

UNIVERSIDADE FEEVALE

FERNANDO GABRIEL BLOEDORN

SISTEMA DE VAREJO DE CALÇADOS E ACESSÓRIOS COM RECURSOS WEB E
ÊNFASE NA UTILIZAÇÃO DE SOFTWARE LIVRE

Novo Hamburgo
2010

FERNANDO GABRIEL BLOEDORN

SISTEMA DE VAREJO DE CALÇADOS E ACESSÓRIOS COM RECURSOS WEB E
ÊNFASE NA UTILIZAÇÃO DE SOFTWARE LIVRE

Trabalho de conclusão de curso
apresentado como requisito parcial à
obtenção do grau de Bacharel em
Sistemas de Informação pela Universidade
Feevale

Orientador: Juliano Varella de Carvalho

Novo Hamburgo
2010

AGRADECIMENTOS

Gostaria de agradecer a todos os que, de alguma maneira, contribuíram para a realização desse trabalho de conclusão, em especial:

A minha namorada Gisele Schoeninger pelo apoio, compreensão e suporte emocional no decorrer desse trabalho.

Aos meus colegas e amigos, pelos conhecimentos e experiências compartilhados durante meus anos de graduação.

Ao meu orientador Juliano Varella de Carvalho, pelo empenho e suporte no desenvolvimento desse trabalho.

Aos demais professores da Universidade Feevale por sua dedicação ao longo do curso de Sistemas de Informação.

RESUMO

Nos últimos anos o setor varejista nacional tem crescido de forma exponencial, levando-o à aquisição de softwares que supram suas necessidades, tanto de negócio como fiscais. Empresas do ramo muitas vezes não possuem sistemas adequados para o controle dos seus negócios. A definição da tecnologia a ser utilizada não é algo trivial, sendo que fatores como tempo e dinheiro estão diretamente envolvidos. Mensurar os benefícios de quebra de paradigmas como migração de sistemas legados e utilização de software livre, por exemplo, não é sempre possível. O desenvolvimento de software utilizando recursos web tem ganhado bastante espaço no desenvolvimento de sistemas empresariais, pois oferecem bom desempenho e baixo custo. Sendo assim, o objetivo deste trabalho é a análise de viabilidade, a modelagem e o desenvolvimento de um sistema de varejo desenvolvido com recursos web, através da utilização exclusiva de software livre: desenvolvido na linguagem PHP e com banco de dados PostgreSQL, por meio de *frameworks* que usam o paradigma de orientação a objetos.

Palavras-chave: Desenvolvimento de Software. Software Livre. Orientação a Objetos. Ambiente Web.

ABSTRACT

In recent years the domestic retail sector has grown exponentially, leading the purchasing of software that meet your needs, both business and tax. Related businesses often do not have adequate systems to control their business. The definition of the technology being used is not trivial, and factors such as time and money are directly involved. Measuring the benefits of breaking paradigms such as migration of legacy systems and use of free software, for example, is not always possible. Software development using web resources has gained enough space to develop business systems, they offer good performance and low cost. Therefore, the objective of this study is the feasibility analysis, modeling and development of a retail web resources developed through the use of only free software: developed using PHP and the database PostgreSQL, through framework using the paradigm of object orientation.

Keywords: Software Development. Free Software. Object Orientation. Web environment.

LISTA DE FIGURAS

Figura 1 - Ao invocar o Cadastro de Consumidor, são executados três aplicativos.....	20
Figura 2 – Arquitetura básica ambiente atual.....	21
Figura 3 – Tratamento de retorno da impressora fiscal através da comunicação <i>serial</i>	22
Figura 4 – Tela de venda de produtos do sistema atual.	22
Figura 5 – Relatório de despesas do sistema atual.	23
Figura 6 – Comunicação com a impressora fiscal através da DLL Bematech.....	24
Figura 7 – Visualização do relatório de despesas através de um gráfico de pizza.....	25
Figura 8 – Visualização do relatório de despesas através de um gráfico de linhas.....	25
Figura 9 – Típica aplicação habilitada para web.	28
Figura 10 - A arquitetura básica de banco de dados web.....	29
Figura 11 – Arquitetura MVC.	34
Figura 12 - Estrutura de diretórios do Zend Framework.	35
Figura 13 – Um <i>web service</i> básico.	39
Figura 14 – Um <i>web service</i> permite o acesso a aplicações.....	39
Figura 15 – <i>Web service</i> fornece uma camada de abstração entre as aplicações.	40
Figura 16 – Mensagem XML.	41
Figura 17 – A estrutura da mensagem SOAP.....	42
Figura 18 – Classificação dos tipos de diagrama da UML.	46
Figura 19 – Diagrama de Caso de Uso do Sistema.	53
Figura 20 – Diagrama de Atividade Gerência de produto.	56
Figura 21 – Diagrama de Atividade Gerência de pessoa.	59
Figura 22 – Diagrama de Atividade Abertura de caixa.	61
Figura 23 – Diagrama de Atividade Fechamento de caixa.	62
Figura 24 – Diagrama de Atividade Emissão de venda.	64
Figura 25 – Diagrama de Classes do sistema.	66
Figura 26 – Diagrama de Sequência Gerência de produto.....	67
Figura 27 – Diagrama de Sequência Gerência de pessoa.	68
Figura 28 – Diagrama de Sequência Abertura de caixa.....	69
Figura 29 – Diagrama de Sequência Fechamento de caixa.....	69
Figura 30 – Diagrama de Sequência Emissão de venda.	70
Figura 31 – Diagrama entidade relacionamento – DER.	72
Figura 32 – Tela de permissão de acesso.	76
Figura 33 – Tela de cadastro de consumidor.....	77

Figura 34 – Tela de cadastro de produto.....	78
Figura 35 – Tela de emissão de venda.	79
Figura 36 – Consulta de produto na emissão de venda.....	80
Figura 37 – Gráfico comparativo de vendas.	81
Figura Anexo 1 – Diagrama de Atividade Gerência de usuário.....	90
Figura Anexo 2 – Diagrama de Sequência Gerência de Usuário.	91
Figura Anexo 3 – Diagrama de Atividade Gerência de rede de lojas.....	93
Figura Anexo 4 – Diagrama de Sequência Gerência de rede de lojas.	94
Figura Anexo 5 – Diagrama de Atividade Gerência de loja.	96
Figura Anexo 6 – Diagrama de Sequência Gerência de loja.....	97
Figura Anexo 7 – Diagrama de Atividade Gerência de terminal.	99
Figura Anexo 8 – Diagrama de Sequência Gerência de terminal.....	100
Figura Anexo 9 – Diagrama de Atividade Gerência de divisão.	102
Figura Anexo 10 – Diagrama de Sequência Gerência de divisão.	103
Figura Anexo 11 – Diagrama de Atividade Gerência de artigo.	105
Figura Anexo 12 – Diagrama de Sequência Gerência de divisão.	106
Figura Anexo 13 – Diagrama de Atividade Gerência de marca.....	108
Figura Anexo 14 – Diagrama de Sequência Gerência de marca.	109
Figura Anexo 15 – Diagrama de Atividade Gerência de modelo.....	111
Figura Anexo 16 – Diagrama de Sequência Gerência de modelo.	112
Figura Anexo 17 – Diagrama de Atividade Gerência de material.....	114
Figura Anexo 18 – Diagrama de Sequência Gerência de material.	115
Figura Anexo 19 – Diagrama de Atividade Gerência de cor.	117
Figura Anexo 20 – Diagrama de Sequência Gerência de cor.....	118
Figura Anexo 21 – Diagrama de Atividade Gerência de classificação fiscal.	120
Figura Anexo 22 – Diagrama de Sequência Gerência de classificação fiscal.....	121
Figura Anexo 23 – Diagrama de Atividade Gerência de unidade de medida.	123
Figura Anexo 24 – Diagrama de Sequência Gerência de unidade de medida.	124
Figura Anexo 25 – Diagrama de Atividade Gerência de preço.....	126
Figura Anexo 26 – Diagrama de Sequência Gerência de preço.	127
Figura Anexo 27 – Diagrama de Atividade Gerência de forma de pagamento.	129
Figura Anexo 28 – Diagrama de Sequência Gerência de forma de pagamento.	130
Figura Anexo 29 – Diagrama de Atividade Gerência de vendedor.....	132
Figura Anexo 30 – Diagrama de Sequência Gerência de vendedor.	133
Figura Anexo 31 – Diagrama de Atividade Gerência de consumidor.	136

Figura Anexo 32 – Diagrama de Sequência Gerência de consumidor.	137
Figura Anexo 33 – Diagrama de Atividade Gerência de fornecedor.	140
Figura Anexo 34 – Diagrama de Sequência Gerência de fornecedor.	141
Figura Anexo 35 – Diagrama de Atividade Emissão de ordem de compra.	143
Figura Anexo 36 – Diagrama de Sequência Emissão de ordem de compra.	144
Figura Anexo 37 – Diagrama de Atividade Recebimento de produto.	145
Figura Anexo 38 – Diagrama de Sequência Recebimento de produto.	146
Figura Anexo 39 – Diagrama de Atividade Emissão da leitura da memória da impressora fiscal.	147
Figura Anexo 40 – Diagrama de Sequência Emissão da leitura da memória da impressora fiscal.	147
Figura Anexo 41 – Diagrama de Atividade Emissão de suprimento de caixa.	148
Figura Anexo 42 – Diagrama de Sequência Emissão de suprimento de caixa.	149
Figura Anexo 43 – Diagrama de Atividade Emissão de sangria de caixa.	150
Figura Anexo 44 – Diagrama de Sequência Emissão de sangria de caixa.	150
Figura Anexo 45 – Diagrama de Atividade Emissão de relatórios.	152
Figura Anexo 46 – Diagrama de Sequência Emissão de relatórios.	152

LISTA DE TABELAS

Tabela 1 – Estrutura do Zend Framework.....	35
Tabela 2 – Questionário de validação do sistema.....	82
Tabela Anexo 1 – Questionário de validação do sistema.	153

LISTA DE QUADROS

Quadro 1 – Caso de Uso Gerência de produto.	55
Quadro 2 – Caso de Uso Gerência de pessoa.	58
Quadro 3 – Caso de Uso Abertura de caixa.	60
Quadro 4 – Caso de Uso Fechamento de caixa.	62
Quadro 5 – Caso de Uso Emissão de venda.	64
Quadro 6 – Declaração do componente BemaFisc.ocx no código HTML.	74
Quadro 7 – Verificação do produto digitado em AJAX.	75
Quadro 8 – Declaração da função de impressão do item na impressora fiscal.	75
Quadro Anexo 1 – Caso de Uso Gerência de usuários.	89
Quadro Anexo 2 – Caso de Uso Gerência de rede de lojas.	93
Quadro Anexo 3 – Caso de Uso Gerência de loja.	96
Quadro Anexo 4 – Caso de Uso Gerência de terminal.	99
Quadro Anexo 5 – Caso de Uso Gerência de divisão.	101
Quadro Anexo 6 – Caso de Uso Gerência de artigo.	105
Quadro Anexo 7 – Caso de Uso Gerência de marca.	108
Quadro Anexo 8 – Caso de Uso Gerência de modelo.	111
Quadro Anexo 9 – Caso de Uso Gerência de material.	114
Quadro Anexo 10 – Caso de Uso Gerência de cor.	117
Quadro Anexo 11 – Caso de Uso Gerência de classificação fiscal.	120
Quadro Anexo 12 – Caso de Uso Gerência de unidade de medida.	123
Quadro Anexo 13 – Caso de Uso Gerência de preço.	126
Quadro Anexo 14 – Caso de Uso Gerência de forma de pagamento.	129
Quadro Anexo 15 – Caso de Uso Gerência de vendedor.	132
Quadro Anexo 16 – Caso de Uso Gerência de consumidor.	135
Quadro Anexo 17 – Caso de Uso Gerência de fornecedor.	139
Quadro Anexo 18 – Caso de Uso Emissão de ordem compra.	143
Quadro Anexo 19 – Caso de Uso Recebimento de produto.	145
Quadro Anexo 20 – Caso de Uso Emissão da leitura da memória da impressora fiscal.	146
Quadro Anexo 21 – Caso de Uso Emissão de suprimento de caixa.	148
Quadro Anexo 22 – Caso de Uso Emissão de sangria de caixa.	150
Quadro Anexo 23 – Caso de Uso Emissão de relatórios.	151

LISTA DE ABREVIATURAS E SIGLAS

AJAX	<i>Asynchronous Javascript And XML</i>
COTEPE	Comissão Técnica Permanente
CSS	<i>Cascading Style Sheet</i>
DBA	<i>DataBase Administrator</i>
DER	Diagrama de Entidade Relacionamento
DLL	<i>Dynamic Link Library</i>
ECF	Emissor de Cupom Fiscal
ERP	<i>Enterprise Resource Planning</i>
HTTP	<i>Hyper Text Transfer Protocol</i>
ICMS	Imposto sobre operações relativas à Circulação de Mercadorias de Serviços de Transporte Interestadual e Intermunicipal e de Comunicação
IP	<i>Internet Protocol</i>
IPi	Imposto sobre Produtos Industrializados
JSON	<i>JavaScript Object Notation</i>
MVC	<i>Model View Controler</i>
PAF	Programa Aplicativo Fiscal
PDF	<i>Portable Document Format</i>
PDV	Ponto de Venda
SGBD	Sistema Gerenciador de Banco de Dados
SGML	<i>Standard Generalized Markup Language</i>
SMTP	<i>Simple Mail Transfer Protocol</i>
SOAP	<i>Simple Object Acess Protocol</i>
SQL	<i>Structure Query Language</i>
SSH	<i>Secure Shell</i>

SSL	<i>Secure Socket Layer</i>
TCP	<i>Transmission Control Protocol</i>
UDDI	<i>Universal Description, Discovery and Integration</i>
UML	<i>Unified Modeling Language</i>
URL	<i>Uniform Resource Locator</i>
W3C	<i>World Wide Web Consortium</i>
WEB	<i>World Wide Web</i>
WSDL	<i>Web Service Description Language</i>
XML	<i>Extensible Markup Language</i>

SUMÁRIO

INTRODUÇÃO.....	15
1 AMBIENTE ATUAL X DESEJADO.....	19
1.1 AMBIENTE ATUAL.....	19
1.1.1 Interface.....	22
1.2 O QUE SE ESPERA DO NOVO AMBIENTE.....	23
1.3 ESTUDO DE VIABILIDADE	26
1.3.1 Viabilidade técnica	26
1.3.2 Viabilidade econômica	27
1.3.3 Viabilidade operacional	27
2 NOVAS TECNOLOGIAS	28
2.1 CARACTERÍSTICAS DO AMBIENTE WEB.....	28
2.2 LINGUAGEM PHP	29
2.2.1 Vantagens	30
2.2.2 Orientação a objetos com PHP	30
2.3 <i>FRAMEWORK</i> ZEND.....	32
2.3.1 Características.....	32
2.3.2 Arquitetura.....	33
2.4 SGBD	36
2.4.1 Vantagens	36
2.4.2 PostgreSQL.....	37
2.5 WEBSERVICES.....	38
2.5.1 Fundamentos	39
2.6 SOAP.....	40
2.6.1 Mensagens SOAP	41
2.6.2 Modelo de troca de mensagens SOAP	42
3 PROJETO DO SOFTWARE	44
3.1 LINGUAGEM DE MODELAGEM UNIFICADA – UML.....	44
3.2 LEVANTAMENTO E ANÁLISE DE REQUISITOS	46
3.2.1 Requisitos funcionais.....	47
3.1.2 Requisitos não funcionais.....	51

3.3 CASOS DE USO	52
3.3.1 Diagrama de casos de uso.....	52
3.3.2 Casos de Uso Estendidos	53
3.4 DIAGRAMA DE CLASSES	65
3.5 DIAGRAMA DE SEQUÊNCIA	67
3.5.1 Diagrama de Sequência – Gerência de produto.....	67
3.5.2 Diagrama de Sequência – Gerência de pessoa	68
3.5.3 Diagrama de Sequência – Abertura de caixa	68
3.5.4 Diagrama de Sequência – Fechamento de caixa	69
3.5.5 Diagrama de Sequência – Emissão de venda.....	69
3.6 DIAGRAMA DE ENTIDADE RELACIONAMENTO – DER	71
4 DESENVOLVIMENTO DO SISTEMA	73
4.1 LINGUAGEM DE PROGRAMAÇÃO	73
4.2 AMBIENTE DE DESENVOLVIMENTO.....	73
4.3 SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS	74
4.4 COMUNICAÇÃO COM A IMPRESSORA FISCAL.....	74
4.5 DESENVOLVIMENTO DO MÓDULO DE FRENTE DE CAIXA	75
4.6 VALIDAÇÃO DO SISTEMA.....	81
CONCLUSÃO.....	83
REFERÊNCIAS BIBLIOGRÁFICAS	85
ANEXO	88

INTRODUÇÃO

A importância do varejo na economia brasileira vem ganhando cada vez mais destaque e reconhecimento. Além de maior gerador de empregos formais no país, o setor apresenta números expressivos de crescimento e consistentes indicadores de modernização. Os resultados favoráveis refletem, principalmente, em sucessivos ganhos reais dos trabalhadores, o aumento das transferências governamentais e as melhores condições de crédito (BOLETIM REGIONAL DO BANCO CENTRAL DO BRASIL, 2008, p. 86).

Para Saiani (2007), o varejo e as áreas de serviços serão os grandes empregadores deste século, pois não há mais perspectiva de que o emprego gerado pelas indústrias de manufatura cresça, desta forma o varejo no Brasil terá papel econômico fundamental nos próximos anos. O cenário atual apresenta um constante crescimento do setor, conforme descrito no Boletim Regional do Banco Central do Brasil (2008, p. 88), “para o decorrer dos próximos anos, espera-se que o comércio varejista mantenha resultados positivos no país, principalmente em razão da esperada continuidade da expansão da economia brasileira, alicerçada no crescimento do crédito, do emprego e da massa salarial real”.

A partir destas informações é compreensível que empresas do setor cada vez mais necessitem de softwares que controlem suas atividades de forma prática e confiável, e ainda que estejam de acordo com a legislação que regula os sistemas de frente de caixa (PDV's): Ato COTEPE/ICMS de setembro de 2008.

A utilização de desenvolvimento de aplicações web através do paradigma de orientação a objetos está cada vez mais presente nas empresas, deixando de ser apenas uma ferramenta para desenvolvimento de sites e passando a ser solução para vários sistemas, como ERP's (*Enterprise Resource Planning*), por exemplo. Tal paradigma e ambiente podem ser empregados no ramo varejista.

Welling e Thomson (2005) defendem que uma das principais linguagens de desenvolvimento web é o PHP, pois oferece alto desempenho, baixo custo, ótimo suporte a orientação a objetos e portabilidade. Converse (2001) afirma que o PHP pode reduzir custos de mão de obra e aumentar a eficiência, além disso, o PHP se destaca pela sua capacidade, confiabilidade e facilidade de uso.

O desenvolvimento de software, a partir do paradigma de orientação a objetos tem a finalidade de diminuir o custo e manutenção da criação de sistemas, além de acelerar seu desenvolvimento (LISBOA, 2008).

A orientação a objetos é uma maneira de programar, que modela os processos de programação de uma maneira próxima a realidade, tratando a cada componente de um programa como um objeto com suas características e funcionalidades. O tratamento de objetos no PHP 5 foi totalmente reescrito, permitindo a utilização de

uma maior quantidade de recursos da Programação Orientada a Objetos, melhor performance e mais vantagens na implementação deste tipo de programação (XAVIER, 2008, p.137).

A adoção de software livre para desenvolvimento de sistemas com o intuito de reduzir custos com tecnologia da informação (TI), torna-se um diferencial para as empresas que adotam tal metodologia.

Do ponto de vista macroeconômico, a adoção de software livre permite reduzir drasticamente o envio de royalties pelo pagamento de licenças de software, gerando maior sustentabilidade do processo de inclusão digital da sociedade brasileira e de informatização e modernização das empresas e instituições (SILVEIRA, 2004, p. 39).

Este trabalho será realizado em uma empresa nacional de grande porte do ramo de produção de calçados e acessórios que, a partir do ano de 2007, iniciou sua atuação no ramo varejista, com a abertura de lojas onde são vendidos exclusivamente produtos produzidos pela própria empresa.

Esta empresa utiliza, desde o ano de 1995, um modelo de sistema de uma só camada (arquitetura servidor – terminal). Este sistema roda na plataforma Unix, a linguagem de programação é o Clipper e a armazenagem de dados é feita através de arquivos simples (.dbf). A empresa não possui programas para integração com outros sistemas, assim como SGBD e ferramentas para auxiliar o desenvolvimento do sistema, como *frameworks*.

O sistema atual atende as necessidades básicas, porém é limitado em vários aspectos como integração com outros sistemas, utilização de recursos visuais, gráficos, interação com protocolos, etc. Outra peculiaridade do módulo de varejo é a integração com hardwares específicos: impressoras fiscais e de cheques, além de maquinetas de cartão de crédito. Esses hardwares usam as bibliotecas do sistema operacional para a comunicação com o software e também nesse aspecto o sistema apresenta dificuldades, pois não consegue interagir com essas bibliotecas. A fim de solucionar esta questão, foram encontradas soluções “paliativas” para utilizar tais recursos de hardware.

A empresa está em expansão nos setores de varejo e produção e, diante deste cenário, tem a intenção de migrar seus sistemas e, assim, surgiu a oportunidade de implementar um sistema de varejo desenvolvido em PHP, com banco de dados PostgreSQL e utilização exclusiva de software livre, usando recursos como orientação a objetos, *framework* e MVC¹.

¹ MVC (*Modelo View Controller*): é um padrão de arquitetura de software composto por três tipos de objetos: o Modelo é o objeto de aplicação, a Visão é a apresentação na tela e o Controlador define a maneira como a interface do usuário reage às entradas do mesmo.

Frameworks estão se tornando cada vez mais comuns e importantes. Eles representam a maneira pela qual sistemas orientados a objetos conseguem um maior grau de reutilização. Aplicações de grande porte, orientadas a objetos, terminarão por constituir-se de camadas de *frameworks* que cooperam uns com os outros para atender as necessidades do domínio da aplicação (GAMMA, 2000, p. 43).

A decisão por usar um framework de desenvolvimento reside na necessidade de estruturar os projetos de software, devido à grande complexidade que os mesmos alcançaram. Um projeto estruturado reduz custos, aumenta a qualidade da aplicação e reduz o tempo de desenvolvimento (LISBOA, 2008, p 16).

Existem vários *frameworks* de desenvolvimento para PHP, um deles, desenvolvido para PHP *open source*, é o Zend Framework. Para Lisboa (2008), este *framework* é baseado na simplicidade, utiliza as melhores práticas de orientação a objetos, tem uma licença corporativamente amigável e uma base de código ágil rigorosamente testada. Ainda segundo Lisboa, o foco do Zend Framework é a construção de aplicações e *web services* mais seguros, confiáveis e modernos.

Web services são ferramentas utilizadas na integração e comunicação entre aplicações distintas e de diferentes plataformas, através de dados em formato XML. “Um *web service* é qualquer serviço de troca de mensagens através da Internet usando o padrão de XML e que não está vinculado a um sistema operacional ou linguagem de programação”. (CERAMI, 2002, p. 3, tradução nossa). A futura troca de informações entre o sistema atual e sistema pretendido poderá ser feita através de *web services* pelo protocolo SOAP, trazendo eficiência e agilidade no processo de comunicação entre os diferentes sistemas. Cunha (2002, p. 1) enfatiza:

SOAP é um protocolo projetado para invocar aplicações remotas através de RPC (*Remote Procedure Calls* – Chamadas Remotas de Procedimento) ou trocas de mensagens, em um ambiente independente de plataforma e linguagem de programação. SOAP é, portanto, um padrão normalmente aceito para utilizar-se com *Web Services*. Desta forma, pretende-se garantir a interoperabilidade e intercomunicação entre diferentes sistemas, através de uma linguagem (XML) e mecanismo de transporte (HTTP) padrões.

A metodologia empregada no desenvolvimento deste trabalho poderá posteriormente abrir caminho para que outros sistemas da empresa possam utilizar dos mesmos recursos, assim como outras empresas.

A motivação para o desenvolvimento deste trabalho, está na importância da aquisição de conhecimento de melhores técnicas e práticas para análise e desenvolvimento de software, bem como a possibilidade de prover a solução para as diversas empresas do ramo que necessitam desse tipo de sistema.

No primeiro capítulo deste trabalho são apresentados os aspectos do cenário tecnológico atual da empresa, assim como a descrição dos objetivos que o projeto pretende alcançar. Procurou-se fazer uma comparação entre o ambiente atual e o ambiente desejado, dando ênfase aos contrastes tecnológicos. Nesse capítulo também é feito estudo de viabilidade técnica, econômica e operacional para o sistema proposto. Já o segundo capítulo trata das características para as tecnologias pretendidas no desenvolvimento deste projeto, onde linguagem de programação, *framework*, banco de dados e *web services* são apresentados. Logo após, o capítulo 3, descreve o projeto do software, abordando o levantamento e análise de requisitos, casos de uso e a modelagem de dados utilizando a UML 2.0 para o sistema proposto. No quarto capítulo, são apresentadas as ferramentas de desenvolvimento para o software, telas do sistema e, além disso, é descrito o resultado da validação do sistema. Finalmente, são apresentados a conclusão do trabalho e os anexos que fazem parte do projeto.

1 AMBIENTE ATUAL X DESEJADO

Nesse capítulo serão apresentados os aspectos do ambiente atual da empresa e as expectativas com relação ao desenvolvimento do sistema projetado, comparando as tecnologias, além de um estudo de viabilidade para a implantação do mesmo.

1.1 AMBIENTE ATUAL

Atualmente a empresa apresenta um ambiente tecnológico em que aspectos como arquitetura, linguagem de programação e gerenciamento de banco de dados, necessitam de atualizações. A busca por novos recursos pode ser vital para o futuro do bom desempenho na área de TI.

O sistema operacional utilizado pela empresa, em seus servidores, é o Unix. Este sistema foi desenvolvido no início da década de 1970 e segundo Tanenbaum (1992, p.184), é um sistema interativo de tempo compartilhado projetado para ser utilizado por usuários com experiência em projetos de desenvolvimento de software, projetos estes muitas vezes complexos.

Salemi (1995, p. 51) cita:

Criticado como um sistema operacional 'técnico', o Unix pode parecer difícil de entender na visão de muitos usuários de computadores. Essa crítica tem algum fundamento, e você deverá levá-la em consideração no seu processo decisório; além do custo extra da plataforma da estação de trabalho Risc, você também poderá ter dificuldade para encontrar pessoal de suporte familiarizado com o sistema operacional.

A linguagem de programação utilizada no sistema da empresa é o Clipper, sendo que a armazenagem de dados é feita através de arquivos simples (dbf). O método de desenvolvimento dos programas Clipper da empresa é procedural, portanto, orientação a objetos não faz parte deste contexto. Camara (2006, p. 3), cita que as principais desvantagens da programação procedural, com relação à programação orientada a objetos, são: a falta de conceitos de reutilização de código, manutenção complexa e códigos desorganizados e de difícil compreensão.

Além disso, esta linguagem possui limitações, como integração com outros sistemas, por exemplo. Isso dificulta bastante o trabalho da equipe de desenvolvimento, pois o caminho para encontrar as soluções necessárias é, muitas vezes, extremamente complexo. O sistema é formado por vários programas executáveis, e para cada um desses programas existe um arquivo código fonte. Cada um desses executáveis executa uma rotina.

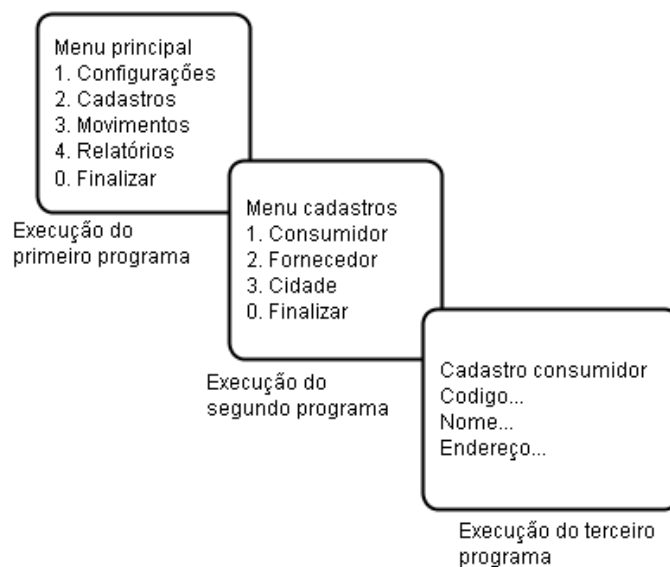


Figura 1 - Ao invocar o Cadastro de Consumidor, são executados três aplicativos.
Fonte: Autor (2009).

Os dados são armazenados em arquivos simples. Cada um destes arquivos representa uma tabela independente, ou seja, não há gerenciamento relacional, assim como integração, verificação de integridade relacional, segurança e compartilhamento de dados, comuns em um SGBD. O controle das informações é implementado totalmente dentro da aplicação. Redundância, inconsistência de dados e falta de padronização são comuns neste cenário.

As estações de trabalho possuem sistema operacional Windows XP e o acesso aos dados é feito através de terminal (arquitetura de uma só camada) utilizando o protocolo de comunicação *Telnet* via SSH. O servidor fica centralizado na empresa e para todas as operações há uma solicitação da estação de trabalho e uma resposta do servidor.

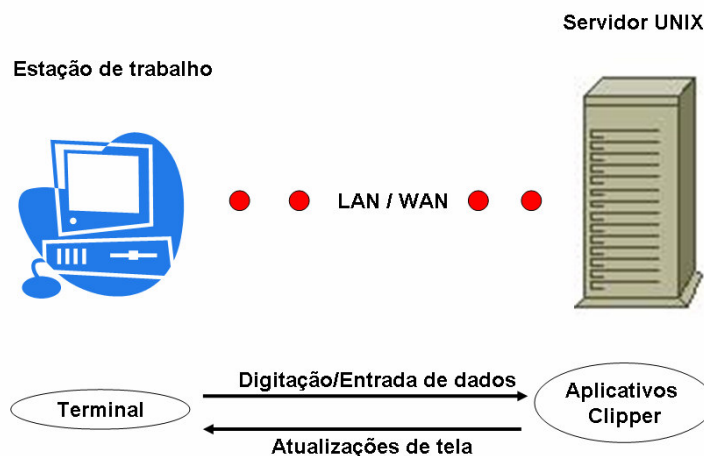


Figura 2 – Arquitetura básica ambiente atual.
Fonte: Autor (2009).

Por fazer o uso de terminal nas estações de trabalho, toda a aplicação é executada no servidor e não é possível a interação da aplicação com o sistema operacional da estação de trabalho. Como hardwares específicos de um estabelecimento de varejo utilizam as bibliotecas do sistema, o uso de tais recursos é limitado. Um exemplo disso é a comunicação com a impressora fiscal. O fornecedor deste hardware oferece de forma gratuita uma DLL (*Dynamic Link Library*) com todas as funções que executam as transações entre o aplicativo e o equipamento. Porém a solução para o sistema que não interage com tais bibliotecas é a comunicação direta, ou seja, é necessário enviar para a impressora uma sequência de comandos através da *interface* serial que corresponde à operação solicitada.

Conforme descrito no manual “Utilizando comunicação direta com a impressora fiscal Bematech” (BEMATECH, 2009), a comunicação serial utiliza o protocolo RTS/CTS padrão RS232 e, neste caso, a impressora fiscal será conectada em uma porta *serial* (livre) do computador, e a impressão do cupom fiscal deverá ser efetuada através de um software/aplicativo usando um *set* de comandos específicos da impressora. Os comandos enviados à impressora, através do protocolo direto, podem utilizar o sistema de numeração decimal ou em hexadecimal, e o envio desses comandos deve ser: *byte a byte*.

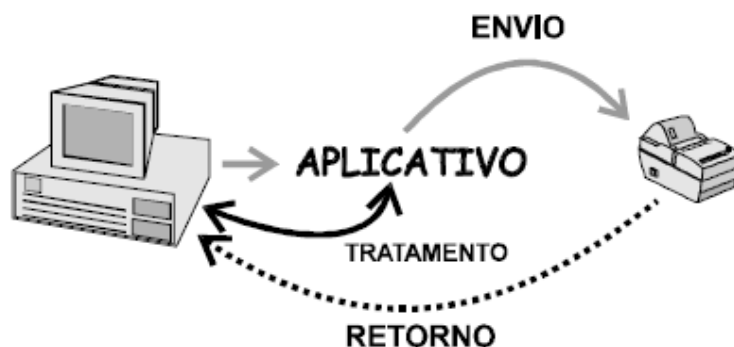


Figura 3 – Tratamento de retorno da impressora fiscal através da comunicação serial.
Fonte: Bematech (2009).

Quando é enviado um comando à impressora, a mesma devolve via interface serial, bytes que correspondem ao estado de execução deste comando. É importante que a aplicação leia esses bytes, pois é através deles que são identificados os estados da impressora, bem como a execução ou não do comando enviado. Este retorno será dado em Hexadecimal, exemplo: 6h 40h 00h (BEMATECH, 2009).

1.1.1 Interface

O sistema atual é desenvolvido em um ambiente caractere, desta forma, não possui aspectos visuais dos elementos das interfaces (cor, tamanho, leiaute). Para Paula Filho (2003, p. 390), “uma interface eficaz é transparente para os usuários. Ela permite a concentração na tarefa que deve ser realizada, e não nos mecanismos oferecidos pelo produto”.

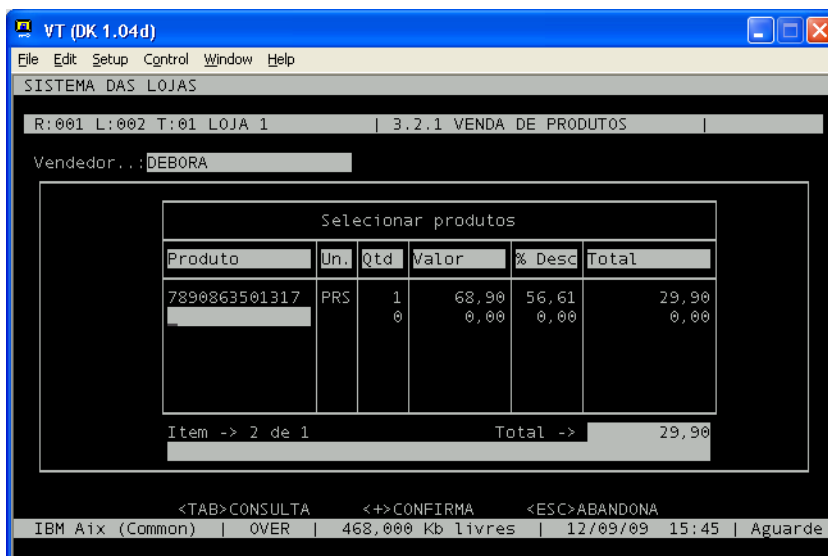


Figura 4 – Tela de venda de produtos do sistema atual.
Fonte: Autor (2009).

Muitos sistemas interativos são difíceis de usar por causa da própria complexidade do domínio da aplicação [...] Os símbolos e ícones devem ser significativos dentro do domínio da aplicação. As tarefas complexas devem ser subdivididas até o nível

necessário para um bom entendimento. Não deve ser oferecida aos usuários uma quantidade excessiva de campos e comandos. É preferível identificar um conjunto de funções de uso frequente e investir em torná-las mais fáceis. As interfaces principais devem conter apenas os campos mais usados, colocando-se os demais em interfaces secundárias, acionadas somente quando necessário (PAULA FILHO, 2003, p. 389).

Atualmente não são utilizados recursos gráficos para a visualização das informações. Para Nascimento e Ferreira (2005, p. 1263), a visualização das informações através de recursos gráficos visa facilitar seu entendimento, assim como auxiliar na descoberta de novas informações contidas nos mesmos. A Figura 5 mostra um relatório extraído do sistema atual.

```

REDE DE LOJAS
CONTAS A PAGAR
3.1 Relatório de despesas - LOJA 1 - Período: 01/05/08 a 30/04/09
-----

```

	ALUGUEL	CONDOMINIO LUZ	TELEFONE	MARKRING	
MAIO	1400,00	630,00	540,00	290,00	650,00
JUNHO	1400,00	700,00	490,00	320,00	300,00
JULHO	1400,00	820,00	475,00	390,00	350,00
AGOSTO	1400,00	750,00	490,00	345,00	400,00
SETEMBRO	1400,00	690,00	600,00	330,00	700,00
OUTUBRO	1500,00	700,00	540,00	330,00	750,00
NOVEMBRO	1500,00	650,00	610,00	320,00	850,00
DEZEMBRO	1500,00	800,00	600,00	330,00	600,00
JANEIRO	1500,00	670,00	475,00	345,00	400,00
FEVEREIRO	1500,00	640,00	510,00	270,00	450,00
MARCO	1500,00	700,00	550,00	300,00	510,00
ABRIL	1500,00	650,00	475,00	350,00	520,00

Figura 5 – Relatório de despesas do sistema atual.
Fonte: Autor (2009).

1.2 O QUE SE ESPERA DO NOVO AMBIENTE

Através do desenvolvimento deste projeto, pretende-se criar um ambiente onde serão utilizadas tecnologias avançadas na área de TI da empresa, visando auxiliar o melhor desenvolvimento de software, a interação homem-máquina e a redução de custos.

Para a área de desenvolvimento de software, busca-se maior produtividade, através da utilização de *framework*, orientação a objetos e do padrão MVC. Um *framework* de desenvolvimento é, conforme descreve Lisboa (2008, apud MINETTO 2007), “uma base onde se pode desenvolver algo maior ou mais específico. É uma coleção de códigos fonte, classes, funções, técnicas, e metodologias que facilitam o desenvolvimento de novos softwares”.

A linguagem de programação utilizada será o PHP 5.0, desenvolvido com Zend Framework, com o uso do paradigma de orientação a objetos. Assim, pretende-se aumentar a produtividade do desenvolvimento de software, além de utilizar uma linguagem de

programação eficiente e que oferece portabilidade: capacidade do software de ser transferido de um ambiente para outro.

O gerenciamento dos dados será realizado através do banco de dados PostgreSQL, desta forma, será usado um SGBD Relacional auxiliando na manipulação, acesso, segurança e integridade das informações dos sistemas da empresa, além da extração de dados relacionados, possibilitando novas descobertas.

Dias (2007, p. 28), descreve que

a usabilidade pode ser considerada como uma qualidade de uso, isto é, qualidade de interação entre usuário e sistema. Para o novo sistema, é desejável uma melhor usabilidade, através da criação de um software mais interativo que o atual, e que seja funcional e eficiente [...] Um sistema interativo é considerado eficaz quando possibilita que os usuários atinjam seus objetivos. A eficácia é a principal motivação que leva um usuário a utilizar um produto ou sistema. Se um sistema é fácil de usar, de aprender e mesmo agradável ao usuário, mas não consegue atender a objetivos específicos, ele não será usado, mesmo que seja oferecido gratuitamente.

Através dos recursos utilizados, será possível que o aplicativo interaja com as bibliotecas do sistema operacional da estação de trabalho, tornando possível à integração com os hardwares específicos de utilização no varejo.



