidades do SisGera.

Tabela 2 – Histórias de usuário do SisGera.

Cadastrar Produto	Gerar B.O.
<ul style="list-style-type: none"> • Como um usuário interno, • eu quero cadastrar um novo produto, • de modo que seja possível realizar o controle de estoque de tal produto. 	<ul style="list-style-type: none"> • Como um usuário externo, • eu quero gerar o boletim de ocorrência da vítima, • de modo que não precise ir até a corporação para pegar o boletim de ocorrência.

Fonte: Elaboradora pelo autor (2018)

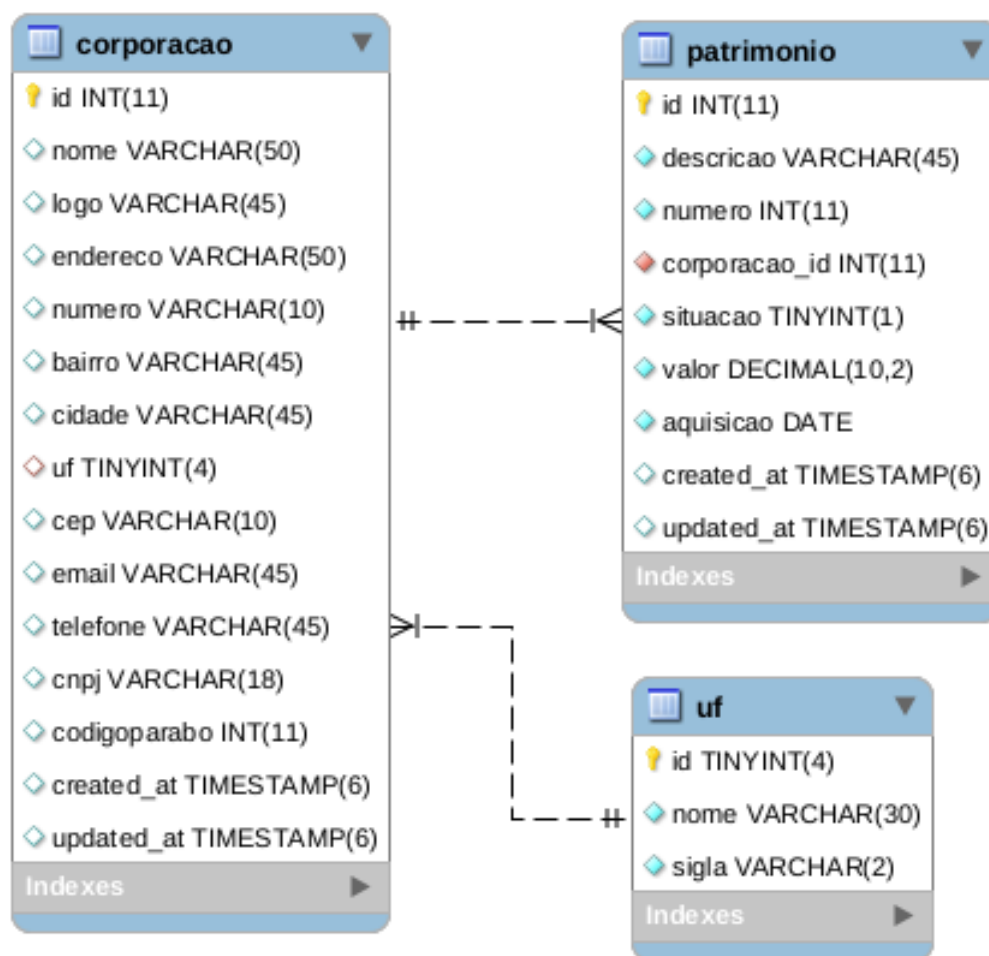
3.3 Projeto

Esta seção apresenta as abordagens para modelagem das camadas de *Model-View-Controller* do sistema, no qual são explicitados os modelos desenvolvidos no trabalho.

3.3.1 Entidade-Relacionamento (ER)

Seguindo a mesma tecnologia de banco de dados utilizado na versão inicial do SisGera desenvolvida por Arantes (2018), foi utilizado o banco de dados relacional *MySQL*. Para realizar o projeto do modelo de banco de dados, foram utilizados diagramas ER, sendo que os diagramas das novas funcionalidades do SisGera foram separados por funcionalidade de modo a facilitar a visualização. A Figura 8 demonstra o diagrama ER da aplicação de controle de bens patrimoniais, sendo que este diagrama e os demais podem ser visualizados na seção seção B.1 do Apêndice B.

Figura 8 – Diagrama ER da funcionalidade de controle de bens patrimoniais



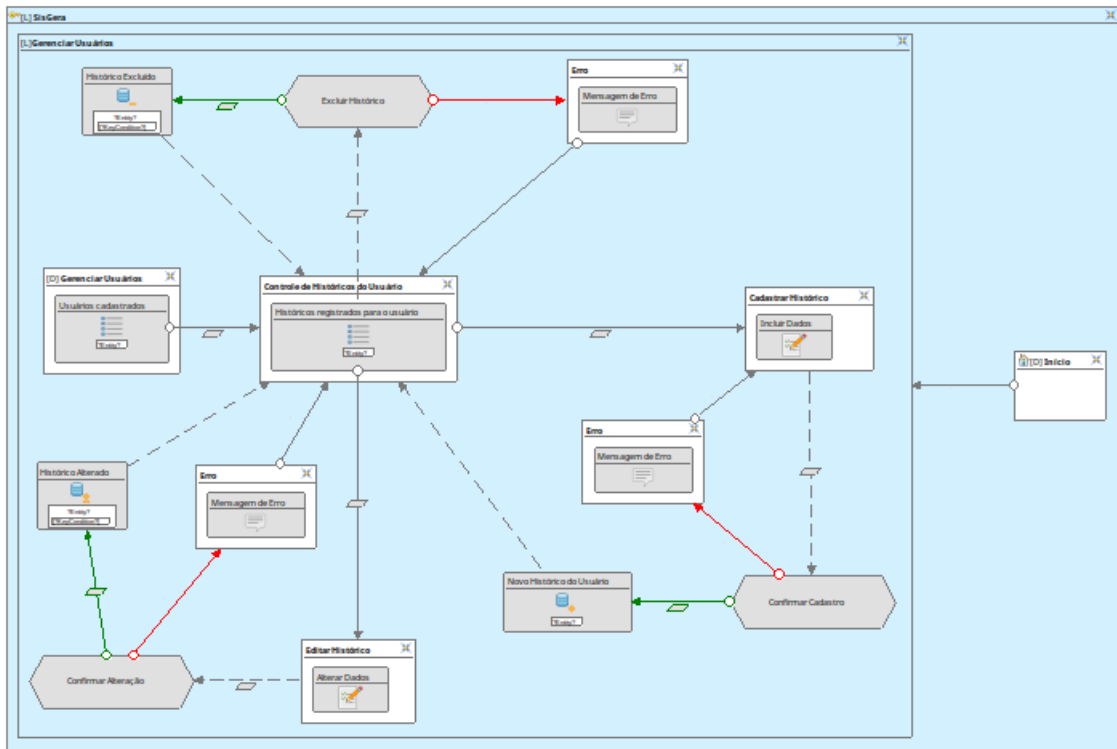
Fonte: Elaborado pelo autor (2018)

3.3.2 Interaction Flow Modeling Language (IFML)

Para modelar a camada de views das novas funcionalidades do SisGera, foi utilizado o modelo *IFML*, no qual foram gerados diagramas com o auxílio da ferramenta *WebRatio Web Platform*. Os diagramas foram gerados para cada funcionalidade, facilitando assim a visualização.

A Figura 9 demonstra a funcionalidade de históricos do usuário, no qual a mesma é acessada a partir de um *link* no menu Gerenciar Usuários, onde usuário tem a possibilidade de adicionar um novo histórico, alterar ou excluir um histórico existente. A Área Gerenciar Usuários está identificada com o tipo [L] *Landmark*, a qual é acessada por um menu geral, que por sua vez é acessado pela Page Início, identificada como uma *Home Page* na Área SisGera. Os demais diagramas *IFML* poderão ser visualizados na seção B.2 do Apêndice B.

Figura 9 – Diagrama *IFML* da funcionalidade de histórico do usuário

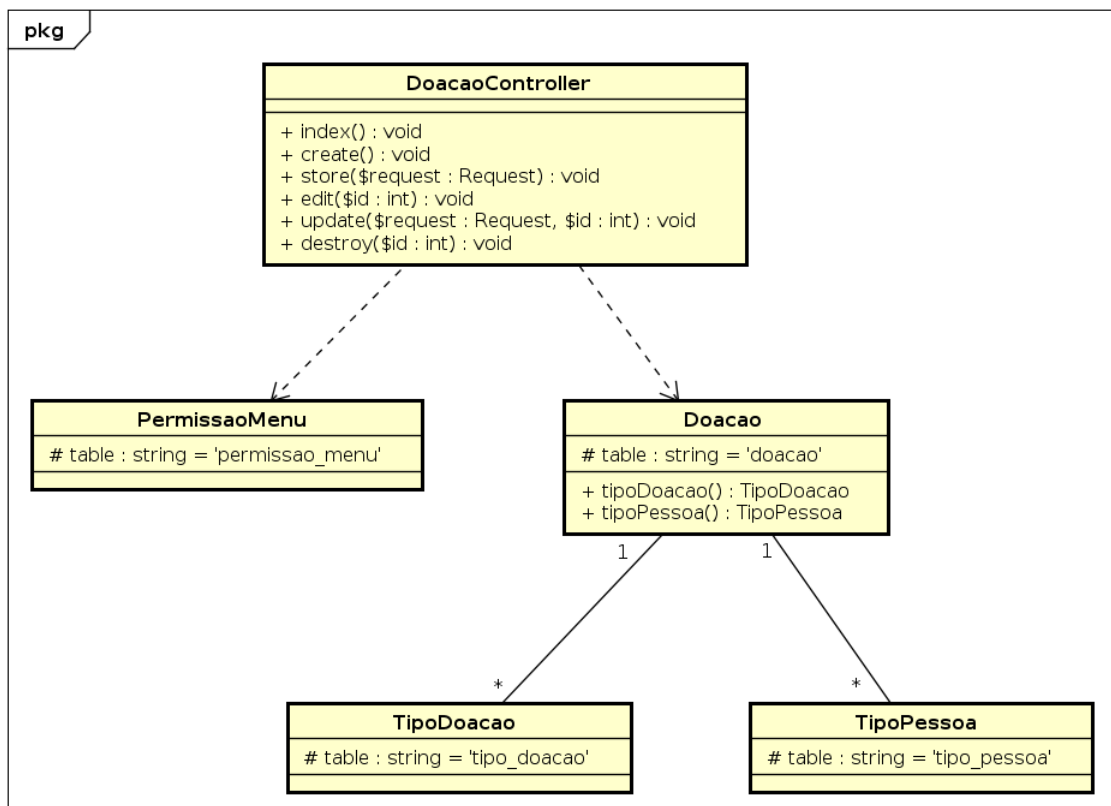


Fonte: Elaborado pelo autor (2018)

3.3.3 Diagrama de Classes

Para a modelagem da camada de controllers no projeto das novas funcionalidades do SisGera, foram utilizados diagramas de classe da linguagem *UML*. A [Figura 10](#) representa o diagrama de classe da aplicação de Controle de Doações do SisGera, e os diagramas de classe das demais funcionalidades estão disponíveis na seção [seção B.3](#) do [Apêndice B](#).

Figura 10 – Diagrama de classe da funcionalidade de controle de doações



Fonte: Elaborado pelo autor (2018)

3.4 Testes

Esta seção apresenta as estratégias de teste utilizadas no trabalho, sendo utilizadas técnicas de caixa-branca e caixa-preta. Logo a seguir são apresentadas as abordagens utilizadas para validação das novas funcionalidades do SisGera.

3.4.1 Teste de Unidade

Os testes de unidades foram implementados com a utilização do *PHPUnit*, que é um *framework* de testes de unidade para a linguagem *PHP* que foi criado por Sebastian Bergmann com base na arquitetura *XUnit* para *frameworks* de testes de unidade, conforme descrito em Bergmann (2018). Embora existam outros *frameworks* de testes de unidade para *PHP*, o *PHPUnit* foi o escolhido para a realização dos testes das novas aplicações do SisGera, pois os *frameworks PHP* mais populares, incluindo o *Laravel*, disponibilizam testes do *PHPUnit*.

Foram criados casos de teste a partir dos métodos das aplicações, certificando assim o funcionamento das partes do sistema de forma isolada, conforme é o foco dos testes de unidade. Para a realização de testes de unidade das novas funcionalidades do SisGera, foram criadas asserções que testam o funcionamento dos métodos de cada aplicação. A [Figura 11](#) retrata dois métodos de teste criados para a aplicação de movimentação de estoque, no qual o primeiro método *testCreateMovimentoEstoqueDataBase* realiza a inclusão de um registro diretamente no banco de dados e verifica se o registro existe no banco de dados a partir da asserção *assertDatabaseHas*, já o método *testCreateMovimentoEstoqueApp* realiza a criação de um novo registro a partir da aplicação e depois verifica se a página será redirecionada para a rota `/listarMovimentos`, que exibe todas as movimentações de estoque após a inclusão de uma nova movimentação. Para definir os dados a serem utilizados nos testes, foram utilizadas *factories* do *Laravel*, que é um recurso que permite criar dados falsos para a realização de testes com o banco de dados¹⁰.

¹⁰ Disponível em: <<https://goo.gl/zwTVbV>>

Figura 11 – Exemplos de testes criados para a funcionalidade de controle de estoque.

```
<?php
namespace Tests\Unit;

use Tests\TestCase;
use Illuminate\Foundation\Testing\DatabaseTransactions;
use App\MovimentoEstoque;
use App\User;

class MovimentoEstoqueTest extends TestCase
{
    use DatabaseTransactions;

    /**
     * @test
     */
    public function testCreateMovimentoEstoqueDataBase()
    {
        $movimento = factory(MovimentoEstoque::class)->create();
        $this->assertDatabaseHas('movimento_estoque', ['id' => $movimento['id']]);
    }

    /**
     * @test
     */
    public function testCreateMovimentoEstoqueApp()
    {
        $user = User::find(1);
        $movimento = factory(MovimentoEstoque::class)->make();
        $request = $this->actingAs($user)->post('estoque', [
            'tipo_movto' => $movimento['tipo_movimento'],
            'qtde' => $movimento['quantidade'],
            'produto' => $movimento['produto_id'],
            'justificativa' => $movimento['justificativa']
        ]);
        $request->assertRedirect('/listarMovimentos');
    }
}
```

Fonte: Elaborado pelo autor (2018)

3.4.2 Teste Funcional

O teste funcional foi implementado com a utilização do *Katalon Recorder*, uma ferramenta para automação de testes criada pela empresa *KMS Technology*, sendo disponibilizada como *plugin* nos navegadores *Google Chrome* e *Mozilla Firefox*¹¹, sendo uma alternativa após a *Selenium IDE* não dar continuidade para versões a partir da 55 do *Mozilla Firefox*¹².

A geração dos casos de teste no *Katalon Recorder* se mostrou bem simples. Nessa ferramenta, podem ser gerados testes de forma manual ou feita a gravação de eventos realizados pelo usuário no navegador, gerando assim os casos de teste. A ferramenta

¹¹ Disponível em: <<https://goo.gl/hqPJ2i>>

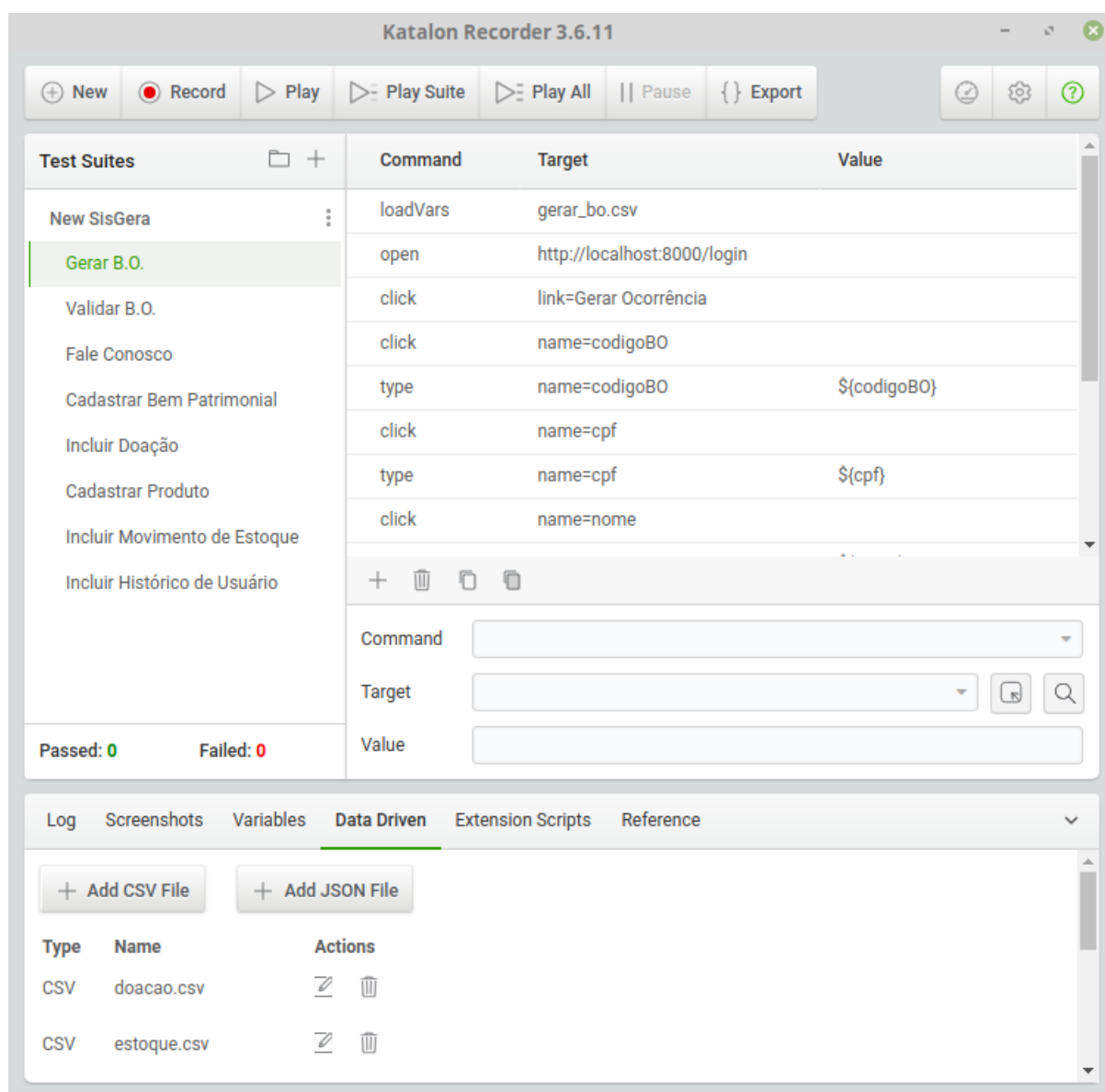
¹² Disponível em: <<https://goo.gl/G5CnLP>>

também disponibiliza a importação de arquivos *.CSV* ou *.JSON* para a execução dos casos de teste, sendo uma opção para quem deseja realizar testes em massa sem ter que se preocupar com a entrada de dados.

Nos testes realizados para as novas funcionalidades do SisGera, foram criados arquivos *.CSV* com dados aleatórios na ferramenta *Mockaroo*, na qual são gerados dados de teste realistas, sendo subdivididos em diversas categorias disponibilizadas pela ferramenta¹³. Foram gerados casos de teste em todas as novas funcionalidades do SisGera que contam com a entrada de dados, avaliando assim o comportamento das aplicações mediante a uma grande massa de dados, de diferentes tipos. A [Figura 12](#) demonstra o caso de teste criado para a funcionalidade de geração do B.O. pelo usuário externo, no qual é apresentada uma coluna com a suíte de casos de testes criados para as novas funcionalidades do projeto SisGera, e no outro lado são demonstradas as ações realizadas no teste, onde primeiro é carregado o arquivo *.CSV* com os dados fictícios gerados pelo *Mockaroo*, posteriormente é aberta a página de *login*, um clique é efetuado no botão Gerar Ocorrência, um clique é efetuado no campo para inserção do código do B.O., entre outras ações, realizando assim todo passo a passo de ações efetuadas pelo usuário.

¹³ Disponível em: <<https://www.mockaroo.com/>>

Figura 12 – Caso de teste criado para a funcionalidade de geração do B.O. pelo usuário externo no SisGera.



Fonte: Elaborado pelo autor (2018)

3.5 Apresentação das funcionalidades

Esta seção apresenta as novas aplicações desenvolvidas para o projeto SisGera, onde é feita uma breve descrição de cada uma delas, bem como um exemplo visual da tela principal de cada uma das aplicações.

O projeto SisGera iniciado por Arantes (2018) e complementado neste trabalho, foi disponibilizado para utilização das corporações, sendo a hospedagem feita em um servidor

comercial e adquirida de forma gratuita, podendo ser acessado a partir do endereço <www.sisgera.com.br>. Logo a seguir, são apresentadas as novas funcionalidades.


3.5.1 Geração e validação do boletim de ocorrência usuário externo

Usuários externos podem gerar o boletim de ocorrência de determinada vítima, sendo necessário a confirmação de dados do código do B.O. e CPF da vítima, e também informar o nome e o *email* do usuário que está gerando o boletim. Para acesso a tal funcionalidade, foi disponibilizado um botão na tela inicial, que caso seja acionado, o usuário será redirecionado para o ambiente que solicita os dados para geração do B.O.

O boletim de ocorrência gerado pelo usuário externo contém um código com 8 caracteres alfanuméricos, que poderá ser utilizado para autenticar se aquele documento é verdadeiro. Para autenticar a validade do documento, foi disponibilizado um botão também na tela inicial que redireciona o usuário para uma tela em que o usuário deverá informar o código do B.O. e o código para autenticação presente no corpo do B.O. gerado pelo usuário.

A [Figura 13](#) demonstra a tela Gerar Ocorrência, no qual o usuário externo pode gerar uma determinada ocorrência, a [Figura 14](#) demonstra um B.O. simplificado gerado pelo usuário externo com a chave de autenticidade, e a [Figura 15](#) demonstra a tela Validar Ocorrência, no qual pode ser atestada a autenticidade de algum B.O. gerado pelo SisGera.

Figura 13 – Tela Gerar Ocorrência.



A imagem mostra a interface de usuário para a geração de um boletim de ocorrência. No topo, há dois logotipos: o do Corpo de Bombeiros Militar do Estado de São Paulo e o do SisGera. Abaixo dos logotipos, o texto "SisGera" e "Sistema de Gerenciamento e Registro de Atividades" é exibido. O formulário principal, intitulado "Geração da Cópia do Boletim de Ocorrência", contém quatro campos de entrada obrigatórios: "Código da Ocorrência:", "CPF da Vítima:", "Nome do Usuário:" (com o placeholder "Digite aqui o seu nome") e "Email do Usuário:" (com o placeholder "Digite aqui o seu email"). Na base do formulário, há dois botões vermelhos: "Voltar" e "Confirmar".

Fonte: Elaborado pelo autor (2018)

Figura 14 – B.O. simplificado com a chave de autenticidade gerado por um usuário externo.



Associação de Bombeiros Voluntários - São Domingos do Prata
Av. Min. Paulino Cícero de Vasconcelos, Nº 500, bairro Dona Julieta - São Domingos do Prata - Minas Gerais
Telefone: (31)38561543 - E-mail: bombeirosdoprata@gmail.com

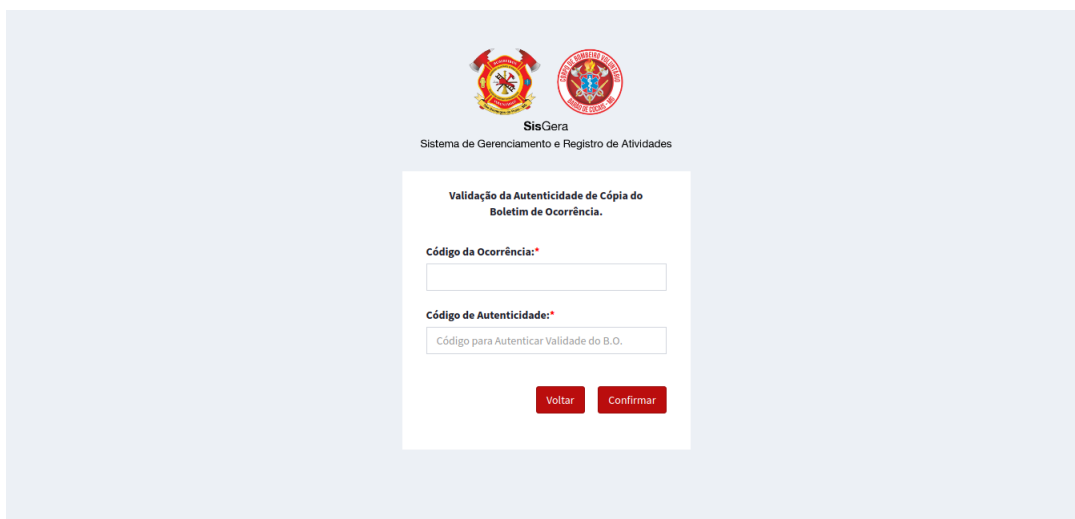
Dados do atendimento	
Código do B.O.: 0018/60-2018	Responsável pela ocorrência: Silvano Sergio Martins Oliveira
Data da ocorrência: 08/10/2018	Meio de acionamento: PRF
Transmissão: 18:39	Saída do local: 19:02
Chegada ao local: 18:53	Liberação da viatura: 19:10
Equipe de plantão: Bráulio Henrique de Vasconcelos Perdigão - Carmem Lucia de Freitas -	
Dados da vítima	
Nome da vítima: Fulano da Silva Filho	Gênero: Masculino
Data de nascimento: 11/11/1911	Idade: 107 anos, 0 meses, 17 dias
CPF: 123.456.789-01	Identidade: 123456789
Telefone de contato: (31)99999-9999	Cartão SUS: 123456789
Mãe: Fulana da Silva	Paí: Fulano da Silva
Endereço: Rua A	Número: 10
Complemento: casa	Cidade: São Domingos do Prata
Bairro: Centro	CEP: Não especificado
Estado: Minas Gerais	
Dados da ocorrência	
Local da ocorrência: Não especificado	Ponto de referência:
Endereço:	Número:
Complemento:	Cidade:
Bairro:	CEP:
Estado: Minas Gerais	
Tipo de ocorrência	
Eventos da defesa civil	
Outras observações	
Resumo da ocorrência: Teste na geração de B.O. Simplificado.	

Boletim de Ocorrência emitido às 19:39 de 16/11/2018

A autenticidade deste documento poderá ser confirmada na página do SisGera na Internet,
no endereço www.sisgera.com.br/validarBO por meio do código **1afr3w2x**

Fonte: Elaborado pelo autor (2018)

Figura 15 – Tela Validar Ocorrência.

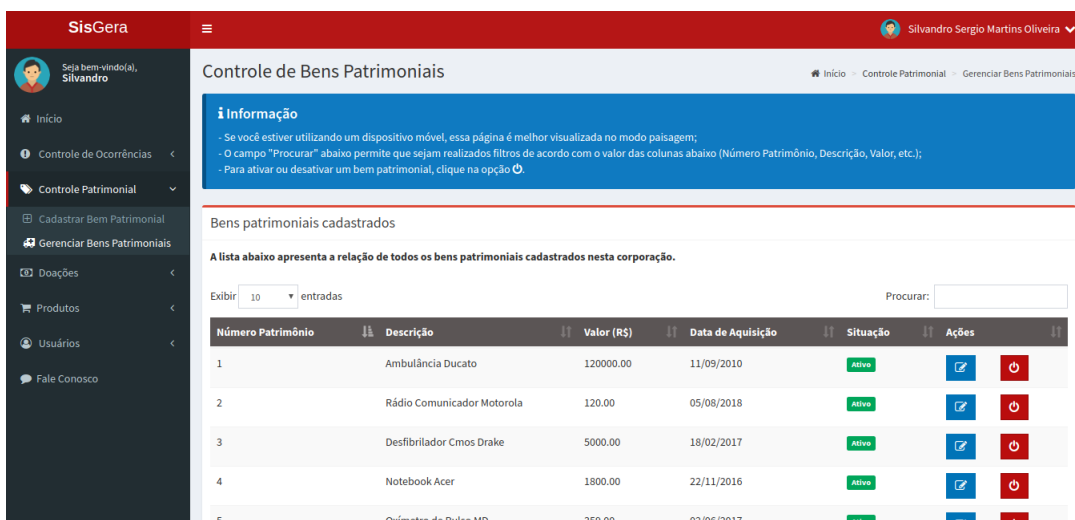


Fonte: Elaborado pelo autor (2018)

3.5.2 Controle de bens patrimoniais

A funcionalidade para controle de bens patrimoniais tem como objetivo manter o cadastro de todos os ativos imobilizados da corporação, na qual o usuário tem a possibilidade de cadastrar novos bens patrimoniais e gerenciar bens patrimoniais já cadastrados. A Figura 16 demonstra a tela Gerenciar Bens Patrimoniais no SisGera.

Figura 16 – Tela Gerenciar Bens Patrimoniais.

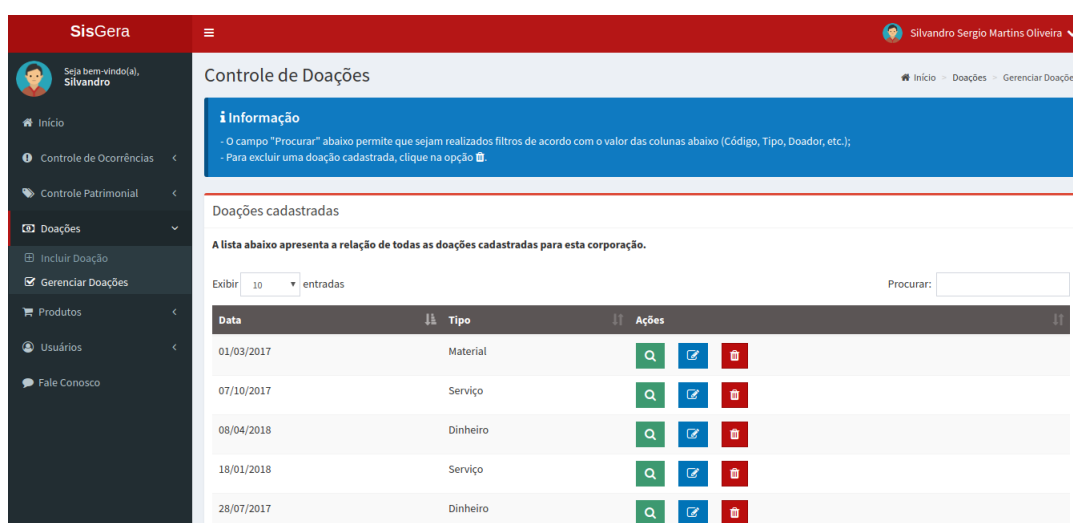


Fonte: Elaborado pelo autor (2018)

3.5.3 Controle de doações

A funcionalidade para controle de doações tem como objetivo manter o cadastro de todas as doações recebidas pela corporação, na qual o usuário tem a possibilidade de incluir uma nova doação e gerenciar doações já cadastradas. A Figura 17 exibe a tela Gerenciar Doações no SisGera.

Figura 17 – Tela Gerenciar Doações.



Data	Tipo	Ações
01/03/2017	Material	[Search] [Edit] [Delete]
07/10/2017	Serviço	[Search] [Edit] [Delete]
08/04/2018	Dinheiro	[Search] [Edit] [Delete]
18/01/2018	Serviço	[Search] [Edit] [Delete]
28/07/2017	Dinheiro	[Search] [Edit] [Delete]

Fonte: Elaborado pelo autor (2018)

3.5.4 Cadastro de produtos

A funcionalidade para cadastro de produtos tem como objetivo manter o cadastro dos materiais de consumo utilizados e mantidos pela corporação, na qual o usuário tem a possibilidade de incluir um novo produto ou gerenciar produtos já cadastrados. A Figura 18 demonstra a tela Gerenciar Produtos no SisGera.

Figura 18 – Tela Gerenciar Produtos.

The screenshot displays the 'Gerenciar Produtos' interface in SisGera. It features a sidebar menu on the left and a main content area. The main content area includes an 'Informação' box with instructions for mobile usage, a 'Produtos cadastrados' section with a search bar, and a table of products.

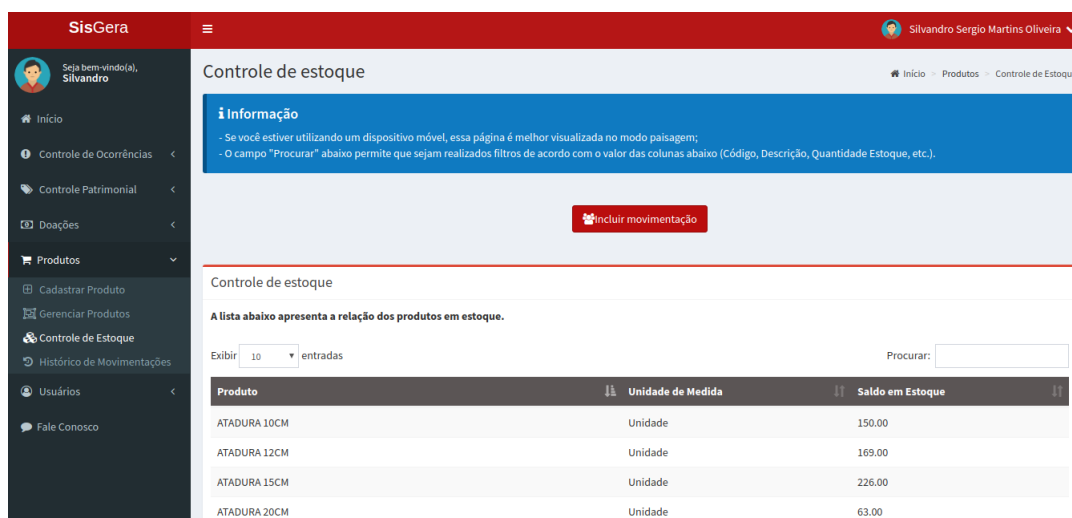
Produto	Preço Unitário (R\$)	Unidade de Medida	Estoque Mínimo	Situação	Ações
ATADURA 10CM	1.00	Unidade	20.00	Ativo	[Editar] [Desativar] [Excluir]
ATADURA 12CM	1.28	Unidade	20.00	Ativo	[Editar] [Desativar] [Excluir]
ATADURA 15CM	1.31	Unidade	20.00	Ativo	[Editar] [Desativar] [Excluir]

Fonte: Elaborado pelo autor (2018)

3.5.5 Controle de estoque

A funcionalidade para controle de estoque tem como objetivo cadastrar e demonstrar as movimentações de estoque realizadas pela corporação, bem como controlar os itens em estoque da corporação, sendo que o usuário tem como possibilidade a inclusão de movimentos de estoque e gerenciamento dos movimentos cadastrados. A Figura 19 exemplifica a tela Controle de Estoque no SisGera.