Figura 6 – Comunicação com a impressora fiscal através da DLL Bematech.
Fonte: Autor (2009).

Existem vários aspectos jurídicos que devem ser observados no desenvolvimento de software voltado ao ramo varejista, principalmente ao sistema de frente de caixa. O sistema deverá estar de acordo com a legislação que regula o Programa Aplicativo Fiscal – Emissor de Cupom Fiscal (PAF-ECF). São dois os documentos publicados pela Secretaria Nacional da Fazenda: Ato COTEPE 06/08 e o Convênio ICMS 15/08. Estes documentos são de abrangência nacional (LUIZE, 2008).

Portanto, implantando este novo projeto, será possível criar uma melhor visualização das informações com o uso de recursos visuais, como gráficos de linhas, barras, pizza, além de outros como mapas e paisagens. Segundo Nascimento e Ferreira (2005, p. 1263), “o processo de visualização está relacionado com a transformação de algo abstrato em imagens

(mentais ou reais) que possam ser visualizadas pelos seres humanos. O objetivo final é auxiliar no entendimento de determinado assunto, o qual, sem uma visualização, exigiria maior esforço para ser compreendido”.

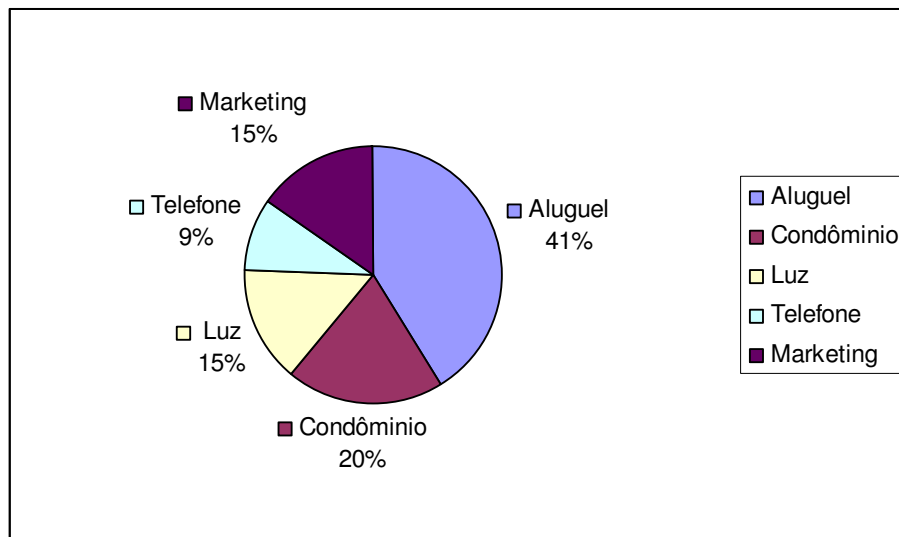


Figura 7 – Visualização do relatório de despesas através de um gráfico de pizza.
 Fonte: adaptado de Nascimento e Ferreira (2005).

Através da Figura 7, é possível analisar com facilidade a representação percentual de cada uma das despesas no período.

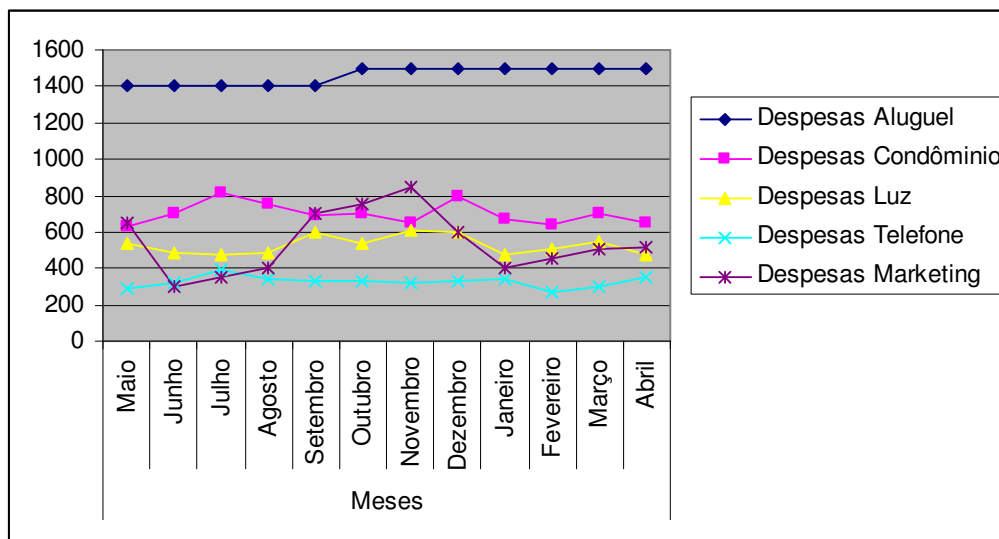


Figura 8 – Visualização do relatório de despesas através de um gráfico de linhas.
 Fonte: adaptado de Nascimento e Ferreira (2005).

A Figura 8 apresenta uma visualização mais detalhada das despesas. Através dele é possível responder com facilidade questões como: “qual despesa oscilou constantemente com o tempo?”, “qual despesa apresentou uma tendência de aumento?” (Adaptado de NASCIMENTO; FERREIRA, 2005).

A constatação de padrões ou de características visuais presentes em imagens contribuem de forma muito mais significativa para o processo de compreensão do que a simples observação dos dados em sua forma bruta. “Em geral, visualizações acabam possibilitando a recuperação de informações relevantes e a construção de novos conhecimentos” (NASCIMENTO; FERREIRA, 2005, p. 1266).

Com a utilização de software livre, este projeto busca-se a redução de custos com as licenças de uso, além de eliminar os custos das ferramentas de desenvolvimento.

Desta forma, o desenvolvimento deste projeto busca trazer novos recursos de TI para a empresa, apresentar novos conceitos e paradigmas, tornar o sistema mais eficiente e funcional, e, além disso, diminuir os custos desta através da adoção de software livre.

1.3 ESTUDO DE VIABILIDADE

Segundo Buarque (1984, p. 25), a realização de um projeto, desde a ideia inicial até o seu funcionamento, é um processo contínuo no tempo, através de sucessivas fases, nas quais se combinam considerações de caráter técnico, econômico e financeiro estudadas através de diferentes etapas. Tonsig (2008, p. 150), afirma que em um estudo de viabilidade encontram-se três etapas distintas: viabilidade técnica, viabilidade econômica e viabilidade operacional.

1.3.1 Viabilidade técnica

Neste quesito, “são avaliados os recursos técnicos disponíveis, próprios ou não, que podem viabilizar as soluções que estão sendo analisadas. Estes recursos restringem-se ao hardware, software e pessoas qualificadas” (TONSIG, 2008, p. 150).

A empresa possui uma excelente infra-estrutura a nível de hardware, e estes atenderão de forma completa as necessidades do ambiente proposto. Todo o software para o desenvolvimento do projeto deverá ser adquirido através de *download* gratuito na internet.

O desenvolvimento do sistema proposto foi iniciado pelo autor durante o desenvolvimento deste trabalho com o módulo de frente de caixa. Para o desenvolvimento dos demais módulos, deverão ser alocadas duas pessoas para o desenvolvimento e manutenção do sistema. Essas pessoas deverão receber treinamento específico das novas ferramentas de desenvolvimento.

1.3.2 Viabilidade econômica

Aqui, “para as possíveis soluções tecnicamente disponíveis e utilizáveis, é verificado o montante financeiro a ser aplicado” (TONSIG, 2008, p. 150).

Não será necessário nenhum investimento a nível de hardware para o projeto. Toda a infra-estrutura já existe na empresa. Todo o software utilizado será *open source*, desta forma não haverá gastos com aquisição e manutenção de software. Além disso, toda sua atualização, feita periodicamente (uma ou duas vezes ao ano), será de forma gratuita.

Já existe uma equipe exclusiva de desenvolvimento da área de varejo da empresa. Essa equipe dará continuidade ao projeto recebendo os mesmos valores atuais.

1.3.3 Viabilidade operacional

Aqui são relacionados os “aspectos das soluções que implicarão mudanças de rotinas existentes ou a serem criadas e seus impactos na organização e sociedade. Entram aqui discussões de ordem ética e moral” (TONSIG, 2008, p. 150).

Haverá quebra de paradigmas na área de desenvolvimento da empresa: migração da programação estruturada para programação orientada a objetos, utilização de SGBD e *framework*. Este poderá ser um dos principais empecilhos para o sucesso deste projeto. Desta forma, as pessoas envolvidas deverão estar cientes destas alterações.

A migração da interface caractere para a interface gráfica deverá tornar o software mais amigável ao usuário, facilitando a usabilidade e aumentando a eficiência dos processos. O sistema deverá trazer ao usuário maior facilidade na interpretação dos dados, auxiliando nas suas tomadas de decisão.

A alteração para um novo sistema operacional terá impacto sobre o trabalho diário do usuário final. Este, portanto, deverá receber o treinamento e a atenção necessária, buscando viabilizar o sucesso do presente projeto.

2 NOVAS TECNOLOGIAS

Neste capítulo serão apresentadas as tecnologias que formam o ambiente do sistema proposto nesta pesquisa.

2.1 CARACTERÍSTICAS DO AMBIENTE WEB

Cummins (2002, p. 9), afirma que uma interface universalmente disponível a todo usuário que possua uma estação de trabalho é fornecida através de navegadores web, isso significa que um sistema habilitado para web pode ser acessado facilmente por qualquer pessoa, sendo esse um recurso de valor dentro da empresa e fora dela.

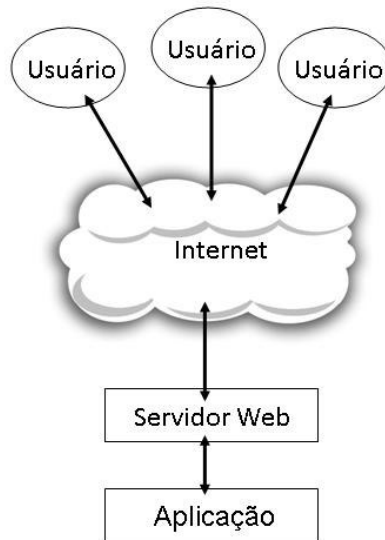


Figura 9 – Típica aplicação habilitada para web.

Fonte: adaptado de Cummins (2002, p. 10).

Para Cummins (2002, p. 9), tornar aplicações habilitadas para web resolve uma porção significativa de dificuldades envolvidas no acesso à informação pela empresa e, além disso, os usuários não se limitam a acessar dados de dispositivos específicos localizados em um escritório, pois podem acessar muitos sistemas de qualquer parte do mundo pela internet.

Conforme descrito por Welling e Thomson (2005, p. 162-163), a operação básica de um servidor web consiste em dois objetos: um navegador web e um servidor web, sendo necessário um link de comunicação entre eles. Essa arquitetura atende bem a um servidor que fornece páginas estáticas, porém para uma típica transação de banco de dados, é um pouco mais complexa e consiste nas seguintes etapas:

1. Um navegador web faz uma solicitação ao servidor;
2. O servidor recebe a solicitação e encaminha para processamento;
3. O script é analisado sintaticamente. Dentro do script há um comando para conectar-se ao banco de dados e executar uma rotina;
4. O servidor recebe a instrução de banco de dados, a processa e devolve os resultados;
5. A execução do script é finalizada, os dados são formatados e retornados para o servidor web;
6. O servidor web repassa o resultado para o navegador, onde as informações podem ser vistas pelo usuário.

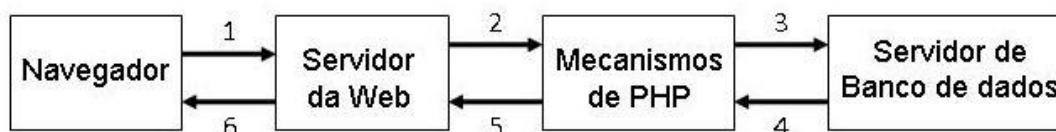


Figura 10 - A arquitetura básica de banco de dados web.
Fonte: adaptado de Welling e Thomson (2005, p. 163).

Welling e Thomson (2005, p. 163), defendem que o processo é basicamente o mesmo independente de quais mecanismos de criação de script ou servidor de banco de dados é utilizado. Frequentemente, o software de servidor web, o mecanismo de PHP e o servidor de banco de dados estão na mesma máquina, porém, também é bastante comum o servidor de banco de dados executar em uma máquina diferente, por motivos como segurança, aumento de capacidade ou distribuição de carga.

Essa arquitetura possui uma camada intermediária entre o cliente e o servidor de banco de dados. Elmasri e Navathe (2005, p. 31), defendem que a camada, chamada de servidor web, desempenha o papel de armazenar as regras de negócio (procedimentos ou restrições) que são usadas para acessar os dados do servidor de banco de dados, além de poder incrementar a segurança do banco de dados checando as credenciais do cliente antes de enviar uma solicitação ao servidor de banco de dados.

2.2 LINGUAGEM PHP

Conforme citado no Manual do PHP (2009), o PHP é uma linguagem script *open source* de ampla utilização, interpretada, utilizada especialmente para o desenvolvimento web e pode ser embutida no código HTML. Para Lerdorf (2000, p. 1), esta linguagem de

programação permite inserir instruções em arquivos HTML a fim de criar conteúdo dinâmico, tais instruções são lidas pelo *web server* porém não serão exibidas pelo *browser*.

2.2.1 Vantagens

Para Welling e Thomson (2005), as principais vantagens do PHP são:

- Alto desempenho: o PHP é muito eficiente. Utilizando um único servidor é possível atender a milhões de acesso diários;
- Integração de banco de dados: o PHP tem conexões nativas disponíveis para muitos sistemas de banco de dados, entre os principais estão PostgreSQL, MySQL, mSQL, Oracle, dbm, FilePro, HyperWave, Informix e Sybase. O PHP 5 também tem uma interface integrada SQL para um arquivo simples, chamada SQLite;
- Bibliotecas integradas: possui várias funções integradas para realizar muitas tarefas úteis relacionadas a web. Pode-se gerar imagens instantâneas em diversos formatos, conectar-se a outros serviços de rede, enviar e-mail, trabalhar com *cookies* e gerar documentos PDF, tudo isso com facilidade;
- PHP: é gratuito. Pode-se fazer o *download* através do site oficial do PHP: <http://www.php.net>;
- Suporte orientado a objetos: a versão 5 do PHP possui recursos orientados a objetos, tal como herança, atributos e métodos privados e protegidos, classes e métodos abstratos, interfaces, construtores e destruidores, muito bem elaborados;
- Portabilidade: disponibilidade para muitos sistemas operacionais diferentes;
- Código fonte: ao contrário dos produtos comerciais e de código fonte fechado, se houver algo a modificar ou adicionar à linguagem, o acesso é livre, sem precisar esperar o fabricante lançar os *patches*;
- Disponibilidade de suporte: a Zend Technologies, a empresa responsável pelo mecanismo que mantém o PHP, financia o desenvolvimento do PHP oferecendo suporte e softwares relacionados em uma base comercial.

2.2.2 Orientação a objetos com PHP

Orientação a Objetos “significa que organizamos o software como uma coleção de objetos distintos, que incorporam estrutura de dados e comportamento” (BLAHA; RUMBAUGH, 2006, p. 1). Ainda de acordo com Blaha e Rumbaugh (2006, p. 3), o desenvolvimento orientado a objetos é um meio de pensar a respeito de software com base em abstrações que existem no mundo real, e também no programa. Nesse contexto, desenvolvimento refere-se ao ciclo de vida de um software: análise, projeto e implementação. A identificação e a organização de conceitos da aplicação são fundamentais para o desenvolvimento orientado a objetos.

Para Camara (2006, p. 6), a programação orientada a objetos “quando corretamente aplicada em projetos de desenvolvimento de software trará resultados qualitativos à programação em manutenção dos códigos fontes”.

Leite (2006, p. 252), afirma que existem dois principais objetivos da programação orientada a objetos: reutilização de código, alcançado através de reaproveitamento direto de códigos já prontos e depurados, especialização de classes e tipos de dados reutilizáveis; e modularização, obtido com a definição da unidade fundamental da tecnologia: o objeto. “Objeto é uma abstração de alguma coisa em um domínio de problemas, exprimindo as capacidades de um sistema de manter informações sobre ela, interagir com ela, ou ambos; um emcapsulamento de valores e atributos e de seus serviços exclusivos” (LEITE, 2006, p. 252 apud COAD; YOURDON, 1991, p. 69).

O tratamento de objetos no PHP 5 foi totalmente reescrito, permitindo a utilização de uma maior quantidade de recursos da programação orientada a objetos, melhor performance e oferecendo vantagens na implementação desse tipo de programação. De acordo com Welling e Thomson (2005), algumas diferenças da versão 5 do PHP para versões anteriores são: passagem de objetos como referências por padrão, assim como a maioria das linguagens orientadas a objetos, o PHP, a partir dessa versão, se incluiu nessa categoria; aplicação de métodos que permitem que sejam chamados sem instanciar a classe; verificação do tipo de objeto, permitindo verificar se um objeto é uma instância de uma classe específica, se herda de uma classe ou se implementa uma interface; permite copiar um objeto existente, garantindo que os atributos de uma classe tratados como referências sejam copiados corretamente; criação de classes abstratas, oferecendo métodos abstratos, que fornecem assinatura para um método. O principal uso de classes e métodos abstratos é uma hierarquia complexa de classes, na qual se deseja garantir que cada subclasse contém e sobescreve algum método específico, isso também pode ser feito com interface; e a capacidade de examinar classes e objetos existentes para descobrir sobre sua estrutura e conteúdo.

2.3 FRAMEWORK ZEND

Zend Framework (ZF) é um framework open source para o desenvolvimento de aplicações web com PHP 5. O ZF é implementado usando 100% orientação a objetos. A estrutura de componentes do ZF é algo único: cada componente é projetado com algumas dependências de outros componentes. Este baixo acoplamento permite que os desenvolvedores utilizem os componentes individualmente” (ZEND TECHNOLOGIES INC, 2009 p. 1, tradução nossa).

“O Zend Framework é uma ferramenta de desenvolvimento para PHP *open source*, a licença do ZF é compatível com a BSD License, o que permite que softwares construídos com o *framework* possam ter código fonte proprietário” (LISBOA, 2008, p. 18), desta forma é possível manter o código fonte proprietário a empresa.

A idéia de utilizar um *framework* e melhores práticas de desenvolvimento é reduzir custos de treinamento e agilizar o tempo de colocação no mercado, dois fatores importantes em tomada de decisão. A construção de aplicações desta forma faz com que você escolha apenas as peças que precisa para o negócio de suas aplicações web (LISBOA, 2008, p. 20).

A empresa mantenedora do Zend Framework é a Zend Technologies. Conforme descrito por Lisboa (2008, p. 19), essa empresa foi fundada por dois contribuintes-chave do PHP: Andi Gutmans e Zeev Suraski, e tornou-se uma empresa líder tanto para a comunidade PHP quanto para outras comunidades *open source*. Ainda segundo Lisboa, a Zend mantém parcerias com grandes empresas de TI, como IBM e Oracle, o que fortalece o uso e o desenvolvimento da linguagem PHP.

2.3.1 Características

Lisboa (2008, p. 20), descreve quatro principais características que incorporam o Zend Framework:

- Extrema simplicidade e produtividade: projetado com a simplicidade em mente, fornece uma biblioteca de componentes leve e fracamente acoplada que fornece a maior parte das funcionalidades que todo o desenvolvedor necessita, permitindo que o mesmo customize aquilo que não atender plenamente. O espírito de simplicidade da programação PHP, implica em uma baixa curva de aprendizado – que, por sua vez, resulta em custos menores de treinamento, possíveis através de: uma base de códigos bem-testada e extensível, uma arquitetura flexível e a dispensa de arquivos de configuração para invocar e rodar a aplicação;

- Últimas características do desenvolvimento web: suporte a AJAX² utilizando o JSON³, edição PHP nativa do mecanismo de busca Lucene, formato de dados de acesso fácil e biblioteca de classes PHP 5 orientada a objetos e com alta qualidade;
- Licença segura e confiável: utiliza uma licença baseada na BSD License, a Zend Framework's License, a qual assegura que seu código é irrepreensível – está em conformidade com as melhores práticas e padrões – e se encontra protegido;
- Segurança extensível: testado unitariamente desde o começo com requisitos de cobertura de código rigorosos para garantir que todo código contribuído tenha não somente sido testado de forma unitária, mas também permaneça estável e fácil de estender, fácil de testar com suas extensões, e fácil de manter.

2.3.2 Arquitetura

Conforme descrito no Programmer's Reference Guide: Zend Framework (2009 p. 1), o ZF oferece uma performance robusta usando a arquitetura Modelo / Visão / Controlador (*Model-View-Controller*) (MVC), uma abstração de banco de dados simples de usar, e uma interface orientada a objetos fácil de usar. Pacheco (2009) cita MVC como “um padrão de arquitetura de software largamente utilizado em soluções para web. Sua principal vantagem é a separação da telas (*view*) da camada de negócio (*model*)”.

A arquitetura MVC é usada para separar diferentes partes de uma aplicação, facilitando o desenvolvimento e a manutenção, e aumentando a flexibilidade e a reutilização. De acordo com Gamma et al. (2000, p. 20), o “MVC é composto por três tipos de objetos. O Modelo é o objeto de aplicação, a Visão é a apresentação na tela e o Controlador define a maneira como a interface do usuário reage às entradas do mesmo”.

MVC separa Visão e Modelos pelo estabelecimento de um protocolo inserção / notificação entre eles. Uma vista deve garantir que a sua aparência reflita o estado do modelo. Sempre que os dados do modelo mudam, o modelo notifica as vistas

² *Asynchronous Javascript And XML* (AJAX) é o uso de tecnologias como Javascript e XML para tornar páginas Web mais interativas, utilizando-se de solicitações assíncronas de informações.

³ *JavaScript Object Notation* (JSON) é um formato para o intercâmbio de informações.

que dependem dele. Esta abordagem permite ligar múltiplas vistas a um modelo para fornecer diferentes apresentações (GAMMA et al., 2000, p. 20-21).

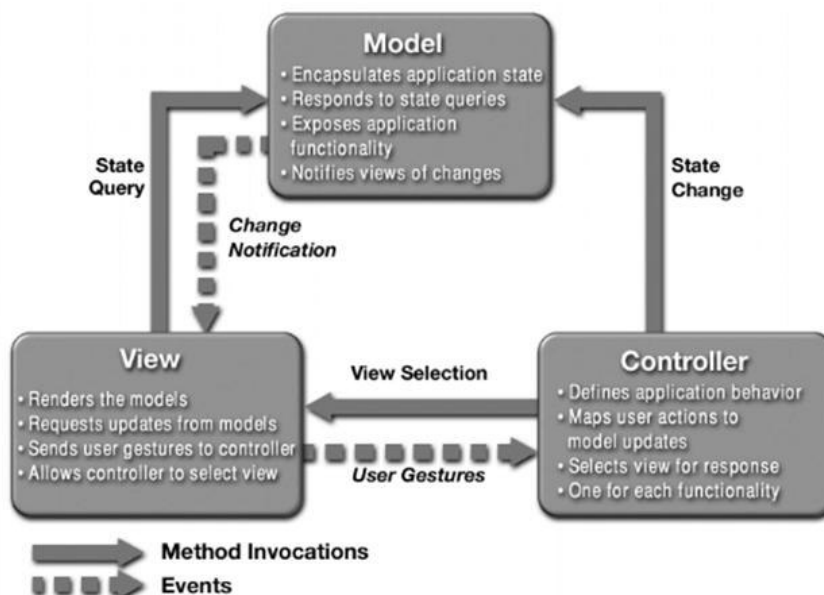


Figura 11 – Arquitetura MVC.

Fonte: Ramachandran (2002).

[...] o MVC tornou-se um padrão no design de aplicações web modernas. A maioria do código de aplicações web cai sobre uma das três seguintes categorias: apresentação, regra de negócios e acesso a dados. Os modelos padrão MVC fazem essa separação. O resultado final é que o código de apresentação pode ser consolidado em uma parte do aplicativo, o código da regra de negócio em outro e o código de acesso a dados em outro ainda. Muitos desenvolvedores definiram essa separação como indispensável para manter seu código organizado, especialmente quando mais de um desenvolvedor está trabalhando na mesma aplicação (ZEND TECHNOLOGIES INC, 2009, tradução nossa).

O Programmer's Reference Guide: Zend Framework (2009, tradução nossa), descreve cada um dos três objetos como:

- **Modelo:** é a parte do aplicativo que define sua funcionalidade básica por trás de um conjunto de abstrações. Rotinas de acesso aos dados e de regra de negócios podem ser definidas pelo Modelo.
- **Visão:** define exatamente o que é apresentado ao usuário. Geralmente a interface é desenvolvida em HTML.
- **Controlador:** manipulam os Modelos, a fim de processar e responder a solicitação do usuário. A maioria dos especialistas recomenda manter o Controlador o mais enxuto possível.

A estrutura do ZF é dividida em diretórios, necessária para o seu correto funcionamento:

```

zf-tutorial/
  /application
    /controllers
    /models
    /views
    /filters
    /helpers
    /scripts
  /library
  /public
    /images
    /scripts
    /styles

```

Figura 12 - Estrutura de diretórios do Zend Framework.

Fonte: Allen (2007, p. 2).

Conforme descrito por Allen (2007, p. 2), existem diretórios separados para o modelo, visão e o controlador do aplicativo, que ficam na pasta *application*. Os demais arquivos, como imagens, scripts e arquivos CSS, são armazenados em pastas específicas do diretório *public*. Os arquivos do Zend Framework são colocados na pasta *library*, assim como outras bibliotecas que possam vir a serem utilizadas.

Tabela 1 – Estrutura do Zend Framework

Diretório ou arquivo	Descrição
<i>application\</i>	Diretório da aplicação.
<i>controllers\</i>	Diretórios onde serão criados os controladores da aplicação.
<i>models\</i>	Diretório onde serão criados os modelos da aplicação.
<i>Views\</i>	Diretório onde serão criados os arquivos de visão.
<i>.htaccess</i>	Arquivo do Apache, com negação de acesso direto aos arquivos pelo navegador.
<i>config.ini</i>	Arquivo de configuração da aplicação.
<i>public\</i>	Diretório com arquivos que podem ser acessados pelo navegador.
<i>images\</i>	Diretório para arquivos de imagem.
<i>scripts\</i>	Diretório para arquivos Javascript.
<i>styles\</i>	Diretórios para arquivos CSS.
<i>.htaccess</i>	Arquivo do Apache, com desabilitação das regras de reescrita de URL.
<i>Index.php</i>	Arquivo inicial do projeto, com todas as definições da aplicação.
<i>.htaccess</i>	Arquivo do Apache com ativação da reescrita da URL e

redirecionamento para o *index.php*.

Fonte: Lisboa (2008, p. 33-34).

2.4 SGBD

Um Sistema Gerenciador de Banco de Dados (SGDB) permite aos usuários criar e manter um banco de dados. Elmasri e Navathe (2005, p. 3), definem SGDB como um sistema de software com o propósito de facilitar os processos de definição, construção, manipulação e compartilhamento de banco de dados entre vários usuários e aplicações.

2.4.1 Vantagens

De acordo com Elmasri e Navathe (2005, p.11-14), um bom SGBD apresenta as seguintes vantagens:

- Controle de redundância: capacidade de controlar as redundâncias, impedindo as inconsistências entre os arquivos, melhorando a performance das consultas;
- Restrição de acesso não autorizado: garantia da segurança de um subsistema de autorização usado pelo DBA para criar contas e definir as restrições de cada uma de forma automática;
- Armazenamento persistente para objetos programas: é uma das principais justificativas para os sistemas de banco de dados orientados a objeto. Esses sistemas oferecem estruturas de dados compatíveis com uma ou mais linguagens de programação orientadas a objetos;
- Armazenamento de estruturas para o processamento eficiente de consultas: funcionalidades para a execução de atualizações e consultas eficientemente, através de estruturas de dados especializadas para aumentar a velocidade de pesquisa no disco dos registros desejados;
- Garantia de *backup* e restauração: subsistema de *backup* e recuperação responsável pela restauração de falhas de hardware ou de software, assegurando que uma transação seja resumida do ponto em que foi interrompida;
- Múltiplas interfaces para os usuários: interfaces distintas aos diversos tipos de usuários de conhecimento técnico diferentes que utilizam o banco de dados.

Essas interfaces incluem linguagens de consulta; linguagens de programação de aplicação; formulários e sequências de comandos; menus e linguagem natural;

- Representação de relacionamentos complexos entre os dados: capacidade de representar a variedade de relacionamentos complexos entre os dados, bem como recuperar e atualizar os dados relacionados fácil e eficientemente;
- Restrições de integridade: funcionalidades para a definição e a garantia de certas restrições de integridade que devem complementar os dados;
- Interferências e ações usando as regras: sistemas de banco de dados dedutivos que oferecem capacidades para definir as regras de dedução por interferência gerando novas informações de fatos armazenados no banco de dados.

Ainda segundo Elmasri e Navathe (2005, p.14-15), existem algumas implicações adicionais da utilização da abordagem do uso de um banco de dados que beneficiam a maioria da organização, são elas: potencial para garantir padrões, redução no tempo de desenvolvimento de aplicações, flexibilidade, disponibilidade para atualizar as informações e economias de escala.

2.4.2 PostgreSQL

Conforme descrito na Documentação do PostgreSQL (2005, p. 15), ele é um sistema gerenciador de banco de dados objeto-relacional (SGBDOR), desenvolvido pelo Departamento de Ciência da Computação da Califórnia em Berkeley. Ele “é derivado do pacote Postgres escrito na Universidade da Califórnia em Berkeley. Com mais de uma década de desenvolvimento por trás, o PostgreSQL é atualmente o mais avançado banco de dados de código aberto disponível em qualquer lugar” (DOCUMENTAÇÃO DO POSTGRES, 2005, p. 15). O POSTGRES for pioneiro em vários conceitos, que somente muito mais tarde se tornaram disponíveis em alguns sistemas de banco de dados comerciais.

De acordo a Documentação do PostgreSQL (2005, p. 15), ele possui sua licença aberta, desta forma pode ser utilizado, modificado e distribuído por qualquer pessoa para qualquer finalidade, privada, comercial ou acadêmica, livre de encargos. Ainda segundo a documentação do PostgreSQL, tem suporte a grande parte do padrão SQL:2003, além de oferecer muitas funcionalidades, como:

- Comandos complexos;

- Chaves estrangeiras;
- Gatilhos;
- Visões;
- Integridade transacional;
- Controle de simultaneidade multiversão.

Além disso, o PostgreSQL pode ser estendido pelo usuário, adicionando novos

- Tipos de dados
- Funções
- Operadores
- Métodos de índice
- Linguagens procedurais

A Documentação do PostgreSQL (2005, p. 23) cita que o PostgreSQL utiliza o modelo cliente-servidor, onde uma sessão do mesmo consiste nos seguintes processos (programas) cooperando entre si:

- Um processo servidor chamado *postmaster*, que gerencia os arquivos de banco de dados, recebe conexões dos aplicativos cliente com o banco de dados, e executa ações no banco de dados em nome dos clientes;
- O aplicativo cliente do usuário chamado *frontend*, que deseja executar operações de banco de dados. Estes aplicativos podem ter naturezas muito diversas: o cliente pode ser uma ferramenta no modo caractere, um aplicativo gráfico, um servidor web que acessa o banco de dados para mostrar páginas Web, ou uma ferramenta especializada para manutenção do banco de dados.

2.5 WEB SERVICES

Web services são ferramentas utilizadas na integração e comunicação entre aplicações distintas e de diferentes plataformas, através de dados em formato XML. Para Cerami (2002, p. 3 tradução nossa), um *web service* “é qualquer serviço de troca de mensagens através da internet usando o padrão de XML e que não está vinculado a um sistema operacional ou linguagem de programação”.

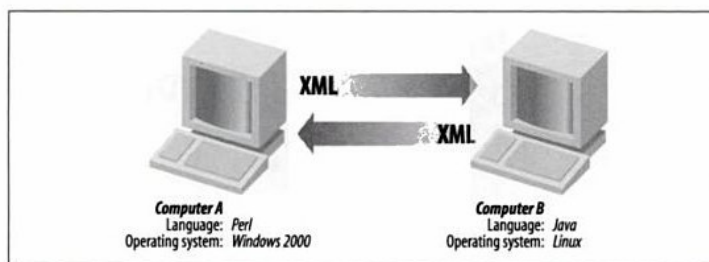


Figura 13 – Um *web service* básico.

Fonte: Cerami (2002, p. 4).

Um *web service* pode ser definido, segundo Snell et al. (2002, p. 1, tradução nossa), como uma funcionalidade para acesso de aplicação através da rede, utilizando tecnologias de internet padrão, assim, se uma aplicação pode ser acessada através da internet usando protocolos como HTTP, XML, SMTP, ou *Jabber*⁴, logo esta é um *web service*.

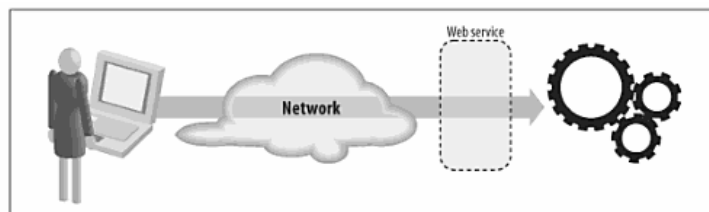


Figura 14 – Um *web service* permite o acesso a aplicações.

Fonte: Snell et al. (2002, p. 1).

2.5.1 Fundamentos

Para Snell et al. (2002, p. 2, tradução nossa), um *web service* é uma interface posicionada entre as aplicações de diferentes linguagens, separando-as através de uma camada de abstração. Essa camada padrão significa que qualquer linguagem com suporte a *web service* poderá acessar a aplicação.

⁴ *Jabber* é um novo *lightweight*, protocolo de transporte assíncrono usado em aplicações *peer-to-peer* (SNELL; TIDWELL; KULCHENKO, 2002, p. 7).

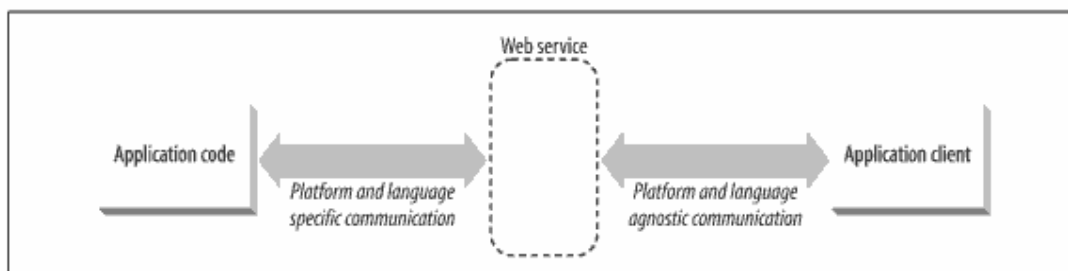


Figura 15 – Web service fornece uma camada de abstração entre as aplicações.

Fonte: Snell et al. (2002, p. 2).

Snell et al. (2002, p. 2, tradução nossa), defende que diferentes serviços – publicações, gestão, pesquisas, e recuperação de conteúdos – são acessados através de protocolos e formatos padrão: HTTP e HTML. Aplicativos (*web browsers*) que entendem esses padrões podem interagir com esses serviços para executar como ordenar livros, enviar mensagens ou ler notícias. Ainda segundo Snell et al. (2002, p. 2, tradução nossa), devido à abstração fornecida por interfaces padrão, não importa se uma aplicação é escrita em Java e o *browser* em C++, ou se aplicação usa a plataforma Unix enquanto que o *browser* o Windows, o *web service* permite torna a plataforma irrelevante.

De acordo com Cerami (2002, p. 5), pode-se resumir um *web service* como qualquer serviço que esteja disponível em redes de internet públicas ou privadas, usando um sistema de mensagens XML padrão, não vinculado a qualquer sistema operacional o linguagem de programação, descrito através de uma gramática XML comum e detectado através de um simples mecanismo de busca.

2.6 SOAP

Welling e Thomson (2005, p. 630), definem SOAP (*Simple Object Access Protocol*) como um protocolo de troca de mensagens controlado por solicitação e resposta, permitindo aos clientes chamarem serviços web e receberem respostas dos servidores, sendo que cada mensagem de SOAP, uma solicitação ou uma resposta, é um documento simples XML.

Na pilha de tecnologias *web service*, SOAP é como um protocolo de pacotes padronizados para mensagens compartilhadas por aplicativos. A especificação define como sendo nada mais que mensagens no formato XML para a transferência de informações, e um conjunto de regras para a transposição de aplicações e tipos de dados específicos de plataformas em formato XML, adequados para uma ampla variedade de mensagens de aplicações e padrões de integração (SNELL; TIDWELL; KULCHENKO, 2002, p. 11, tradução nossa).

Snell et al. (2002, p. 11, tradução nossa), afirma que SOAP é uma aplicação da especificação XML e depende fortemente dos padrões XML para sua definição e função.

Mensagens XML é onde ocorre a troca de informações entre aplicações usando arquivos XML, fornece flexibilidade para a comunicação entre aplicações e constitui a base do SOAP.



Figura 16 – Mensagem XML.
Fonte: Snell et al. (2002, p. 5).

Snell et al. (2002, p. 11, tradução nossa), defende que por não ser amarrado a um aplicativo particular, sistema operacional ou linguagem de programação, mensagens XML podem ser usadas em todos os ambientes. A ideia fundamental é que duas aplicações, indiferente de sistema operacional, linguagem de programação, ou qualquer outro detalhe técnico, possa compartilhar abertamente informações usando não mais que uma simples mensagem codificada de uma forma que ambas entendam. SOAP fornece um padrão para a estrutura de mensagens XML.

Em geral, de acordo com Welling e Thomson (2005, p. 631), quando geradas as solicitações de SOAP, estas interpretam as respostas programaticamente utilizando uma biblioteca de SOAP, independente da linguagem de programação utilizada. Desta forma, não há a necessidade de construir a solicitação de SOAP e interpretar a resposta manualmente.

2.6.1 Mensagens SOAP

Snell et al. (2002, p. 13, tradução nossa), cita que uma mensagem SOAP “consiste em um envelope contendo um cabeçalho opcional e um corpo de requisições. O cabeçalho contém blocos de informação relevantes a como a mensagem está a ser processada. Isso inclui definições de roteamento e entrega, afirmações de autenticação e autorização, e contextos de operações. O corpo contém a mensagem atual a ser entregue processada. Qualquer informação que possa ser expressada na sintaxe XML, pode estar no corpo da mensagem”.

A mensagem de SOAP, segundo Welling e Thomson (2005, p. 631), inicia com a declaração de que é um documento de XML. O elemento raiz de todas as mensagens é um envelope de SOAP e dentro dele localiza-se o elemento *body* que contém a solicitação real.