Figura 19 – Tela Controle de Estoque.

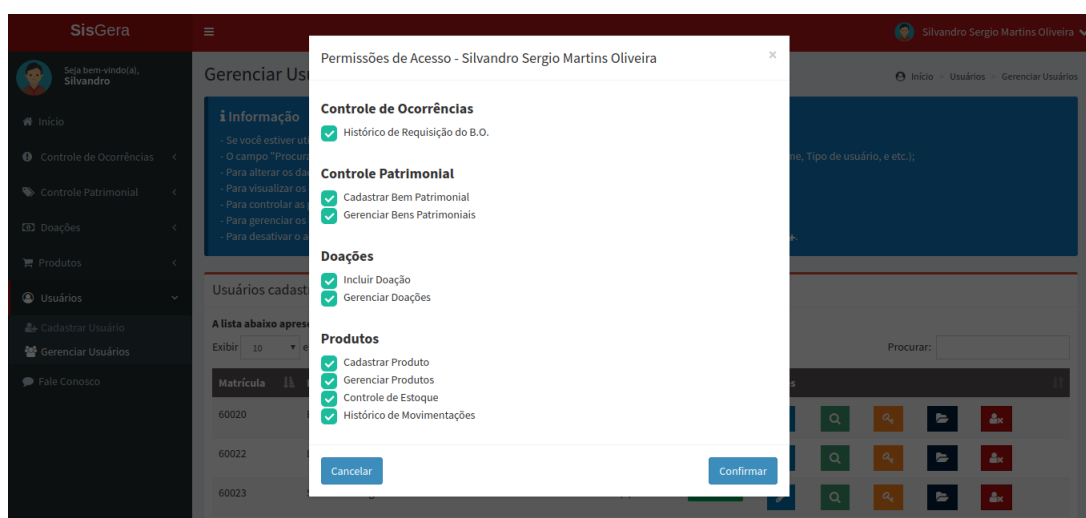


Fonte: Elaborado pelo autor (2018)

3.5.6 Controle de permissões do usuário

A funcionalidade para controle de permissões de acesso do usuário tem como objetivo limitar o acesso as novas funcionalidades do SisGera a determinados usuários, ficando a cargo do Coordenador Geral definir qual usuário terá acesso a determinada tela. A [Figura 20](#) demonstra como são cedidas as permissões de acesso aos usuários do SisGera.

Figura 20 – Tela Permissões de Acesso.

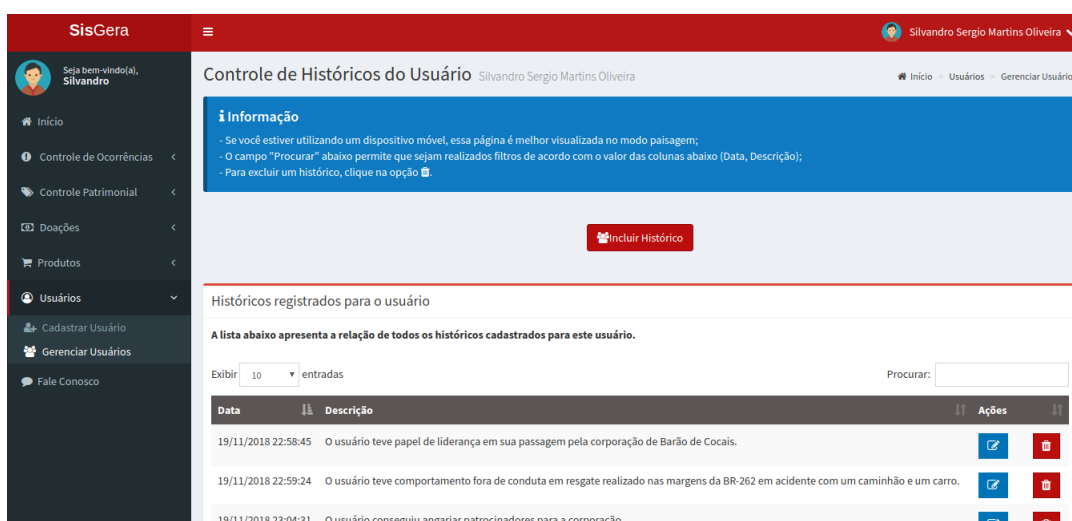


Fonte: Elaborado pelo autor (2018)

3.5.7 Controle de históricos do usuário

A funcionalidade para controle de históricos do usuário tem como objetivo manter um histórico comportamental do usuário na corporação atual e em outras corporações que o usuário tenha trabalhado, se for o caso. Na funcionalidade é possível realizar a inclusão de novo histórico para determinado usuário e o gerenciamento de históricos já cadastrados. A [Figura 21](#) demonstra a tela Controle de Históricos do Usuário no SisGera.

Figura 21 – Tela Controle de Históricos do Usuário.

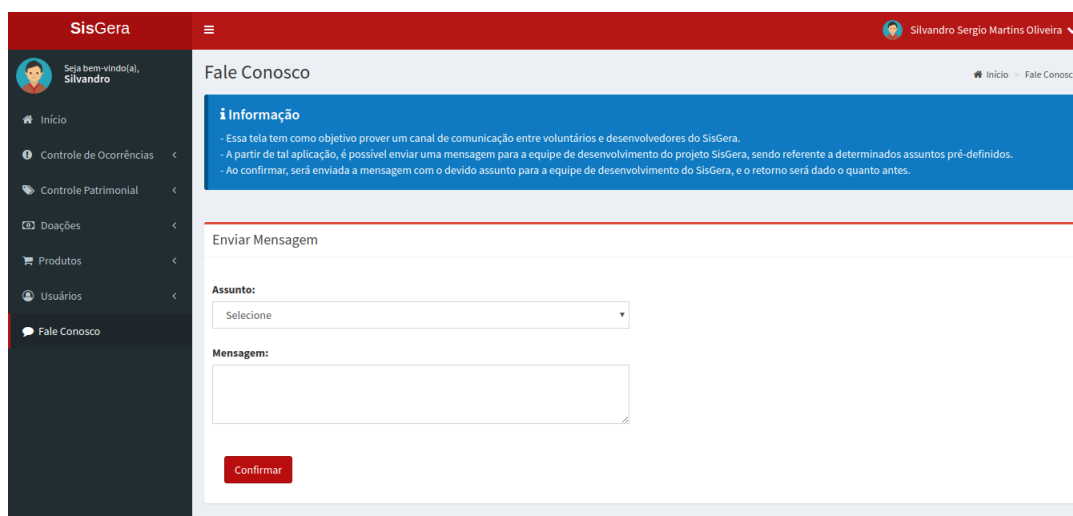


Fonte: Elaborado pelo autor (2018)

3.5.8 Fale Conosco

Com o objetivo de se ter um canal de comunicação mais formal entre os voluntários das corporações e a equipe de desenvolvimento do SisGera, foi criada a funcionalidade Fale Conosco, na qual os usuários do SisGera poderão enviar mensagens de assuntos pertinentes ao sistema, sendo que tais mensagens poderão ser visualizadas pela equipe de desenvolvimento do SisGera. A [Figura 22](#) exhibe a tela Fale Conosco no SisGera.

Figura 22 – Tela Fale Conosco.



Fonte: Elaborado pelo autor (2018)

3.6 Sistemas Similares

As funcionalidades desenvolvidas para o Sisgera no desenvolvimento deste trabalho podem ser obtidas em outros sistemas disponíveis no mercado, sendo assim, esta seção irá fazer uma breve demonstração de alguns sistemas.

3.6.1 Economato

O Economato é um *software* de gestão financeira de instituições do terceiro setor, no qual foi desenvolvido pela empresa bhbit de Belo Horizonte/MG e está disponível no mercado a mais de 10 anos. O *software* é totalmente *online* e disponibiliza um acesso gratuito por um período de 14 dias, sendo que após tal período a instituição terá de realizar um pagamento mensal para utilização do *software*.

O *software* busca solucionar a gestão financeira de instituições do terceiro setor a partir dos pilares de Organização das Finanças, Prestação de Contas e Planejamento Financeiro.

3.6.2 ANY3

O ANY3 é um sistema para gestão de recursos financeiros de instituições do terceiro setor, sendo desenvolvido pela empresa FP2 Tecnologia de Santa Maria/RS. O *software* conta com uma versão totalmente *online* e outra *desktop*, e conta atualmente com 6

módulos, como o módulo Contribuições que gerencia as doações, e o módulo Patrimônio que gerencia os bens patrimoniais.

O *software* conta com diferentes planos para ser adquirido, sendo que conta também com uma versão gratuita, porém tal versão gratuita tem recursos e validade limitados, o que não pode ser interessante para determinadas instituições. A Figura 23 demonstra a tela de pesquisa de patrimônios do ANY3.

Figura 23 – Funcionalidade de gerenciamento de patrimônios no ANY3

The screenshot shows the 'Patrimônio - Pesquisa' interface. At the top, there is a navigation bar with the ANY3 logo and 'Empresa Fictícia'. The main area contains a search form with the following fields:

- Projeto: Seleccione...
- Número da Etiqueta: [input field]
- Forma de Aquisição: Seleccione...
- Pessoa Responsável: Seleccione...
- Grupo de Tipo de Objeto: Seleccione...
- Tipo de Objeto: Seleccione...
- Descrição: [input field]
- Data da Aquisição (inicial): [input field]
- Data da Aquisição (final): [input field]

Below the form are buttons for 'Pesquisar' and 'Limpar'. The results table has the following structure:

Nr da Etiqueta	Data de Aquisição	Objeto	Descrição	Valor Aquisição R\$
Nenhum registro encontrado				

The URL at the bottom is <https://app.any3.com.br/Patrimonio#>.

Fonte: Any3 (2018)

3.6.3 HYB

O HYB é um *software* criado para instituições do terceiro setor, no qual disponibiliza funcionalidades para gestão de entidades sem fins lucrativos. O *software* foi desenvolvido pela empresa Conplan Sistemas de Santa Maria/RS, sendo uma ferramenta *Web* totalmente *online*, não sendo necessário a instituição contratante manter uma infraestrutura de servidores.

O *software* tem como objetivo proporcionar benefícios como transparência das operações, modernização da gestão, e organização de áreas como saúde, educação e assistência social. O *software* atualmente disponibiliza 23 tipos de recursos diferentes, como gestão de doações, patrimônio, almoxarifado, veículos, entre outros, porém nenhum deles é gratuito, sendo necessário realizar um pagamento mensal para adquirir a licença

de uso do *software*. A Figura 24 demonstra a funcionalidade de doações do HYB.

Figura 24 – Funcionalidade de gestão de doações do HYB

Doador	Centro de Resultados	Cobrança	Data de Vencimento/Pgto	Valor	Situação	Ação
ALTAISA DELMONDES DE LIMA	Contribuições Comunitaria		01/06/2018	600,00	Vencida	Detalhada
Doador de Exemplo	Receita C/Mensalidades		02/06/2018	500,00	Vencida	Detalhada
doador 1000	Doações Pessoa Juridica		05/06/2018	2.500,00	Paga	
Diogenes	doacoes pessoas fisicas		05/06/2018	12.333,33	Paga	
CENTRAL DE DOACOES	Contribuições Comunitaria		06/06/2018	1.000,00	Vencida	Detalhada
Diego Ribas Adiers	Contribuições Comunitaria		07/06/2018	10,00	Vencida	Detalhada
CENTRAL DE DOACOES	Receita C/Mensalidades		10/06/2018	100,00	Vencida	Detalhada
Doador de Exemplo	Doações Pessoa Juridica		11/06/2018	20,00	Vencida	Detalhada
Doador de Exemplo	Doações Pessoa Juridica		11/06/2018	25,00	Vencida	Detalhada
ALTAISA DELMONDES DE LIMA	doacoes pessoas fisicas		12/06/2018	100,00	Vencida	Detalhada
Doador testando	Contribuições Comunitaria		13/06/2018	100,00	Vencida	Detalhada
Nailson	doacoes pessoas fisicas		15/06/2018	50,00	Vencida	Detalhada
doador 1000	Contribuições Comunitaria		23/06/2018	100,00	Aguardando pagamento	Detalhada
CENTRAL DE DOACOES	Receitas Prestacional		24/06/2018	100,00	Aguardando pagamento	Detalhada
CENTRAL DE DOACOES	Contribuições Comunitaria		26/06/2018	100,00	Aguardando pagamento	Detalhada
CENTRAL DE DOACOES	doacoes pessoas fisicas		28/06/2018	100,00	Aguardando pagamento	Detalhada
				17.738,33		

Fonte: HYB (2018)

3.7 Considerações finais

Este capítulo apresentou as etapas da fase de desenvolvimento do trabalho, onde foram demonstradas as abordagens e tecnologias utilizadas para o projeto, desenvolvimento e validação das funcionalidades. Também foram mencionados sistemas correlatos disponíveis no mercado, sendo que a Tabela 3 demonstra um comparativo das funcionalidades de tais sistemas com as funcionalidades desenvolvidas neste trabalho. Nenhum dos sistemas correlatos apresentados é gratuito, sendo que disponibilizam apenas um período para testes de forma gratuita. Dos sistemas correlatos descritos, o HYB se mostrou mais completo em termos de funcionalidades, já o Economato é o mais simples deles, tendo foco apenas para operações financeiras. Em relação à funcionalidade e usabilidade, os sistemas correlatos se mostraram bem completos e com uma boa usabilidade, já o SisGera tem como vantagem ser um sistema personalizado conforme a realidade das corporações de Barão de Cocais e

São Domingos do Prata, oferecendo funcionalidades condizentes ao processo interno de tais corporações, além de ser gratuito.

Tabela 3 – Comparativo de funcionalidades entre o SisGera e os sistemas correlatos.

	SisGera	Economato	ANY3	HYB
Controle de Doações	X			X
Contabilidade			X	X
Financeiro		X	X	X
Cadastro de Produtos e Estoque	X			X
Controle de Permissões de Usuário	X			
Controle de Frota				X

Fonte – Elaborado pelo autor (2018)

4 Conclusão

Este trabalho apresentou o desenvolvimento de novas funcionalidades no projeto SisGera, sendo que tais funcionalidades visam o apoio a atividades administrativas das corporações. O sistema já foi implantado e a expectativa é que ele otimize e organize as operações administrativas das corporações, bem como traga comodidade as vítimas na geração do boletim de ocorrência sem necessidade de comparecer na sede da corporação. É esperado um grande impacto social positivo com o sistema, uma vez que o mesmo poderá ser usado por qualquer corporação de resgate e combate a incêndio. Com o desenvolvimento do trabalho também foi possível revisar importantes assuntos da literatura, bem como o aprendizado sobre a *IFML* que é uma boa proposta para modelagem da camada de *views* em aplicações *Web*, obtendo assim um enorme ganho acadêmico, pessoal e profissional.

O trabalho iniciou com uma revisão da literatura acerca dos assuntos pertinentes, como os Sistemas de Informações Gerenciais (*SIG*), o processo de desenvolvimento de *software* no qual foram abordados assuntos da engenharia de requisitos, projeto de *software* e teste de *software*, o desenvolvimento de sistemas *Web*, apresentando a *Interaction Flow Modeling Language (IFML)* que é uma nova proposta para modelagem e representação de interações de uma aplicação, e também apresentados sistemas correlatos voltados para o terceiro setor.

No desenvolvimento do trabalho, foram apresentadas as tecnologias e abordagens utilizadas para a criação das funcionalidades, bem como as técnicas utilizadas para modelagem das camadas de *Model-View-Controller* das aplicações, e a utilização de técnicas de testes de unidade e funcional para validação das funcionalidades desenvolvidas. Com base na análise realizada no desenvolvimento das funcionalidades, foram levantados alguns pontos como propostas para trabalhos futuros que são apresentados na seção 4.1.

4.1 Trabalhos futuros

Durante o desenvolvimento deste trabalho foram identificados alguns pontos para trabalhos futuros, sendo os seguintes:

- Testes e avaliações de usabilidade com os usuários do sistema;
- Desenvolvimento de funcionalidade para controle da frota de veículos da corporação;
- Criação de página *Web* para divulgação de informações das corporações como localização, prestação de contas, com integração a redes sociais como *Facebook*.

- Criação de página para recebimento de doações *online* integrada ao sistema.

Referências

- ADMINLTE. *AdminLTE2 - Dashboard*. [S.l.], 2018. Disponível em: <<https://goo.gl/89m4kC>>. Citado na página 35.
- ANY3. *Any3 - Sistema de Gestão para qualquer tipo de entidade do Terceiro Setor*. [S.l.], 2018. Disponível em: <<http://www.any3.com.br/>>. Citado na página 54.
- ARANTES, V. M. Um sistema de informação para apoio ao registro de ocorrências atendidas por grupos de bombeiros voluntários. João Monlevade, MG, 2018. Disponível em: <<https://goo.gl/Yu2SFe>>. Citado 7 vezes nas páginas 16, 17, 19, 31, 33, 38 e 45.
- BECK, K.; FOWLER, M. *Planning extreme programming*. 2000. Citado na página 22.
- BELL, D. *Fundamentos básicos de uml: O diagrama de classes*. 2016. Citado na página 26.
- BERGMANN, S. *PHPUnit - The PHP Testing Framework*. [S.l.], 2018. Disponível em: <<https://phpunit.de/>>. Citado na página 41.
- BERRE, A. J.; BRAMBILLA, M. *Interaction Flow Modeling Language - Model-Driven Development of Software Front Ends*. [S.l.], 2014. Citado na página 31.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *Sistemas de banco de dados*. 2. ed. Rio de Janeiro: [s.n.], 2006. Citado 2 vezes nas páginas 25 e 26.
- CASARIN, N. V. Um panorama da modelagem de software web por empresas e instituições de ensino do sudoeste do paran . Francisco Beltr o, PR, 2014. Citado 2 vezes nas páginas 30 e 31.
- CERI, S.; FRATERNALI, P.; BONGIO, A. Web modeling language (webml): a modeling language for designing web sites. *Computer Networks*, v. 33, p. 137–157, 2000. Citado na página 29.
- COHN, M. *User Stories Applied: For Agile Software Development*. [S.l.: s.n.], 2004. Citado 3 vezes nas páginas 22, 23 e 37.
- DELAMARO, M. E.; MALDONADO, J. C.; JINO, M. *Introdu o ao Teste de Software*. Rio de Janeiro: [s.n.], 2007. Citado 2 vezes nas páginas 27 e 28.
- ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados*. 6. ed. S o Paulo: [s.n.], 2010. Citado 2 vezes nas páginas 24 e 25.
- FERREIRA, A. Projeto de aplica o web por meio da linguagem webml. Medianeira, PR, 2013. Citado na página 29.
- HYB. *Software para o Terceiro Setor - HYB*. [S.l.], 2018. Disponível em: <<https://hyb.com.br>>. Citado na página 55.
- KOSCIANSKI, A.; SOARES, M. dos S. *Engenharia de Software*. 2. ed. S o Paulo: [s.n.], 2007. Citado na página 27.

- LAUDON, K.; LAUDON, J. *Sistemas de Informações Gerenciais*. 9. ed. São Paulo: [s.n.], 2011. Citado na página 20.
- LONGO, H. E. R.; SILVA, M. P. da. A utilização de histórias de usuários no levantamento de requisitos Ágeis. *International Journal of Knowledge Engineering and Management*, 2014. Citado 2 vezes nas páginas 22 e 37.
- MARINHO, E. H.; RESENDE, R. F. Extensão de um metamodelo de aplicações baseadas na web considerando ajax. *VII Simpósio Brasileiro de Sistemas de Informação*, 2011. Citado na página 28.
- PLANTULLO, V. L.; HOFFMAN, A. R. *Sistemas de Informação - Fundamentos - Do Sistema de Informações Gerenciais - SIG ao Planejamento de Recursos Empresariais - ERP*. Curitiba: [s.n.], 2012. Citado na página 20.
- PRESSMAN, R. S. *Engenharia de Software - Uma Abordagem Profissional*. 7. ed. Porto Alegre: [s.n.], 2011. Citado 3 vezes nas páginas 22, 27 e 28.
- SOMMERVILLE, I. *Engenharia de Software*. 9. ed. São Paulo: [s.n.], 2011. Citado 4 vezes nas páginas 21, 22, 25 e 27.
- WEBRATIO. *Training and Documentation*. [S.l.], 2018. Disponível em: <<https://goo.gl/89cB5y>>. Citado na página 32.

Apêndices

APÊNDICE A – Levantamento de Requisitos

Este capítulo apresenta as histórias de usuários geradas no levantamento de requisitos das novas funcionalidades do SisGera.

A.1 Geração do B.O.

Gerar B.O.

Como um usuário externo,
eu quero gerar o boletim de ocorrência da vítima,
de modo que não precise de ir até a corporação para pegar o boletim de ocorrência.

Autenticar Validade do B.O.

Como um usuário externo,
eu quero autenticar a validade do B.O.,
de modo que se certifique que o documento foi gerado pelo SisGera.

Histórico de Requisições do B.O.

Como um administrador,
eu quero visualizar as gerações de boletim de ocorrência pelo usuário externo,
de modo que possa verificar os boletins gerados pelo usuário externo e os responsáveis pela geração.

A.2 Controle Patrimonial

Cadastrar Bem Patrimonial

Como um usuário interno,
eu quero cadastrar um novo bem patrimonial,
de modo que se tenha um controle de todo ativo fixo da corporação.

Editar Bem Patrimonial

Como um usuário interno,
eu quero editar o cadastro de um bem patrimonial,
de modo que os dados do bem patrimonial em questão possam ser atualizados.

Desativar Bem Patrimonial

Como um usuário interno,
eu quero desativar o cadastro de um bem patrimonial,
de modo que seja sinalizado que o bem patrimonial em questão não está mais em uso na corporação.

Ativar Bem Patrimonial

Como um usuário interno,
eu quero ativar o cadastro de um bem patrimonial,
de modo que seja sinalizado que o bem patrimonial em questão voltou a estar em uso na corporação.

A.3 Doações

Cadastrar Doação

Como um usuário interno,
eu quero cadastrar uma nova doação,
de modo que se tenha um controle de todas as doações realizadas para a corporação.

Editar Doação

Como um usuário interno,
eu quero editar o registro de uma doação,
de modo que os dados da doação em questão possam ser atualizados.

Excluir Doação

Como um usuário interno,
eu quero excluir o registro de uma doação,
de modo que a doação referente seja eliminada do sistema.

A.4 Produtos

Cadastrar Produto

Como um usuário interno,
eu quero cadastrar um novo produto,
de modo que seja possível realizar o controle de estoque de tal produto.

Editar Produto

Como um usuário interno,
eu quero editar o cadastro de um produto,

de modo que os dados do produto em questão possam ser atualizados.

Desativar Produto

Como um usuário interno,

eu quero desativar o cadastro de um produto,

de modo que seja sinalizado que o produto em questão não está mais em uso na corporação.

Ativar Produto

Como um usuário interno,

eu quero ativar o cadastro um produto,

de modo que seja sinalizado que o produto em questão voltou a estar em uso na corporação.

Excluir Produto

Como um usuário interno,

eu quero excluir um produto cadastrado,

de modo que o produto referente seja eliminado do sistema.

A.5 Estoque

Incluir Movimentação

Como um usuário interno,

eu quero inserir uma movimentação de estoque,

de modo que eu possa controlar o estoque de um determinado produto.

Editar Movimentação

Como um usuário interno,

eu quero editar o registro de uma movimentação de estoque,

de modo que os dados da movimentação de estoque em questão possam ser atualizados.

Excluir Movimentação

Como um usuário interno,

eu quero excluir o registro de uma movimentação de estoque,

de modo que a movimentação referente seja eliminada do sistema.

A.6 Usuários

Alterar Permissões

Como um administrador,
eu quero alterar as permissões de acesso de um usuário,
de modo que seja controlado as telas no qual o usuário terá acesso.

Incluir Histórico do Usuário

Como um administrador,
eu quero incluir histórico para um usuário da corporação,
de modo que seja possível controlar o histórico comportamental do usuário na corporação ou em outras corporações.

Alterar Histórico do Usuário

Como um administrador,
eu quero editar o registro de um histórico do usuário,
de modo que os dados do registro de histórico em questão possam ser atualizados.

Excluir Histórico do Usuário

Como um administrador,
eu quero excluir o registro de um histórico do usuário,
de modo que o histórico referente seja eliminado do sistema.

A.7 Fale Conosco

Enviar Mensagem

Como um usuário interno,
eu quero enviar mensagens a equipe de desenvolvimento do projeto SisGera,
de modo que possam ser tratados assuntos pertinentes ao sistema via canal de comunicação entre voluntários e desenvolvedores.

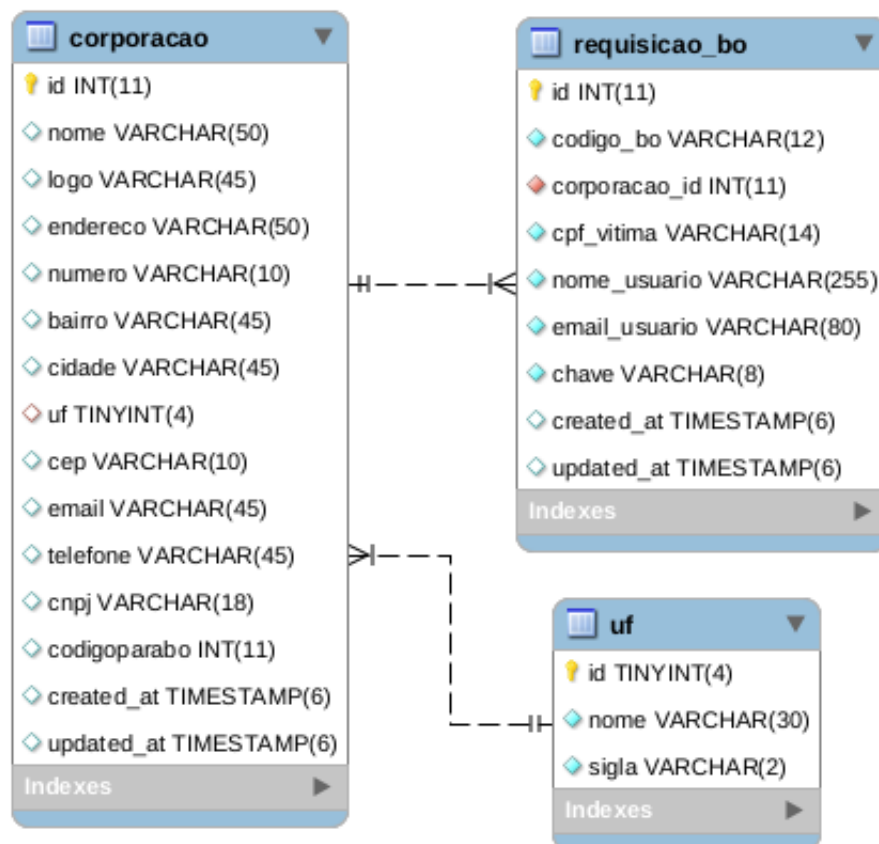
APÊNDICE B – Projeto

Este capítulo apresenta os artefatos gerados na fase de projeto de *software*, no qual são apresentados os diagramas que modelam as camadas *Model-View-Controller* da aplicação.

B.1 Entidade-Relacionamento

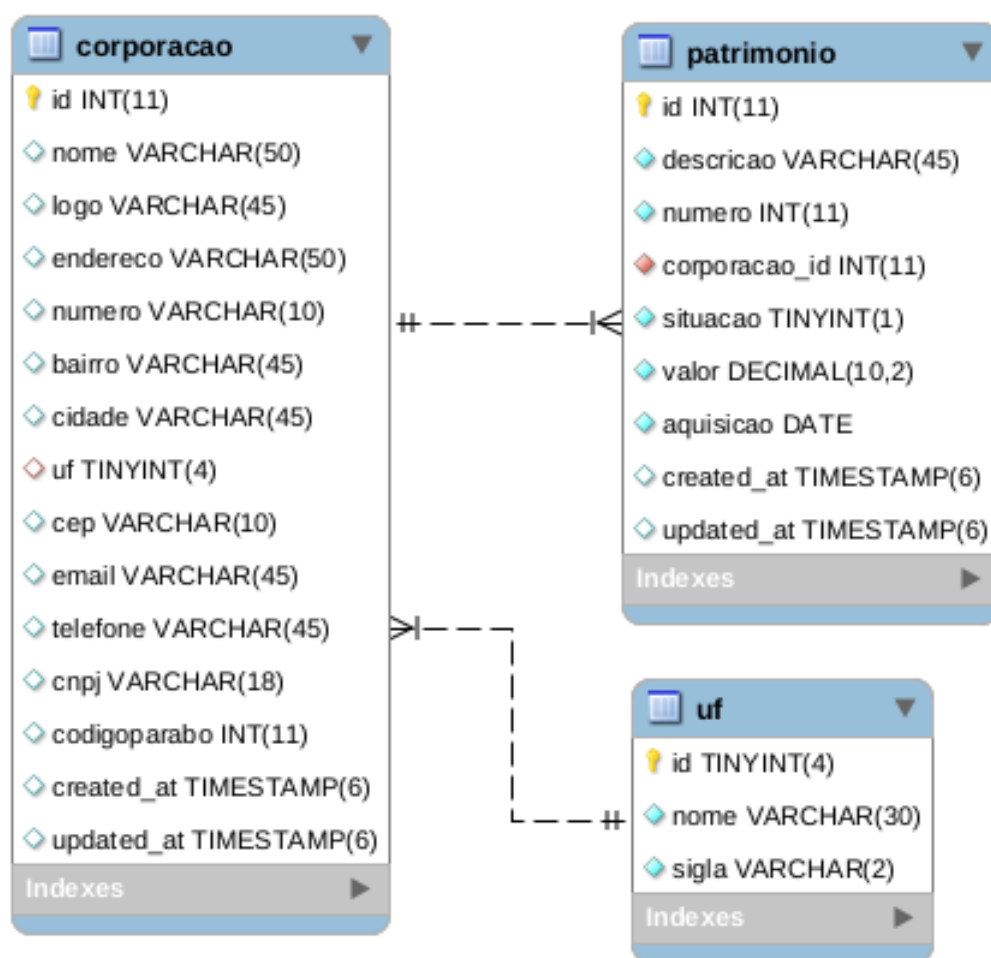
Logo abaixo são apresentados os diagramas ER que modelam a camada *Model* das novas funcionalidades agregadas ao SisGera.

Figura 25 – Diagrama ER da funcionalidade de geração e validação do B.O. pelo usuário externo.



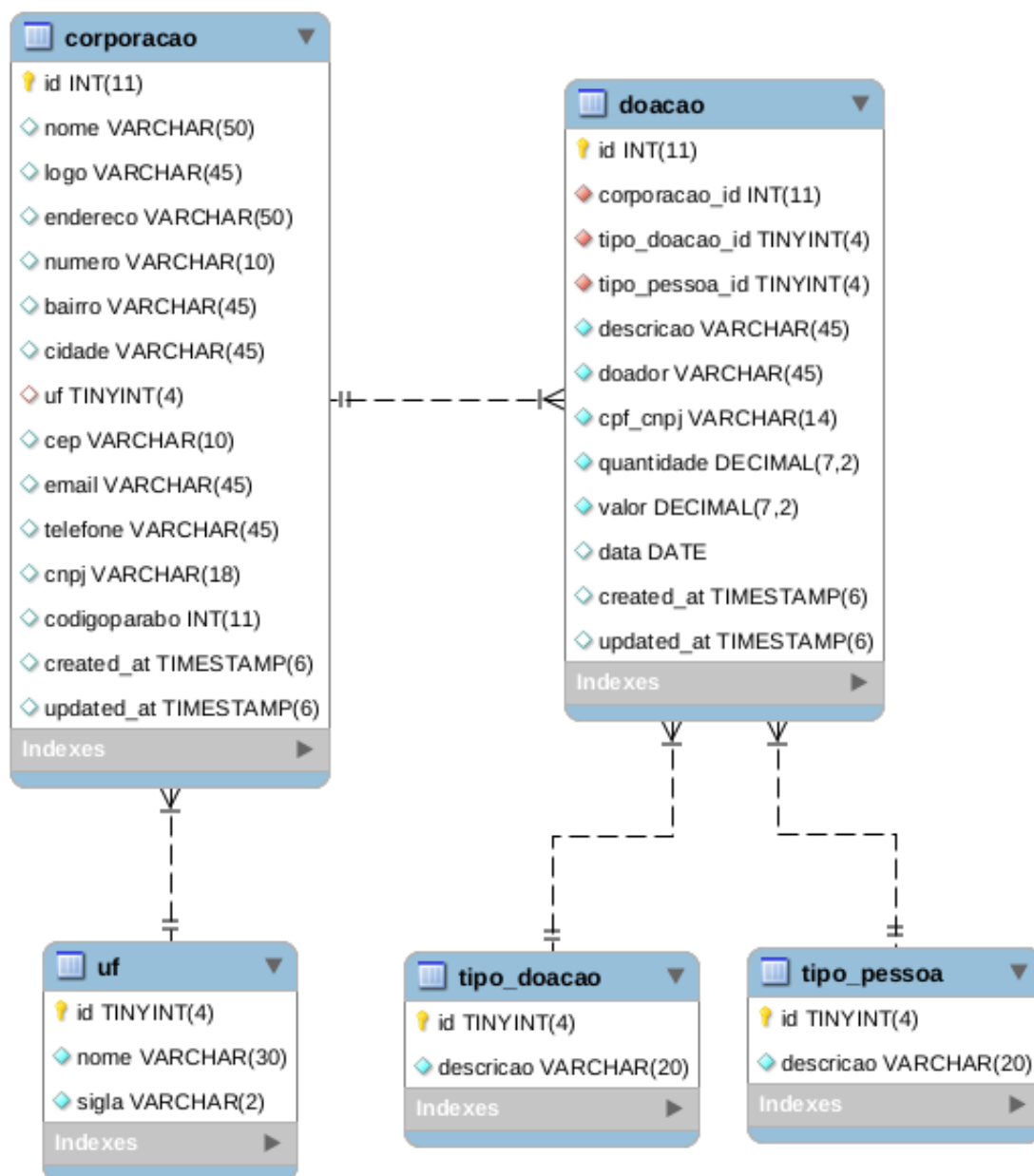
Fonte: Elaborado pelo autor (2018)

Figura 26 – Diagrama ER da funcionalidade de controle de bens patrimoniais.