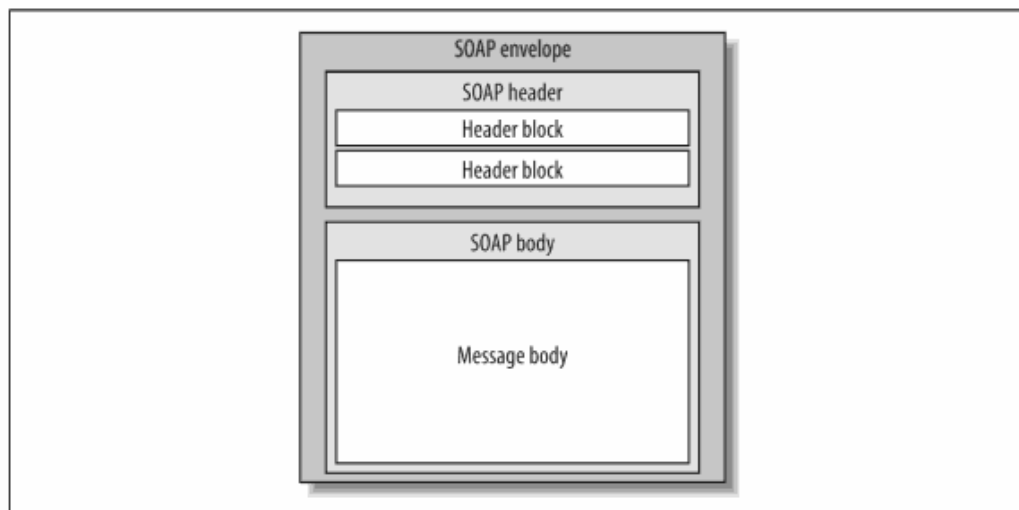


Figura 17 – A estrutura da mensagem SOAP.

Fonte: Snell et al. (2002, p. 13).

De acordo com Snell et al. (2002, p. 14, tradução nossa), se um envelope contém um cabeçalho, deve ter não mais que um e deve aparecer no início do envelope, antes do corpo. O cabeçalho, assim como o corpo, conterá um conteúdo válido que o criador da mensagem deseja inserir. Cada elemento contido no cabeçalho é chamado *header block*. O propósito de um *header block* é informar informação contextual relevante ao processo de uma mensagem SOAP.

Já os conteúdos do corpo da mensagem, chamados elementos *body*, são as mensagens. “Cada elemento de um envelope precisa conter exatamente um elemento *body*. O elemento *body* poderá conter quantos nós filhos forem necessários” (SNELL; TIDWELL; KULCHENKO, 2002, p. 14, tradução nossa).

2.6.2 Modelo de troca de mensagens SOAP

Para Snell et al. (2002, p. 22, tradução nossa), processar uma mensagem SOAP envolve separar o envelope e fazendo algo com a informação que carrega. SOAP define um quadro geral para o processamento, mas deixa os detalhes reais de como o processamento é executado até a aplicação.

Snell et al. (2002, p. 22, tradução nossa), defende que a essência do modelo troca de mensagens SOAP, é a ideia de que a transmissão de um envelope de um emitente a um destinatário pode passar por vários processos intermediários, onde cada por sua vez faz algo com a ela. Um SOAP intermediário é um *web service* especialmente concebido para estar

entre o cliente e o provedor e adicionar valor ou funcionalidade na transação entre as duas. O conjunto intermediário que a mensagem percorre é chamado *path*, e cada intermediário ao longo do *path* é chamado de um ator.

Ainda segundo Snell et al. (2002, p. 22, tradução nossa), a construção de um *path* (a definição de quais nós a mensagem passa) não abrange a especificação SOAP. O que SOAP especifica, contudo, é um mecanismo para identificar quais as partes da mensagem são destinadas para os processamentos específicos em seu trajeto. Esse mecanismo é conhecido como “segmentação” e só pode ser usado relacionado com blocos de cabeçalho (o corpo de um envelope SOAP não pode ser explicitamente segmentado de um nó específico).

3 PROJETO DO SOFTWARE

Para desenvolver um software é preciso garantir a sua qualidade e atender as necessidades dos seus usuários, este deve ser desenvolvido de maneira previsível e em determinado período, com a utilização eficiente e eficaz de recursos (BOOCH; RUMBAUGH; JACOBSON; 2005, p. 4).

Blaha e Rumbaugh (2006, p. 17) defendem que para desenvolver sistemas complexos, é necessário extrair diferentes visões do sistema, montar modelos com notações exatas, verificar se os modelos satisfazem aos requisitos do sistema e acrescentar detalhes gradualmente, a fim de transformar os modelos em uma implementação.

No presente trabalho a modelagem do software proposto será baseada na Linguagem de Modelagem Unificada, ou *Unified Modeling Language* (UML).

3.1 LINGUAGEM DE MODELAGEM UNIFICADA – UML

A UML é uma família de notações gráficas, apoiada por um metamodelo único, que auxilia na descrição e no projeto de sistemas, principalmente os construídos utilizando o paradigma orientado a objetos (FOWLER, 2004, p. 25).

De acordo com Guedes (2008, p. 18), a UML “é uma linguagem visual utilizada para modelar sistemas computacionais por meio do paradigma de orientação a objetos. Esta linguagem tornou-se, nos últimos anos a linguagem padrão de modelagem de software adotada internacionalmente pela indústria de engenharia de software”.

Para Booch et al. (2005), a UML é uma linguagem bastante expressiva, abrangendo todas as visões necessárias ao desenvolvimento e implantação de sistemas complexos de software, empregada de maneira efetiva em domínios como: sistemas de informações corporativos; serviços bancários e financeiros; telecomunicações; transportes; defesa/espaço aéreo; vendas de varejo; eletrônica médica; científicos; e serviços distribuídos baseados na web.

A UML 2.0 descreve treze tipos de diagramas oficiais para modelagem de sistemas (FOWLER, 2004, p.33), a fim de “fornecer múltiplas visões do sistema a ser modelado, analisando-o e modelando-o sob diversos aspectos, procurando atingir a completude da modelagem, permitindo que cada diagrama complemente os outros” (GUEDES, 2008, p. 27). Desta forma, os diagramas modelam características distintas da estrutura ou do

comportamento do sistema. Para melhor detalhamento sobre os diagramas, sugere-se consultar (GUEDES, 2008; BOOCH, RUMBAUGH, JACOBSON, 2005). No presente trabalho foram utilizados os diagramas descritos abaixo:

- Diagrama de Casos de Uso: apresenta uma idéia geral de como o sistema irá se comportar, buscando identificar os atores (usuários, outros sistemas ou até mesmo um hardware especial) que utilizarão de alguma forma o software, bem como os serviços, ou seja, as opções que o sistema disponibilizará aos atores (GUEDES, 2008, p. 28);
- Diagrama de Classes: define a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos que cada classe possui, além de estabelecer como as classes se relacionam e trocam informações entre si (GUEDES, 2008, p. 29);
- Diagrama de Sequência: preocupa-se com a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos em um determinado processo, identificando o evento gerador do processo modelado, bem como o ator responsável por esse evento, e determinando como o processo deve se desenrolar e ser concluído, por meio da chamada de métodos disparados por mensagens enviadas entre objetos (GUEDES, 2008, p. 31);
- Diagrama de Atividade: descreve os passos a serem percorridos para a conclusão de uma atividade específica, muitas vezes representada por um método com certo grau de complexidade, concentrando-se na representação do fluxo de controle de uma atividade (GUEDES, 2008, p. 33);

Os diagramas da UML 2.0 dividem-se em diagramas Estruturais e Diagramas Comportamentais, conforme Figura 18.

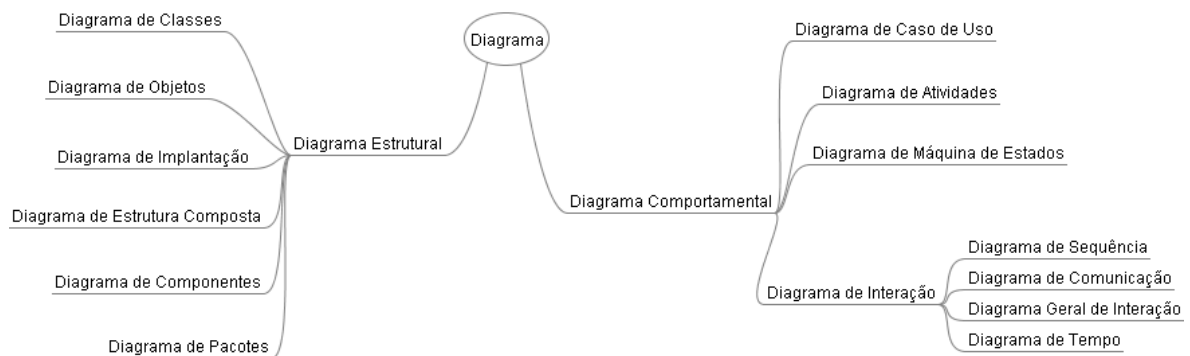


Figura 18 – Classificação dos tipos de diagrama da UML.
Fonte: adaptado de Guedes (2008).

Os Diagramas Estruturais abrangem os Diagramas de Classes. Já os Comportamentais englobam os de Casos de Uso, Atividade e Sequência. Os Diagramas Estruturais apresentam as características estáticas do sistema, não se alteram em função do tempo, servem para visualizar, especificar, construir e documentar. Os diagramas comportamentais têm como função detalhar as interações entre os sistemas estáticos (GUEDES, 2008).

3.2 LEVANTAMENTO E ANÁLISE DE REQUISITOS

A atividade de análise de requisitos, de acordo com Fowler (2004, p. 47), procura descobrir o que os usuários e clientes de um produto de software querem que o sistema faça. A partir dessa etapa são determinadas as reais necessidades de um sistema de informação. Para Paula Filho (2003, p. 5), os requisitos podem ser dos seguintes tipos:

- Explícitos, são aqueles descritos em um documento que arrola os requisitos de um produto, ou seja, um documento de especificação de requisitos;
- Normativos, são aqueles que decorrem de leis, regulamentos padrões e outros tipos de normas a que o tipo de produto deve obedecer;
- Implícitos, são expectativas dos clientes e usuários, que são cobradas por esses, embora não-documentadas.

Paula Filho (2003, p. 5) defende que os requisitos são as características que definem os critérios de aceitação de um produto, sendo que a engenharia de software tem por objetivo colocar nos produtos as características que são requisitos, e elas costumam ser divididas em duas:

- Funcionais: representam os comportamentos que um programa ou sistema deve apresentar diante de certas ações de seus usuários;
- Não-funcionais: expressam como deve ser feito o desenvolvimento do software.

Os requisitos funcionais e não funcionais do sistema de frente de caixa desenvolvido no presente trabalho foram classificados em essenciais – imprescindíveis, devem ser implementados impreterivelmente – e desejáveis – que não comprometem as funcionalidades básicas do sistema. Os requisitos funcionais e não funcionais são descritos a seguir.

3.2.1 Requisitos funcionais

REQF1 – [Cadastrar usuário]

Requisito que corresponde ao cadastro de usuário: nome completo do usuário, tipo de usuário (administrador, gerente ou caixa), data do cadastro, identificação de usuário (*login*), senha e situação do cadastro (ativo ou inativo).

Classificação: essencial.

REQF2 – [Cadastrar rede de lojas]

Requisito que corresponde ao cadastro de rede de lojas, no qual deve constar: código da rede de lojas, descrição e situação do cadastro (ativo ou inativo).

Classificação: essencial.

REQF3 – [Cadastrar loja]

Requisito que corresponde ao cadastro da loja, deve constar: código da loja, rede, nome, CNPJ, inscrição estadual, logradouro, número, complemento, bairro, município e situação do cadastro (ativo ou inativo).

Classificação: essencial.

REQF4 – [Cadastrar terminal]

Requisito que corresponde ao cadastro do terminal, consistindo de: código do terminal, rede, loja, descrição, número de série do terminal e situação do cadastro (ativo ou inativo).

Classificação: essencial.

REQF5 – [Cadastrar divisão]

Requisito que corresponde ao cadastro de divisão (calçados, acessórios, etc.): código da divisão do produto, descrição.

Classificação: essencial.

REQF6 – [Cadastrar artigo]

Requisito que corresponde ao cadastro de artigo (sandália, bota, bolsa, cinto, etc.), deve constar: código do artigo do produto e descrição.

Classificação: essencial.

REQF7 – [Cadastrar marca]

Requisito que corresponde ao cadastro de marca, constando: código da marca do produto e descrição.

Classificação: essencial.

REQF8 – [Cadastrar modelo]

Requisito que corresponde ao cadastro de modelo, no qual deve constar: código do modelo do produto e descrição.

Classificação: essencial.

REQF9 – [Cadastrar material]

Requisito que corresponde ao cadastro de material: código do material do produto e descrição.

Classificação: essencial.

REQF10 – [Cadastrar cor]

Requisito que corresponde ao cadastro de cor: código da cor do produto e descrição.

Classificação: essencial.

REQF11 – [Cadastrar classificação fiscal]

Requisito que corresponde ao cadastro de classificação fiscal das mercadorias, que determina os tributos envolvidos nas operações de entrada e saída dos produtos, consistindo de: código da classificação fiscal, descrição, espécie, tipo e percentual de Imposto sobre Produtos Industrializados (IPI).

Classificação: desejável.

REQF12 – [Cadastrar unidade de medida]

Requisito que corresponde ao cadastro de unidade de medida, onde deve constar: código da unidade de medida do produto, descrição e abreviação.

Classificação: essencial.

REQF13 – [Cadastrar produto]

Requisito que corresponde ao cadastro de produto, constando: código do produto, descrição, divisão, artigo, modelo, material, cor, classificação fiscal, unidade de medida, tamanho do produto e situação do cadastro (ativo ou inativo).

Classificação: essencial.

REQF14 – [Cadastrar preço]

Requisito que corresponde ao cadastro de preço, consistindo: código da rede, código da loja, código do produto, preço e percentual de ICMS para o produto na loja.

Classificação: essencial.

REQF15 – [Cadastrar forma de pagamento]

Requisito que corresponde ao cadastro de formas de pagamento: código da forma de pagamento, descrição e situação do cadastro (ativo ou inativo).

Classificação: essencial.

REQF16 – [Cadastrar pessoa]

Requisito que corresponde ao cadastro de pessoa, onde deve constar: código da pessoa, nome, CPF, CNPJ, inscrição estadual, logradouro, número, complemento, bairro, município, número de telefone, número de telefone celular, e-mail, data do cadastro e situação do cadastro (ativo ou inativo).

Classificação: essencial.

REQF17 – [Cadastrar vendedor]

Requisito que corresponde ao cadastro de vendedor, no qual deve constar: código do vendedor, rede, loja, código de pessoa, apelido, percentual de comissão sobre venda e situação (ativo ou inativo).

Classificação: essencial.

REQF18 – [Cadastrar consumidor]

Requisito que corresponde ao cadastro de consumidor, constando: código do consumidor, rede, loja, código de pessoa e situação do cadastro (ativo ou inativo).

Classificação: essencial.

REQF19 – [Cadastrar fornecedor]

Requisito que corresponde ao cadastro de fornecedor, onde deve constar: código do fornecedor, razão social, código de pessoa e situação do cadastro (ativo ou inativo).

Classificação: essencial.

REQF20 – [Emitir ordem de compra]

Requisito que corresponde a emissão de ordem de compra. Realiza o pedido de compra dos produtos que serão vendidos na loja, que consiste de: rede, loja, fornecedor, data de emissão, data de entrega, produto, quantidade, preço de compra, percentual de desconto e situação (aberto ou recebido).

Classificação: essencial.

REQF21 – [Receber produto]

Requisito que corresponde ao recebimento de produtos. Realiza a entrada dos produtos através da ordem de compra, consistindo: rede, loja, fornecedor, ordem de compra, nota fiscal, data de emissão da nota fiscal, data de recebimento, produto, quantidade e valor.

Classificação: essencial.

REQF22 – [Abrir caixa]

Requisito que corresponde à abertura de caixa e deve: emitir relatório de abertura de caixa na impressora fiscal e gravar as informações na base de dados.

Classificação: essencial.

REQF23 – [Fechar caixa]

Requisito que corresponde ao fechamento de caixa e deve: emitir relatório de fechamento de caixa na impressora fiscal e gravar as informações na base de dados.

Classificação: essencial.

REQF24 – [Emitir Leitura da Memória Fiscal]

Requisito que corresponde à Leitura da Memória Fiscal e deve: emitir relatório de Leitura da Memória da Impressora Fiscal por intervalo de período e de Redução Z⁵.

Classificação: essencial.

REQF25 – [Emitir venda]

Requisito que corresponde à emissão de venda e deve: realizar venda de produto através da emissão de Cupom Fiscal. Deve constar: rede, loja, terminal, vendedor, consumidor, produto, quantidade, desconto, valor total e forma de pagamento.

Classificação: essencial.

REQF26 – [Emitir suprimento de caixa]

Requisito que corresponde à emissão de suprimento de caixa e deve: emitir relatório de suprimento de caixa (entrada de valor no caixa) e gravar as informações na base de dados.

Classificação: essencial.

REQF27 – [Emitir sangria de caixa]

Requisito que corresponde à emissão de sangria de caixa deve: emitir relatório de sangria de caixa (saída de valor no caixa) e gravar as informações na base de dados.

Classificação: essencial.

⁵ A redução Z é um relatório emitido pelo equipamento fiscal, indicando a totalização dos valores acumulados nos contadores e totalizadores (SOUZA, 2007, p. 112).

REQF28 – [Gerar relatórios]

Requisito que corresponde a geração de relatórios. Deve constar:

- Compras;
- Vendas;
- Estoque.

Classificação: essencial.

REQF29 – [Controlar acesso ao sistema]

Requisito que corresponde ao controle de acesso às opções do sistema: o usuário terá acesso às opções disponíveis de acordo com seu perfil.

Classificação: essencial.

REQF30 – [Comunicação com impressora fiscal]

Requisito que corresponde à comunicação com a impressora fiscal Bematech modelo MP2100⁶ ou superior.

Classificação: essencial.

3.1.2 Requisitos não funcionais**REQNF1 – [Utilização de banco de dados relacional]**

Requisito que corresponde à utilização de um Sistema Gerenciador de Banco de Dados Relacional. Será utilizado o PostgreSQL 8.4 no ambiente Windows.

Classificação: essencial.

REQNF2 – [Utilização de servidor web]

Requisito que corresponde à utilização de servidor web. Será utilizado o Apache 2.2 no ambiente Windows como servidor de aplicação.

Classificação: essencial.

REQNF3 – [Links/ícones/botões para acesso às principais funções]

Requisito que corresponde ao acesso as principais funções, através da utilização de links/ícones/botões para facilitar a manipulação do sistema.

Classificação: essencial.

REQNF4 – [Conformidade com a legislação]

⁶ A Impressora Fiscal Bematech MP2100 atende as exigências fiscais da legislação que especifica os equipamentos fiscais e é a impressora fiscal mais vendida no mercado brasileiro (www.bematech.com.br).

Requisito que corresponde à conformidade com a legislação. O sistema deverá estar de acordo à legislação que regula os sistemas de frente de caixa.

Classificação: essencial.

REQNF5 – [Cópia de segurança]

Requisito que corresponde ao backup de segurança da base de dados do sistema.

Classificação: desejável.

3.3 CASOS DE USO

Para Paula Filho (2003), os Casos de Uso representam funções do sistema, realizando um aspecto maior da funcionalidade do produto e gerando benefícios para o cliente ou os usuários, sendo representados através de Diagramas de Casos de Uso.

Os Casos de Uso são uma técnica para captar os requisitos funcionais de um sistema, de forma a descrever as interações típicas entre os usuários de um sistema e o próprio sistema, fornecendo uma narrativa sobre como o sistema é utilizado (FOWLER, 2004, p. 104).

Os casos de uso podem ser aplicados para captar o comportamento pretendido pelo sistema que está sendo desenvolvido, sem ser necessário especificar como esse comportamento é implementado, [...] bem estruturados denotam somente o comportamento essencial do sistema ou subsistema e não amplamente gerais, nem muito específicos (BOOCH, RUMBAUGH, JACOBSON, 2005, p. 227).

3.3.1 Diagrama de casos de uso

Guedes (2008) descreve que o Diagrama de Casos de Uso tem por objetivo representar uma visão externa geral das funções e serviços que o sistema deverá oferecer, concentrando-se em dois itens principais: Atores e Casos de Uso. Os atores representam os papéis desempenhados pelos diversos usuários que poderão utilizar os serviços e funções do sistema, podendo, eventualmente, representar algum hardware específico ou, mesmo, um outro software que interaja com o sistema. Os Casos de Uso referem-se aos serviços, tarefas ou funções que podem ser utilizados de alguma maneira pelos usuários do sistema e também servem para expressar e documentar os comportamentos pretendidos para as funções do sistema.

Na Figura 19 é apresentado o Diagrama de Caso de Uso referente às atividades do sistema.



Figura 19 – Diagrama de Caso de Uso do Sistema.
Fonte: Autor (2010).

O sistema proposto apresenta três atores: o Administrador que gerencia os cadastros e o estoque; o Gerente que administra o cadastro dos vendedores e consumidores, gera relatórios, emite venda, além de realizar as operações de caixa; e, finalmente, o Caixa, que cadastra consumidores, emite venda e interage com as opções de caixa.

3.3.2 Casos de Uso Estendidos

Os casos de uso a seguir descrevem de forma mais detalhada as interações entre o sistema e os usuários.

Abaixo são apresentados os casos de uso referente ao cadastro de produto, cadastro de pessoa, abertura de caixa, fechamento de caixa e emissão de venda, bem como seus

diagramas. Estes casos de uso foram escolhidos devido a ocorrerem com maior frequência no sistema de frente de caixa. Os demais casos de uso e diagramas encontram-se no Anexo do presente trabalho.

UC013 – Gerência de produto

Breve descritivo: este caso de uso descreve o processo de gerenciamento de produto.

Requisitos funcionais:	REQF5 – [Cadastrar divisão] REQF6 – [Cadastrar artigo] REQF7 – [Cadastrar marca] REQF8 – [Cadastrar modelo] REQF9 – [Cadastrar material] REQF10 – [Cadastrar cor] REQF11 – [Cadastrar classificação fiscal] REQF12 – [Cadastrar unidade de medida] REQF13 – [Cadastrar produto] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções]
Atores:	Administrador.
Pré-condições:	Ser usuário com permissão de administrador.
Fluxo principal:	O usuário administrador acessa a tela de gerenciamento de produto. Os subfluxos são apresentados pelo sistema: Inclusão, Alteração, Consulta e Exclusão.
Subfluxos:	1. Inclusão: 1.1 O usuário seleciona a opção de inclusão; 1.2 O usuário informa dos dados de um novo produto: código, descrição, divisão, artigo, modelo, material, cor, classificação fiscal, unidade de medida, tamanho e situação; 1.3 O usuário clica no botão/link “Salvar”; 1.4 O sistema verifica os dados; 1.5 O sistema grava as informações no banco de dados; 1.6 O sistema retorna a mensagem “Dados incluídos com sucesso”. 2. Alteração: 2.1 O usuário seleciona a opção de alteração;

	<p>2.2 O usuário informa o código do produto que deseja alterar;</p> <p>2.3 O sistema preenche os dados do produto;</p> <p>2.4 O usuário altera os dados desejados;</p> <p>2.5 O usuário clica no botão/link “Salvar”;</p> <p>2.6 O sistema verifica os dados;</p> <p>2.7 O sistema grava as informações no banco de dados;</p> <p>2.8 O sistema retorna a mensagem “Dados alterados com sucesso”.</p> <p>3. Consulta:</p> <p>3.1 O usuário seleciona a opção de consulta;</p> <p>3.2 O usuário informa o código do produto que deseja consultar;</p> <p>3.3 O sistema preenche os dados do produto.</p> <p>4. Exclusão:</p> <p>4.1 O usuário seleciona a opção de exclusão;</p> <p>4.2 O usuário informa o código do produto que deseja excluir;</p> <p>4.3 O sistema preenche os dados do produto;</p> <p>4.4 O usuário clica no botão/link “Excluir”;</p> <p>4.5 O sistema verifica os dados;</p> <p>4.6 O sistema exclui as informações no banco de dados;</p> <p>4.7 O sistema retorna a mensagem “Dados excluídos com sucesso”.</p>
Tratamento de exceções:	<p>1.4.1 O sistema deve verificar se os campos obrigatórios foram preenchidos e retornar a mensagem “Preencha todos os campos obrigatórios”, caso algum não tenha sido preenchido;</p> <p>2.2.1 Se o código informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p> <p>2.6.1 Recebe o mesmo tratamento de exceção do item 1.4.1;</p> <p>3.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.5.1 Recebe mesmo tratamento de exceção do item 1.4.1.</p>
Pós-condições:	<p>1. Um novo produto está cadastrado;</p> <p>2. Um produto está alterado;</p> <p>3. Um produto está consultado;</p> <p>4. Um produto está excluído.</p>

Quadro 1 – Caso de Uso Gerência de produto.

Fonte: autor (2010).

O Diagrama de Atividade referente à atividade de gerência de produto é apresentado na Figura 20.

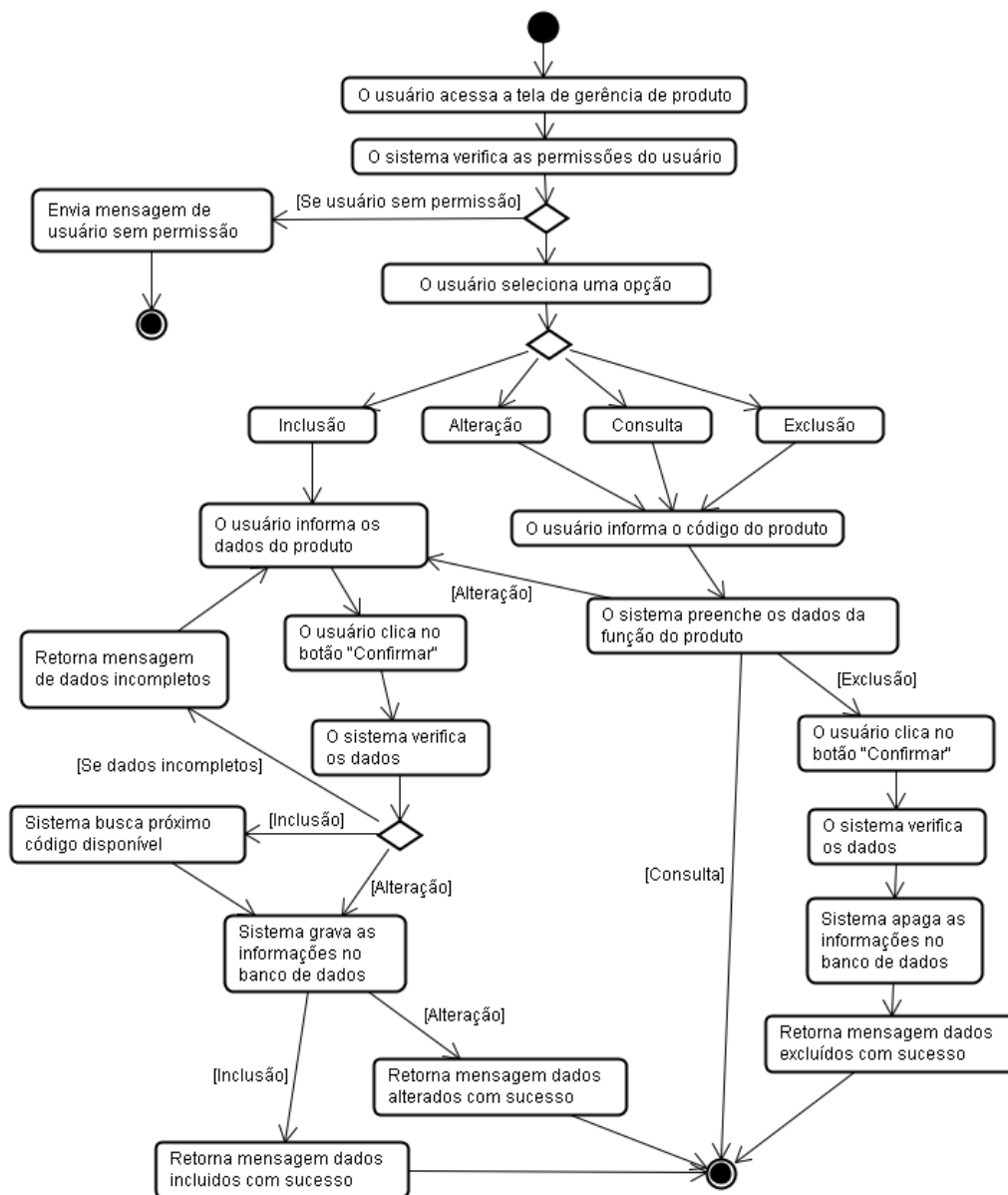


Figura 20 – Diagrama de Atividade Gerência de produto.

Fonte: autor (2010).

UC016 – Gerência de pessoa

Breve descritivo: este caso de uso descreve o processo de gerenciamento de pessoa.

Requisitos funcionais:	REQF16 – [Cadastrar pessoa] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções]

Atores:	Administrador, Gerente, Caixa.
Pré-condições:	Ser usuário com permissão de administrador/gerente/caixa.
Fluxo principal:	<p>O usuário administrador/gerente/caixa acessa a tela de gerenciamento de vendedor.</p> <p>Os subfluxos são apresentados pelo sistema: Inclusão, Alteração, Consulta e Exclusão.</p>
Subfluxos:	<p>1. Inclusão:</p> <p>1.1 O usuário seleciona a opção de inclusão;</p> <p>1.2 O usuário informa dos dados de uma nova pessoa: nome, CPF, CNPJ, inscrição estadual, logradouro, número, complemento, bairro, município, número de telefone, número de telefone, celular, e-mail, data do cadastro e situação do cadastro;</p> <p>1.3 O usuário clica no botão/link “Salvar”;</p> <p>1.4 O sistema verifica os dados;</p> <p>1.5 O sistema preenche o código com o próximo disponível</p> <p>1.6 O sistema grava as informações no banco de dados;</p> <p>1.7 O sistema retorna a mensagem “Dados incluídos com sucesso”.</p> <p>2. Alteração:</p> <p>2.1 O usuário seleciona a opção de alteração;</p> <p>2.2 O usuário informa:</p> <p>2.2.1 O código da pessoa que deseja alterar;</p> <p>2.2.2 O CPF da pessoa que deseja alterar;</p> <p>2.2.3 O CNPJ da pessoa que deseja alterar;</p> <p>2.3 O sistema preenche os dados do vendedor;</p> <p>2.4 O usuário altera os dados desejados;</p> <p>2.5 O usuário clica no botão/link “Salvar”;</p> <p>2.6 O sistema verifica os dados;</p> <p>2.7 O sistema grava as informações no banco de dados;</p> <p>2.8 O sistema retorna a mensagem “Dados alterados com sucesso”.</p> <p>3. Consulta:</p> <p>3.1 O usuário seleciona a opção de consulta;</p> <p>3.2 O usuário informa:</p> <p>3.2.1 O código da pessoa que deseja consultar;</p> <p>3.2.2 O CPF da pessoa que deseja consultar;</p> <p>3.2.3 O CNPJ da pessoa que deseja consultar;</p> <p>3.3 O sistema preenche os dados do vendedor.</p> <p>4. Exclusão:</p>

	<p>4.1 O usuário seleciona a opção de exclusão;</p> <p>4.2 O usuário informa;</p> <p> 4.2.1 O código da pessoa que deseja excluir;</p> <p> 4.2.2 O CPF da pessoa que deseja excluir;</p> <p> 4.2.3 O CNPJ da pessoa que deseja excluir;</p> <p>4.3 O sistema preenche os dados do vendedor;</p> <p>4.4 O usuário clica no botão/link “Excluir”;</p> <p>4.5 O sistema verifica os dados;</p> <p>4.6 O sistema exclui as informações no banco de dados;</p> <p>4.7 O sistema retorna a mensagem “Dados excluídos com sucesso”.</p>
Tratamento de exceções:	<p>1.4.1 O sistema deve verificar se os campos obrigatórios foram preenchidos e retornar a mensagem “Preencha todos os campos obrigatórios.” caso algum não tenha sido preenchido;</p> <p>2.2.1.1 Se o código informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p> <p>2.2.1.2 Se o CPF informado não estiver registrado no banco de dados, o sistema retorna a mensagem “CPF não cadastrado”;</p> <p>2.2.1.3 Se o CNPJ informado não estiver registrado no banco de dados, o sistema retorna a mensagem “CNPJ não cadastrado”;</p> <p>2.6.1 Recebe o mesmo tratamento de exceção do item 1.4.1;</p> <p>3.2.1.1 Recebe mesmo tratamento de exceção do item 2.2.1.1;</p> <p>3.2.1.2 Recebe mesmo tratamento de exceção do item 2.2.1.2;</p> <p>3.2.1.3 Recebe mesmo tratamento de exceção do item 2.2.1.3;</p> <p>4.2.1.1 Recebe mesmo tratamento de exceção do item 2.2.1.1;</p> <p>4.2.1.2 Recebe mesmo tratamento de exceção do item 2.2.1.2;</p> <p>4.2.1.3 Recebe mesmo tratamento de exceção do item 2.2.1.3;</p> <p>4.5.1 Recebe mesmo tratamento de exceção do item 1.4.1.</p>
Pós-condições:	<p>1. Uma nova pessoa está cadastrada.</p> <p>2. Uma pessoa está alterada;</p> <p>3. Uma pessoa está consultada;</p> <p>4. Uma pessoa está excluída.</p>

Quadro 2 – Caso de Uso Gerência de pessoa.

Fonte: autor (2010).

A Figura 21 mostra o Diagrama de Atividade referente à atividade de gerência de pessoa.

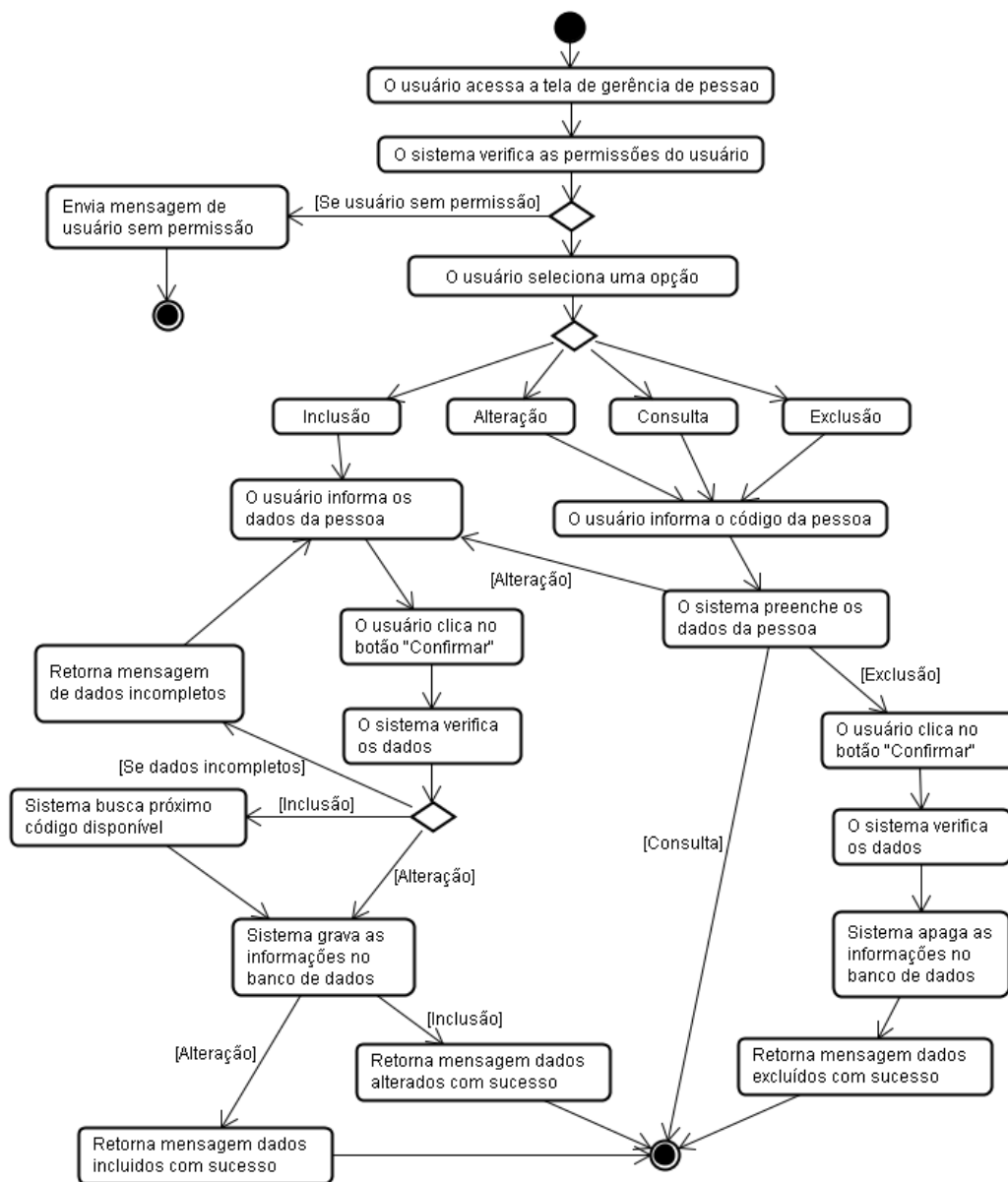


Figura 21 – Diagrama de Atividade Gerência de pessoa.
 Fonte: autor (2010).

UC022 – Abertura de caixa

Breve descritivo: este caso de uso descreve o processamento de abertura de caixa.

Requisitos funcionais:	REQF22 – [Abrir caixa] REQF29 – [Controlar acesso ao sistema] REQF30 – [Comunicação com a impressora fiscal]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções]

	REQNF4 – [Conformidade com a legislação]
Atores:	Gerente, Caixa.
Pré-condições:	Ser usuário com permissão de gerente/caixa; Caixa deve estar fechado.
Fluxo principal:	<ol style="list-style-type: none"> 1. O usuário gerente/caixa acessa a tela de abertura de caixa; 2. O usuário informa o valor do suprimento de caixa; 3. O usuário clica no botão “Abrir caixa”; 4. O sistema verifica os dados; 5. O sistema imprime o relatório de abertura de caixa; 6. O sistema grava as informações no banco de dados.
Tratamento de exceções:	3.1 O sistema deve verificar qual a situação do caixa. Se este estiver aberto, o sistema deve retornar a mensagem “O caixa está aberto.”
Pós-condições:	O Caixa está aberto e apto a emitir venda.

Quadro 3 – Caso de Uso Abertura de caixa.

Fonte: autor (2010).

O Diagrama de Atividade referente à abertura de caixa é apresentado na Figura 22.