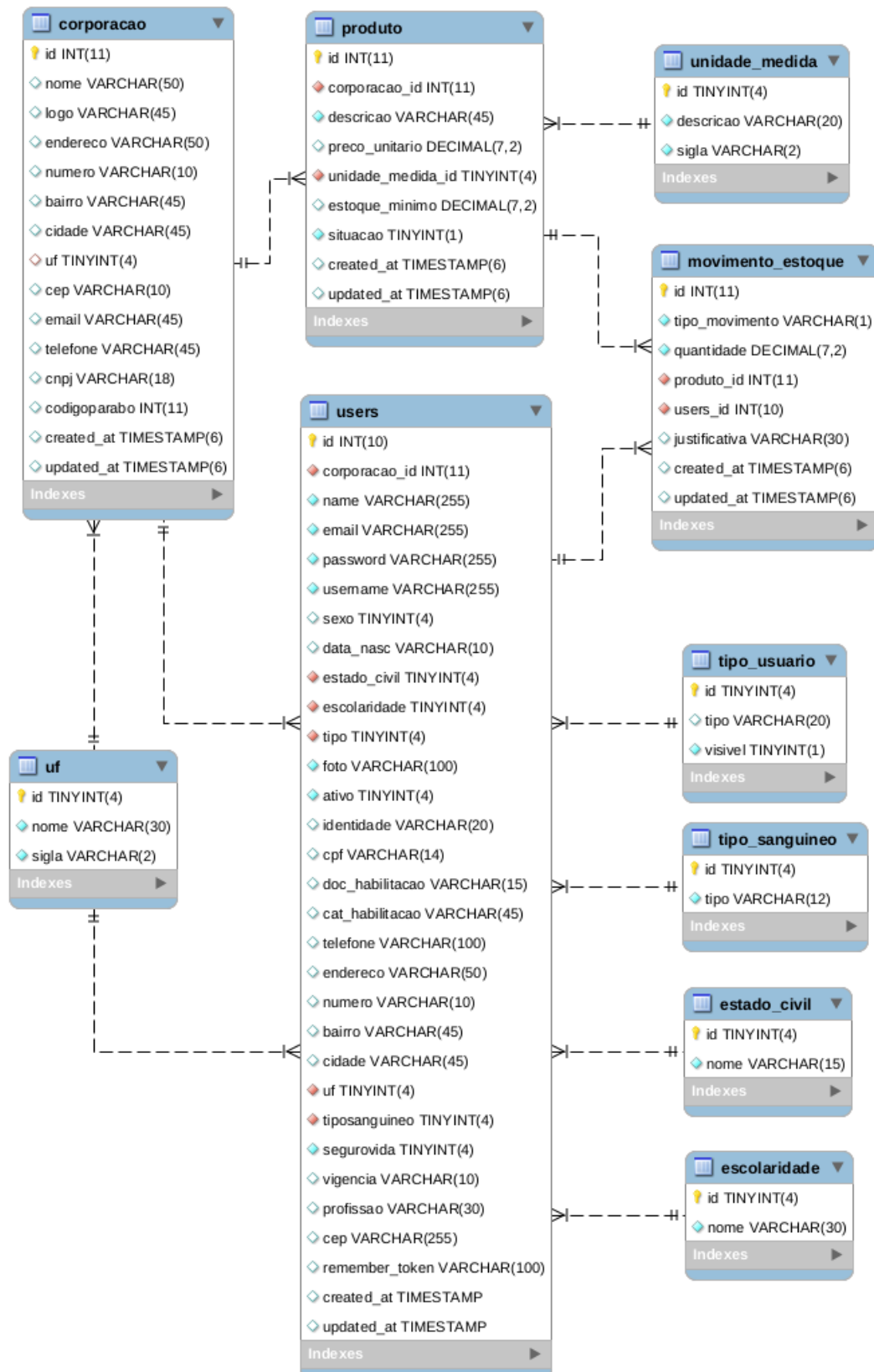
Fonte: Elaborado pelo autor (2018)

Figura 27 – Diagrama ER da funcionalidade de controle de doações.



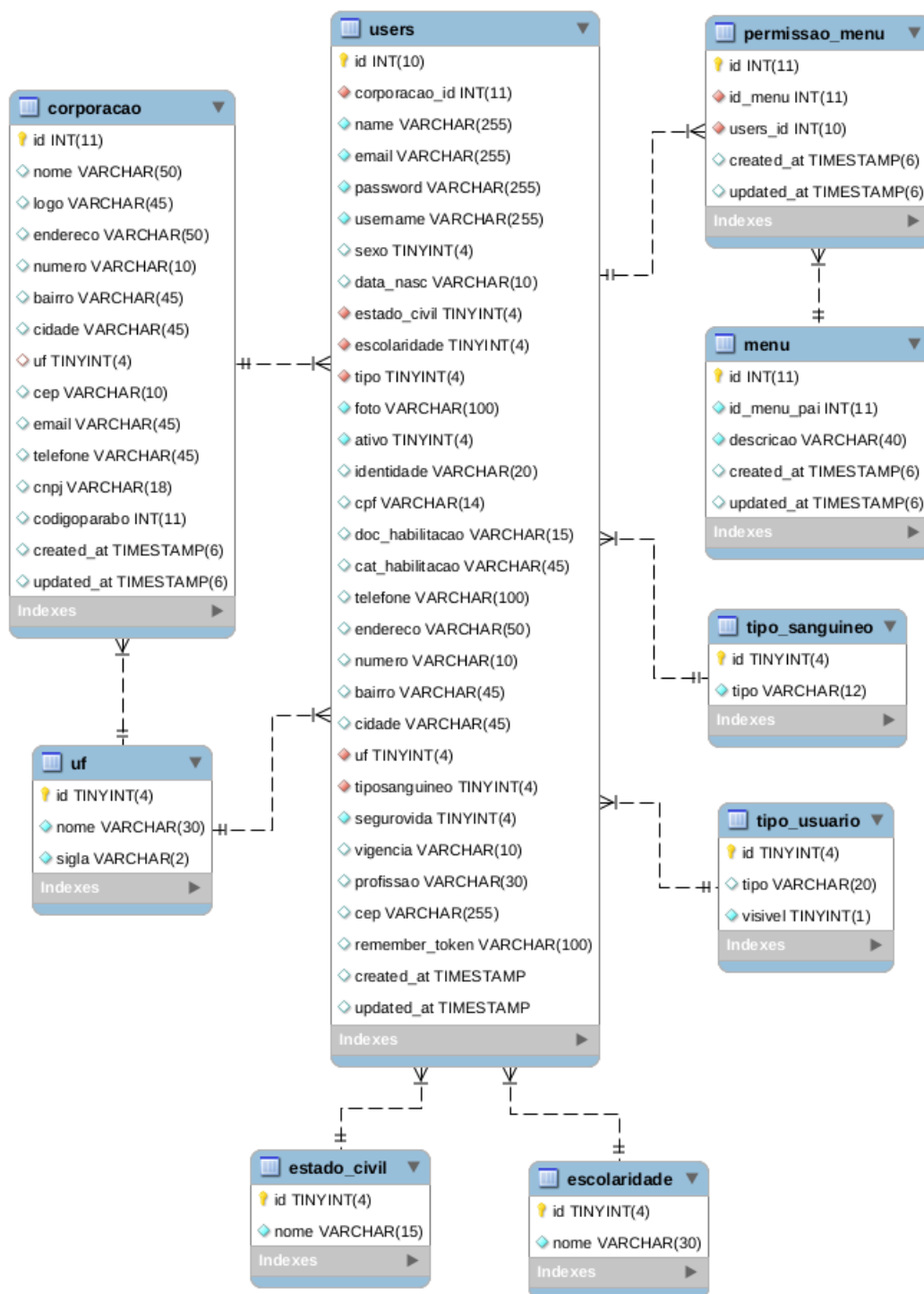
Fonte: Elaborado pelo autor (2018)

Figura 28 – Diagrama ER das funcionalidades de cadastro de produtos e controle de estoque.



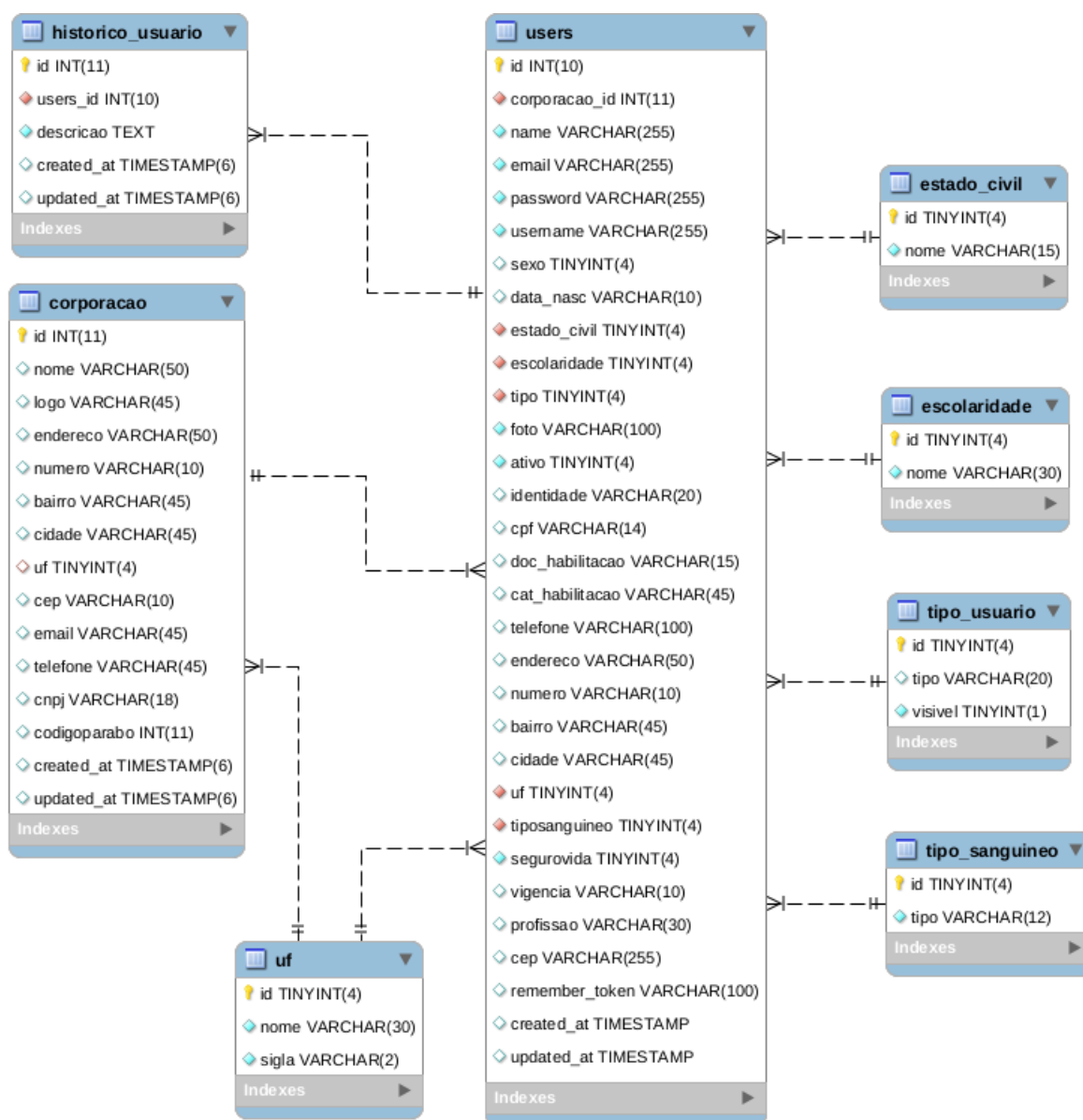
Fonte: Elaborado pelo autor (2018)

Figura 29 – Diagrama ER da funcionalidade de permissões de acesso.



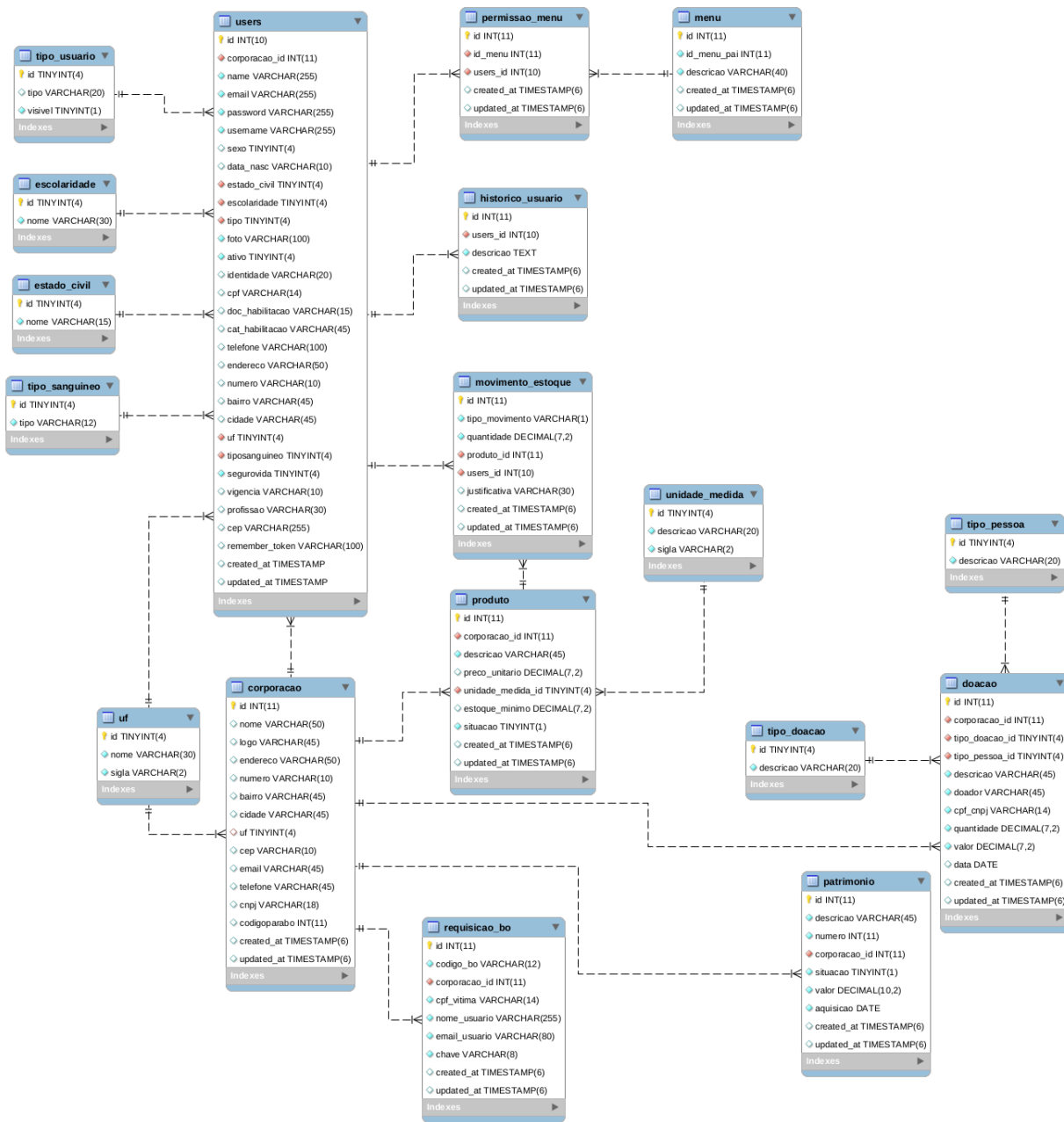
Fonte: Elaborado pelo autor (2018)

Figura 30 – Diagrama ER da funcionalidade de histórico do usuário.



Fonte: Elaborado pelo autor (2018)

Figura 31 – Esboço completo das tabelas utilizadas nas novas funcionalidades do SisGera.