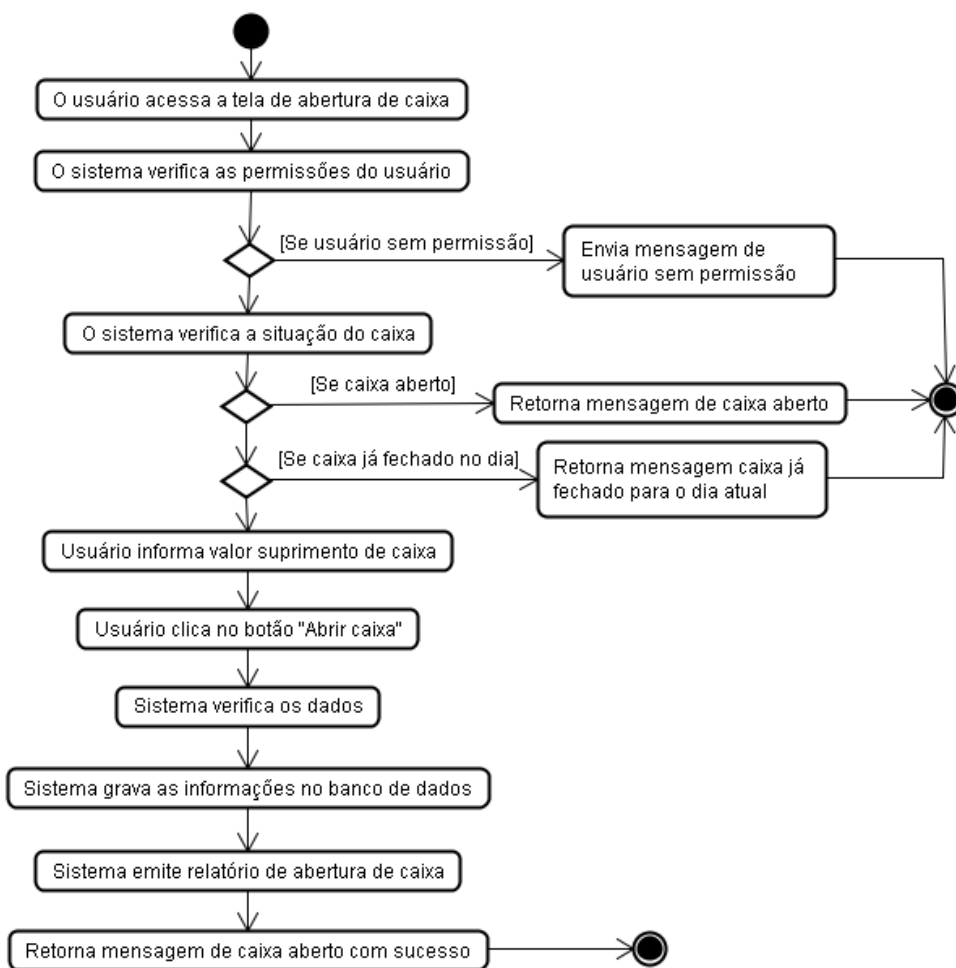


Figura 22 – Diagrama de Atividade Abertura de caixa.

Fonte: autor (2010).

UC023 – Fechamento de caixa

Breve descritivo: este caso de uso descreve o processamento de fechamento de caixa.

Requisitos funcionais:	REQF3 – [Cadastrar terminal] REQF23 – [Fechar caixa] REQF29 – [Controlar acesso ao sistema] REQF30 – [Comunicação com a impressora fiscal]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções] REQNF4 – [Conformidade com a legislação]
Atores:	Gerente, Caixa.

Pré-condições:	Ser usuário com permissão de gerente/caixa; Caixa deve estar aberto.
Fluxo principal:	1. O usuário gerente/caixa acessa a tela de fechamento de caixa; 2. O usuário clica no botão “Fechar caixa”; 3. O sistema verifica os dados; 4. O sistema imprime o relatório de fechamento de caixa; 5. O sistema grava as informações no banco de dados.
Tratamento de exceções:	3.1 O sistema deve verificar qual a situação do caixa. Se este estiver fechado, o sistema deve retornar a mensagem “O caixa está fechado.”
Pós-condições:	O Caixa está fechado e não permite emitir venda.

Quadro 4 – Caso de Uso Fechamento de caixa.

Fonte: autor (2010).

A Figura 23 mostra o Diagrama de Atividade referente à atividade de abertura de caixa.

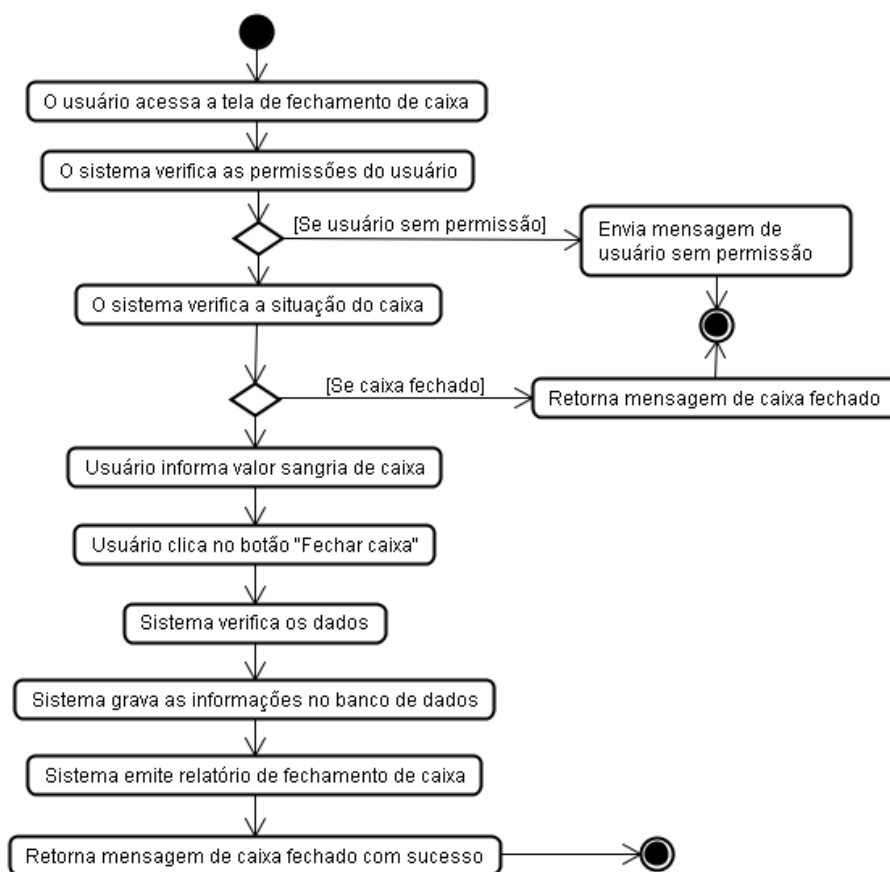


Figura 23 – Diagrama de Atividade Fechamento de caixa.

Fonte: autor (2010).

UC025 – Emissão de venda

Breve descritivo: este caso de uso descreve o processamento de emissão de venda.

Requisitos funcionais:	<p>REQF3 – [Cadastrar terminal]</p> <p>REQF13 – [Cadastrar produto]</p> <p>REQF14 – [Cadastrar preço]</p> <p>REQF15 – [Cadastrar forma de pagamento]</p> <p>REQF16 – [Cadastrar pessoa]</p> <p>REQF17 – [Cadastrar vendedor]</p> <p>REQF18 – [Cadastrar consumidor]</p> <p>REQF25 – [Emitir venda]</p> <p>REQF29 – [Controlar acesso ao sistema]</p> <p>REQF30 – [Comunicação com a impressora fiscal]</p>
Requisitos não funcionais:	<p>REQNF1 – [Utilização de banco de dados relacional]</p> <p>REQNF2 – [Utilização de servidor web]</p> <p>REQNF3 – [Links/ícones/botões para acesso as principais funções]</p> <p>REQNF4 – [Conformidade com a legislação]</p>
Atores:	Gerente, Caixa.
Pré-condições:	<p>Ser usuário com permissão de caixa;</p> <p>Caixa deve estar aberto.</p>
Fluxo principal:	<ol style="list-style-type: none"> 1. O usuário gerente/caixa acessa a tela de emissão de venda; 2. O usuário informa o vendedor e consumidor da venda; 3. O sistema verifica os dados; 4. O sistema inicia a impressão do cupom fiscal; 5. O usuário informa o código do produto e a quantidade a ser vendido; 6. O sistema verifica os dados; 7. O sistema preenche na tela a descrição e preço do produto; 8. O item é impresso no cupom fiscal; 9. O usuário seleciona a forma de pagamento; 10. O usuário clica no botão “Finalizar venda”; 11. O sistema verifica os dados; 12. O sistema finaliza a impressão do cupom fiscal; 13. O sistema grava as informações no banco de dados.
Tratamento de exceções:	<ol style="list-style-type: none"> 3.1 O sistema deve verificar se foi selecionado o vendedor. Se não foi selecionado, o sistema deve retornar a mensagem “Selecione o vendedor”. 6.1 O sistema deve verificar se o produto informado está cadastrado. Se não

	<p>estiver, o sistema deve retornar a mensagem “Produto não cadastrado”.</p> <p>11.1 O sistema deve verificar se foi selecionada a forma de pagamento. Se não foi selecionado, o sistema deve retornar a mensagem “Selecione a forma de pagamento”.</p>
Pós-condições:	Uma nova venda é realizada.

Quadro 5 – Caso de Uso Emissão de venda.

Fonte: autor (2010).

O Diagrama de Atividade referente à abertura de caixa é apresentado na Figura 24.

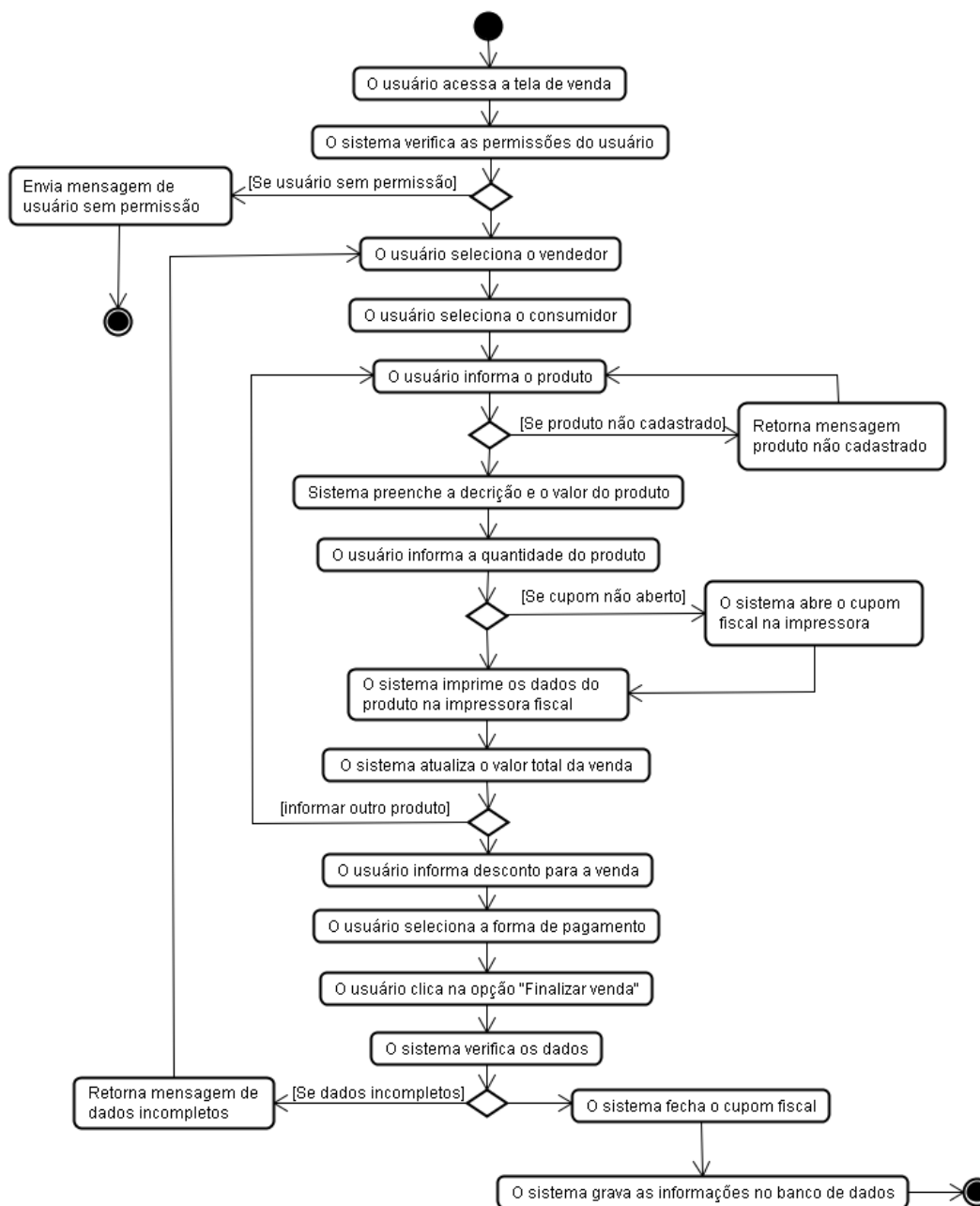


Figura 24 – Dia grama de Atividade Emissão de venda.

Fonte: autor (2010).

3.4 DIAGRAMA DE CLASSES

Fowler (2004, p. 52), cita que um diagrama de classes descreve os tipos de objetos presentes no sistema e os vários tipos de relacionamentos estáticos entre eles, mostrando as propriedades e as operações de uma classe e as restrições que se aplicam à maneira como os objetos estão conectados.

Segundo Booch et al. (2005, p.107), um diagrama de classes mostra um conjunto de classes, interfaces e colaborações e seus relacionamentos, e é importante para a visualização, a especificação e a documentação dos modelos estruturais, além da construção de sistemas executáveis por intermédio de engenharia de produção reversa.

O Diagrama de Classes apresenta uma visão estática de como as classes estão organizadas, definindo sua estrutura lógica. O principal enfoque desse diagrama, para Guedes (2008, p. 75), está em permitir a visualização das classes que farão parte do sistema com seus respectivos atributos e métodos, assim como demonstrar como as classes do diagrama se relacionam, complementam e transmitem informações entre si. A Figura 25 apresenta o Diagrama de Classes do sistema proposto.

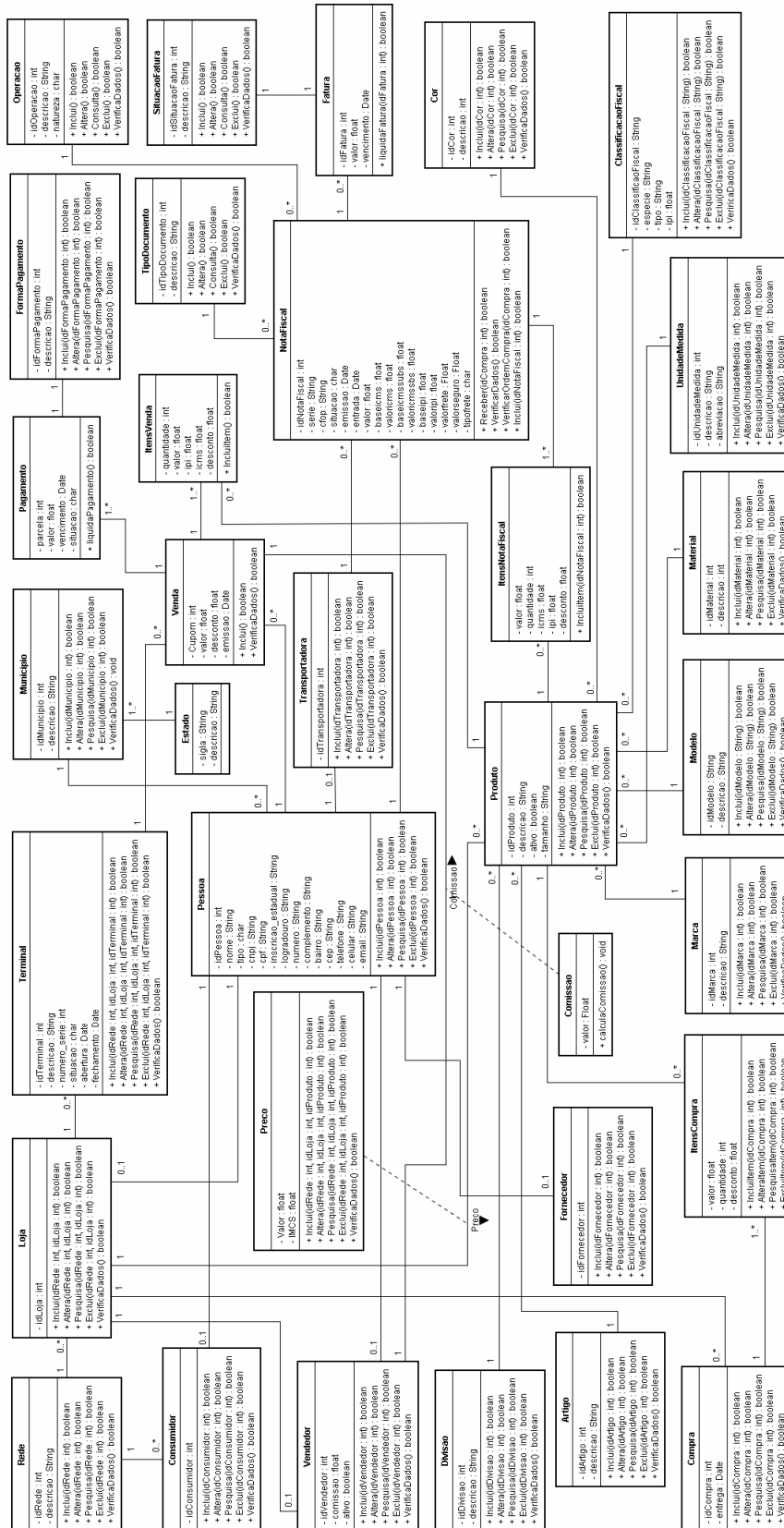


Figura 25 – Diagrama de Classes do sistema.
Fonte: autor (2010).

3.5 DIAGRAMA DE SEQUÊNCIA

O Diagrama de Sequência determina “a sequência de eventos que ocorrem em um determinado processo, ou seja, quais as condições devem ser satisfeitas e quais métodos devem ser disparados entre os objetos envolvidos e em que ordem durante um processo específico” (GUEDES, 2008, p. 113).

Os diagramas de sequência enfatizam o ordenamento temporal das ações (PAULA FILHO, 2003, p. 135), apresentando vários exemplos de objetos e mensagens que são passadas entre esses objetos dentro de um caso de uso (FOWLER, 2004, p. 67). Segundo Guedes (2008, p. 113), o principal objetivo deste diagrama é determinar a ordem em que os eventos ocorrem, as mensagens que são enviadas, os métodos que são chamados e como os objetos interagem dentro de um determinado processo.

3.5.1 Diagrama de Sequência – Gerência de produto

Na Figura 26 é apresentado o diagrama de sequência referente à gerência de produto.

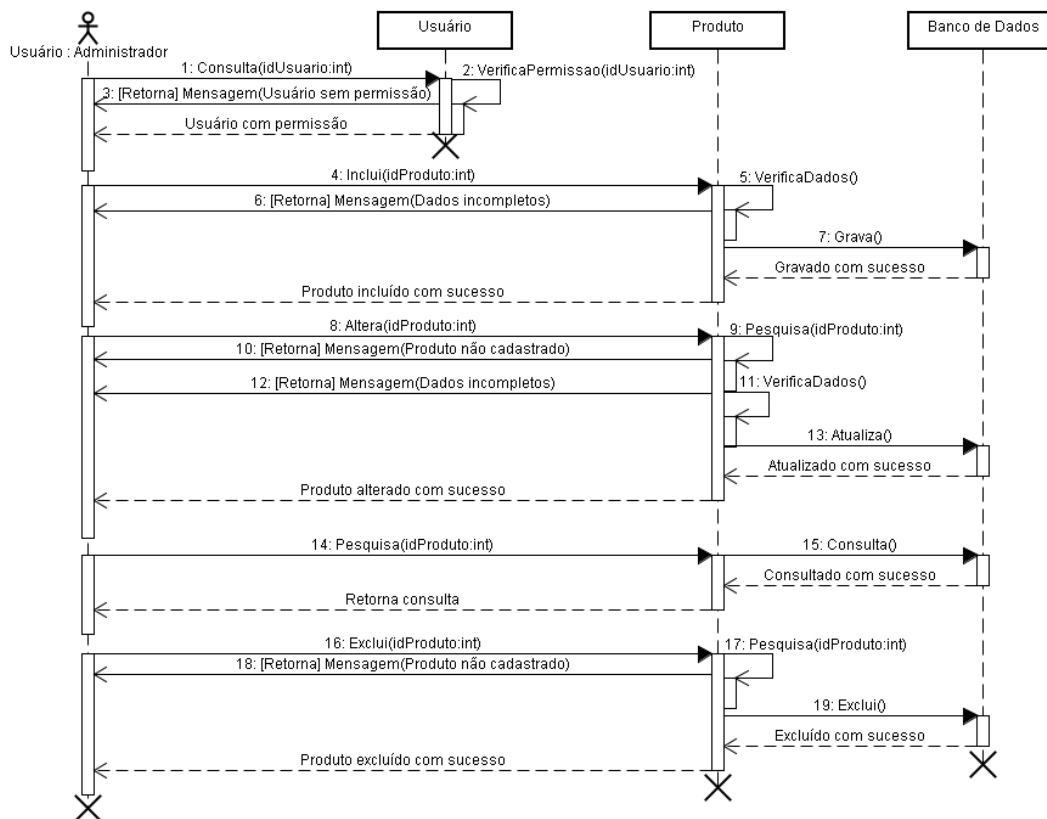


Figura 26 – Diagrama de Sequência Gerência de produto.

Fonte: autor (2010).

3.5.2 Diagrama de Sequência – Gerência de pessoa

O diagrama de sequência referente à gerência de pessoa é apresentado na Figura 27.

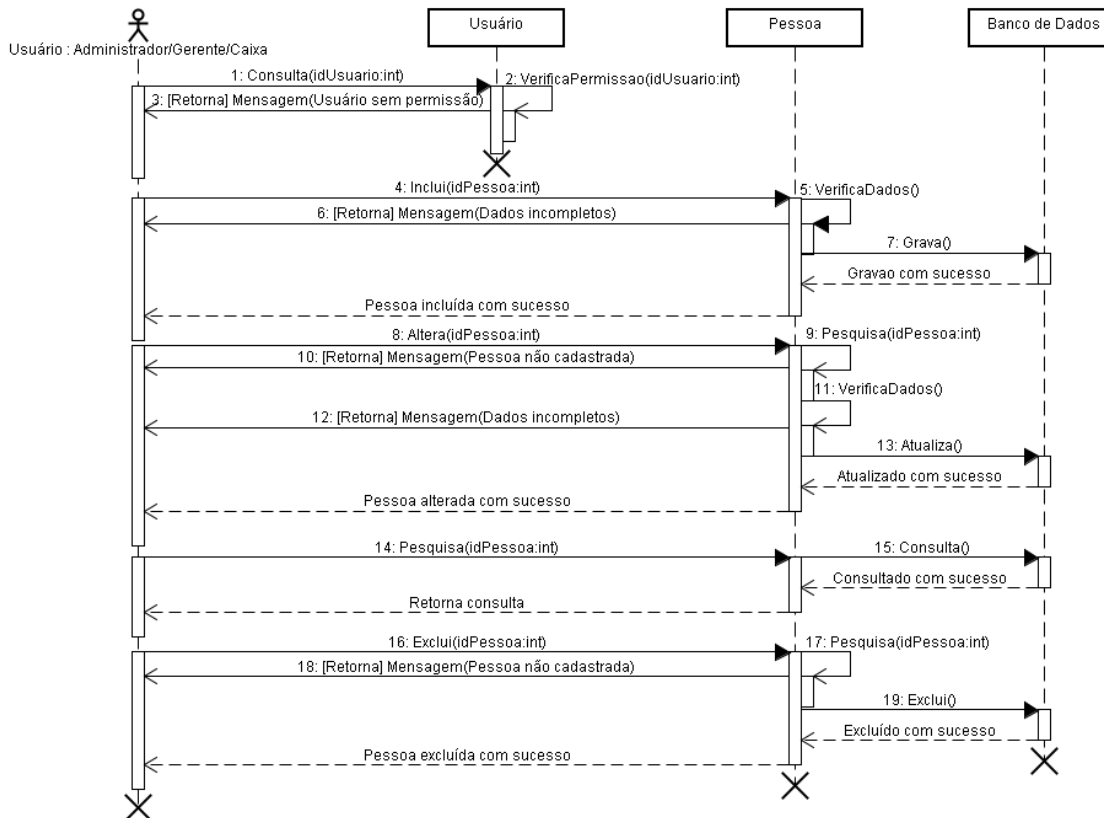


Figura 27 – Diagrama de Sequência Gerência de pessoa.
 Fonte: autor (2010).

3.5.3 Diagrama de Sequência – Abertura de caixa

A Figura 28 é apresenta o diagrama de sequência referente à abertura de caixa.

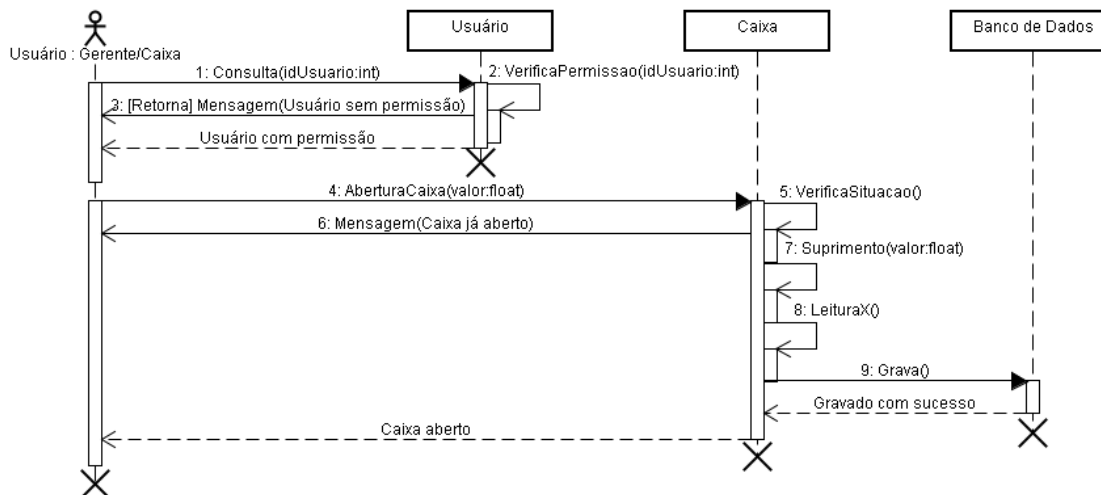


Figura 28 – Diagrama de Sequência Abertura de caixa.
 Fonte: autor (2010).

3.5.4 Diagrama de Sequência – Fechamento de caixa

O diagrama de sequência referente ao fechamento de caixa é mostrado na Figura 29.

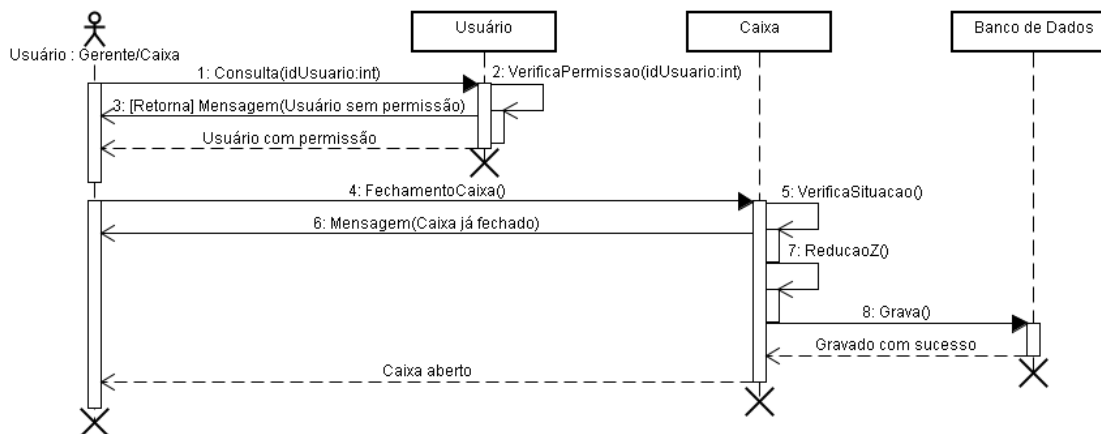


Figura 29 – Diagrama de Sequência Fechamento de caixa.
 Fonte: autor (2010).

3.5.5 Diagrama de Sequência – Emissão de venda

A Figura 30 apresenta o diagrama de sequência referente à emissão de venda.

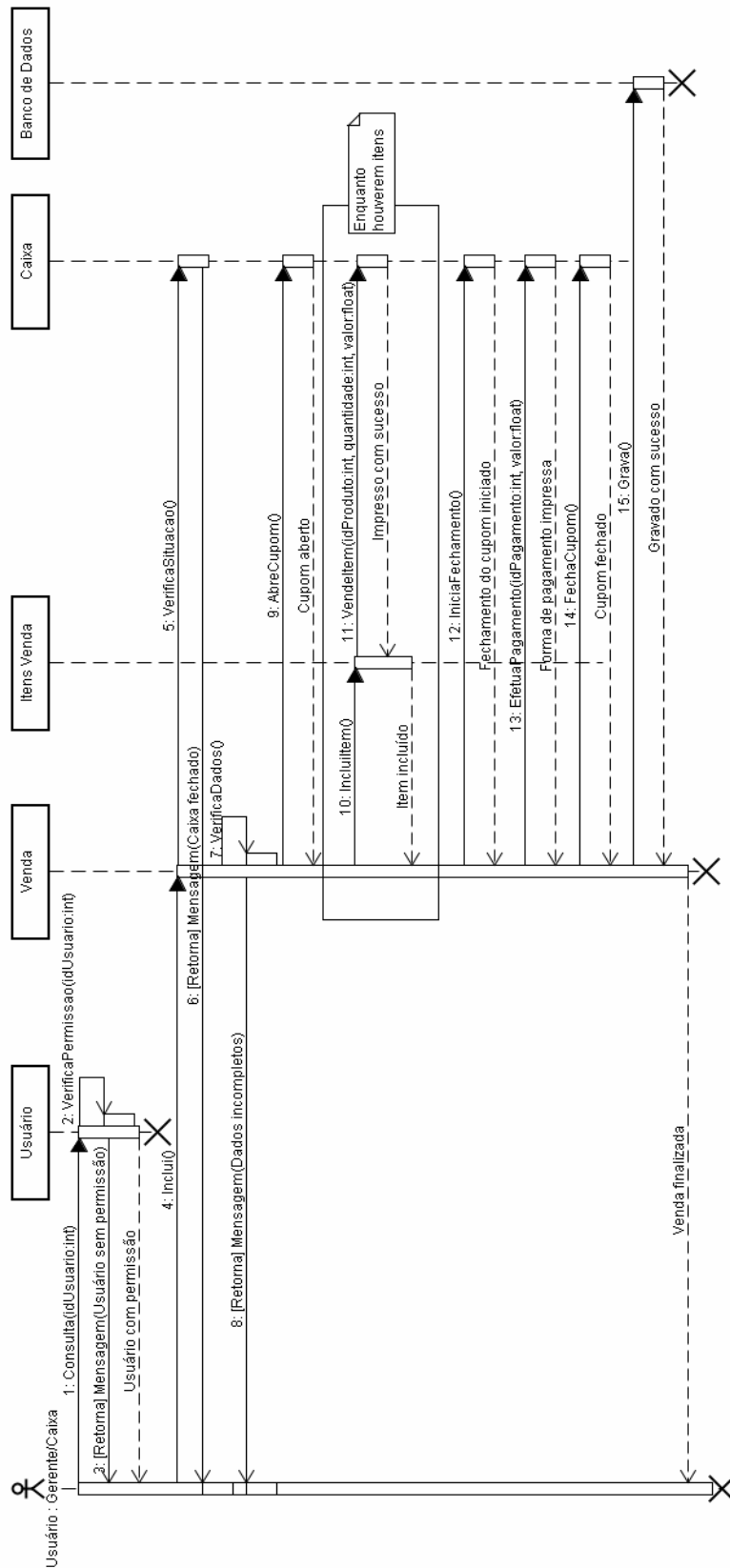


Figura 30 – Diagrama de Sequência Emissão de venda.
 Fonte: autor (2010).

3.6 DIAGRAMA DE ENTIDADE RELACIONAMENTO – DER

Para Dennis e Wixom (2005, p. 174), um diagrama de entidade relacionamento (DER) é um retrato que mostra as informações que serão criadas, armazenadas e usadas pelo sistema. Em um DER, tipos de informações semelhantes são listados juntos e colocados dentro de caixas denominadas entidades. Linhas são desenhadas entre as entidades para representar os relacionamentos entre os dados e símbolos especiais são adicionados ao diagrama para transmitir regras operacionais que devem ser suportadas pelo sistema.

Yourdon (1990) descreve que o DER é um modelo de rede que descreve a diagramação dos dados armazenados de um sistema em alto nível de abstração, sendo representados através de quatro componentes: tipos de objetos, relacionamentos, indicadores associativos de tipos de objetos e indicadores de subtipos. Além disso, ele também apresenta informações como tipo de chaves e índices ou ponteiros que são eficientes para obter acesso eficiente aos registros do banco de dados. A seguir, a Figura 31 mostra o diagrama de entidade relacionamento do sistema proposto.

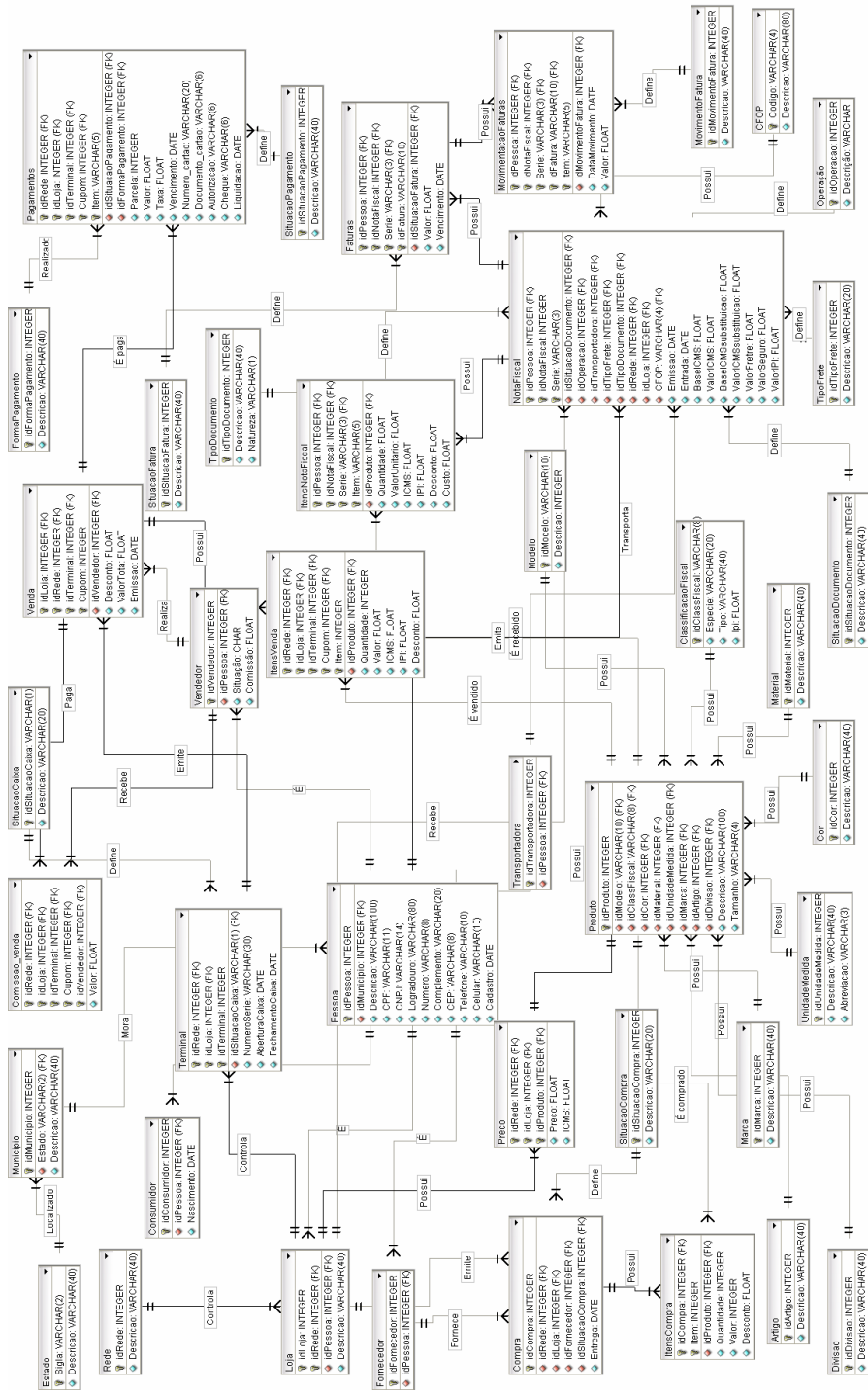


Figura 31 – Diagrama entidade relacionamento – DER.
 Fonte: autor (2010).

Através dos casos de uso, diagramas de atividades, diagrama de classes, diagramas de sequência e do diagrama de entidade relacionamento descritos nesse capítulo, é possível a implementação do sistema proposto. O próximo capítulo apresenta o desenvolvimento do módulo de frente de caixa do sistema.

4 DESENVOLVIMENTO DO SISTEMA

Este trabalho apresenta a análise, a modelagem e a implementação do módulo de frente de caixa de um sistema de varejo para calçados e acessórios, que atenda às necessidades de uma empresa do ramo.

4.1 LINGUAGEM DE PROGRAMAÇÃO

Para o desenvolvimento do módulo de frente de caixa do sistema proposto, foi utilizada a linguagem de programação PHP 5, através do paradigma de orientação a objetos, além de JavaScript e AJAX. O JavaScript foi criado como um modo de controlar a entrada de dados do usuário e acrescentar interesse e interatividade às páginas web, sendo uma linguagem de programação de scripts, poderosa, que pode ser usada para aumentar a interatividade. Já o *Asynchronous JavaScript and XML* (AJAX) não é uma tecnologia e sim o uso de tecnologias incorporadas tendo como as principais o JavaScript e o XML, onde juntos são capazes de tornar o navegador mais interativo, utilizando-se de solicitações assíncronas de informações (GONÇALVES, 2006).

4.2 AMBIENTE DE DESENVOLVIMENTO

O ambiente de desenvolvimento utilizado foi o EasyEclipse for PHP 1.2.2.2⁷. O EasyEclipse é um dos vários projetos derivados do Eclipse padrão, sendo este, um dos projetos Java para *desktop* de maior sucesso. De acordo com Lisboa (2008, p. 173), o projeto Eclipse é *open source*, que permite a customização para o uso em diferentes tipos de projeto java (*desktop*, web, dispositivos móveis) e para outras linguagens de programação, como Python, Ruby e PHP. Como *framework* de desenvolvimento foi utilizado o Zend Framework 1.5.0⁸. As características e vantagens do Zend Framework estão descritas na seção 2.3. Como servidor da aplicação, foi utilizado o Apache HTTP Server 2.2.11⁹, que é um servidor *open source* muito estável e seguro (XAVIER, 2008, p. 18).

⁷ <http://www.easyeclipse.org/site/distributions/php.html>

⁸ <http://framework.zend.com/download>

⁹ <http://www.apache.org/dist/httpd/binaries/win32>

4.3 SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS

Como Sistema de Gerenciamento de Banco de Dados Objeto-Relacional foi usado o PostgreSQL 8.4.0¹⁰, pois, além de ser *open source*, possui as características descritas na seção 2.4.2. A ferramenta de administração do PostgreSQL foi o PgAdmin III 1.8.0¹¹, que também é uma ferramenta que possui a característica de possuir o código fonte aberto.

4.4 COMUNICAÇÃO COM A IMPRESSORA FISCAL

A comunicação com a impressora fiscal foi realizada através da instalação, configuração e registro na máquina do cliente da DLL BemaFI32.dll¹², que é “considerada de alto nível, pela versatilidade e facilidade de programação, reduzindo em até 80% o tempo de desenvolvimento do aplicativo” (BEMATECH, 2009), e do componente para as aplicações web BemaFisc.ocx¹³. O uso da DLL de comunicação tornou simples o desenvolvimento das rotinas de interação com a impressora fiscal, conforme descrito nos exemplos dos Quadros 6, 7 e 8.

```
<object classid="clsid:2F9082AF-E2A7-4F20-9D6B-3855D79A3D86" id="BemaWeb">
```

Quadro 6 – Declaração do componente BemaFisc.ocx no código HTML.

Fonte: autor (2010).

O Quadro 7 descreve a rotina de verificação de um produto, usando AJAX, que foi digitado pelo usuário na tela de venda do sistema. A função “verificaProduto” recebe dois parâmetros: o código do produto e quantidade vendida deste item. Quando esta função é invocada, o sistema verifica se o produto digitado é válido buscando o código no banco de dados. Caso seja encontrado, a rotina chama a função “imprimeItem” passando como

¹⁰ <http://www.postgresql.org/download>

¹¹ <http://www.pgadmin.org/download>

¹² http://www2.bematech.com.br/suporte/dwn_fisc_windows.asp

¹³ http://www2.bematech.com.br/suporte/dwn_fisc_windows.asp

parâmetro os atributos do produto. Caso contrário, o sistema retorna a mensagem “Produto não cadastrado”.

```
function verificaProduto(produto, qtd) {
    ajax = ajaxInit();
    if(ajax) {
        ajax.open("GET", "/loja/public/ajax/produtos.php?codigo="+escape(
            produto), true);
        ajax.setRequestHeader("Content-Type", "application/x-www-form-
            urlencoded");
        ajax.onreadystatechange = function() {
            if(ajax.readyState == 4) {
                if(ajax.responseXML) {
                    var dataArray =
                        ajax.responseXML.getElementsByTagName("produto");
                    if(dataArray.length > 0) {
                        for(var i = 0 ; i < dataArray.length ; i++) {
                            var item = dataArray[i];
                            imprimeItem(produto, item.getElementsByTagName(
                                "descricao")[0].firstChild.nodeValue, 'II', 'I', qtd,
                                '2', item.getElementsByTagName("preco")[0].
                                    firstChild.nodeValue, '%', '0')
                        }
                    } else {
                        alert("Produto não cadastrado.");
                    }
                } else {
                    alert("Problema na leitura do produto.");
                }
            }
        }
        ajax.send(null);
    }
}
```

Quadro 7 – Verificação do produto digitado em AJAX.
Fonte: autor (2010).

Finalmente, o Quadro 8 apresenta a declaração da função “imprimeItem” no código JavaScript do sistema, fazendo referência ao “id” do componente BemaFisc.ocx. A função chama o método “vendeItem” do componente, que realiza a impressão dos dados na impressora fiscal e retorna o seu status.

```
function imprimeItem(CodigoProduto, Descricao, Aliquota, sTipoQtde,
    Quantidade, iDecimal, Valor, sTipoDesconto, Desconto){
    iRetorno = BemaWeb.VendeItem(CodigoProduto, Descricao, Aliquota,
        sTipoQtde, Quantidade, iDecimal, Valor, sTipoDesconto,
        Desconto);
}
```

Quadro 8 – Declaração da função de impressão do item na impressora fiscal.
Fonte: autor (2010).

4.5 DESENVOLVIMENTO DO MÓDULO DE FRENTE DE CAIXA

O sistema de frente de caixa foi desenvolvido com base no Diagrama Entidade Relacionamento da Figura 31 e os requisitos desenvolvidos no presente trabalho foram

REQF13 – Cadastrar produto, REQF15 – Cadastrar forma de pagamento, REQF17 – Cadastrar vendedor, REQF18 – Cadastrar consumidor, REQF22 – Abrir caixa, REQF23 – Fechar caixa, REQF25 – Emitir venda, REQF26 – Emitir suprimento de caixa, REQF27 – Emitir sangria de caixa, REQF29 – Gerar relatórios, REQF30 – Controlar acesso ao sistema e REQF31 – Comunicação com impressora fiscal. A seguir são apresentadas algumas das telas do módulo de frente de caixa desenvolvidas.

Tela de permissão de acesso ao sistema.

A screenshot of a login form. The form is titled 'Login' and has a grey background. It contains two input fields: 'Usuário:' with the text 'administrador' and 'Senha:' with a masked password of ten dots. Below the fields is a button labeled 'Enviar'.

Empresa de calçados e acessórios

Figura 32 – Tela de permissão de acesso.
Fonte: autor (2010).

Tela de cadastro de consumidor.

The screenshot shows a web browser window with the URL `http://localhost/loja/cadastropessoa`. The page has a blue header with the text "Rede de lojas" and a navigation menu with links: "Home", "Configurações", "Caixa", "Cadastros", "Movimentos", and "Relatórios". Below the header is a form titled "Cadastro de consumidor". The form contains the following fields and values:

Código:	00000002		
Nome:	Fernando Bloedom		
CPF:	001.098.290-69		
Logradouro:	Rua Canela		
Número:	256	Complemento:	Apartamento 101
Bairro:	Pousada da Neve	CEP:	95150-000
Estado:	RS	Cidade:	Nova Petrópolis
Telefone:	(54)3281-8000	Celular:	(54)9142-8320
E-mail:	fernandobloedom@gmail.com		

At the bottom of the form are five buttons: "Incluir", "Excluir", "Alterar", "Consultar", and "Limpar". Below the form is a red footer bar with the text "Empresa de calçados e acessórios".

Figura 33 – Tela de cadastro de consumidor.
Fonte: autor (2010).

A tela de cadastro de consumidor, refere-se ao caso de uso UC018 – Gerência de consumidor, e aos diagramas de atividade da Figura Anexo 31 e de sequência da Figura Anexo 32.

A seguir a tela de cadastro de produto.

tórico Favoritos Ferramentas Ajuda

http://localhost/loja/cadastroproduto

passos Últimas notícias

Rede de lojas

Home Configurações Caixa Cadastros Movimentos Relatórios

Cadastro de produto

Código:	7891013441989		
Descrição:	Tamenco L1141 Preto 38		
Preço:	R\$ 49,90		
Divisão:	Calçados	Artigo:	Tamenco
Marca:	Marca	Modelo:	L1141
Tamanho:	38		
NCM:	64.03.99.00	Unidade medida:	Pares

Incluir Excluir Alterar Consultar Limpar

Empresa de calçados e acessórios

Figura 34 – Tela de cadastro de produto.

Fonte: autor (2010).

O cadastro de produto foi desenvolvido baseado no caso de uso UC013 – Gerência de produto, no diagrama de atividade da Figura 20 e no diagrama de seqüência da Figura 26.

A Tela do sistema referente a emissão de venda é apresentada na Figura 35.

The screenshot shows a web browser window with the URL `http://localhost/loja/venda`. The page has a blue header with the text "Rede de lojas" and a navigation menu with items: Home, Configurações, Caixa, Cadastros, Movimentos, Relatórios. Below the menu is a "Venda" form. The form contains a table with the following data:

Produto	?	Descrição	Qtd	Valor
7891502325790	?	Bota N0944 Preto 37	1	149.90
7890123401234	?	Sapato N931 Preto 34	1	99.90
7890964470550	?	Sapatilha M0123 Natural 33	2	39.90
<input type="text"/>	?			
<input type="text"/>	?			
<input type="text"/>	?			
<input type="text"/>	?			
<input type="text"/>	?			
<input type="text"/>	?			

Below the table is a "Consultar:" field and a "Qtd: 2" field. To the right of the table are fields for "Vendedor" (Maria da Silva), "Consumidor" (Fernando Bloedorn), and "Forma pagamento" (Dinheiro). Below these are summary fields: "Total a pagar: 329.60", "Valor pago: 350.00", and "Troco: 20.40". At the bottom right are buttons for "Finalizar venda" and "Cancelar". A red bar at the bottom of the page contains the text "Empresa de calçados e acessórios".

Figura 35 – Tela de emissão de venda.
Fonte: autor (2010).

A operação de emissão de venda foi implementada de acordo com o caso de uso UC025 – Emissão de venda e dos diagramas de atividade da Figura 24 e de sequência da Figura 30. A tela acima apresenta uma venda sendo realizada. À esquerda, a coluna “Produto” mostra que três itens diferentes estão sendo vendidos. A coluna “?” é um botão que realiza a consulta dos produtos através da descrição de acordo com a especificação do usuário no campo “Consultar” (parte inferior da tela), conforme mostrado na tela da Figura 36. Já as colunas “Descrição”, “Qtd” e “Valor” informam a descrição, a quantidade vendida e o preço unitário de cada produto, respectivamente. Já à direita constam a seleção de vendedor, consumidor (onde também existe a opção de consulta pelo nome) e forma de pagamento, além das informações referentes da operação: “Total a pagar” (atualizado pelo sistema), “Valor Pago” (preenchido pelo usuário) e “Troco” (também atualizado automaticamente pelo sistema no momento em que é informado o valor pago). O botão “Finalizar venda” confirma a emissão de venda, registrando as informações no banco de dados e finalizando a impressão do cupom fiscal. Já o botão “Cancelar venda” cancela a emissão da venda em qualquer momento da operação.

The screenshot displays a web browser window with a modal dialog box open. The modal dialog, titled 'Mozilla Firefox', contains a table with the following data:

Codigo	Descrição	Preço	Selecionar
7891013529496	Sapato N0361 Areia 37	149.90	Selecionar
7891013486447	Sapato N0552 Nude 36	79.90	Selecionar
7891013953710	Sapato N0614 Branco 37	109.90	Selecionar
7891502043694	Sapato N0941 Vermelho 35	149.90	Selecionar
7890123477889	Sapato N1012 Fibra 35	114.90	Selecionar
7890123401234	Sapato N931 Preto 34	99.90	Selecionar

Below the table is a 'Fechar' button. The background of the browser shows a search form with 'Sapato' entered in the 'Consultar:' field and a 'Finalizar venda' button. The footer of the page reads 'Empresa de calçados e acessórios'.

Figura 36 – Consulta de produto na emissão de venda.

Fonte: autor (2010).

A Figura 37 apresenta a tela referente ao gráfico de análise de vendas. O gráfico faz o comparativo das vendas de dois anos – 2008 e 2009 – em uma mesma loja. O eixo vertical representa o valor de vendas e o eixo horizontal representa todos os meses do ano. Desta forma é possível analisar em qual ano houve um maior valor de venda e as tendências de aumento e diminuição de vendas nos meses do período.



Figura 37 – Gráfico comparativo de vendas.
Fonte: autor (2010).

4.6 VALIDAÇÃO DO SISTEMA

Com a finalidade de validar o sistema de frente de caixa, foram elaboradas oito questões relacionadas à interface, usabilidade, desempenho, visualização dos dados e eficácia, comparando o sistema desenvolvido com o utilizado atualmente, conforme questionário no Anexo do presente trabalho. O questionário foi aplicado a um universo de 8 colaboradores da empresa que utilizam a aplicação diariamente. Para classificar cada uma das respostas, usou-se uma escala decimal de -5 até +5, sendo que a escala de pontuação negativa julgou que o sistema apresentado foi pior e a escala positiva representou melhoria com relação ao atual. Desta forma, o usuário foi instruído a assinalar cada uma das respostas de acordo com seu entendimento.

O mesmo teste de validação foi aplicado para cada um dos usuários, de acordo com as atividades diárias que eles operam no sistema. As atividades que o teste procurou avaliar foram as seguintes:

1. Apresentação da tela principal do sistema;
2. Abertura de caixa;

3. Apresentação dos cadastros e operações de inclusão, consulta, alteração, e exclusão de dados;
4. Apresentação da opção de venda, com processos de inclusão e cancelamento de venda;
5. Apresentação dos relatórios;
6. Fechamento de caixa.

A fim de obter o resultado final do questionário, foi calculada a média aritmética de cada uma das respostas. O resultado é apresentado na tabela abaixo:

Tabela 2 – Questionário de validação do sistema.

Quesito	Questão	Média respostas
Interface	1. O sistema lhe é amigável?	3,63
	2. As funções do sistema são apresentadas de forma clara?	3,63
Usabilidade	3. O sistema é fácil de operar?	3,13
	4. Sua interação com o sistema é clara e entendível?	3,00
Desempenho	5. O tempo de resposta com a impressora fiscal é eficiente?	3,25
	6. O ritmo de operação é satisfatório?	3,88
Visualização de dados	7. As informações apresentadas nos relatórios são compreensíveis?	1,75
Eficácia	8. O sistema é eficaz, no sentido de atender os objetivos que se propõe?	-1,00

Fonte: autor (2010).

A partir destas informações é correto afirmar que o sistema desenvolvido apresentou melhorias com relação ao sistema atual nos quesitos interface, usabilidade e desempenho. Já a visualização de dados apresentou uma média abaixo dos demais quesitos, mostrando a necessidade de detalhar mais as informações dos relatórios do sistema desenvolvido. O quesito eficácia foi considerado inferior ao sistema atual, pois, apesar da nova aplicação atender as necessidades do usuário, existem recursos específicos no sistema atual que não foram desenvolvidas. Desta forma, a metodologia aplicada, bem como o uso das ferramentas de desenvolvimento e banco de dados, tornou o sistema viável para a necessidade da empresa, tendo em vista que obteve um resultado satisfatório nos testes realizados.

CONCLUSÃO

O varejo tem se tornado cada vez mais importante no cenário nacional. Diversas empresas do ramo crescem em níveis acima dos registrados na economia, tornando-as fundamentais no ambiente econômico atual, e com a perspectiva de manter os resultados positivos nos próximos anos. Desta forma, empresas do setor buscam aquisição de softwares que as auxiliem nos processos e no controle de seus dados de forma prática e confiável.

A definição de novas tecnologias a serem utilizadas é algo vital para empresas que buscam o sucesso, sendo necessário um estudo técnico de acordo com o contexto, antes da tomada de decisão. Fatores como adoção de software livre, migração de sistemas legados e alteração de paradigmas no desenvolvimento de software, como orientação a objetos e *framework*, por exemplo, merecem de um prévio estudo de viabilidade técnica, econômica e operacional.

Sistemas web proporcionam uma interface universal de acesso à informação aos usuários de qualquer parte do mundo através de navegadores web, desta forma, ganharam bastante espaço, pois, além disso, oferecem excelentes vantagens como bom desempenho, baixo custo, portabilidade e flexibilidade de acesso aos dados pela internet.

Logo, a presente pesquisa insere uma empresa de calçados e acessórios em um melhor ambiente tecnológico, através da análise de viabilidade, a modelagem e o desenvolvimento do módulo de frente de caixa de um sistema de varejo com recursos web, desenvolvido exclusivamente com software livre, por meio de tecnologias como orientação a objetos e *framework*, trazendo novos conceitos para a área de TI da empresa.

O trabalho realizado trouxe vários benefícios, além de melhorar a usabilidade e a interação homem-computador através do sistema desenvolvido, apresentou soluções que expandem as tecnologias utilizadas pela empresa e, além disso, trazem novos conceitos para o desenvolvimento, a manutenção e a aquisição de software.

Com base no estudo de viabilidade técnica, econômica e operacional, e do resultado satisfatório do questionário aplicado aos usuários, o projeto mostrou-se aplicável à empresa de calçados e acessórios, sendo uma solução válida às suas necessidades. Além disso, a empresa mostrou-se interessada no resultado do trabalho desenvolvido, de modo que deseja fazer deste sistema um projeto piloto e implantá-lo em uma de suas lojas.

Como trabalhos futuros, sugere-se o aprofundamento teórico sobre a inserção de *web services* através de SOAP e a utilização destes recursos para a troca de informações entre o

sistema desenvolvido e o sistema atual da empresa, integrando as diferentes plataformas, tornando possível a migração por módulos do ambiente atual e, desta forma, aplicando a mesma metodologia apresentada neste trabalho a todo o sistema da empresa.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALLEN, Rob. **Getting Starded with the Zend Framework**. 2007. Disponível em: <<http://www.zend.com/community/downloads>>. Acesso em 24 de outubro de 2009.
- BEMATECH Suporte & Soluções. **Utilizando comunicação direta com a impressora fiscal Bematench**. Disponível em <http://www.bematech.com.br/suporte/dwn_fisc.asp>. Acesso em 07 de novembro de 2009.
- BLAHA, Michael; RUMBAUGH, James. **Modelagem e projetos baseados em objetos com UML 2**. 2. ed. São Paulo: Elsevier, 2006. p. 496.
- BOLETIM REGIONAL DO BANCO CENTRAL DO BRASIL. **Evolução do Comércio Varejista Regional**. 2008 Disponível em: <<http://www.bcb.gov.br/pec/boletimregional/port/2008/07/br200807b2p.pdf>>. Acesso em 24 de maio de 2009.
- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: guia do usuário**. 2.ed. Rio de Janeiro: Elsevier, 2005. 462 p.
- CAMARA, Fábio. **Orientação a objeto com .NET**. 2.ed. Florianópolis: Visual Books, 2006. 124 p.
- BUARQUE, Cristovam. **Avaliação econômica de projetos**. Rio de Janeiro: Campus, 1984. 266 p.
- CERAMI, Ethan. **Web Sevices Essentials**. O`Reilly & Assoc, 2002. 280 p.
- CONVERSE, Tim; PARK, Joyce. **PHP 4: a Bíblia**. Rio de Janeiro: Campus, 2001. 697 p.
- CUMMINS, Fred A. **Integração de sistemas: EAI – Enterprise Application Integration ...** Rio de Janeiro: Campus, 2002. 457 p.
- CUNHA, Davi. **Web Services, SOAP e Aplicações web**. 2002. Disponível em <http://devedge-temp.mozilla.org/viewsource/2002/soap-overview/index_pt_br.html>. Acesso em: 7 de junho de 2009.
- DENNIS, Alan; WIXOM, Barbara Haley. **Análise e projeto de sistemas**. 2. ed. Rio de Janeiro: Livros Técnicos e Científicos Editora S. A., 2005. 456 p.
- DIAS, Cláudia. **Usabilidade na web: criando portais mais acessíveis**. 2. ed. Rio de Janeiro: Alta Books, 2007. 296 p.
- DOCUMENTAÇÃO DO POSTGRESQL 8.0.0**. Rio de Janeiro, 2005. Disponível em: <<http://www.postgresql.org/docs>>. Acesso em: 24 de outubro de 2009.
- ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de bando de dados**. 4. ed. São Paulo: Pearson Addison Wesley, 2005. 724 p.
- FLOWLER, Martin. **UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos**. 3. ed. Porto Alegre: Artmed Editora S. A., 2004. 160 p.
- GAMMA, Erich et al. **Padrões de Projeto: Soluções reutilizáveis de software orientado a objetos**. Porto Alegre: Bookman, 2000. 364 p.
- GUEDES, Gilleanes T. A. **UML: uma abordagem prática**. 3. ed. São Paulo: Novatec Editora Ltda, 2008. p. 332.

- GONÇALVES, Edson. **Dominando AJAX**. Rio de Janeiro: Editora Ciência Moderna Ltda, 2006. 356 p.
- LEITE, Mário. **Técnicas de programação: uma abordagem moderna**. Rio de Janeiro: Brasport, 2006. 395 p.
- LERDORF, Rasmus. **PHP pocket reference**. Estados Unidos: O'Reilly & Associates, 2000. 114 p.
- LISBOA, Flávio Gomes da Silva. **Zend Framework: Desenvolvendo em PHP 5 orientado a objetos com MVC**. São Paulo: Novatec Editora Ltda, 2008. 184 p.
- LUIZE, Luis A. **PAF-ECF: O que é isso?** Curitiba, 2008. Disponível em: <<http://partners.bematech.com.br/bemacast/ac/2008/06/paf-ecf-o-que-e-isso/>>. Acesso em: 15 de setembro de 2009.
- MANUAL DO PHP**. PHP Documentation Group. 2009. Disponível em <http://php.net/manual/pt_BR/index.php>. Acesso em: 20 de setembro de 2009.
- NASCIMENTO, Hugo A. D. Do; FERREIRA, Cristiane B. R. **Visualização de Informações: uma abordagem prática**. XXV Congresso da Sociedade Brasileira de Computação. São Leopoldo: UNISINOS, 2005. Disponível em: <www.sbc.org.br/bibliotecadigital/download.php?paper=127>. Acesso em: 10 de outubro de 2009.
- PACHECO, Diego. **Qualidade e produtividade com MVC em PHP**. 2009. Disponível em: <http://imasters.uol.com.br/artigo/14723/php/qualidade_e_produtividade_com_mvc_em_php/>. Acesso em: 07 de outubro de 2009.
- PAULA FILHO, Wilson de Pádua. **Engenharia de software: fundamentos, métodos e padrões**. 2. ed. Rio de Janeiro: Livros Técnicos e Científicos Editora S.A., 2003. 600 p.
- ZEND TECHNOLOGIES INC. **Programmer's Reference Guide**. 2009 Disponível em: <<http://zendframework.com/manual/>>. Acesso em: 10 de outubro de 2009.
- RAMACHANDRAN, Vijay. **Design Patterns for Building Flexible and Maintainable J2EE Applications**. 2002. Disponível em: <<http://java.sun.com/developer/technicalArticles/J2EE/despat/>>. Acesso em: 23 de outubro de 2009.
- SALEMI, Joe. **Guia para banco de dados cliente/servidor**. Rio de Janeiro, RJ: Infobook, 1995. 376 p.
- SAIANI, Edmour. **Loja viva: Revolução no pequeno varejo brasileiro**. Rio de Janeiro: SENAC Rio, 2007. 151 p.
- SILVEIRA, Sérgio Amadeu da. **Software livre: a luta pela liberdade do conhecimento**. São Paulo, SP: Editora Fundação Perseu Abramo, 2004. 79 p.
- SNELL, James; TIDWELL, Doug, KULCHENKO, Pavel. **Programming Web Services with SOAP**. O'Reilly & Assoc, 2002. 237 p.
- TANENBAUM, Andrew S. **Sistemas operacionais modernos**. Rio de Janeiro: LTC – Livros Técnicos e Científicos Editora S.A., 1992. 487 p.
- TONSIG, Sérgio Luiz. **Engenharia de software: análise e projeto de sistemas. – 2ª ed. revista e ampliada**. Rio de Janeiro: Ciência Moderna Ltda., 2008. 319 p.

XAVIER, Fabrício S. V. **PHP – Do básico à orientação a objetos**. Rio de Janeiro: Ciência Moderna Ltda, 2008. 234 p.

WELLING, Luke; THOMSON, Laura. **PHP e MySQL: desenvolvimento web**. Rio de Janeiro: Elsevier Editora Ltda, 2005. 712 p.

YOURDON, Edward. **Análise estruturada moderna**. Rio de Janeiro: Campus, 1990. 836 p.

ANEXO

UC001 – Gerência de usuários

Breve descritivo: este caso de uso descreve o processo de gerenciamento de usuários no sistema.

Requisitos funcionais:	REQF1 – [Cadastrar usuário] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções]
Atores:	Administrador.
Pré-condições:	Ser usuário com permissão de administrador.
Fluxo principal:	O usuário administrador acessa a tela de gerenciamento de usuários. Os subfluxos são apresentados pelo sistema: Inclusão, Alteração, Consulta e Exclusão.
Subfluxos:	<ol style="list-style-type: none">1. Inclusão:<ol style="list-style-type: none">1.1 O usuário seleciona a opção de inclusão;1.2 O usuário informa dos dados de um novo usuário: nome completo, <i>login</i> (identificação), senha, tipo de acesso e situação do cadastro.1.3 O usuário clica no botão/link “Salvar”;1.4 O sistema verifica os dados;1.5 O sistema preenche o código com o próximo disponível;1.6 O sistema grava as informações no banco de dados;1.7 O sistema retorna a mensagem “Dados incluídos com sucesso”.2. Alteração:<ol style="list-style-type: none">2.1 O usuário seleciona a opção de alteração;2.2 O usuário informa o código do usuário que deseja alterar;2.3 O sistema preenche os dados do usuário;2.4 O usuário altera os dados desejados;2.5 O usuário clica no botão/link “Salvar”;2.6 O sistema verifica os dados;2.7 O sistema grava as informações no banco de dados;2.8 O sistema retorna a mensagem “Dados alterados com sucesso”.3. Consulta:<ol style="list-style-type: none">3.1 O usuário seleciona a opção de consulta;3.2 O usuário informa o código do usuário que deseja consultar;

	<p>3.3 O sistema preenche os dados do usuário.</p> <p>4. Exclusão:</p> <p>4.1 O usuário seleciona a opção de exclusão;</p> <p>4.2 O usuário informa o código do usuário que deseja excluir;</p> <p>4.3 O sistema preenche os dados do usuário;</p> <p>4.4 O usuário clica no botão/link “Excluir”;</p> <p>4.5 O sistema verifica os dados;</p> <p>4.6 O sistema exclui as informações no banco de dados;</p> <p>4.7 O sistema retorna a mensagem “Dados excluídos com sucesso”.</p>
Tratamento de exceções:	<p>1.4.1 O sistema deve verificar se os campos obrigatórios foram preenchidos e retornar a mensagem “Preencha todos os campos obrigatórios”, caso algum não tenha sido preenchido;</p> <p>2.2.1 Se o código informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p> <p>2.6.1 Recebe o mesmo tratamento de exceção do item 1.4.1;</p> <p>3.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.5.1 Recebe mesmo tratamento de exceção do item 1.4.1.</p>
Pós-condições:	O usuário cadastrado possui permissões de acesso ao sistema de acordo com o tipo de acesso definido pelo administrador.

Quadro Anexo 1 – Caso de Uso Gerência de usuários.

Fonte: autor (2010).

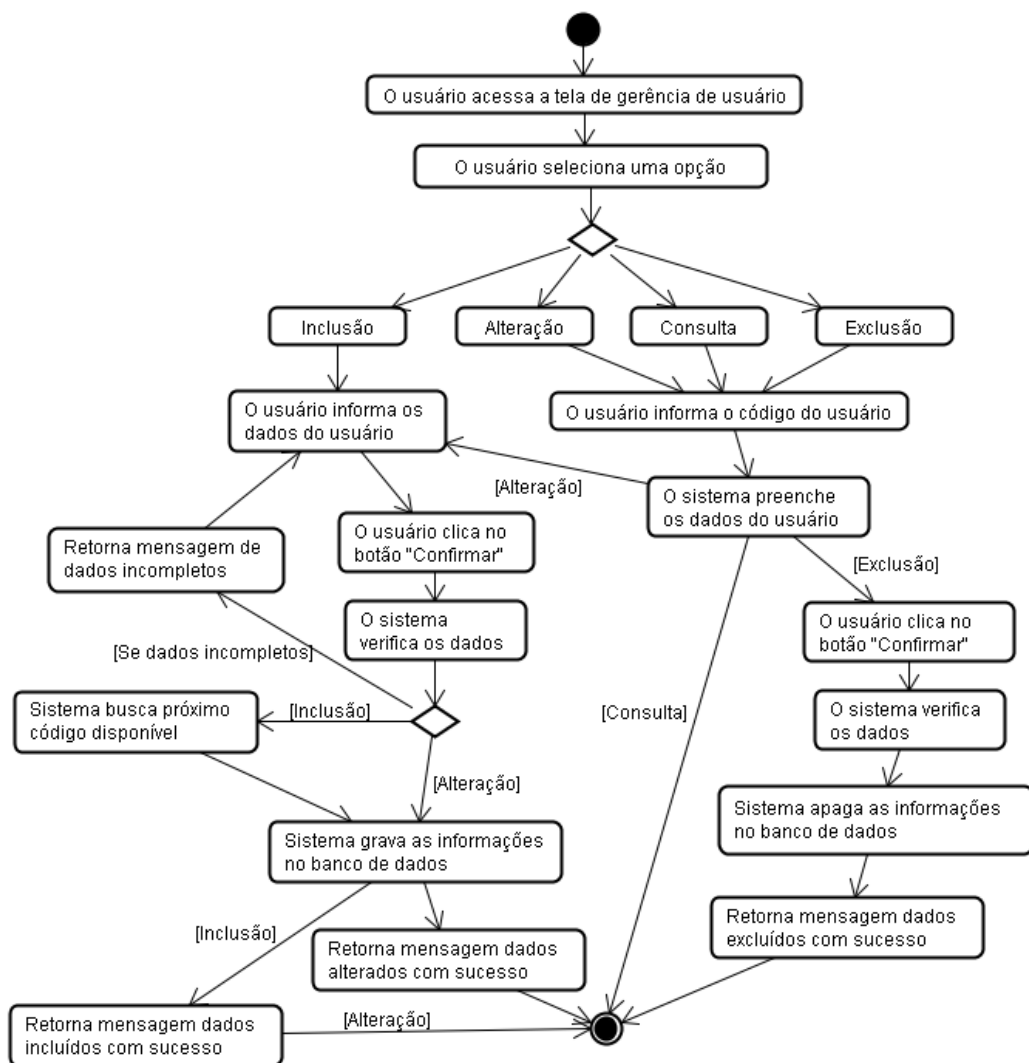


Figura Anexo 1 – Diagrama de Atividade Gerência de usuário.
Fonte: autor (2010).

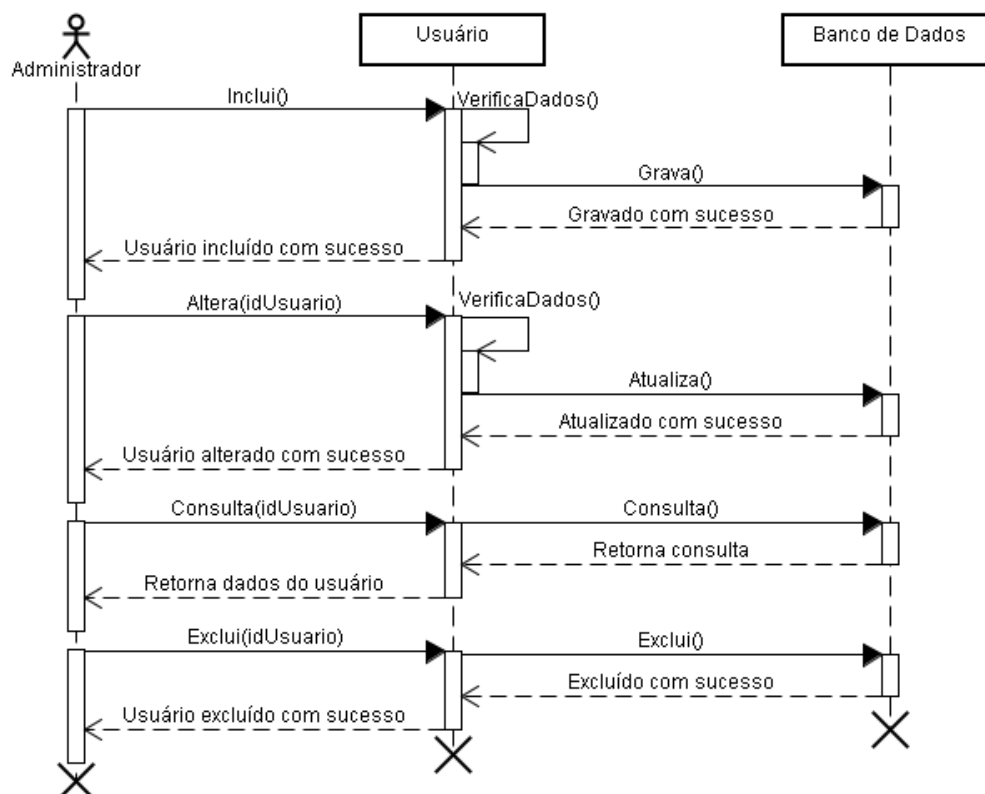


Figura Anexo 2 – Diagrama de Sequência Gerência de Usuário.

Fonte: autor (2010).

UC002 – Gerência de rede de loja

Breve descritivo: este caso de uso descreve o processo de gerenciamento de rede de lojas.

Requisitos funcionais:	REQF2 – [Cadastrar rede de lojas] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções]
Atores:	Administrador.
Pré-condições:	Ser usuário com permissão de administrador.
Fluxo principal:	O usuário administrador acessa a tela de gerenciamento de rede de lojas. Os subfluxos são apresentados pelo sistema: Inclusão, Alteração, Consulta e Exclusão.
Subfluxos:	1. Inclusão: 1.1 O usuário seleciona a opção de inclusão; 1.2 O usuário informa dos dados de uma nova rede de lojas: descrição e situação do cadastro;

	<p>1.3 O usuário clica no botão/link “Salvar”;</p> <p>1.4 O sistema verifica os dados;</p> <p>1.5 O sistema preenche o código com o próximo disponível;</p> <p>1.6 O sistema grava as informações no banco de dados;</p> <p>1.7 O sistema retorna a mensagem “Dados incluídos com sucesso”.</p> <p>2. Alteração:</p> <p>2.1 O usuário seleciona a opção de alteração;</p> <p>2.2 O usuário informa o código da rede de lojas que deseja alterar;</p> <p>2.3 O sistema preenche os dados da rede de lojas;</p> <p>2.4 O usuário altera os dados desejados;</p> <p>2.5 O usuário clica no botão/link “Salvar”;</p> <p>2.6 O sistema verifica os dados;</p> <p>2.7 O sistema grava as informações no banco de dados;</p> <p>2.8 O sistema retorna a mensagem “Dados alterados com sucesso”.</p> <p>3. Consulta:</p> <p>3.1 O usuário seleciona a opção de consulta;</p> <p>3.2 O usuário informa o código da rede de lojas que deseja consultar;</p> <p>3.3 O sistema preenche os dados da rede de lojas.</p> <p>4. Exclusão:</p> <p>4.1 O usuário seleciona a opção de exclusão;</p> <p>4.2 O usuário informa o código da rede de lojas que deseja excluir;</p> <p>4.3 O sistema preenche os dados da rede de lojas;</p> <p>4.4 O usuário clica no botão/link “Excluir”;</p> <p>4.5 O sistema verifica os dados;</p> <p>4.6 O sistema exclui as informações no banco de dados;</p> <p>4.7 O sistema retorna a mensagem “Dados excluídos com sucesso”.</p>
Tratamento de exceções:	<p>1.4.1 O sistema deve verificar se os campos obrigatórios foram preenchidos e retornar a mensagem “Preencha todos os campos obrigatórios”, caso algum não tenha sido preenchido;</p> <p>2.2.1 Se o código informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p> <p>2.6.1 Recebe o mesmo tratamento de exceção do item 1.4.1;</p> <p>3.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.5.1 Recebe mesmo tratamento de exceção do item 1.4.1.</p>
Pós-condições:	<p>1. Uma nova rede de lojas está cadastrada;</p> <p>2. Uma rede de lojas está alterada;</p> <p>3. Uma rede de lojas está consultada;</p>

4. Uma rede de lojas está excluída.

Quadro Anexo 2 – Caso de Uso Gerência de rede de lojas.
Fonte: autor (2010).

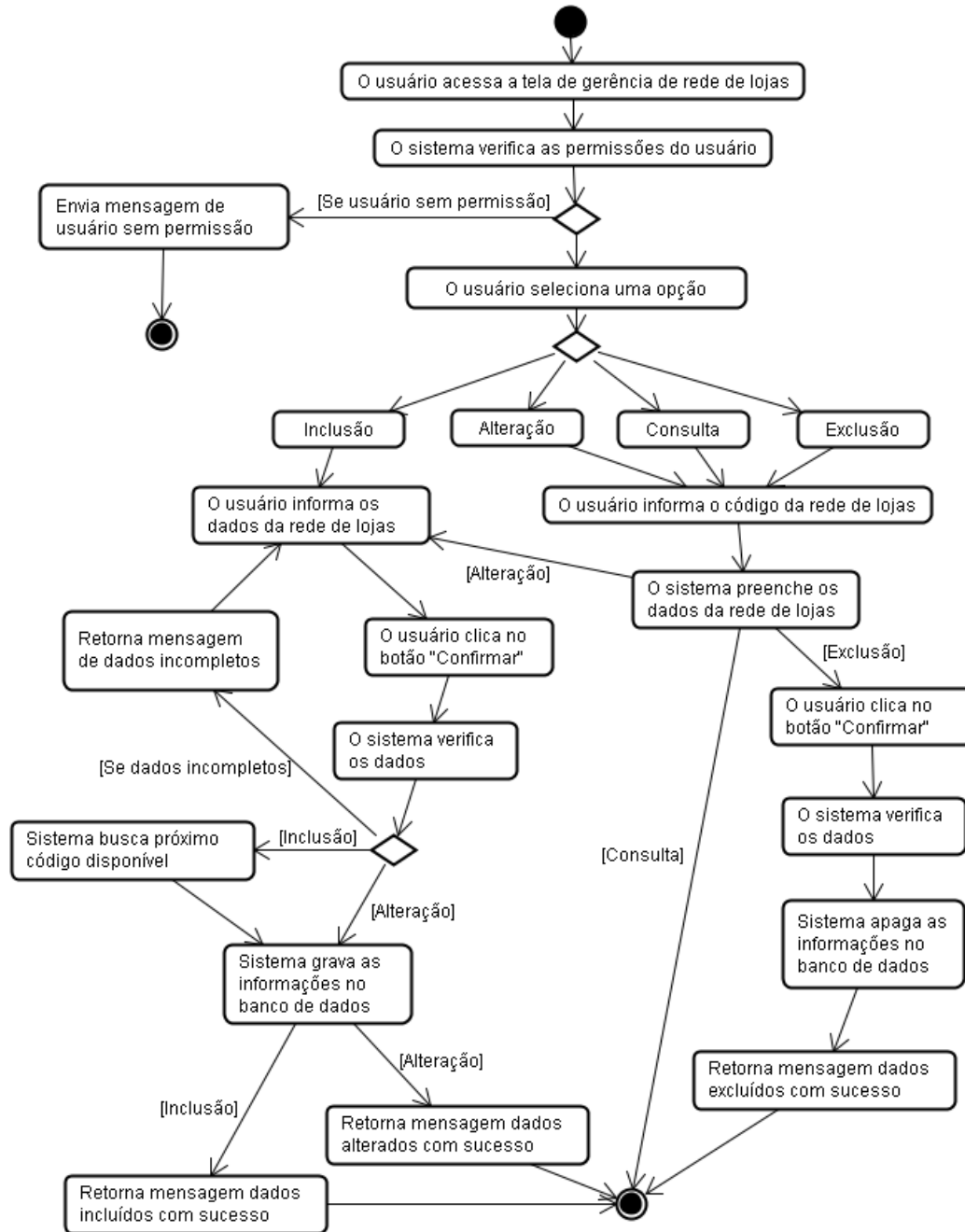


Figura Anexo 3 – Diagrama de Atividade Gerência de rede de lojas.
Fonte: autor (2010).