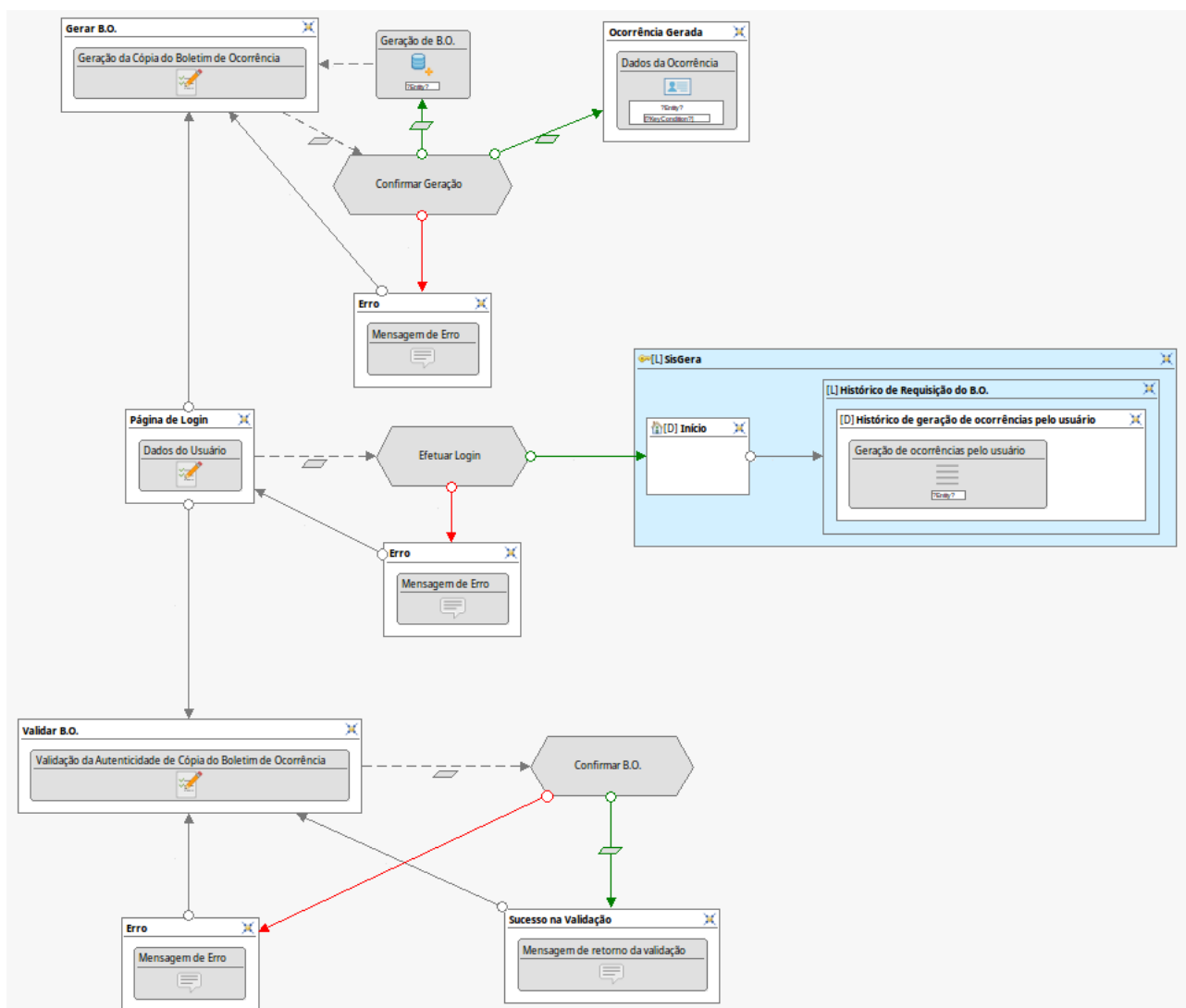


Fonte: Elaborado pelo autor (2018)

B.2 Diagramas *IFML*

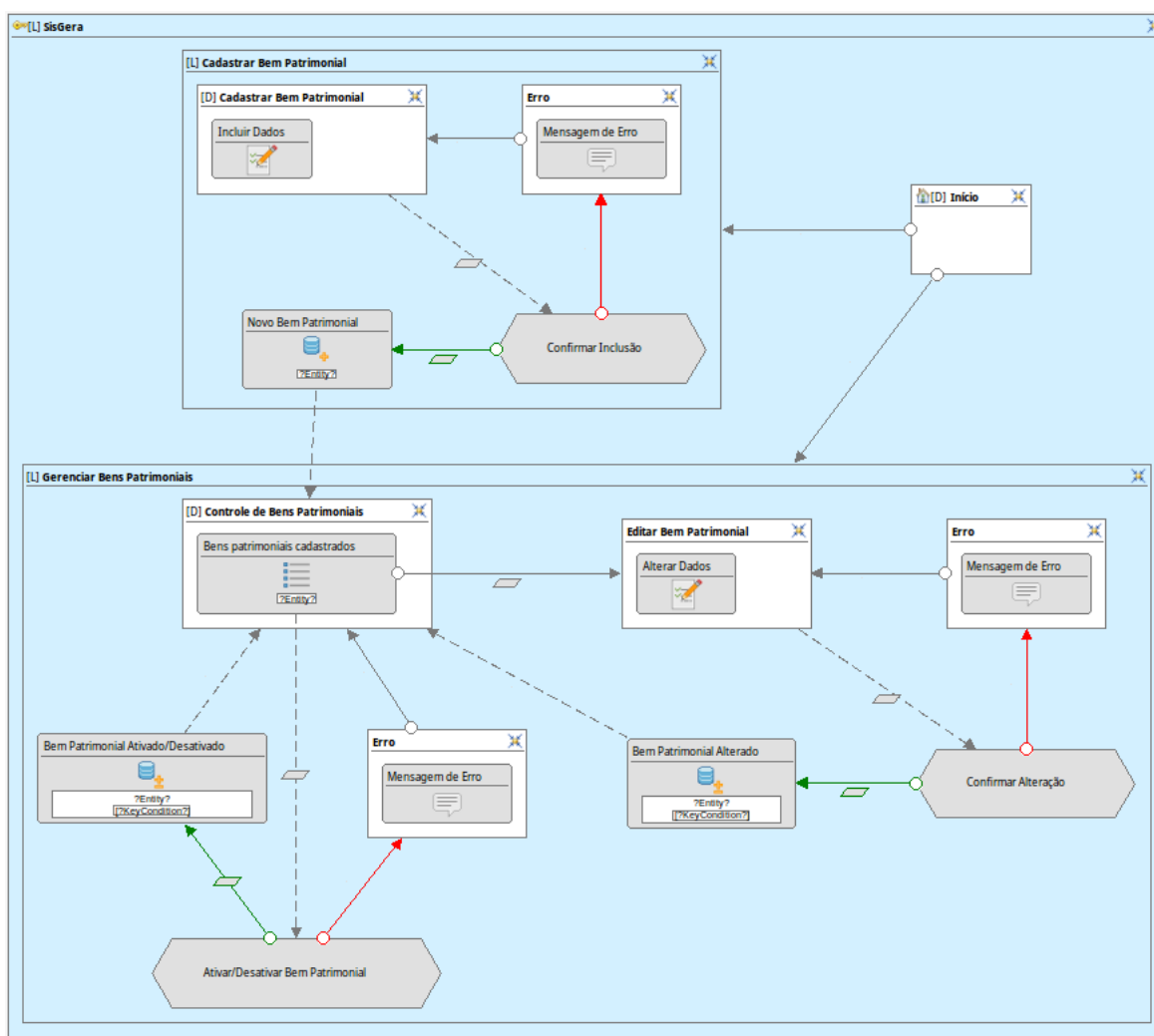
Nesta seção são apresentados os diagramas *IFML* que modelam a camada *View* das novas funcionalidades agregadas ao SisGera.

Figura 32 – Diagrama *IFML* das funcionalidades de geração e validação do B.O. pelo usuário externo.



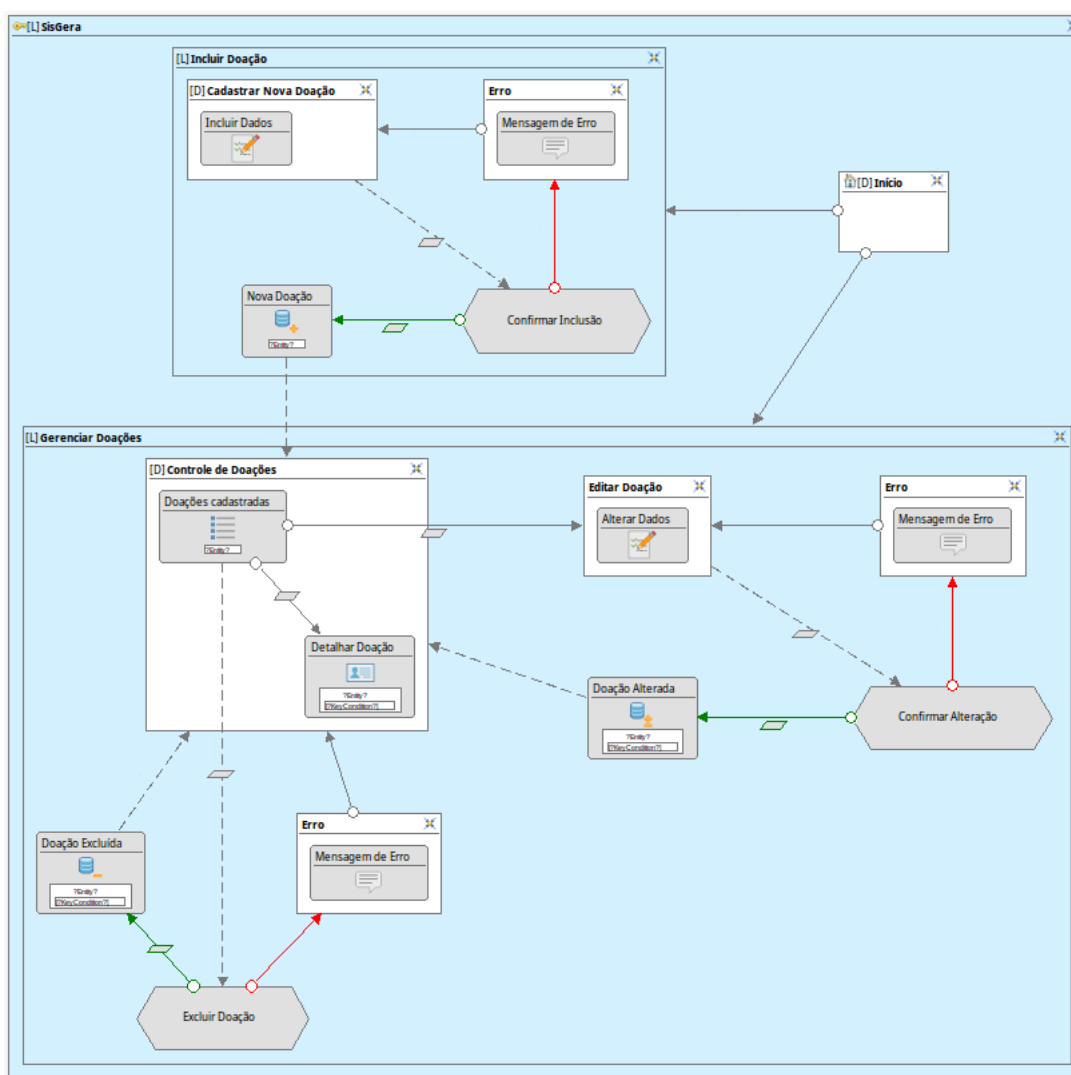
Fonte: Elaborado pelo autor (2018)

Figura 33 – Diagrama *IFML* da funcionalidade de controle de bens patrimoniais.



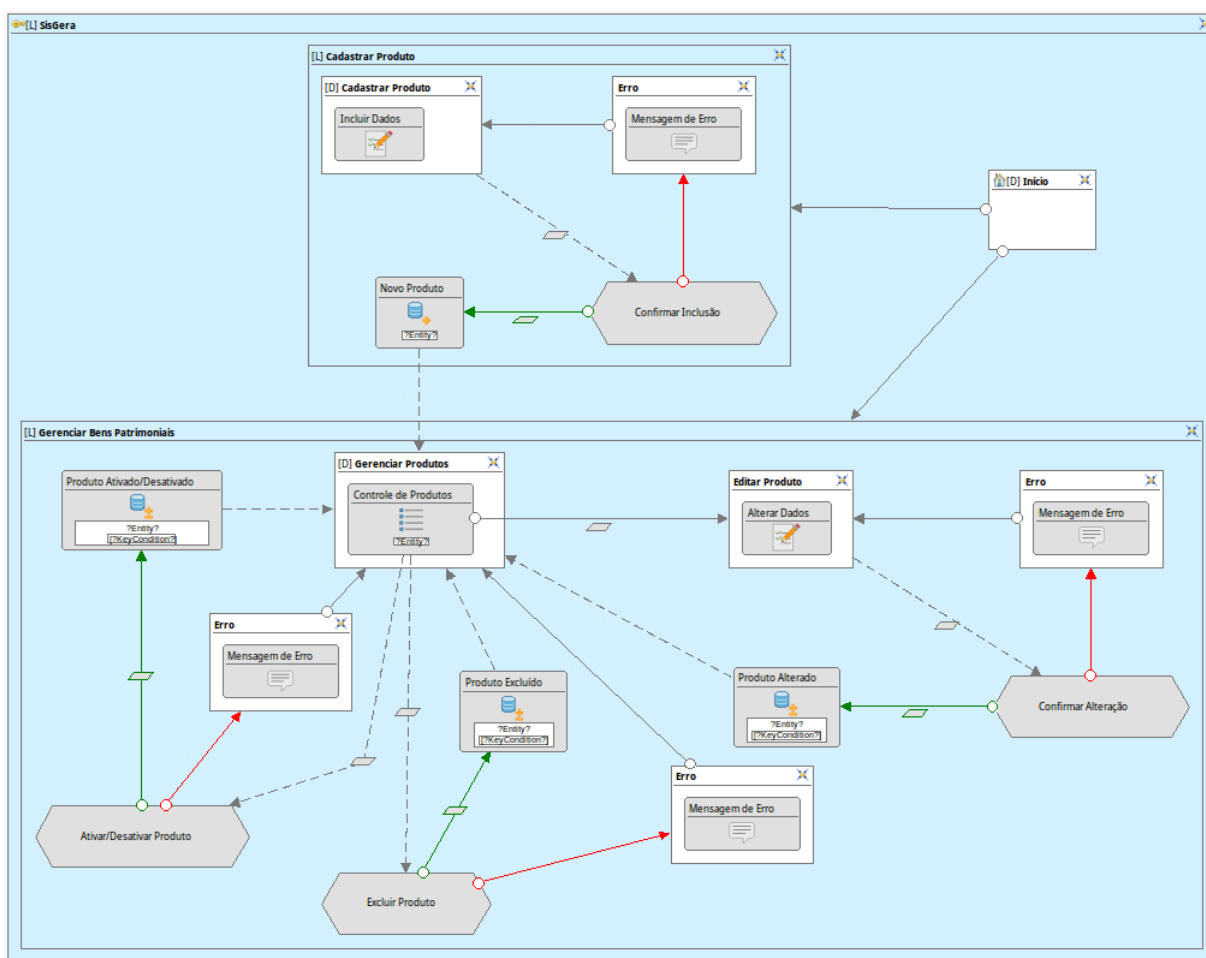
Fonte: Elaborado pelo autor (2018)

Figura 34 – Diagrama *IFML* da funcionalidade de gestão de doações.



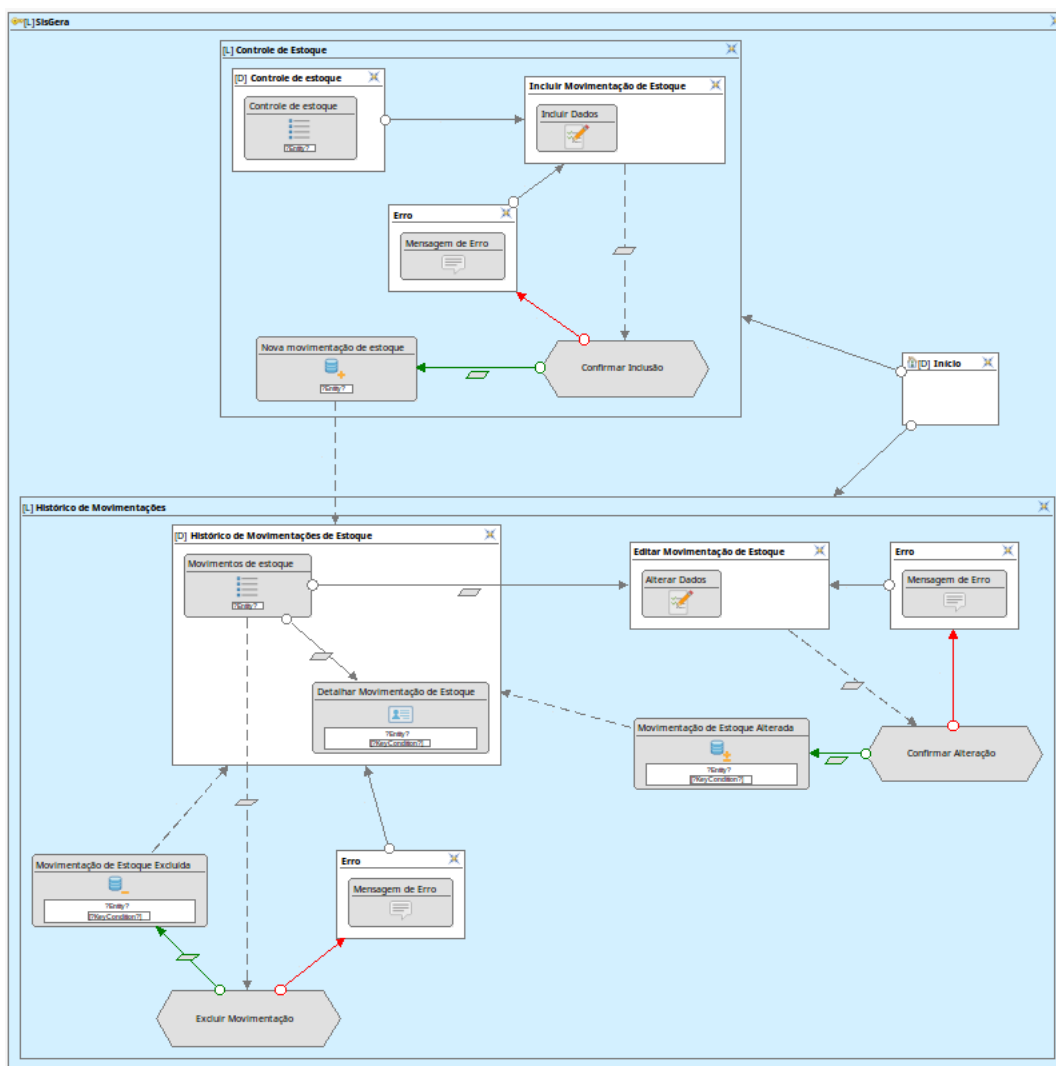
Fonte: Elaborado pelo autor (2018)

Figura 35 – Diagrama *IFML* da funcionalidade de cadastro de produto.