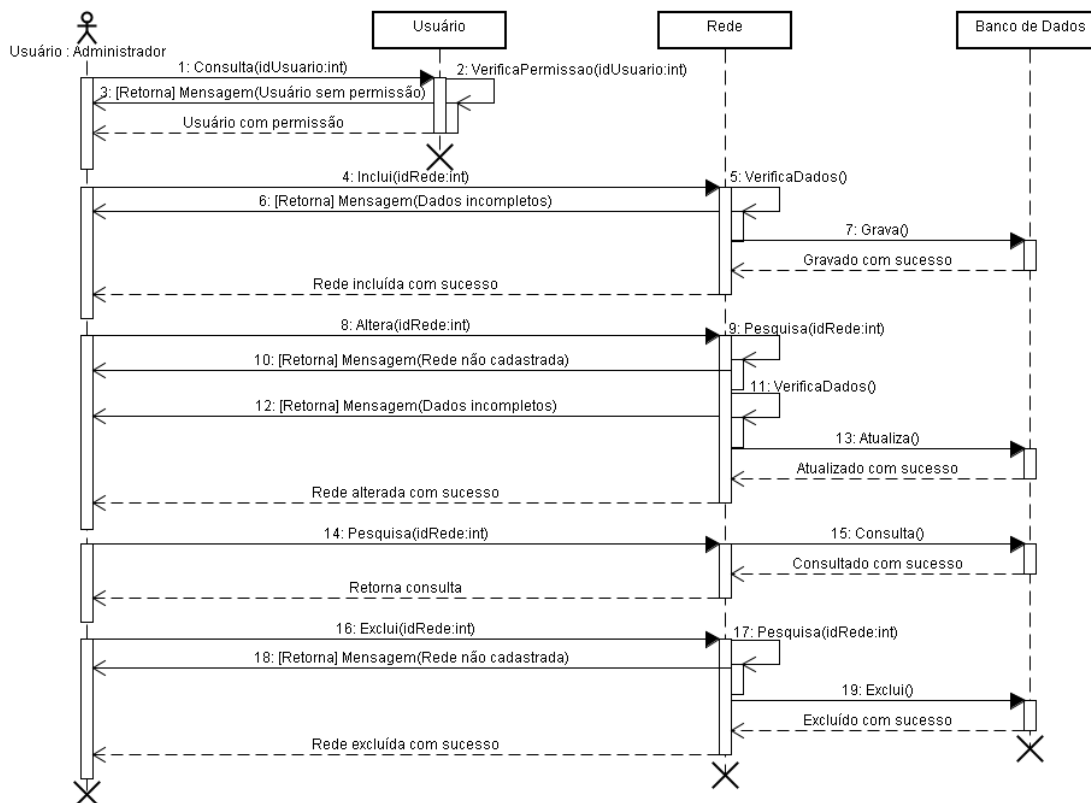


Figura Anexo 4 – Diagrama de Sequência Gerência de rede de lojas.

Fonte: autor (2010).

UC003 – Gerência de loja

abreviado descritivo: este caso de uso descreve o processo de gerenciamento de loja.

Requisitos funcionais:	REQF2 – [Cadastrar rede de lojas] REQF3 – [Cadastrar loja] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções]
Atores:	Administrador.
Pré-condições:	Ser usuário com permissão de administrador.
Fluxo principal:	O usuário administrador acessa a tela de gerenciamento de loja. Os subfluxos são apresentados pelo sistema: Inclusão, Alteração, Consulta e Exclusão.
Subfluxos:	1. Inclusão: 1.1 O usuário seleciona a opção de inclusão; 1.2 O usuário informa dos dados de uma nova loja: rede de lojas, nome, CNPJ, inscrição estadual, logradouro, número, complemento, bairro, município e

	<p>situação do cadastro;</p> <p>1.3 O usuário clica no botão/link “Salvar”;</p> <p>1.4 O sistema verifica os dados;</p> <p>1.5 O sistema preenche o código com o próximo disponível;</p> <p>1.6 O sistema grava as informações no banco de dados;</p> <p>1.7 O sistema retorna a mensagem “Dados incluídos com sucesso”.</p> <p>2. Alteração:</p> <p>2.1 O usuário seleciona a opção de alteração;</p> <p>2.2 O usuário informa o código da loja que deseja alterar;</p> <p>2.3 O sistema preenche os dados da loja;</p> <p>2.4 O usuário altera os dados desejados;</p> <p>2.5 O usuário clica no botão/link “Salvar”;</p> <p>2.6 O sistema verifica os dados;</p> <p>2.7 O sistema grava as informações no banco de dados;</p> <p>2.8 O sistema retorna a mensagem “Dados alterados com sucesso”.</p> <p>3. Consulta:</p> <p>3.1 O usuário seleciona a opção de consulta;</p> <p>3.2 O usuário informa o código da loja que deseja consultar;</p> <p>3.3 O sistema preenche os dados da loja.</p> <p>4. Exclusão:</p> <p>4.1 O usuário seleciona a opção de exclusão;</p> <p>4.2 O usuário informa o código da loja que deseja excluir;</p> <p>4.3 O sistema preenche os dados da loja;</p> <p>4.4 O usuário clica no botão/link “Excluir”;</p> <p>4.5 O sistema verifica os dados;</p> <p>4.6 O sistema exclui as informações no banco de dados;</p> <p>4.7 O sistema retorna a mensagem “Dados excluídos com sucesso”.</p>
Tratamento de exceções:	<p>1.4.1 O sistema deve verificar se os campos obrigatórios foram preenchidos e retornar a mensagem “Preencha todos os campos obrigatórios” caso algum não tenha sido preenchido;</p> <p>2.2.1 Se o código informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p> <p>2.6.1 Recebe o mesmo tratamento de exceção do item 1.4.1;</p> <p>3.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.5.1 Recebe mesmo tratamento de exceção do item 1.4.1.</p>
Pós-condições:	<p>1. Uma nova loja está cadastrada;</p> <p>2. Uma loja está alterada;</p>

	3. Uma loja está consultada;
	4. Uma loja está excluída.

Quadro Anexo 3 – Caso de Uso Gerência de loja.
Fonte: autor (2010).

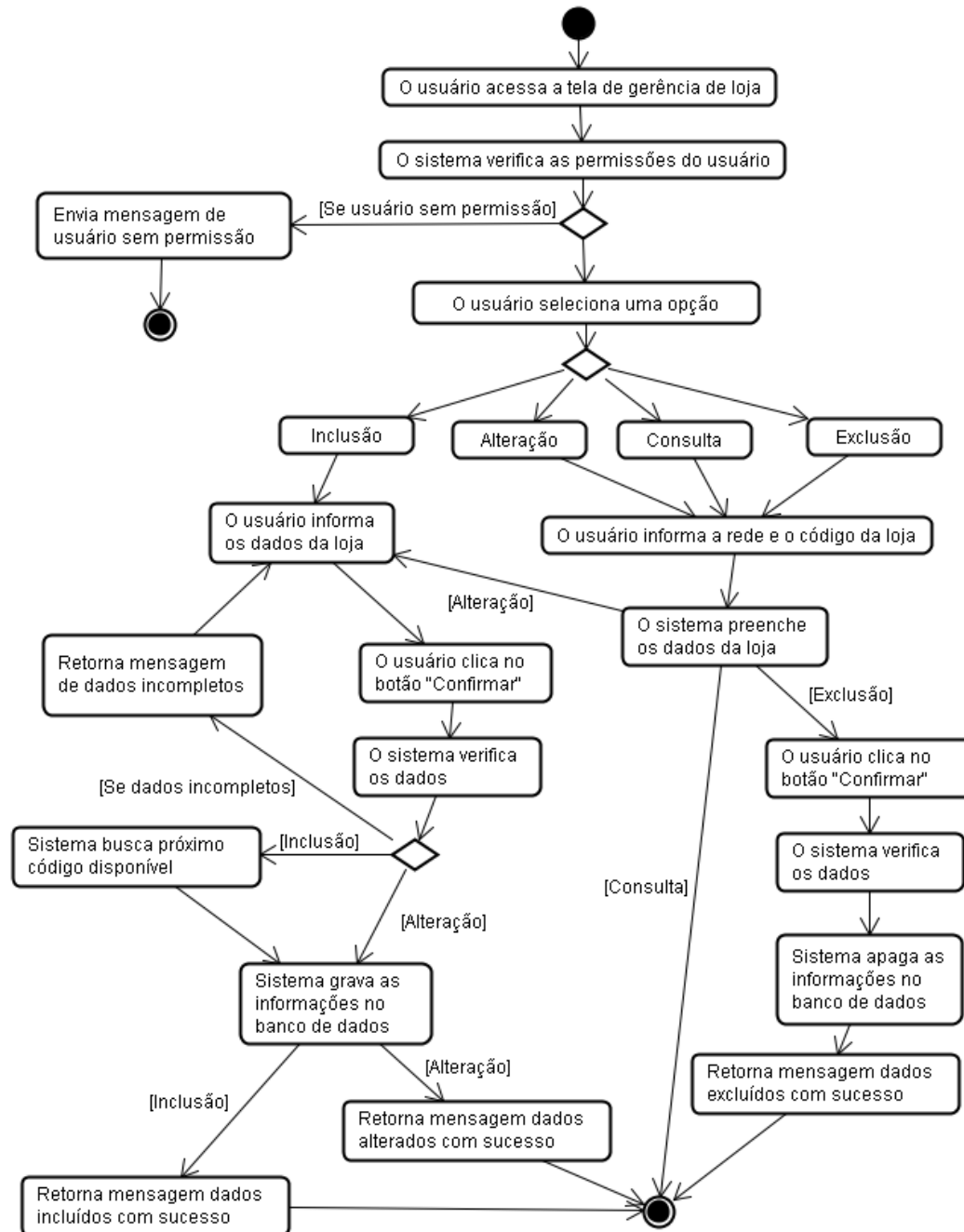


Figura Anexo 5 – Diagrama de Atividade Gerência de loja.
Fonte: autor (2010).

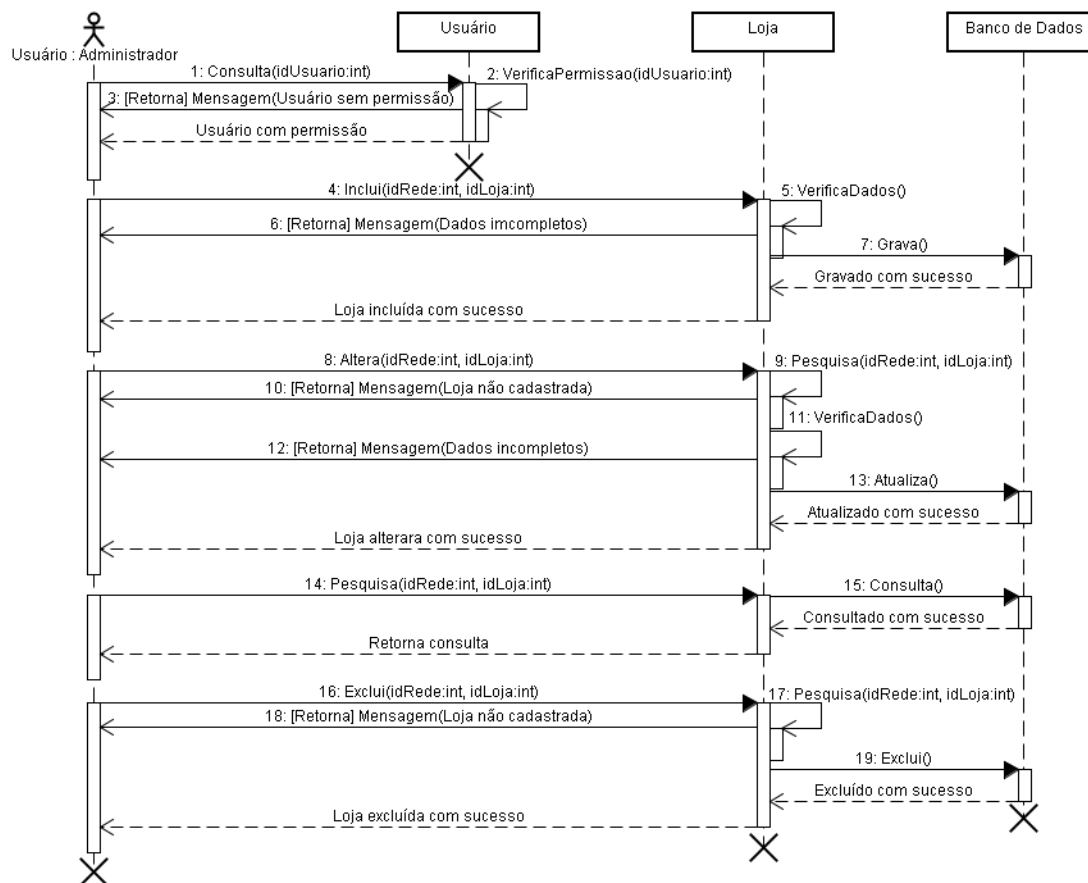


Figura Anexo 6 – Diagrama de Sequência Gerência de loja.

Fonte: autor (2010).

UC004 – Gerência de terminal

Breve descritivo: este caso de uso descreve o processo de gerenciamento de terminal.

Requisitos funcionais:	REQF2 – [Cadastrar rede de lojas] REQF3 – [Cadastrar loja] REQF4 – [Cadastrar terminal] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções]
Atores:	Administrador.
Pré-condições:	Ser usuário com permissão de administrador.
Fluxo principal:	O usuário administrador acessa a tela de gerenciamento de terminal. Os subfluxos são apresentados pelo sistema: Inclusão, Alteração, Consulta e Exclusão.

Subfluxos:	<p>1. Inclusão:</p> <p>1.1 O usuário seleciona a opção de inclusão;</p> <p>1.2 O usuário informa dos dados de um novo terminal: rede de lojas, loja, descrição, número de série e situação do cadastro;</p> <p>1.3 O usuário clica no botão/link “Salvar”;</p> <p>1.4 O sistema verifica os dados;</p> <p>1.5 O sistema preenche o código com o próximo disponível;</p> <p>1.6 O sistema grava as informações no banco de dados;</p> <p>1.7 O sistema retorna a mensagem “Dados incluídos com sucesso”.</p> <p>2. Alteração:</p> <p>2.1 O usuário seleciona a opção de alteração;</p> <p>2.2 O usuário informa o código do terminal que deseja alterar;</p> <p>2.3 O sistema preenche os dados do terminal;</p> <p>2.4 O usuário altera os dados desejados;</p> <p>2.5 O usuário clica no botão/link “Salvar”;</p> <p>2.6 O sistema verifica os dados;</p> <p>2.7 O sistema grava as informações no banco de dados;</p> <p>2.8 O sistema retorna a mensagem “Dados alterados com sucesso”.</p> <p>3. Consulta:</p> <p>3.1 O usuário seleciona a opção de consulta;</p> <p>3.2 O usuário informa o código do terminal que deseja consultar;</p> <p>3.3 O sistema preenche os dados do terminal.</p> <p>4. Exclusão:</p> <p>4.1 O usuário seleciona a opção de exclusão;</p> <p>4.2 O usuário informa o código do terminal que deseja excluir;</p> <p>4.3 O sistema preenche os dados do terminal;</p> <p>4.4 O usuário clica no botão/link “Excluir”;</p> <p>4.5 O sistema verifica os dados;</p> <p>4.6 O sistema exclui as informações no banco de dados;</p> <p>4.7 O sistema retorna a mensagem “Dados excluídos com sucesso”.</p>
Tratamento de exceções:	<p>1.4.1 O sistema deve verificar se os campos obrigatórios foram preenchidos e retornar a mensagem “Preencha todos os campos obrigatórios”, caso algum não tenha sido preenchido;</p> <p>2.2.1 Se o código informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p> <p>2.6.1 Recebe o mesmo tratamento de exceção do item 1.4.1;</p> <p>3.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p>

	4.5.1 Recebe mesmo tratamento de exceção do item 1.4.1.
Pós-condições:	<ol style="list-style-type: none"> 1. Um novo terminal está cadastrado; 2. Um terminal está alterado; 3. Um terminal está consultado; 4. Um terminal está excluído.

Quadro Anexo 4 – Caso de Uso Gerência de terminal.

Fonte: autor (2010).

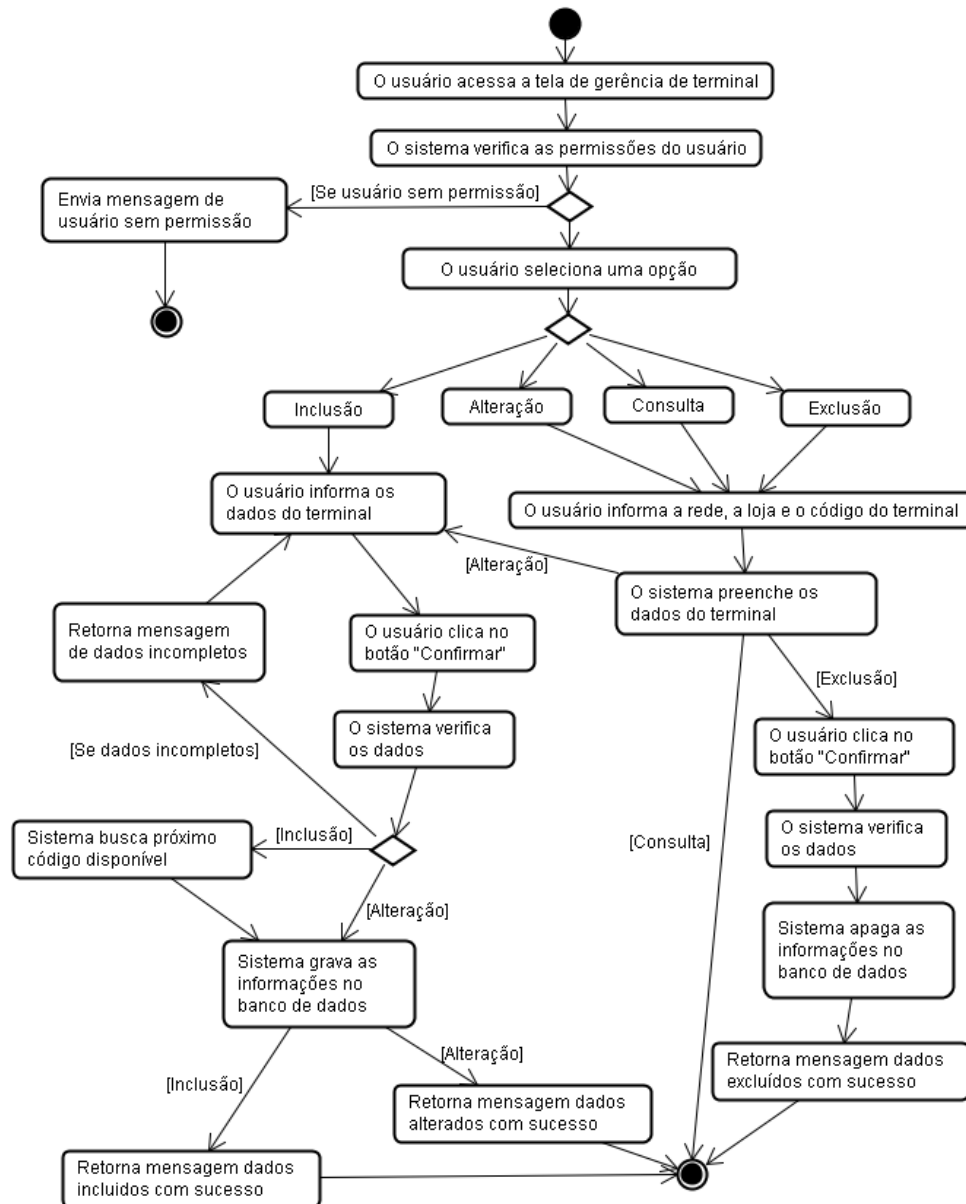


Figura Anexo 7 – Diagrama de Atividade Gerência de terminal.

Fonte: autor (2010).

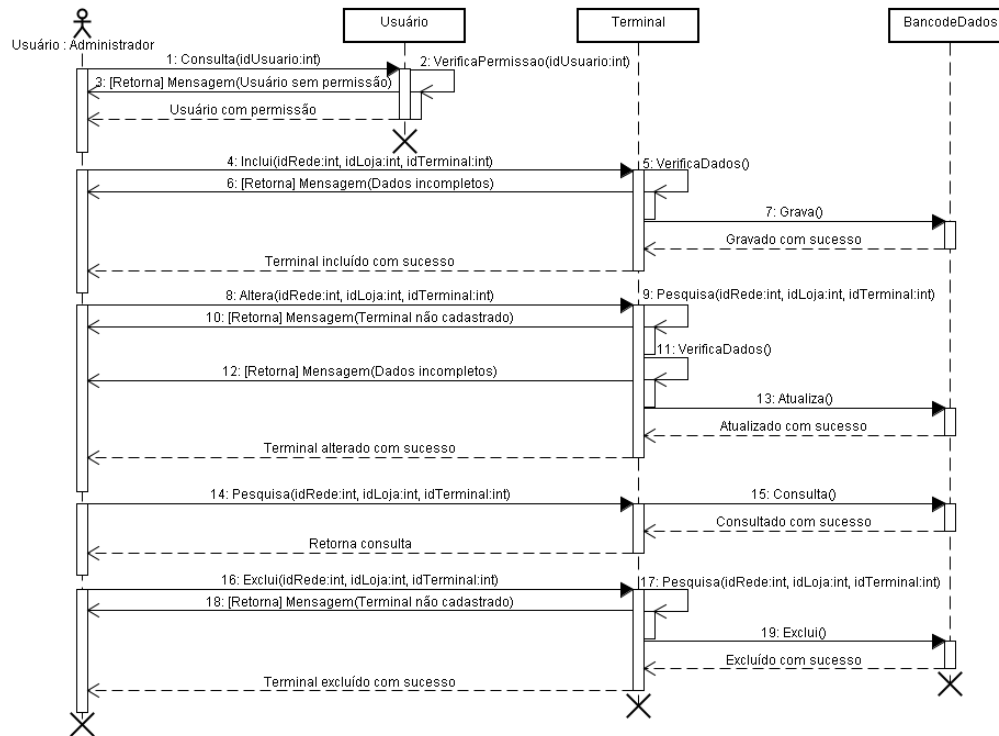


Figura Anexo 8 – Diagrama de Sequência Gerência de terminal.

Fonte: autor (2010).

UC005 – Gerência de divisão

Breve descritivo: este caso de uso descreve o processo de gerenciamento de divisão.

Requisitos funcionais:	REQF5 – [Cadastrar divisão] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções]
Atores:	Administrador.
Pré-condições:	Ser usuário com permissão de administrador.
Fluxo principal:	O usuário administrador acessa a tela de gerenciamento de terminal. Os subfluxos são apresentados pelo sistema: Inclusão, Alteração, Consulta e Exclusão.
Subfluxos:	1. Inclusão: 1.1 O usuário seleciona a opção de inclusão; 1.2 O usuário informa dos dados de uma nova divisão: descrição; 1.3 O usuário clica no botão/link “Salvar”; 1.4 O sistema verifica os dados; 1.5 O sistema preenche o código com o próximo disponível; 1.6 O sistema grava as informações no banco de dados;

	<p>1.7 O sistema retorna a mensagem “Dados incluídos com sucesso”.</p> <p>2. Alteração:</p> <p>2.1 O usuário seleciona a opção de alteração;</p> <p>2.2 O usuário informa o código da divisão que deseja alterar;</p> <p>2.3 O sistema preenche os dados da divisão;</p> <p>2.4 O usuário altera os dados desejados;</p> <p>2.5 O usuário clica no botão/link “Salvar”;</p> <p>2.6 O sistema verifica os dados;</p> <p>2.7 O sistema grava as informações no banco de dados;</p> <p>2.8 O sistema retorna a mensagem “Dados alterados com sucesso”.</p> <p>3. Consulta:</p> <p>3.1 O usuário seleciona a opção de consulta;</p> <p>3.2 O usuário informa o código da divisão que deseja consultar;</p> <p>3.3 O sistema preenche os dados da divisão.</p> <p>4. Exclusão:</p> <p>4.1 O usuário seleciona a opção de exclusão;</p> <p>4.2 O usuário informa o código da divisão que deseja excluir;</p> <p>4.3 O sistema preenche os dados da divisão;</p> <p>4.4 O usuário clica no botão/link “Excluir”;</p> <p>4.5 O sistema verifica os dados;</p> <p>4.6 O sistema exclui as informações no banco de dados;</p> <p>4.7 O sistema retorna a mensagem “Dados excluídos com sucesso”.</p>
Tratamento de exceções:	<p>1.4.1 O sistema deve verificar se os campos obrigatórios foram preenchidos e retornar a mensagem “Preencha todos os campos obrigatórios” caso algum não tenha sido preenchido;</p> <p>2.2.1 Se o código informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p> <p>2.6.1 Recebe o mesmo tratamento de exceção do item 1.4.1;</p> <p>3.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.5.1 Recebe mesmo tratamento de exceção do item 1.4.1.</p>
Pós-condições:	<p>1. Uma nova divisão está cadastrada;</p> <p>2. Uma divisão está alterada;</p> <p>3. Uma divisão está consultada;</p> <p>4. Uma divisão está excluída.</p>

Quadro Anexo 5 – Caso de Uso Gerência de divisão.

Fonte: autor (2010).

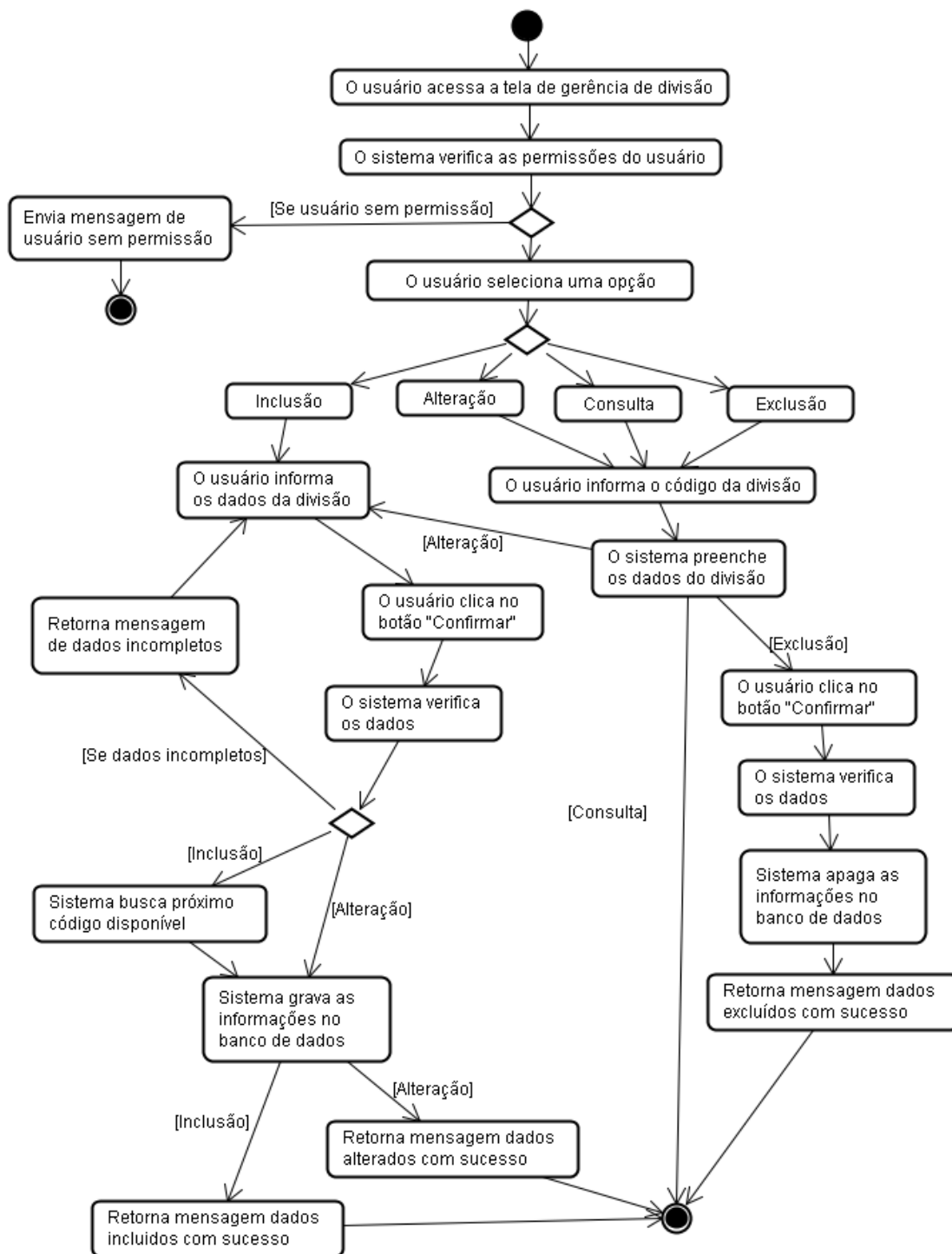


Figura Anexo 9 – Diagrama de Atividade Gerência de divisão.

Fonte: autor (2010).

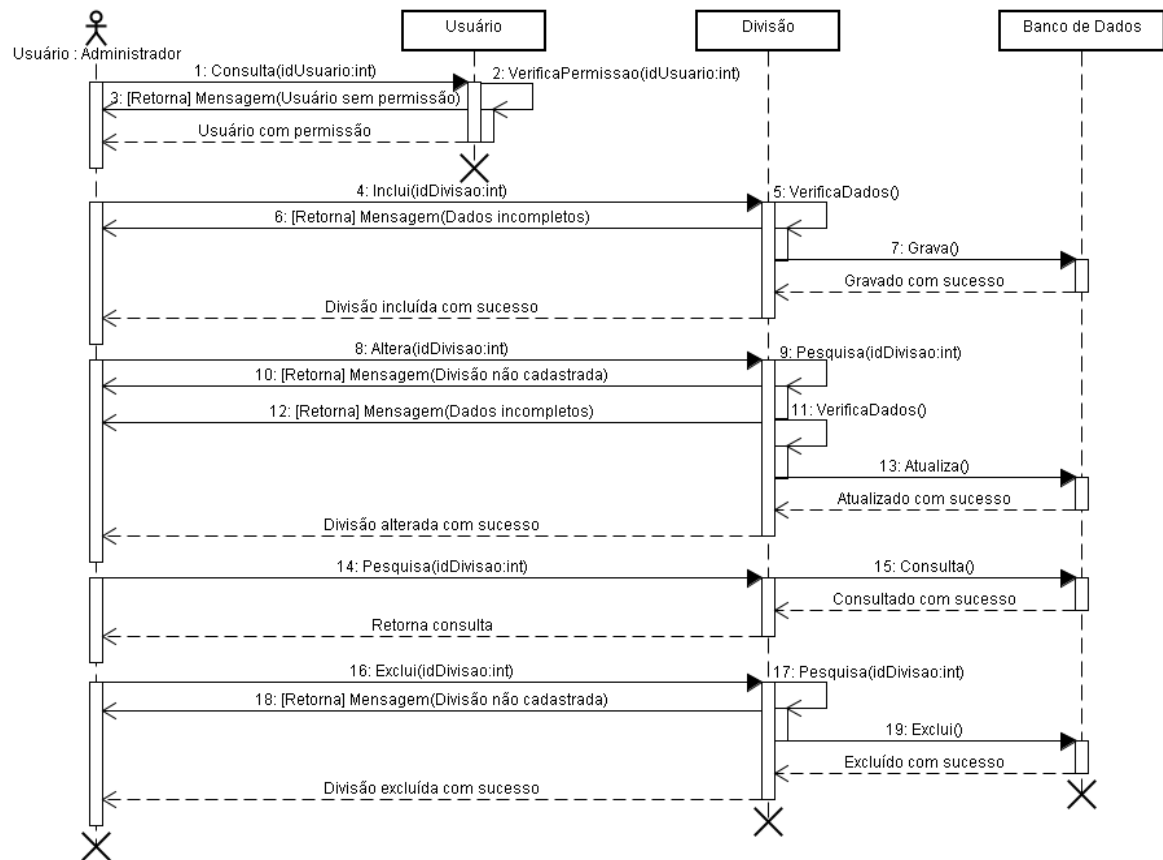


Figura Anexo 10 – Diagrama de Sequência Gerência de divisão.

Fonte: autor (2010).

UC006 – Gerência de artigo

Breve descritivo: este caso de uso descreve o processo de gerenciamento de artigo.

Requisitos funcionais:	REQF6 – [Cadastrar artigo] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções]
Atores:	Administrador.
Pré-condições:	Ser usuário com permissão de administrador.
Fluxo principal:	O usuário administrador acessa a tela de gerenciamento de artigo. Os subfluxos são apresentados pelo sistema: Inclusão, Alteração, Consulta e Exclusão.
Subfluxos:	1. Inclusão: 1.1 O usuário seleciona a opção de inclusão; 1.2 O usuário informa dos dados de um novo artigo: descrição;

	<p>1.3 O usuário clica no botão/link “Salvar”;</p> <p>1.4 O sistema verifica os dados;</p> <p>1.5 O sistema preenche o código com o próximo disponível;</p> <p>1.6 O sistema grava as informações no banco de dados;</p> <p>1.7 O sistema retorna a mensagem “Dados incluídos com sucesso”.</p> <p>2. Alteração:</p> <p>2.1 O usuário seleciona a opção de alteração;</p> <p>2.2 O usuário informa o código do artigo que deseja alterar;</p> <p>2.3 O sistema preenche os dados do artigo;</p> <p>2.4 O usuário altera os dados desejados;</p> <p>2.5 O usuário clica no botão/link “Salvar”;</p> <p>2.6 O sistema verifica os dados;</p> <p>2.7 O sistema grava as informações no banco de dados;</p> <p>2.8 O sistema retorna a mensagem “Dados alterados com sucesso”.</p> <p>3. Consulta:</p> <p>3.1 O usuário seleciona a opção de consulta;</p> <p>3.2 O usuário informa o código do artigo que deseja consultar;</p> <p>3.3 O sistema preenche os dados do artigo.</p> <p>4. Exclusão:</p> <p>4.1 O usuário seleciona a opção de exclusão;</p> <p>4.2 O usuário informa o código do artigo que deseja excluir;</p> <p>4.3 O sistema preenche os dados do artigo;</p> <p>4.4 O usuário clica no botão/link “Excluir”;</p> <p>4.5 O sistema verifica os dados;</p> <p>4.6 O sistema exclui as informações no banco de dados;</p> <p>4.7 O sistema retorna a mensagem “Dados excluídos com sucesso”.</p>
Tratamento de exceções:	<p>1.4.1 O sistema deve verificar se os campos obrigatórios foram preenchidos e retornar a mensagem “Preencha todos os campos obrigatórios”, caso algum não tenha sido preenchido;</p> <p>2.2.1 Se o código informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p> <p>2.6.1 Recebe o mesmo tratamento de exceção do item 1.4.1;</p> <p>3.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.5.1 Recebe mesmo tratamento de exceção do item 1.4.1.</p>
Pós-condições:	<p>1. Um novo artigo está cadastrado;</p> <p>2. Um artigo está alterado;</p> <p>3. Um artigo está consultado;</p>

	4. Um artigo está excluído.
--	-----------------------------

Quadro Anexo 6 – Caso de Uso Gerência de artigo.

Fonte: autor (2010).

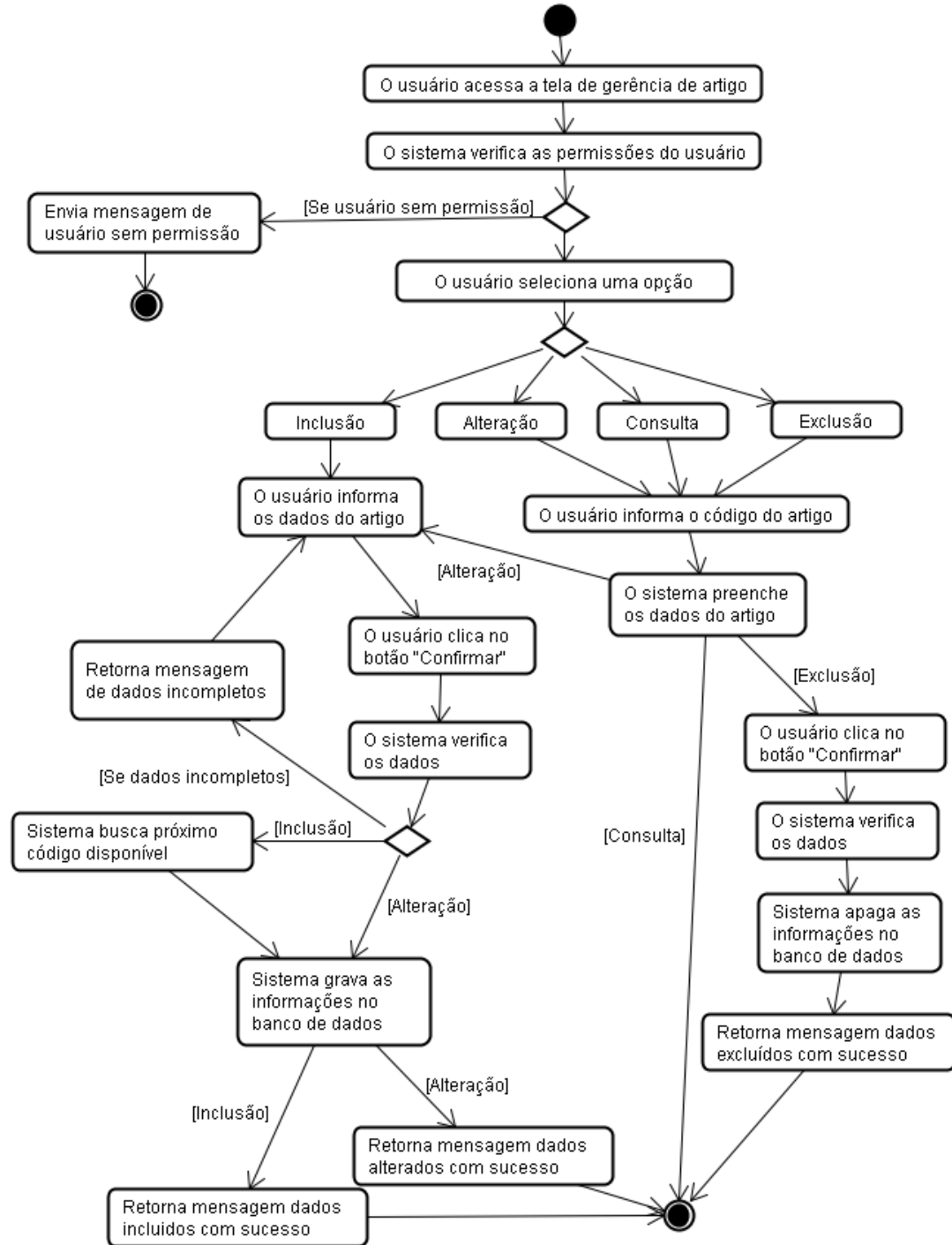


Figura Anexo 11 – Diagrama de Atividade Gerência de artigo.

Fonte: autor (2010).

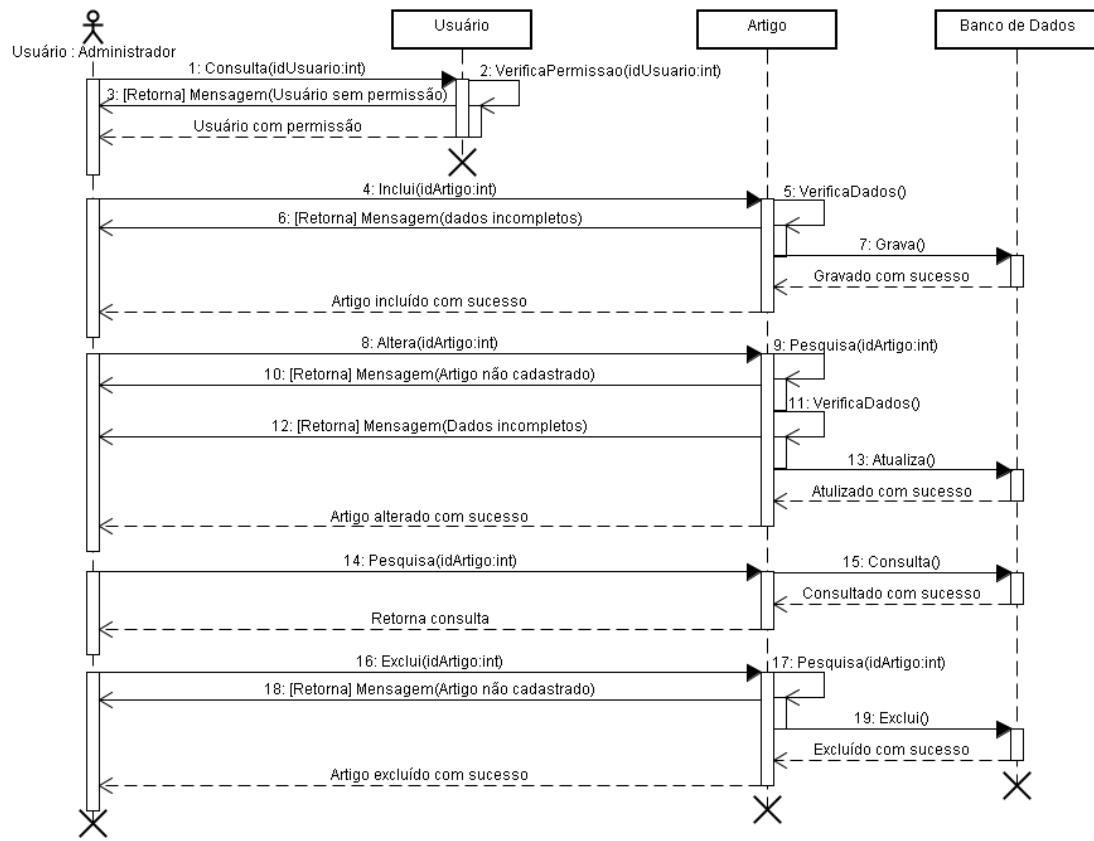


Figura Anexo 12 – Diagrama de Sequência Gerência de divisão.
Fonte: autor (2010).

UC007 – Gerência de marca

Breve descritivo: este caso de uso descreve o processo de gerenciamento de marca.

Requisitos funcionais:	REQF7 – [Cadastrar marca] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções]
Atores:	Administrador.
Pré-condições:	Ser usuário com permissão de administrador.
Fluxo principal:	O usuário administrador acessa a tela de gerenciamento de marca. Os subfluxos são apresentados pelo sistema: Inclusão, Alteração, Consulta e Exclusão.
Subfluxos:	1. Inclusão: 1.1 O usuário seleciona a opção de inclusão; 1.2 O usuário informa dos dados de uma nova marca: descrição;

	<p>1.3 O usuário clica no botão/link “Salvar”;</p> <p>1.4 O sistema verifica os dados;</p> <p>1.5 O sistema preenche o código com o próximo disponível;</p> <p>1.6 O sistema grava as informações no banco de dados;</p> <p>1.7 O sistema retorna a mensagem “Dados incluídos com sucesso”.</p> <p>2. Alteração:</p> <p>2.1 O usuário seleciona a opção de alteração;</p> <p>2.2 O usuário informa o código da marca que deseja alterar;</p> <p>2.3 O sistema preenche os dados da marca;</p> <p>2.4 O usuário altera os dados desejados;</p> <p>2.5 O usuário clica no botão/link “Salvar”;</p> <p>2.6 O sistema verifica os dados;</p> <p>2.7 O sistema grava as informações no banco de dados;</p> <p>2.8 O sistema retorna a mensagem “Dados alterados com sucesso”.</p> <p>3. Consulta:</p> <p>3.1 O usuário seleciona a opção de consulta;</p> <p>3.2 O usuário informa o código da marca que deseja consultar;</p> <p>3.3 O sistema preenche os dados da marca.</p> <p>4. Exclusão:</p> <p>4.1 O usuário seleciona a opção de exclusão;</p> <p>4.2 O usuário informa o código da marca que deseja excluir;</p> <p>4.3 O sistema preenche os dados da marca;</p> <p>4.4 O usuário clica no botão/link “Excluir”;</p> <p>4.5 O sistema verifica os dados;</p> <p>4.6 O sistema exclui as informações no banco de dados;</p> <p>4.7 O sistema retorna a mensagem “Dados excluídos com sucesso”.</p>
Tratamento de exceções:	<p>1.4.1 O sistema deve verificar se os campos obrigatórios foram preenchidos e retornar a mensagem “Preencha todos os campos obrigatórios”, caso algum não tenha sido preenchido;</p> <p>2.2.1 Se o código informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p> <p>2.6.1 Recebe o mesmo tratamento de exceção do item 1.4.1;</p> <p>3.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.5.1 Recebe mesmo tratamento de exceção do item 1.4.1.</p>
Pós-condições:	<p>1. Uma nova marca está cadastrada;</p> <p>2. Uma marca está alterada;</p> <p>3. Uma marca está consultada;</p>

	4. Uma marca está excluída.
--	-----------------------------

Quadro Anexo 7 – Caso de Uso Gerência de marca.

Fonte: autor (2010).

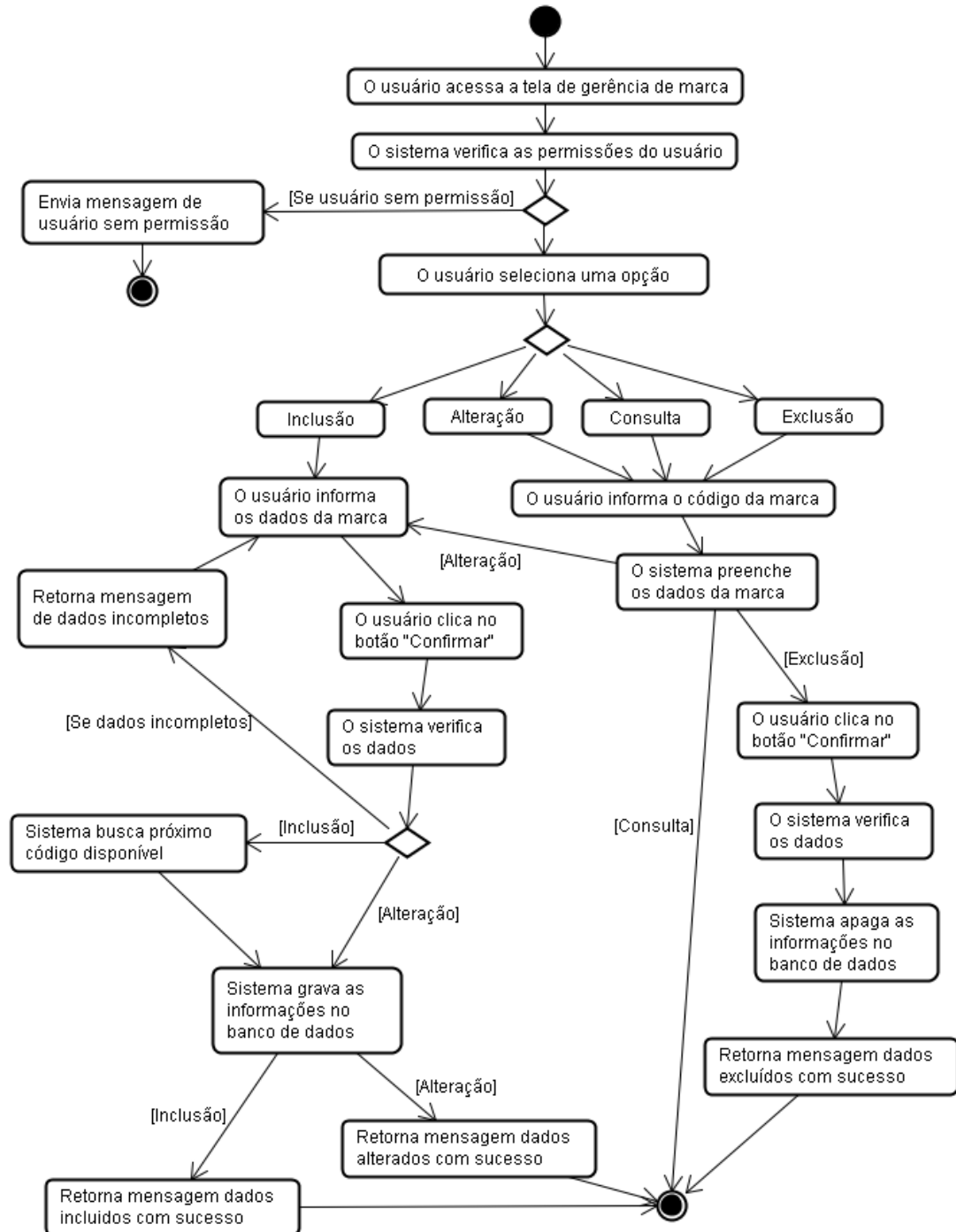


Figura Anexo 13 – Diagrama de Atividade Gerência de marca.

Fonte: autor (2010).

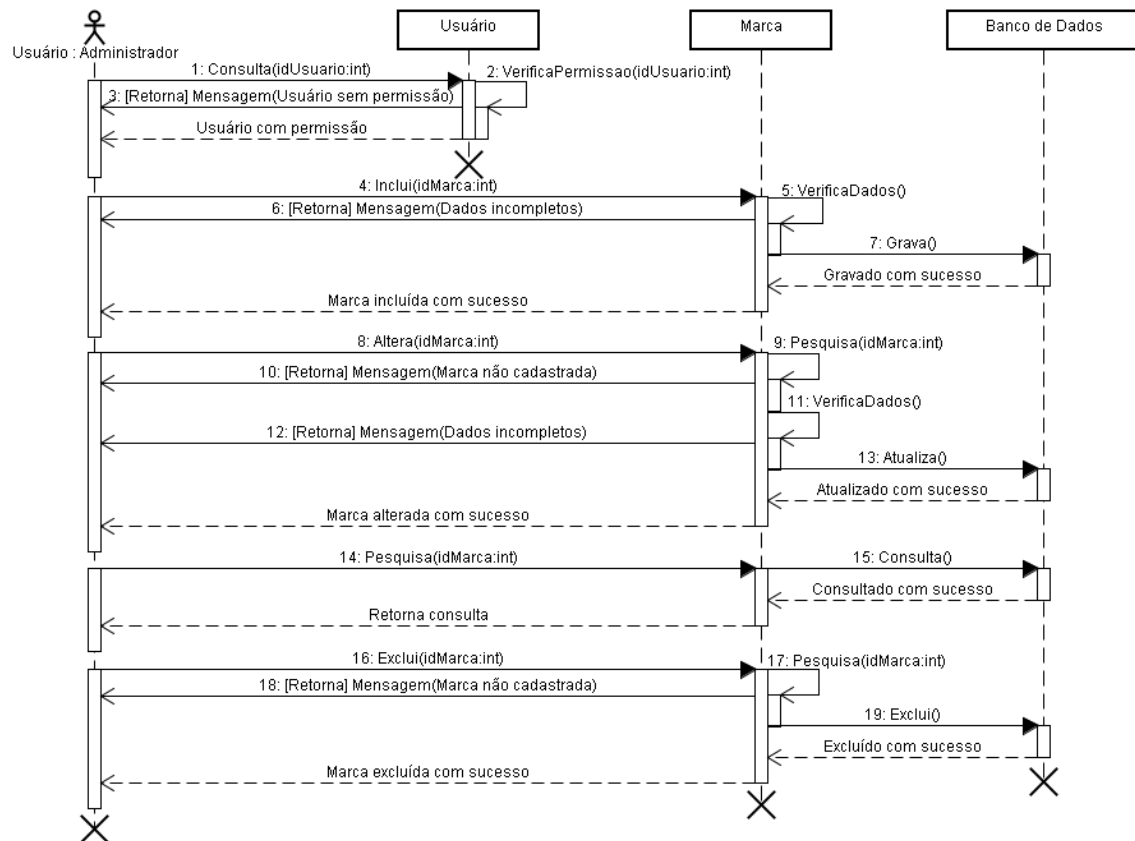


Figura Anexo 14 – Diagrama de Sequência Gerência de marca.
 Fonte: autor (2010).

UC008 – Gerência de modelo

Breve descritivo: este caso de uso descreve o processo de gerenciamento de modelo.

Requisitos funcionais:	REQF8 – [Cadastrar modelo] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções]
Atores:	Administrador.
Pré-condições:	Ser usuário com permissão de administrador.
Fluxo principal:	O usuário administrador acessa a tela de gerenciamento de modelo. Os subfluxos são apresentados pelo sistema: Inclusão, Alteração, Consulta e Exclusão.
Subfluxos:	1. Inclusão: 1.1 O usuário seleciona a opção de inclusão; 1.2 O usuário informa dos dados de um novo modelo: descrição;

	<p>1.3 O usuário clica no botão/link “Salvar”;</p> <p>1.4 O sistema verifica os dados;</p> <p>1.5 O sistema preenche o código com o próximo disponível;</p> <p>1.6 O sistema grava as informações no banco de dados;</p> <p>1.7 O sistema retorna a mensagem “Dados incluídos com sucesso”.</p> <p>2. Alteração:</p> <p>2.1 O usuário seleciona a opção de alteração;</p> <p>2.2 O usuário informa o código do modelo que deseja alterar;</p> <p>2.3 O sistema preenche os dados do modelo;</p> <p>2.4 O usuário altera os dados desejados;</p> <p>2.5 O usuário clica no botão/link “Salvar”;</p> <p>2.6 O sistema verifica os dados;</p> <p>2.7 O sistema grava as informações no banco de dados;</p> <p>2.8 O sistema retorna a mensagem “Dados alterados com sucesso”.</p> <p>3. Consulta:</p> <p>3.1 O usuário seleciona a opção de consulta;</p> <p>3.2 O usuário informa o código do modelo que deseja consultar;</p> <p>3.3 O sistema preenche os dados do modelo.</p> <p>4. Exclusão:</p> <p>4.1 O usuário seleciona a opção de exclusão;</p> <p>4.2 O usuário informa o código do modelo que deseja excluir;</p> <p>4.3 O sistema preenche os dados do modelo;</p> <p>4.4 O usuário clica no botão/link “Excluir”;</p> <p>4.5 O sistema verifica os dados;</p> <p>4.6 O sistema exclui as informações no banco de dados;</p> <p>4.7 O sistema retorna a mensagem “Dados excluídos com sucesso”.</p>
Tratamento de exceções:	<p>1.4.1 O sistema deve verificar se os campos obrigatórios foram preenchidos e retornar a mensagem “Preencha todos os campos obrigatórios” caso algum não tenha sido preenchido;</p> <p>2.2.1 Se o código informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p> <p>2.6.1 Recebe o mesmo tratamento de exceção do item 1.4.1;</p> <p>3.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.5.1 Recebe mesmo tratamento de exceção do item 1.4.1.</p>
Pós-condições:	<p>1. Um novo modelo está cadastrado;</p> <p>2. Um modelo está alterado;</p> <p>3. Um modelo está consultado;</p>

4. Um modelo está excluído.

Quadro Anexo 8 – Caso de Uso Gerência de modelo.
Fonte: autor (2010).

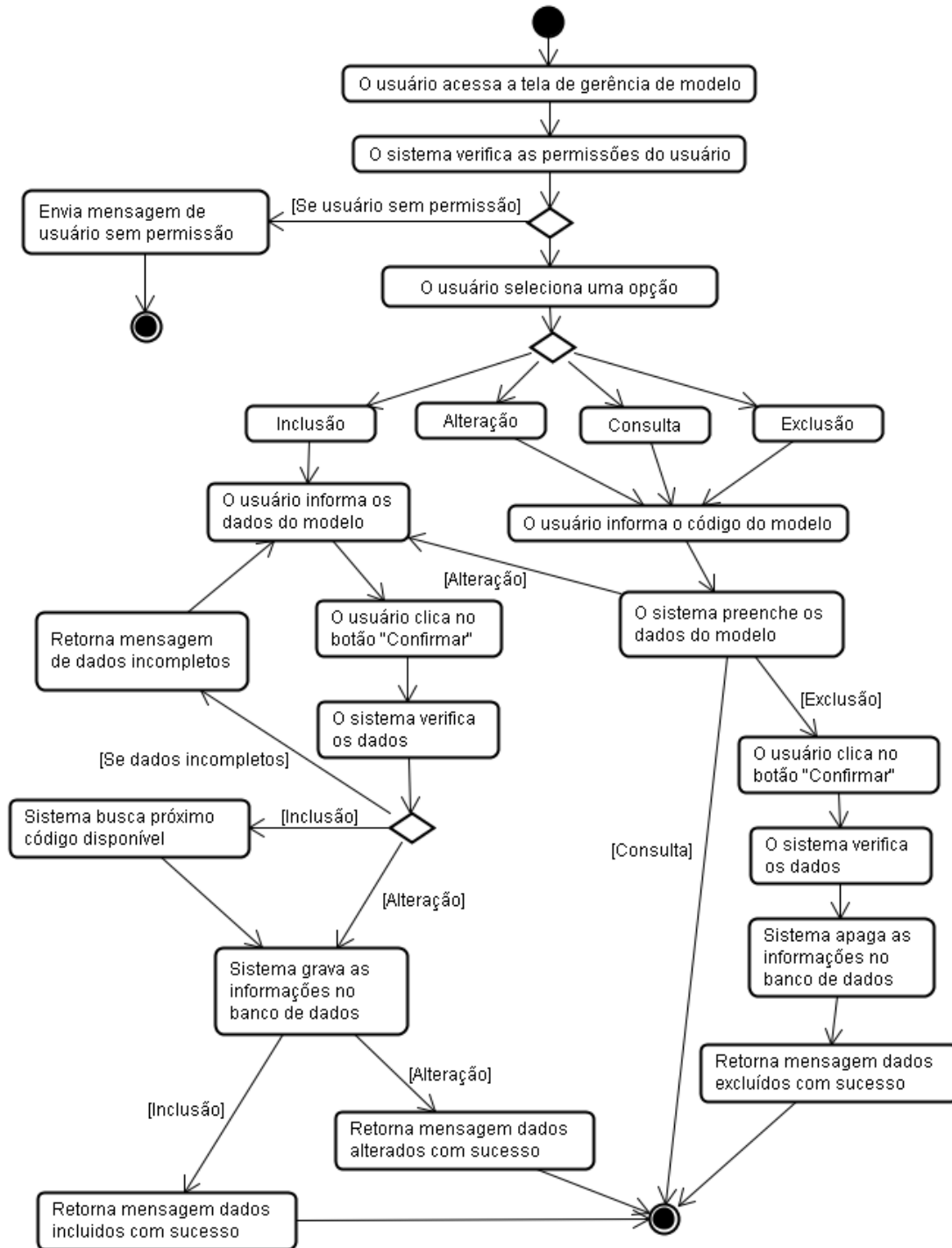


Figura Anexo 15 – Diagrama de Atividade Gerência de modelo.
Fonte: autor (2010).

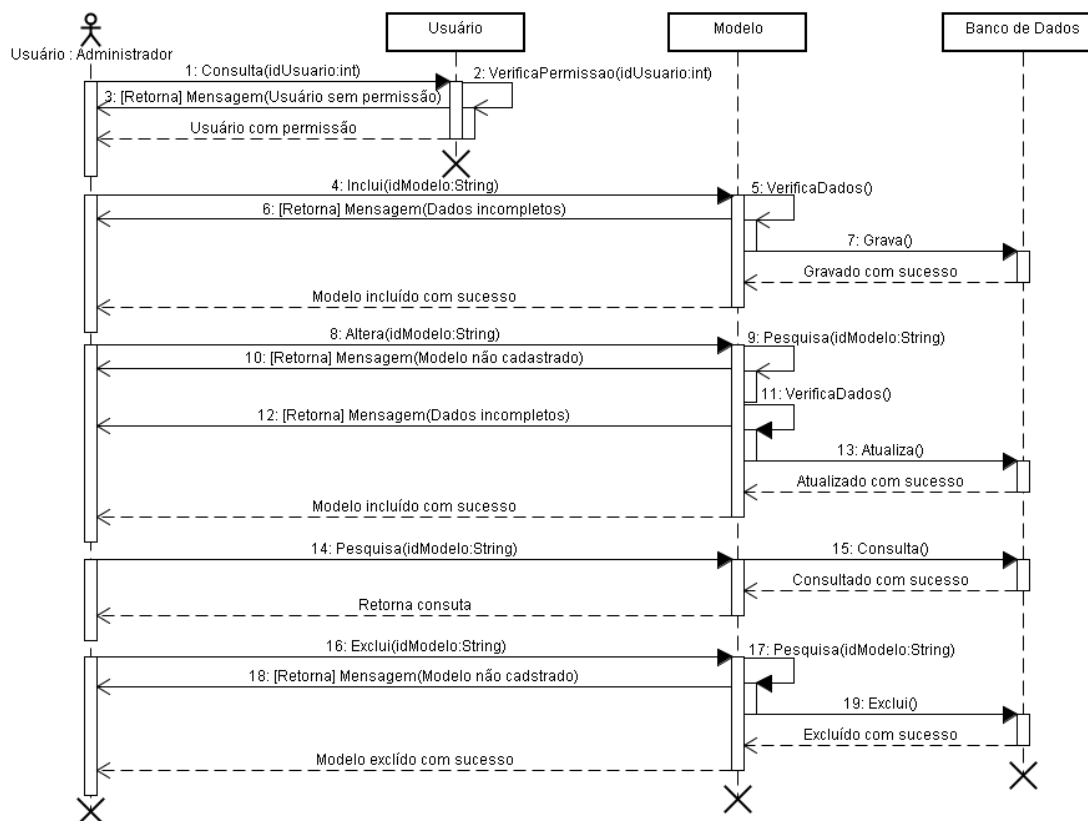


Figura Anexo 16 – Diagrama de Sequência Gerência de modelo.

Fonte: autor (2010).

UC009 – Gerência de material

Breve descritivo: este caso de uso descreve o processo de gerenciamento de material.

Requisitos funcionais:	REQF9 – [Cadastrar material] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções]
Atores:	Administrador.
Pré-condições:	Ser usuário com permissão de administrador.
Fluxo principal:	O usuário administrador acessa a tela de gerenciamento de material. Os subfluxos são apresentados pelo sistema: Inclusão, Alteração, Consulta e Exclusão.
Subfluxos:	1. Inclusão: 1.1 O usuário seleciona a opção de inclusão;

	<p>1.2 O usuário informa dos dados de um novo material: descrição;</p> <p>1.3 O usuário clica no botão/link “Salvar”;</p> <p>1.4 O sistema verifica os dados;</p> <p>1.5 O sistema preenche o código com o próximo disponível;</p> <p>1.6 O sistema grava as informações no banco de dados;</p> <p>1.7 O sistema retorna a mensagem “Dados incluídos com sucesso”.</p> <p>2. Alteração:</p> <p>2.1 O usuário seleciona a opção de alteração;</p> <p>2.2 O usuário informa o código do material que deseja alterar;</p> <p>2.3 O sistema preenche os dados do material;</p> <p>2.4 O usuário altera os dados desejados;</p> <p>2.5 O usuário clica no botão/link “Salvar”;</p> <p>2.6 O sistema verifica os dados;</p> <p>2.7 O sistema grava as informações no banco de dados;</p> <p>2.8 O sistema retorna a mensagem “Dados alterados com sucesso”.</p> <p>3. Consulta:</p> <p>3.1 O usuário seleciona a opção de consulta;</p> <p>3.2 O usuário informa o código do material que deseja consultar;</p> <p>3.3 O sistema preenche os dados do material.</p> <p>4. Exclusão:</p> <p>4.1 O usuário seleciona a opção de exclusão;</p> <p>4.2 O usuário informa o código do material que deseja excluir;</p> <p>4.3 O sistema preenche os dados do material;</p> <p>4.4 O usuário clica no botão/link “Excluir”;</p> <p>4.5 O sistema verifica os dados;</p> <p>4.6 O sistema exclui as informações no banco de dados;</p> <p>4.7 O sistema retorna a mensagem “Dados excluídos com sucesso”.</p>
Tratamento de exceções:	<p>1.4.1 O sistema deve verificar se os campos obrigatórios foram preenchidos e retornar a mensagem “Preencha todos os campos obrigatórios”, caso algum não tenha sido preenchido;</p> <p>2.2.1 Se o código informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p> <p>2.6.1 Recebe o mesmo tratamento de exceção do item 1.4.1;</p> <p>3.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.5.1 Recebe mesmo tratamento de exceção do item 1.4.1.</p>
Pós-condições:	<p>1. Um novo material está cadastrado;</p> <p>2. Um material está alterado;</p>

	3. Um material está consultado;
	4. Um material está excluído.

Quadro Anexo 9 – Caso de Uso Gerência de material.
Fonte: autor (2010).

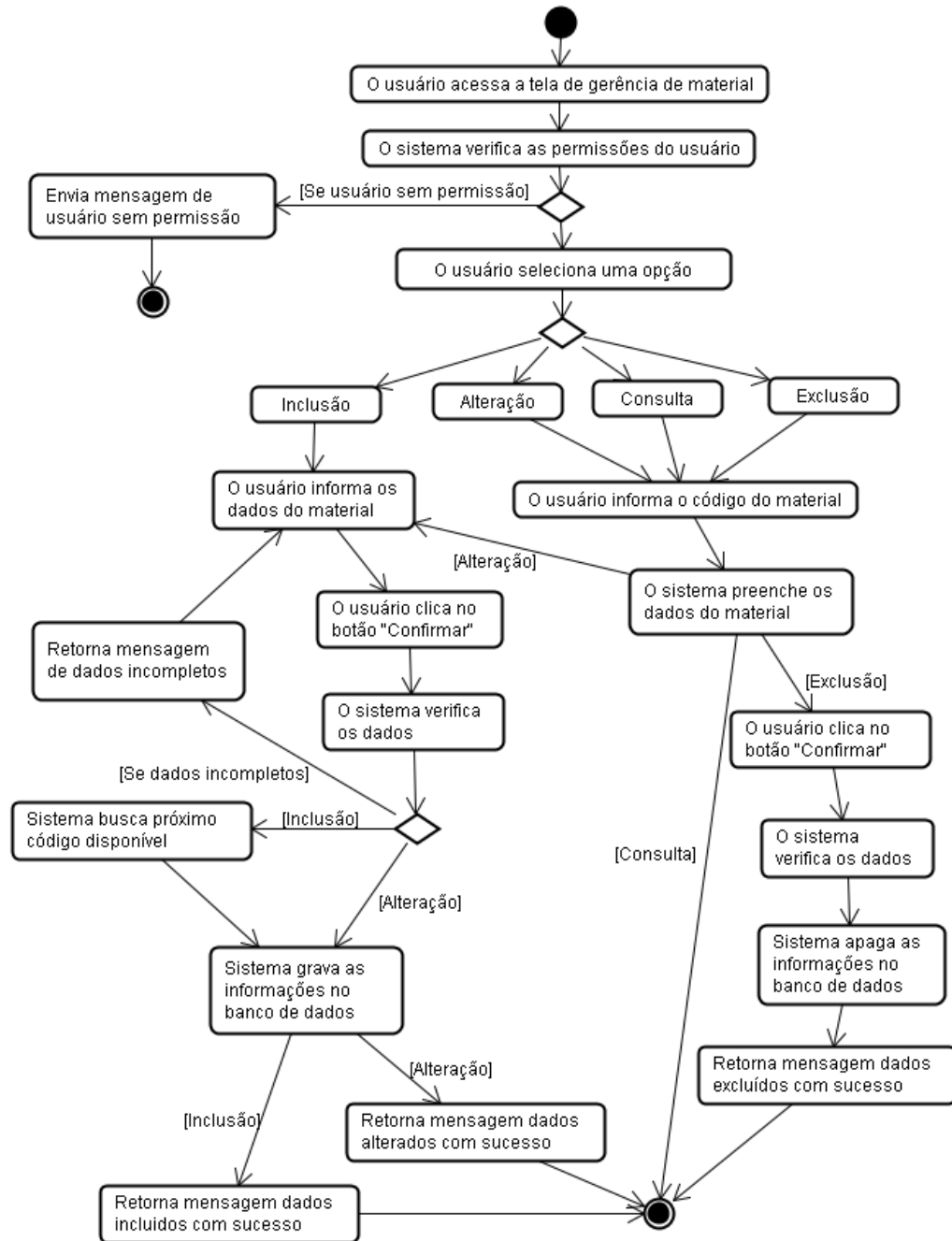


Figura Anexo 17 – Diagrama de Atividade Gerência de material.
Fonte: autor (2010).

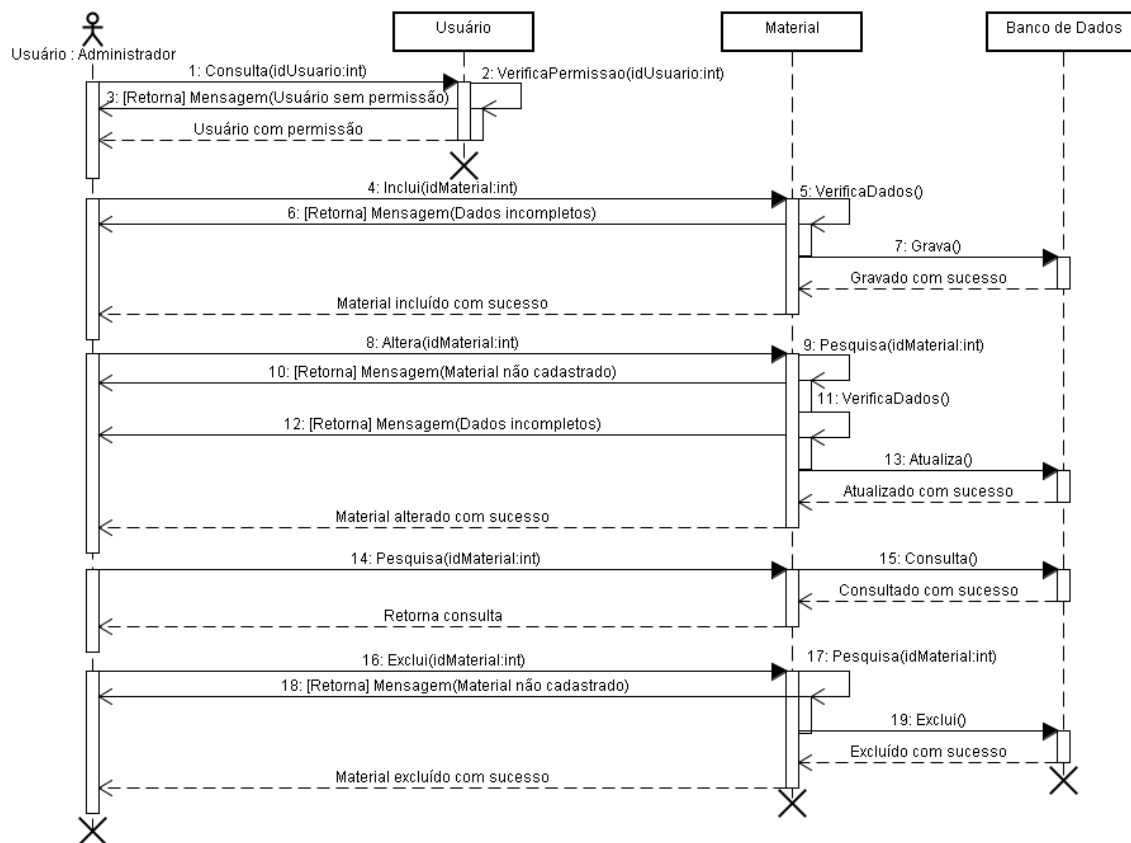


Figura Anexo 18 – Diagrama de Sequência Gerência de material.
Fonte: autor (2010).

UC010 – Gerência de cor

Breve descritivo: este caso de uso descreve o processo de gerenciamento de cor.

Requisitos funcionais:	REQF10 – [Cadastrar cor] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções]
Atores:	Administrador.
Pré-condições:	Ser usuário com permissão de administrador.
Fluxo principal:	O usuário administrador acessa a tela de gerenciamento de cor. Os subfluxos são apresentados pelo sistema: Inclusão, Alteração, Consulta e Exclusão.
Subfluxos:	1. Inclusão: 1.1 O usuário seleciona a opção de inclusão; 1.2 O usuário informa dos dados de uma nova cor: descrição;

	<p>1.3 O usuário clica no botão/link “Salvar”;</p> <p>1.4 O sistema verifica os dados;</p> <p>1.5 O sistema preenche o código com o próximo disponível;</p> <p>1.6 O sistema grava as informações no banco de dados;</p> <p>1.7 O sistema retorna a mensagem “Dados incluídos com sucesso”.</p> <p>2. Alteração:</p> <p>2.1 O usuário seleciona a opção de alteração;</p> <p>2.2 O usuário informa o código da cor que deseja alterar;</p> <p>2.3 O sistema preenche os dados da cor;</p> <p>2.4 O usuário altera os dados desejados;</p> <p>2.5 O usuário clica no botão/link “Salvar”;</p> <p>2.6 O sistema verifica os dados;</p> <p>2.7 O sistema grava as informações no banco de dados;</p> <p>2.8 O sistema retorna a mensagem “Dados alterados com sucesso”.</p> <p>3. Consulta:</p> <p>3.1 O usuário seleciona a opção de consulta;</p> <p>3.2 O usuário informa o código da cor que deseja consultar;</p> <p>3.3 O sistema preenche os dados da cor.</p> <p>4. Exclusão:</p> <p>4.1 O usuário seleciona a opção de exclusão;</p> <p>4.2 O usuário informa o código da cor que deseja excluir;</p> <p>4.3 O sistema preenche os dados da cor;</p> <p>4.4 O usuário clica no botão/link “Excluir”;</p> <p>4.5 O sistema verifica os dados;</p> <p>4.6 O sistema exclui as informações no banco de dados;</p> <p>4.7 O sistema retorna a mensagem “Dados excluídos com sucesso”.</p>
Tratamento de exceções:	<p>1.4.1 O sistema deve verificar se os campos obrigatórios foram preenchidos e retornar a mensagem “Preencha todos os campos obrigatórios” caso algum não tenha sido preenchido;</p> <p>2.2.1 Se o código informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p> <p>2.6.1 Recebe o mesmo tratamento de exceção do item 1.4.1;</p> <p>3.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.5.1 Recebe mesmo tratamento de exceção do item 1.4.1.</p>
Pós-condições:	<p>1. Uma nova cor está cadastrada;</p> <p>2. Uma cor está alterada;</p> <p>3. Uma cor está consultada;</p>

	4. Uma cor está excluída.
--	---------------------------

Quadro Anexo 10 – Caso de Uso Gerência de cor.

Fonte: autor (2010).

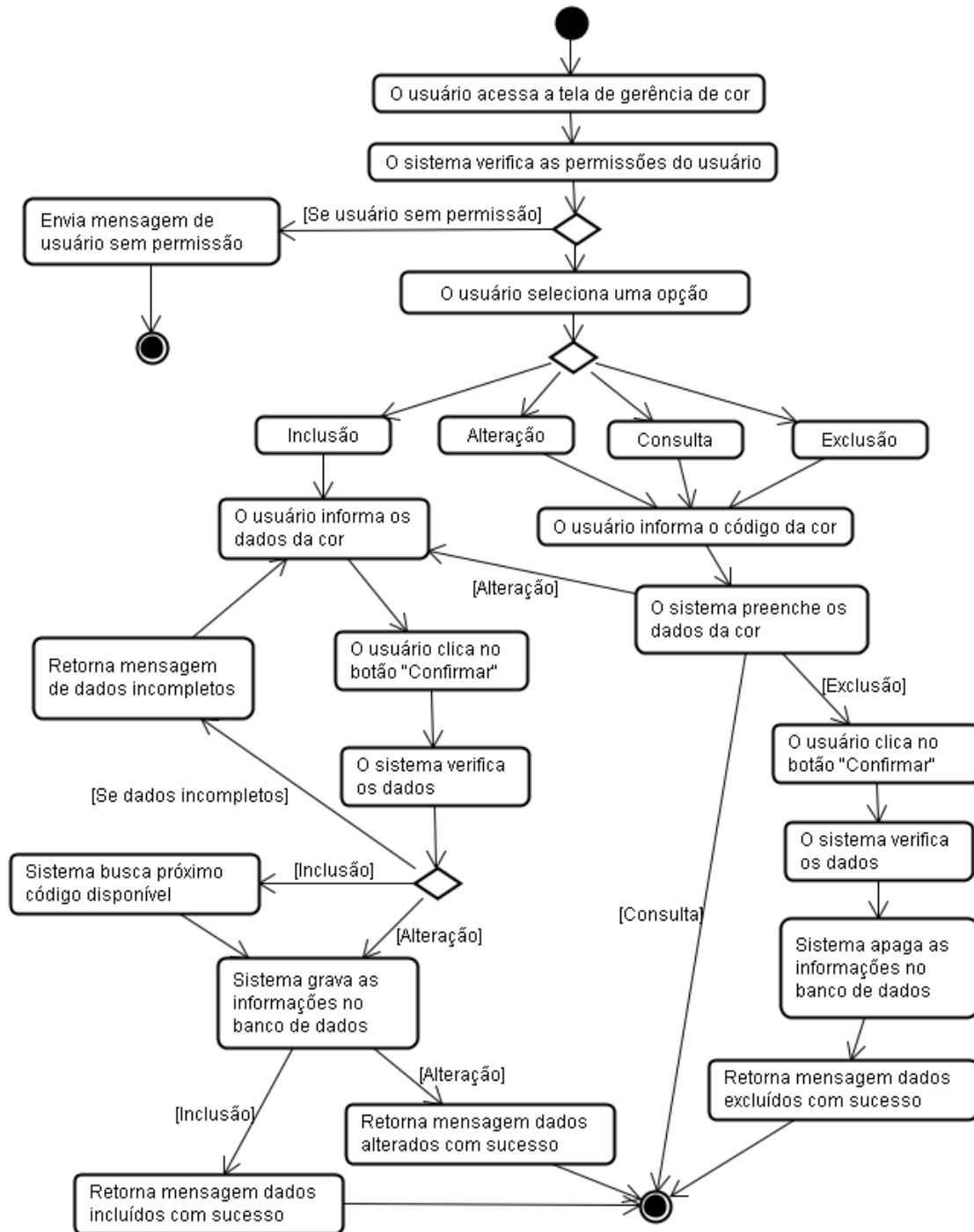


Figura Anexo 19 – Diagrama de Atividade Gerência de cor.

Fonte: autor (2010).