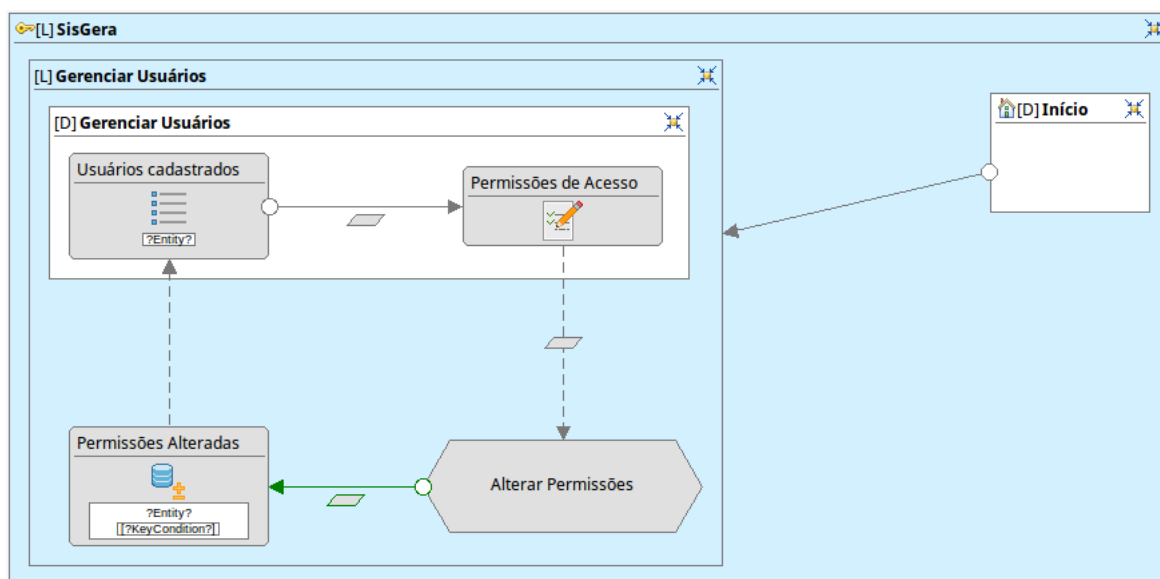
Fonte: Elaborado pelo autor (2018)

Figura 36 – Diagrama *IFML* da funcionalidade de controle de estoque.



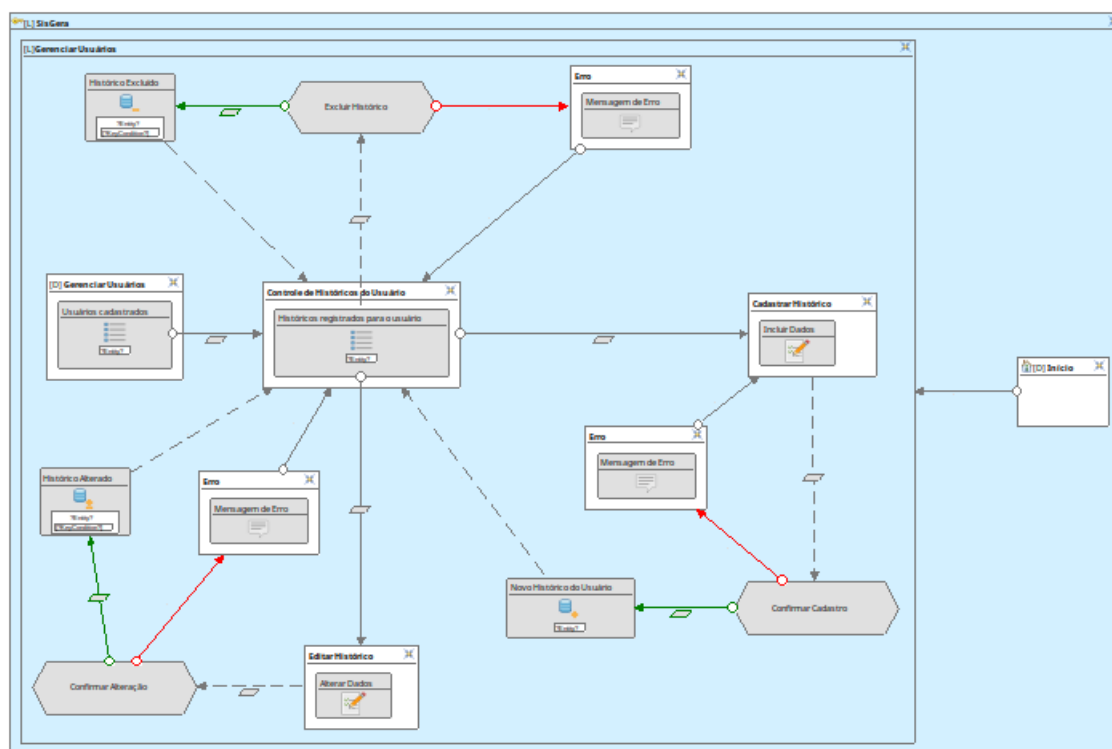
Fonte: Elaborado pelo autor (2018)

Figura 37 – Diagrama *IFML* da funcionalidade de permissões de acesso.



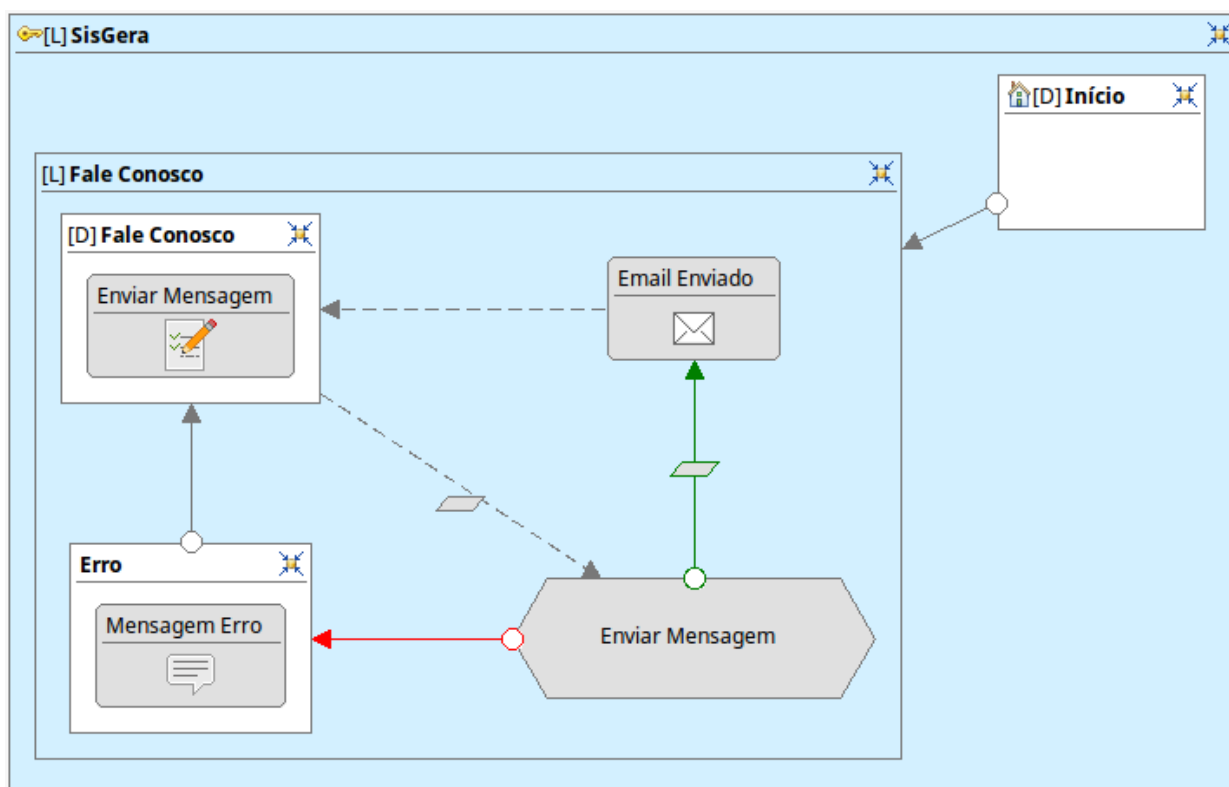
Fonte: Elaborado pelo autor (2018)

Figura 38 – Diagrama *IFML* da funcionalidade de histórico do usuário.



Fonte: Elaborado pelo autor (2018)

Figura 39 – Diagrama *IFML* da funcionalidade Fale Conosco.

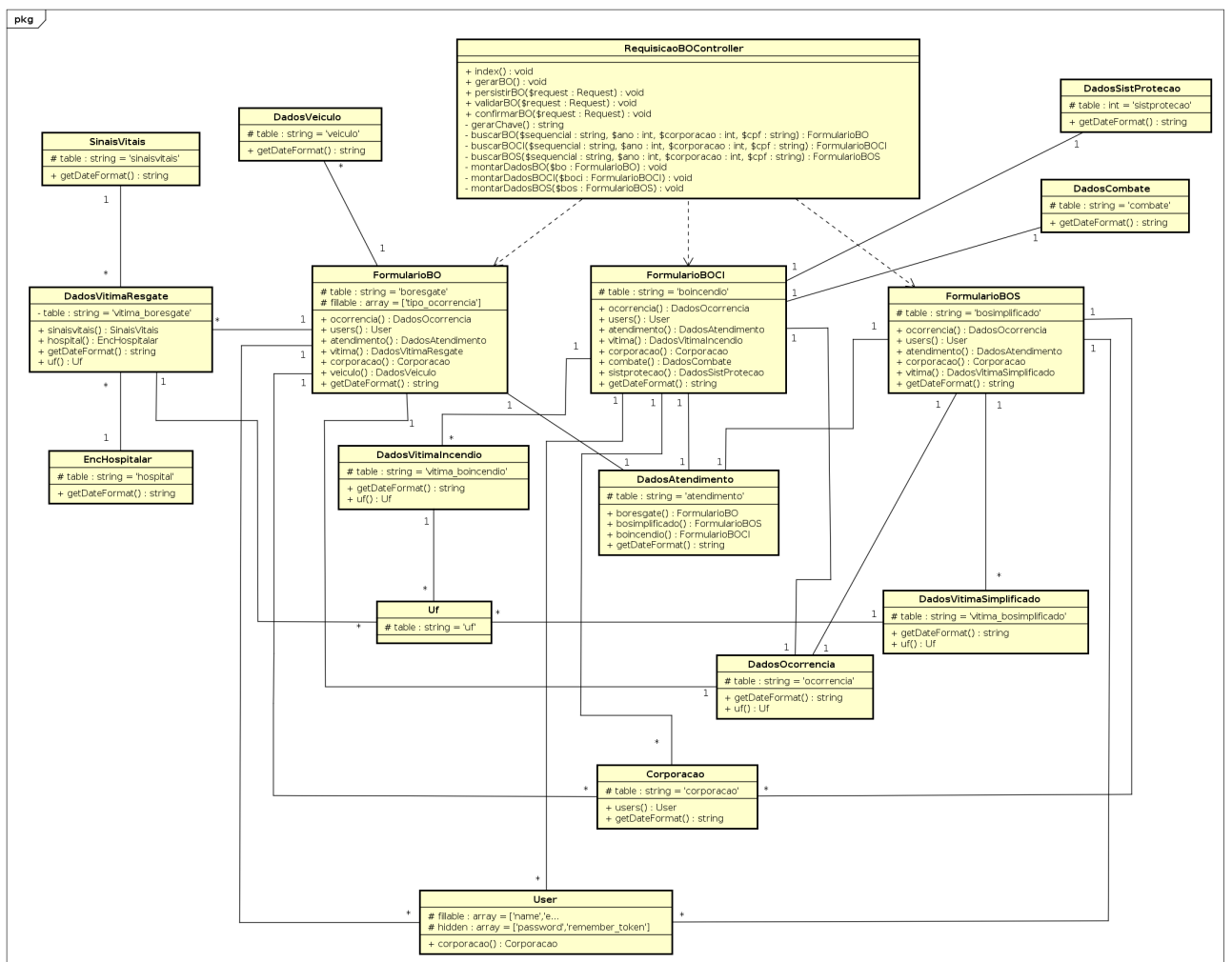


Fonte: Elaborado pelo autor (2018)

B.3 Diagramas de Classe

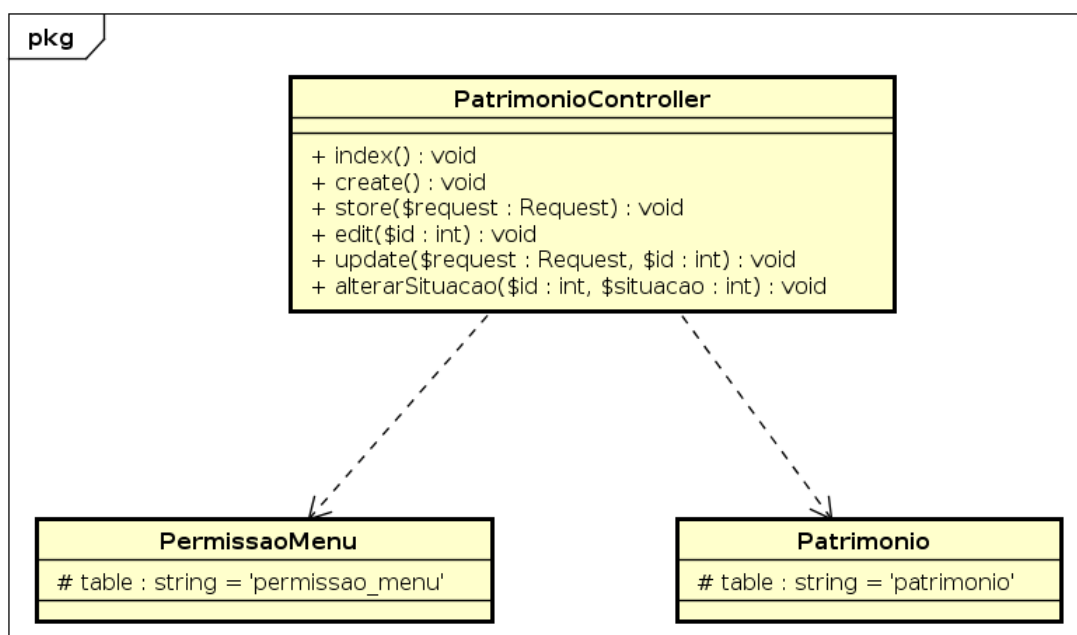
E por último, nesta seção são apresentados os diagramas de classe que modelam a camada *Controller* das novas funcionalidades agregadas ao SisGera. Os diagramas seguem logo abaixo:

Figura 40 – Diagrama de classes funcionalidade de histórico de requisição do B.O. pelo usuário externo.



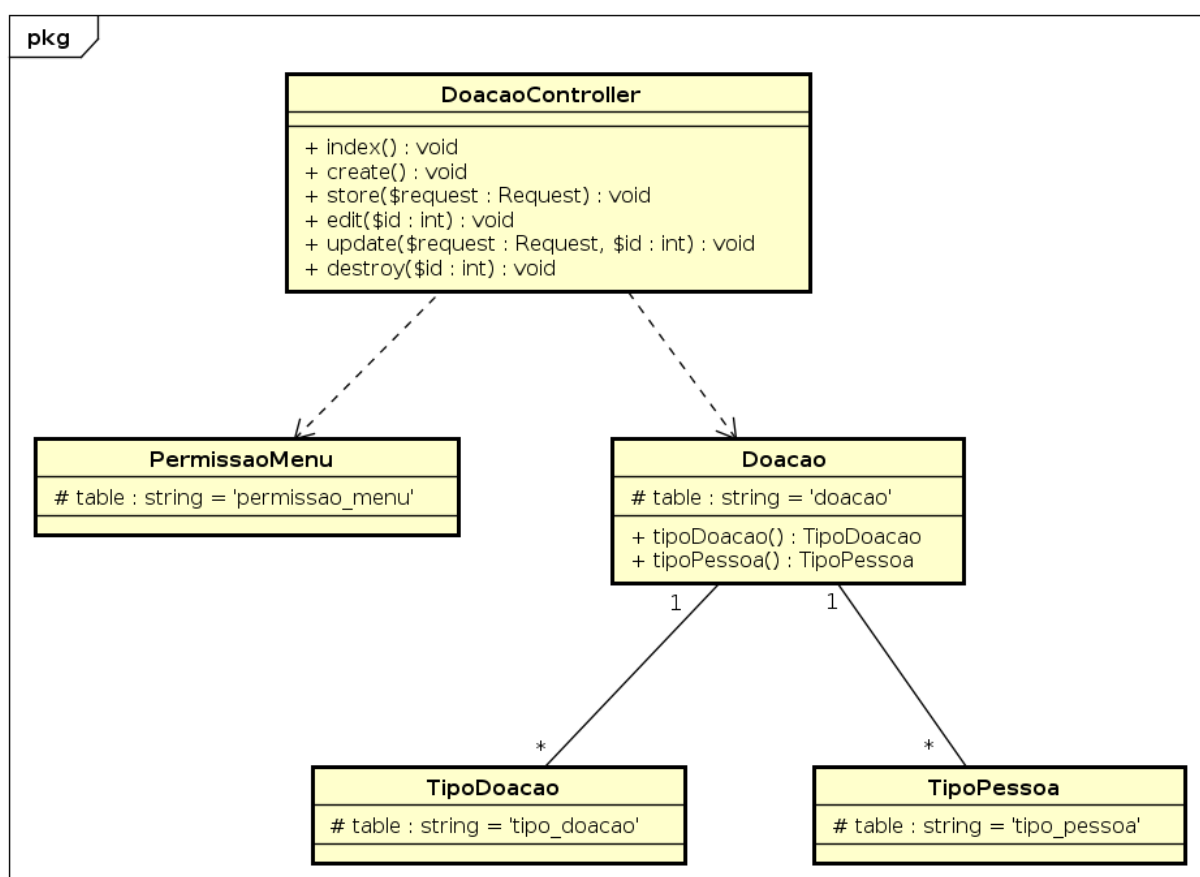
Fonte: Elaborado pelo autor (2018)

Figura 41 – Diagrama de classes da funcionalidade de controle de bens patrimoniais.



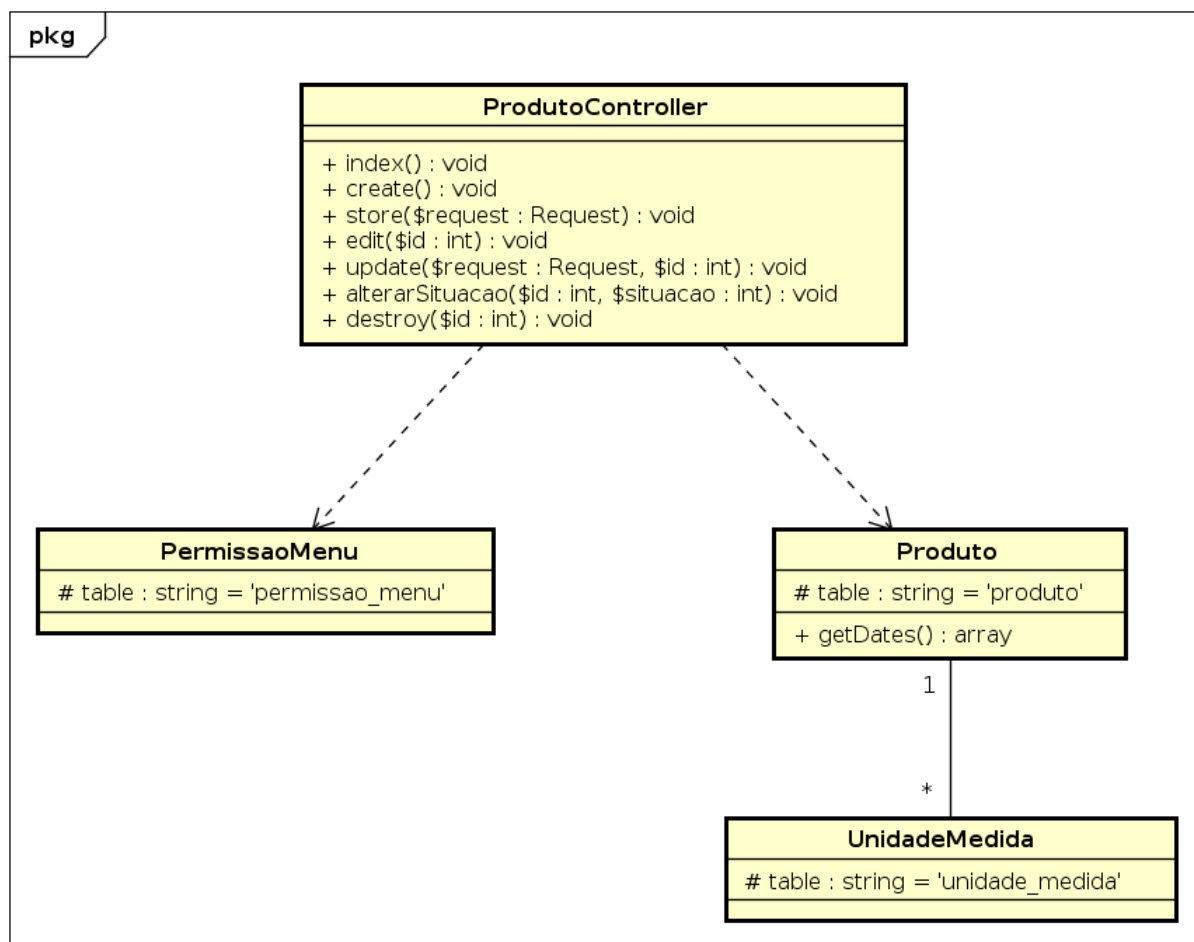
Fonte: Elaborado pelo autor (2018)

Figura 42 – Diagrama de classes da funcionalidade de controle de doações.



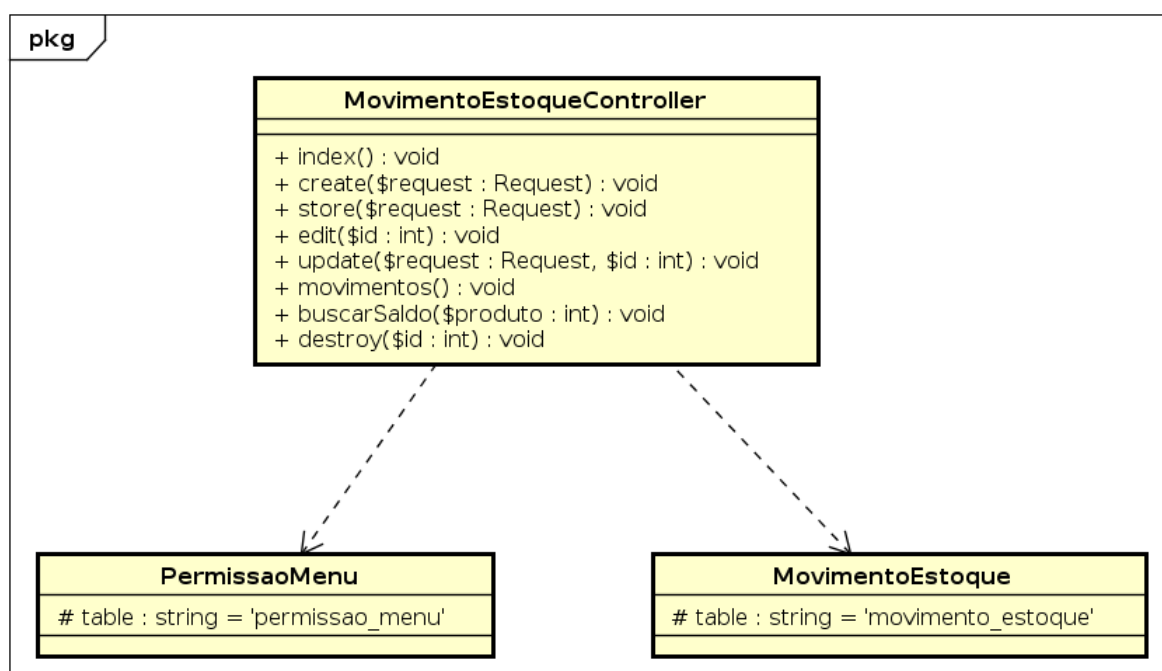
Fonte: Elaborado pelo autor (2018)

Figura 43 – Diagrama de classes da funcionalidade de cadastro de produtos.



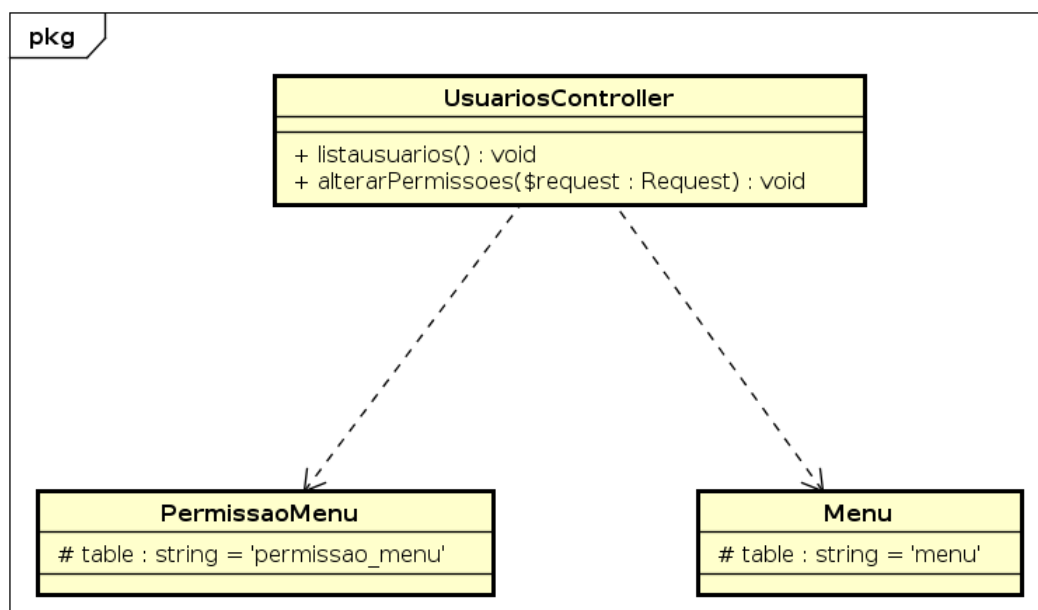
Fonte: Elaborado pelo autor (2018)

Figura 44 – Diagrama de classes da funcionalidade de controle de estoque.



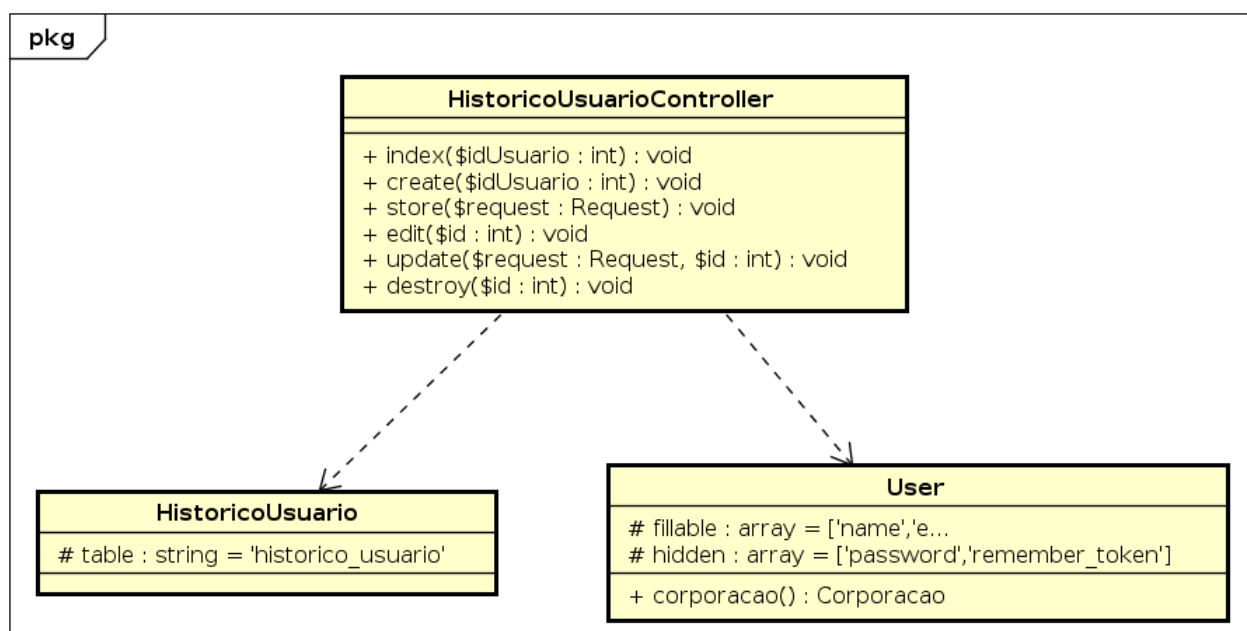
Fonte: Elaborado pelo autor (2018)

Figura 45 – Diagrama de classes da funcionalidade de permissões de acesso.



Fonte: Elaborado pelo autor (2018)

Figura 46 – Diagrama de classes da funcionalidade de histórico do usuário.



Fonte: Elaborado pelo autor (2018)

Anexos



ASSOCIAÇÃO DE BOMBEIROS VOLUNTÁRIOS

São Domingos do Prata – Dionísio – São José do Goiabal

“Sua vida, nosso bem mais precioso”

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO – TCLE

A Associação de Bombeiros Voluntários de São Domingos Do Prata autoriza o aluno da Universidade Federal de Ouro Preto (UFOP) **Silvandro Sergio Martins Oliveira** a desenvolver seu Trabalho de Conclusão de Curso (TCC) intitulado “Sistema de informação para controle de materiais e doações dos bombeiros voluntários” utilizado como objeto de pesquisa a própria organização.

O presente termo autoriza o aluno e a Universidade a utiliza e divulgar o nome da organização somente dentro do conteúdo do TCC para fins acadêmicos.

Por se tratar de um trabalho acadêmico, o desenvolvimento do trabalho e qualquer software decorrente do mesmo não acarretarão nenhum custo ou vínculo para com a organização.

João Monlevade, 29 de Novembro de 2018.

Bráulio Henrique de Vasconcelos Perdigão

Associação de Bombeiros Voluntários de São Domingos do Prata
Bráulio Henrique de Vasconcelos Perdigão
Diretor-Presidente

Bráulio H. Vasconcelos Perdigão
Diretor Presidente/Operacional
CPF: 059.985.606-80 - Mat.: 001
BOMBEIROS VOLUNTÁRIOS
São Domingos do Prata - MG



TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO – TCLE

A Associação de Bombeiros Voluntários de Barão de Cocais, autoriza o aluno da Universidade Federal de Ouro Preto (UFOP) **Silvandro Sergio Martins Oliveira** a desenvolver seu Trabalho de Conclusão de Curso (TCC) intitulado “Sistema de informação para controle de materiais e doações dos bombeiros voluntários” utilizado como objeto de pesquisa a própria organização.

O presente termo autoriza o aluno e a Universidade a utiliza e divulgar o nome da organização somente dentro do conteúdo do TCC para fins acadêmicos.

Por se tratar de um trabalho acadêmico, o desenvolvimento do trabalho e qualquer software decorrente do mesmo não acarretarão nenhum custo ou vínculo para com a organização.

João Monlevade, 30 de Novembro de 2018.



Associação de Bombeiros Voluntários de Barão de Cocais
Uelinton Rodrigues - Diretor - Presidente



UFOP
Universidade Federal
de Ouro Preto

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS
COLEGIADO DO CURSO DE SISTEMAS DE INFORMAÇÃO

TERMO DE RESPONSABILIDADE

Eu, Silvano Sergio Martins Oliveira declaro que o texto do trabalho de conclusão de curso intitulado "*Sistema de Informação para controle de materiais e doações aos bombeiros voluntários*" é de minha inteira responsabilidade e que não há utilização de texto, material fotográfico, código fonte de programa ou qualquer outro material pertencente a terceiros sem as devidas referências ou consentimento dos respectivos autores.

João Monlevade, 03 de dezembro de 2018.


Silvano Sergio Martins Oliveira



UFOP
Universidade Federal
de Ouro Preto

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS
COLEGIADO DO CURSO DE SISTEMAS DE INFORMAÇÃO

DECLARAÇÃO DE CONFORMIDADE

Certifico que o(a) aluno(a) Silvandro Sergio Martins Oliveira, autor do trabalho de conclusão de curso intitulado "Sistema de Informação para controle de materiais e doações aos bombeiros voluntários" efetuou as correções sugeridas pela banca examinadora e que estou de acordo com a versão final do trabalho.

João Monlevade, 20 de dezembro de 2018.

Euler Horta Marinho
Professor (a) Orientador (a)