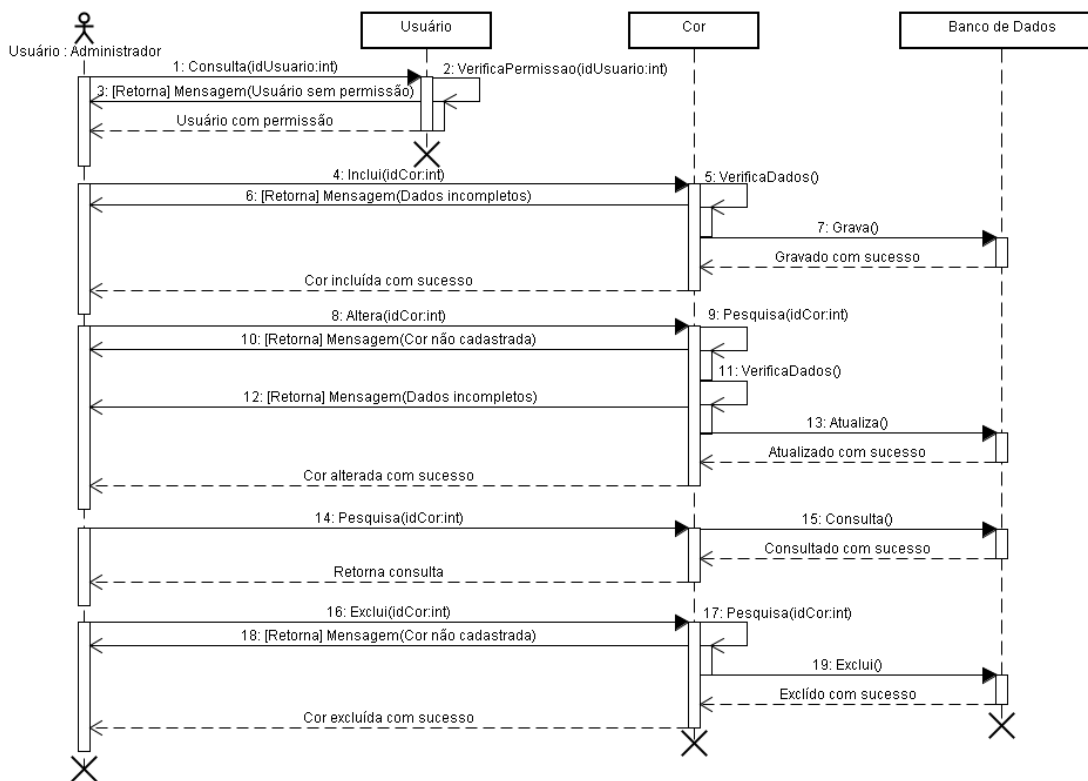


Figura Anexo 20 – Diagrama de Sequência Gerência de cor.
 Fonte: autor (2010).

UC011 – Gerência de classificação fiscal

Breve descritivo: este caso de uso descreve o processo de gerenciamento de classificação fiscal.

Requisitos funcionais:	REQF11 – [Cadastrar classificação fiscal] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões] para acesso às principais funções]
Atores:	Administrador.
Pré-condições:	Ser usuário com permissão de administrador.
Fluxo principal:	O usuário administrador acessa a tela de gerenciamento de classificação fiscal. Os subfluxos são apresentados pelo sistema: Inclusão, Alteração, Consulta e Exclusão.
Subfluxos:	1. Inclusão: 1.1 O usuário seleciona a opção de inclusão;

	<p>1.2 O usuário informa dos dados de uma nova classificação fiscal: descrição, espécie, tipo e percentual de IPI;</p> <p>1.3 O usuário clica no botão/link “Salvar”;</p> <p>1.4 O sistema verifica os dados;</p> <p>1.5 O sistema preenche o código com o próximo disponível;</p> <p>1.6 O sistema grava as informações no banco de dados;</p> <p>1.7 O sistema retorna a mensagem “Dados incluídos com sucesso”.</p> <p>2. Alteração:</p> <p>2.1 O usuário seleciona a opção de alteração;</p> <p>2.2 O usuário informa o código da cor que deseja alterar;</p> <p>2.3 O sistema preenche os dados da classificação fiscal;</p> <p>2.4 O usuário altera os dados desejados;</p> <p>2.5 O usuário clica no botão/link “Salvar”;</p> <p>2.6 O sistema verifica os dados;</p> <p>2.7 O sistema grava as informações no banco de dados;</p> <p>2.8 O sistema retorna a mensagem “Dados alterados com sucesso”.</p> <p>3. Consulta:</p> <p>3.1 O usuário seleciona a opção de consulta;</p> <p>3.2 O usuário informa o código da classificação fiscal que deseja consultar;</p> <p>3.3 O sistema preenche os dados da classificação fiscal.</p> <p>4. Exclusão:</p> <p>4.1 O usuário seleciona a opção de exclusão;</p> <p>4.2 O usuário informa o código da classificação fiscal que deseja excluir;</p> <p>4.3 O sistema preenche os dados da classificação fiscal;</p> <p>4.4 O usuário clica no botão/link “Excluir”;</p> <p>4.5 O sistema verifica os dados;</p> <p>4.6 O sistema exclui as informações no banco de dados;</p> <p>4.7 O sistema retorna a mensagem “Dados excluídos com sucesso”.</p>
Tratamento de exceções:	<p>1.4.1 O sistema deve verificar se os campos obrigatórios foram preenchidos e retornar a mensagem “Preencha todos os campos obrigatórios”, caso algum não tenha sido preenchido;</p> <p>2.2.1 Se o código informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p> <p>2.6.1 Recebe o mesmo tratamento de exceção do item 1.4.1;</p> <p>3.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.5.1 Recebe mesmo tratamento de exceção do item 1.4.1.</p>
Pós-condições:	1. Uma nova classificação fiscal está cadastrada;

	2. Uma classificação fiscal está alterada; 3. Uma classificação fiscal está consultada; 4. Uma classificação fiscal está excluída.
--	------------------------------------------------------------------------------------------------------------------------------------------

Quadro Anexo 11 – Caso de Uso Gerência de classificação fiscal.

Fonte: autor (2010).

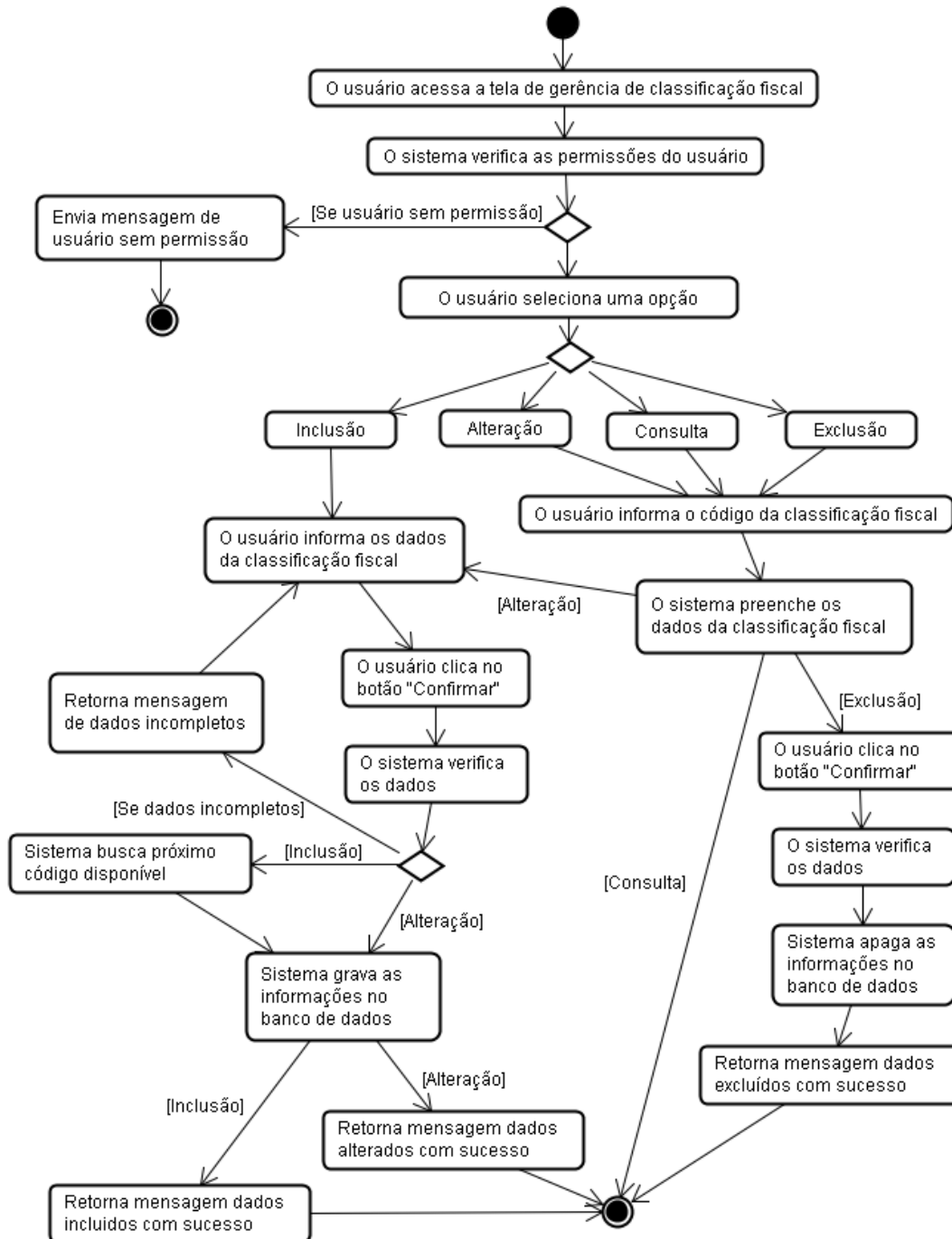


Figura Anexo 21 – Diagrama de Atividade Gerência de classificação fiscal.

Fonte: autor (2010).

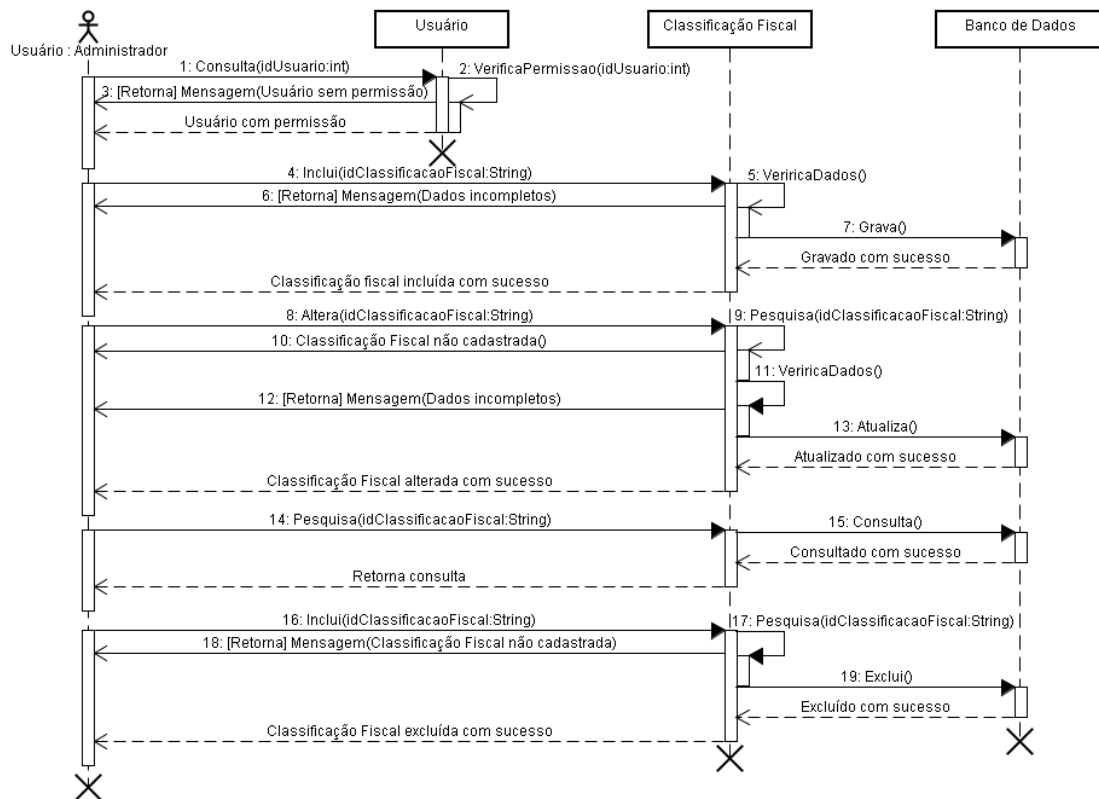


Figura Anexo 22 – Diagrama de Sequência Gerência de classificação fiscal.

Fonte: autor (2010).

UC012 – Gerência de unidade de medida

Breve descritivo: este caso de uso descreve o processo de gerenciamento de unidade de medida.

Requisitos funcionais:	REQF12 – [Cadastrar unidade de medida] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões] para acesso às principais funções]
Atores:	Administrador.
Pré-condições:	Ser usuário com permissão de administrador.
Fluxo principal:	O usuário administrador acessa a tela de gerenciamento de unidade de medida. Os subfluxos são apresentados pelo sistema: Inclusão, Alteração, Consulta e Exclusão.
Subfluxos:	1. Inclusão:

	<p>1.1 O usuário seleciona a opção de inclusão;</p> <p>1.2 O usuário informa dos dados de uma nova unidade de medida: descrição e abreviação;</p> <p>1.3 O usuário clica no botão/link “Salvar”;</p> <p>1.4 O sistema verifica os dados;</p> <p>1.5 O sistema preenche o código com o próximo disponível;</p> <p>1.6 O sistema grava as informações no banco de dados;</p> <p>1.7 O sistema retorna a mensagem “Dados incluídos com sucesso”.</p> <p>2. Alteração:</p> <p>2.1 O usuário seleciona a opção de alteração;</p> <p>2.2 O usuário informa o código da unidade de medida que deseja alterar;</p> <p>2.3 O sistema preenche os dados da unidade de medida;</p> <p>2.4 O usuário altera os dados desejados;</p> <p>2.5 O usuário clica no botão/link “Salvar”;</p> <p>2.6 O sistema verifica os dados;</p> <p>2.7 O sistema grava as informações no banco de dados;</p> <p>2.8 O sistema retorna a mensagem “Dados alterados com sucesso”.</p> <p>3. Consulta:</p> <p>3.1 O usuário seleciona a opção de consulta;</p> <p>3.2 O usuário informa o código da unidade de medida que deseja consultar;</p> <p>3.3 O sistema preenche os dados da unidade de medida.</p> <p>4. Exclusão:</p> <p>4.1 O usuário seleciona a opção de exclusão;</p> <p>4.2 O usuário informa o código da unidade de medida que deseja excluir;</p> <p>4.3 O sistema preenche os dados da unidade de medida;</p> <p>4.4 O usuário clica no botão/link “Excluir”;</p> <p>4.5 O sistema verifica os dados;</p> <p>4.6 O sistema exclui as informações no banco de dados;</p> <p>4.7 O sistema retorna a mensagem “Dados excluídos com sucesso”.</p>
Tratamento de exceções:	<p>1.4.1 O sistema deve verificar se os campos obrigatórios foram preenchidos e retornar a mensagem “Preencha todos os campos obrigatórios” caso algum não tenha sido preenchido;</p> <p>2.2.1 Se o código informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p> <p>2.6.1 Recebe o mesmo tratamento de exceção do item 1.4.1;</p> <p>3.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.5.1 Recebe mesmo tratamento de exceção do item 1.4.1.</p>

Pós-condições:	<ol style="list-style-type: none"> 1. Uma nova unidade de medida está cadastrada; 2. Uma unidade de medida está alterada; 3. Uma unidade de medida está consultada; 4. Uma unidade de medida está excluída.
----------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Quadro Anexo 12 – Caso de Uso Gerência de unidade de medida.
Fonte: autor (2010).

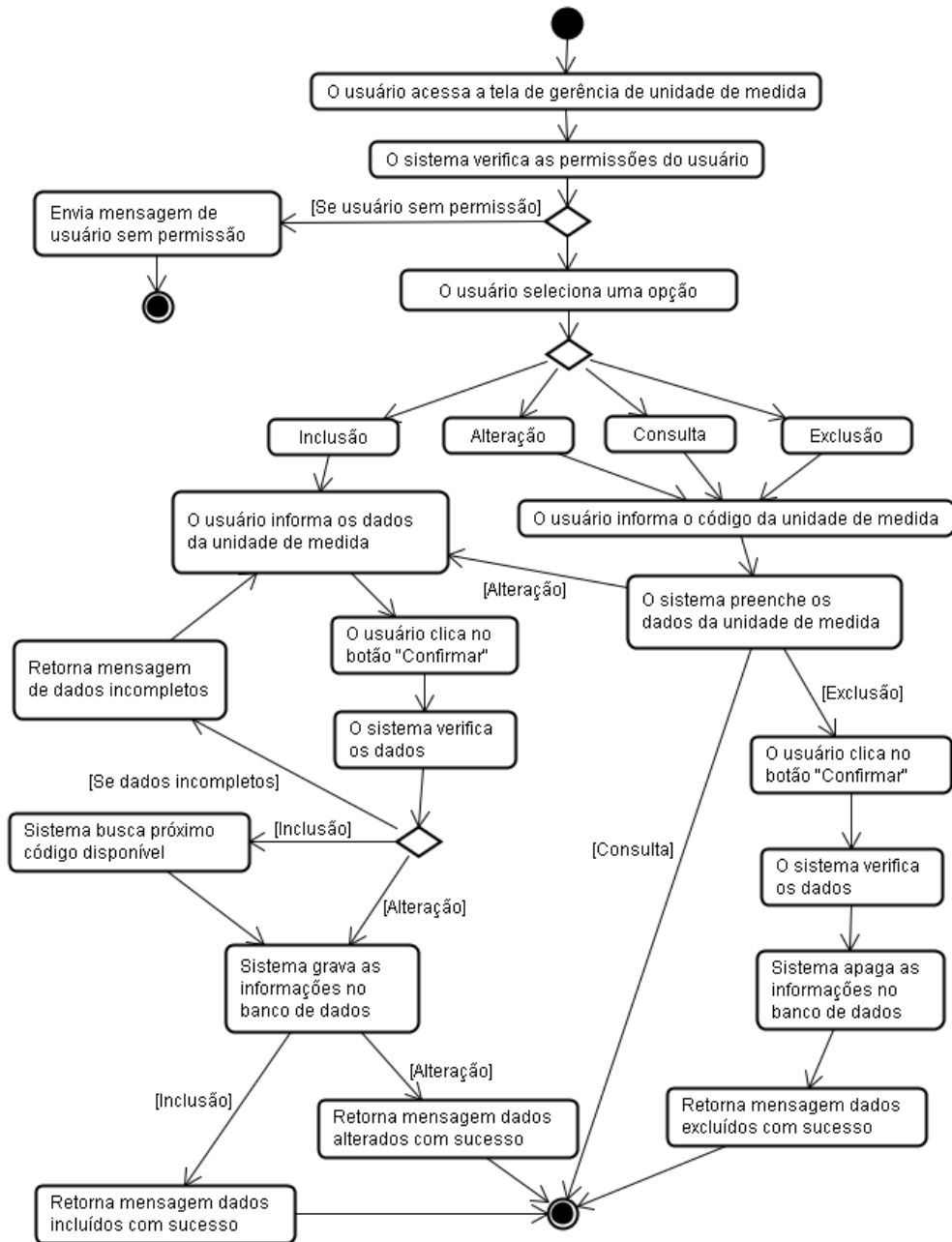


Figura Anexo 23 – Diagrama de Atividade Gerência de unidade de medida.
Fonte: autor (2010).

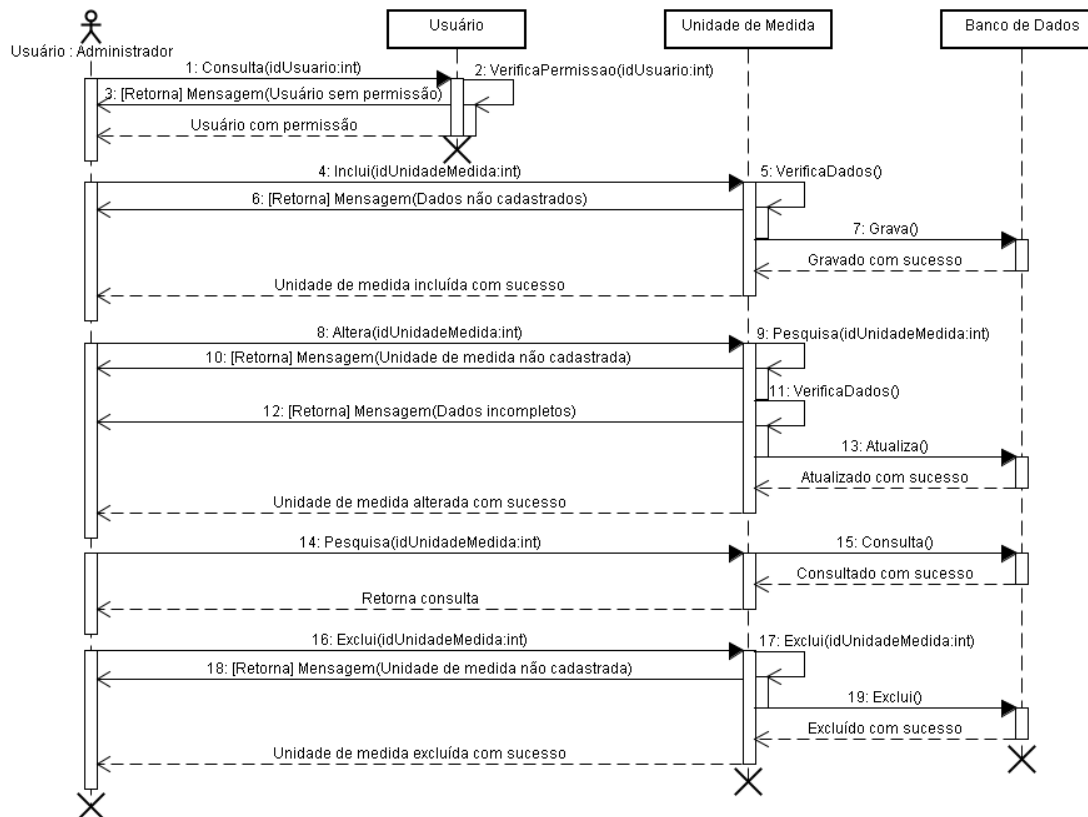


Figura Anexo 24 – Diagrama de Sequência Gerência de unidade de medida.

Fonte: autor (2010).

UC014 – Gerência de preço

Breve descritivo: este caso de uso descreve o processo de gerenciamento de preço.

Requisitos funcionais:	REQF2 – [Cadastrar rede de lojas] REQF3 – [Cadastrar loja] REQF13 – [Cadastrar produto] REQF14 – [Cadastrar preço] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões] para acesso às principais funções]
Atores:	Administrador.
Pré-condições:	Ser usuário com permissão de administrador.
Fluxo principal:	O usuário administrador acessa a tela de gerenciamento de preço. Os subfluxos são apresentados pelo sistema: Inclusão, Alteração, Consulta e Exclusão.
Subfluxos:	1. Inclusão: 1.1 O usuário seleciona a opção de inclusão;

	<p>1.2 O usuário informa dos dados de um novo preço do produto: rede, loja, produto, preço e percentual de ICMS;</p> <p>1.3 O usuário clica no botão/link “Salvar”;</p> <p>1.4 O sistema verifica os dados;</p> <p>1.5 O sistema grava as informações no banco de dados;</p> <p>1.6 O sistema retorna a mensagem “Dados incluídos com sucesso”.</p> <p>2. Alteração:</p> <p>2.1 O usuário seleciona a opção de alteração;</p> <p>2.2 O usuário informa o código da rede, da loja e do produto que deseja alterar o preço;</p> <p>2.3 O sistema preenche os dados do preço do produto;</p> <p>2.4 O usuário altera os dados desejados;</p> <p>2.5 O usuário clica no botão/link “Salvar”;</p> <p>2.6 O sistema verifica os dados;</p> <p>2.7 O sistema grava as informações no banco de dados;</p> <p>2.8 O sistema retorna a mensagem “Dados alterados com sucesso”.</p> <p>3. Consulta:</p> <p>3.1 O usuário seleciona a opção de consulta;</p> <p>3.2 O usuário informa o código da rede, da loja e do produto que deseja consultar o preço;</p> <p>3.3 O sistema preenche os dados do preço do produto.</p> <p>4. Exclusão:</p> <p>4.1 O usuário seleciona a opção de exclusão;</p> <p>4.2 O usuário informa o código da rede, da loja e do produto que deseja excluir;</p> <p>4.3 O sistema preenche os dados do preço do produto;</p> <p>4.4 O usuário clica no botão/link “Excluir”;</p> <p>4.5 O sistema verifica os dados;</p> <p>4.6 O sistema exclui as informações no banco de dados;</p> <p>4.7 O sistema retorna a mensagem “Dados excluídos com sucesso”.</p>
Tratamento de exceções:	<p>1.4.1 O sistema deve verificar se os campos obrigatórios foram preenchidos e retornar a mensagem “Preencha todos os campos obrigatórios” caso algum não tenha sido preenchido;</p> <p>2.2.1 Se o código informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p> <p>2.6.1 Recebe o mesmo tratamento de exceção do item 1.4.1;</p> <p>3.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.5.1 Recebe mesmo tratamento de exceção do item 1.4.1.</p>
Pós-condições:	1. Um novo preço está cadastrado;

	2. Um preço está alterado; 3. Um preço está consultado; 4. Um preço está excluído.
--	------------------------------------------------------------------------------------------

Quadro Anexo 13 – Caso de Uso Gerência de preço.

Fonte: autor (2010).

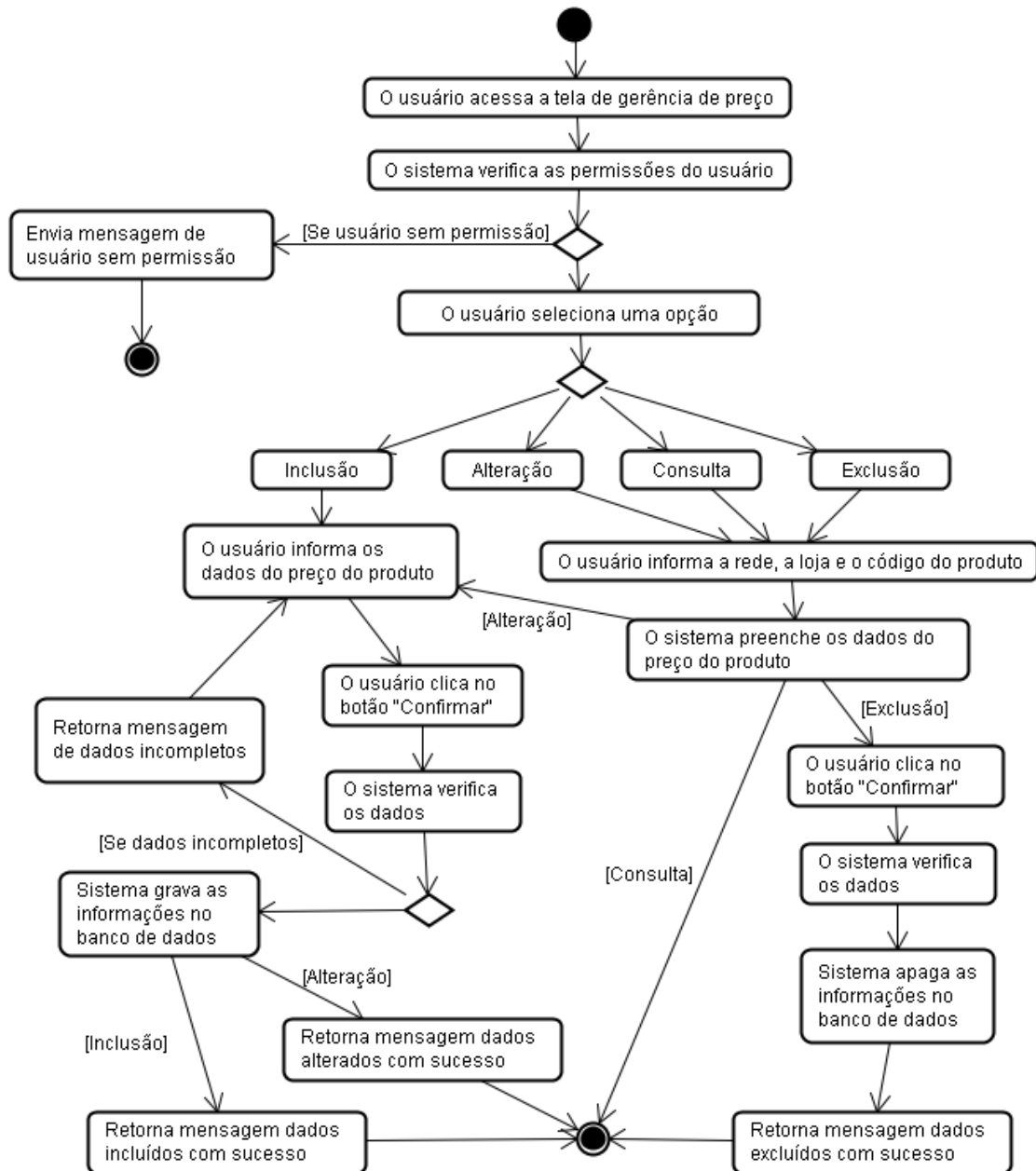


Figura Anexo 25 – Diagrama de Atividade Gerência de preço.

Fonte: autor (2010).

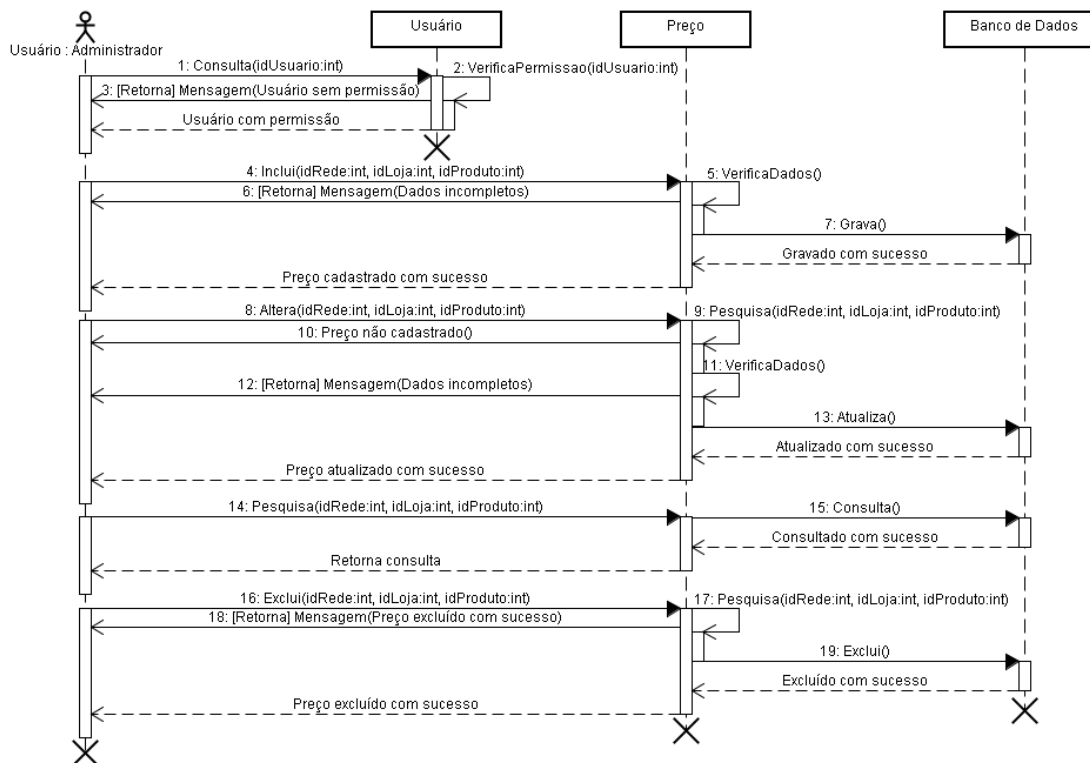


Figura Anexo 26 – Diagrama de Sequência Gerência de preço.

Fonte: autor (2010).

UC015 – Gerência de forma de pagamento

Breve descritivo: este caso de uso descreve o processo de gerenciamento de forma de pagamento.

Requisitos funcionais:	REQF15 – [Cadastrar forma de pagamento] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções]
Atores:	Administrador.
Pré-condições:	Ser usuário com permissão de administrador.
Fluxo principal:	O usuário administrador acessa a tela de gerenciamento de forma de pagamento. Os subfluxos são apresentados pelo sistema: Inclusão, Alteração, Consulta e Exclusão.
Subfluxos:	1. Inclusão: 1.1 O usuário seleciona a opção de inclusão; 1.2 O usuário informa dos dados de uma nova forma de pagamento: descrição e situação;

	<p>1.3 O usuário clica no botão/link “Salvar”;</p> <p>1.4 O sistema verifica os dados;</p> <p>1.5 O sistema preenche o código com o próximo disponível</p> <p>1.6 O sistema grava as informações no banco de dados;</p> <p>1.7 O sistema retorna a mensagem “Dados incluídos com sucesso”.</p> <p>2. Alteração:</p> <p>2.1 O usuário seleciona a opção de alteração;</p> <p>2.2 O usuário informa o código da forma de pagamento que deseja alterar;</p> <p>2.3 O sistema preenche os dados da forma de pagamento;</p> <p>2.4 O usuário altera os dados desejados;</p> <p>2.5 O usuário clica no botão/link “Salvar”;</p> <p>2.6 O sistema verifica os dados;</p> <p>2.7 O sistema grava as informações no banco de dados;</p> <p>2.8 O sistema retorna a mensagem “Dados alterados com sucesso”.</p> <p>3. Consulta:</p> <p>3.1 O usuário seleciona a opção de consulta;</p> <p>3.2 O usuário informa o código da forma de pagamento que deseja consultar;</p> <p>3.3 O sistema preenche os dados do preço da forma de pagamento.</p> <p>4. Exclusão:</p> <p>4.1 O usuário seleciona a opção de exclusão;</p> <p>4.2 O usuário informa o código da forma de pagamento que deseja excluir;</p> <p>4.3 O sistema preenche os dados da forma de pagamento;</p> <p>4.4 O usuário clica no botão/link “Excluir”;</p> <p>4.5 O sistema verifica os dados;</p> <p>4.6 O sistema exclui as informações no banco de dados;</p> <p>4.7 O sistema retorna a mensagem “Dados excluídos com sucesso”.</p>
Tratamento de exceções:	<p>1.4.1 O sistema deve verificar se os campos obrigatórios foram preenchidos e retornar a mensagem “Preencha todos os campos obrigatórios” caso algum não tenha sido preenchido;</p> <p>2.2.1 Se o código informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p> <p>2.6.1 Recebe o mesmo tratamento de exceção do item 1.4.1;</p> <p>3.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.5.1 Recebe mesmo tratamento de exceção do item 1.4.1.</p>
Pós-condições:	<p>1. Uma nova forma de pagamento está cadastrada;</p> <p>2. Uma forma de pagamento está alterada;</p> <p>3. Uma forma de pagamento está consultada;</p>

4. Uma forma de pagamento está excluída.

Quadro Anexo 14 – Caso de Uso Gerência de forma de pagamento.

Fonte: autor (2010).

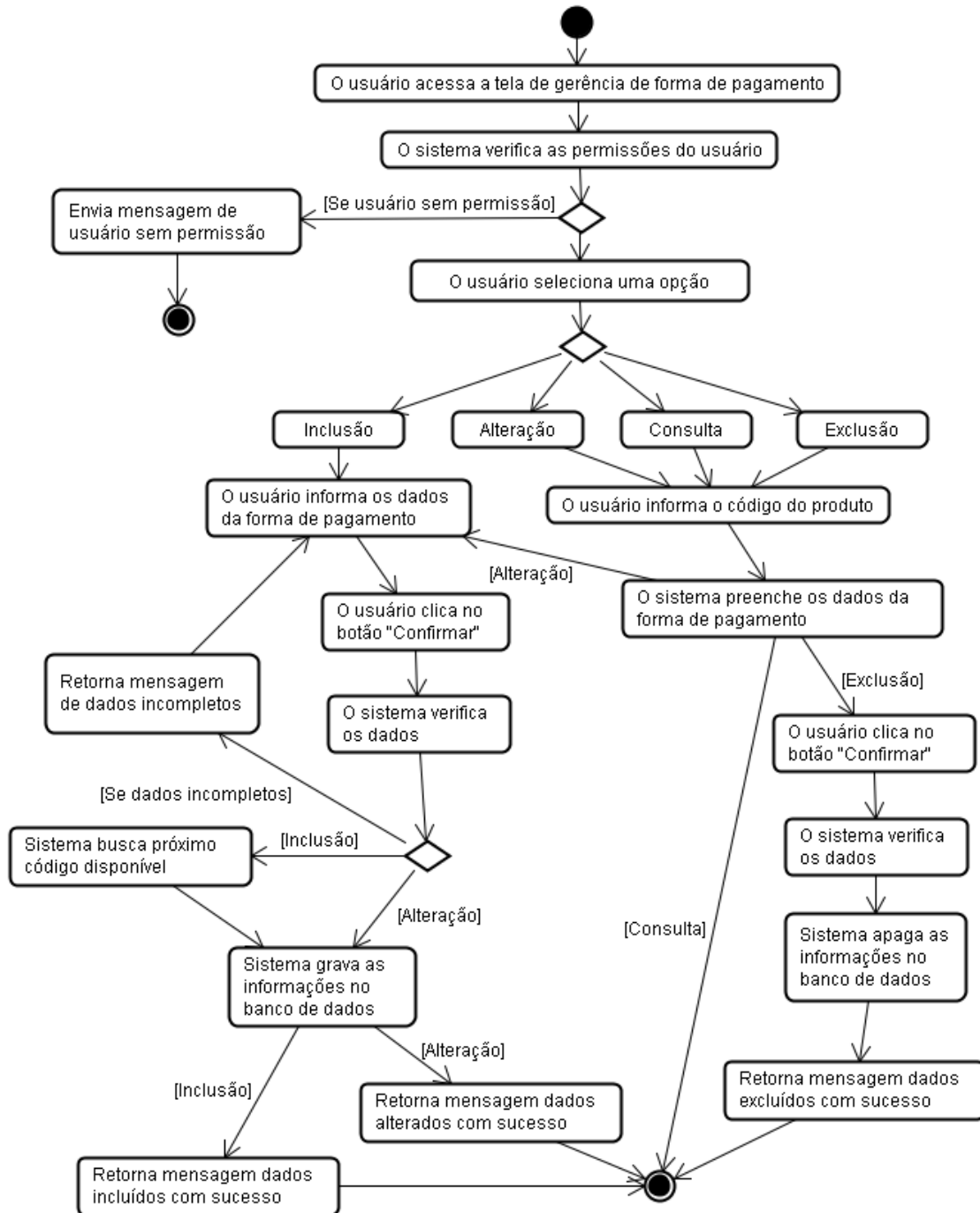


Figura Anexo 27 – Diagrama de Atividade Gerência de forma de pagamento.

Fonte: autor (2010).

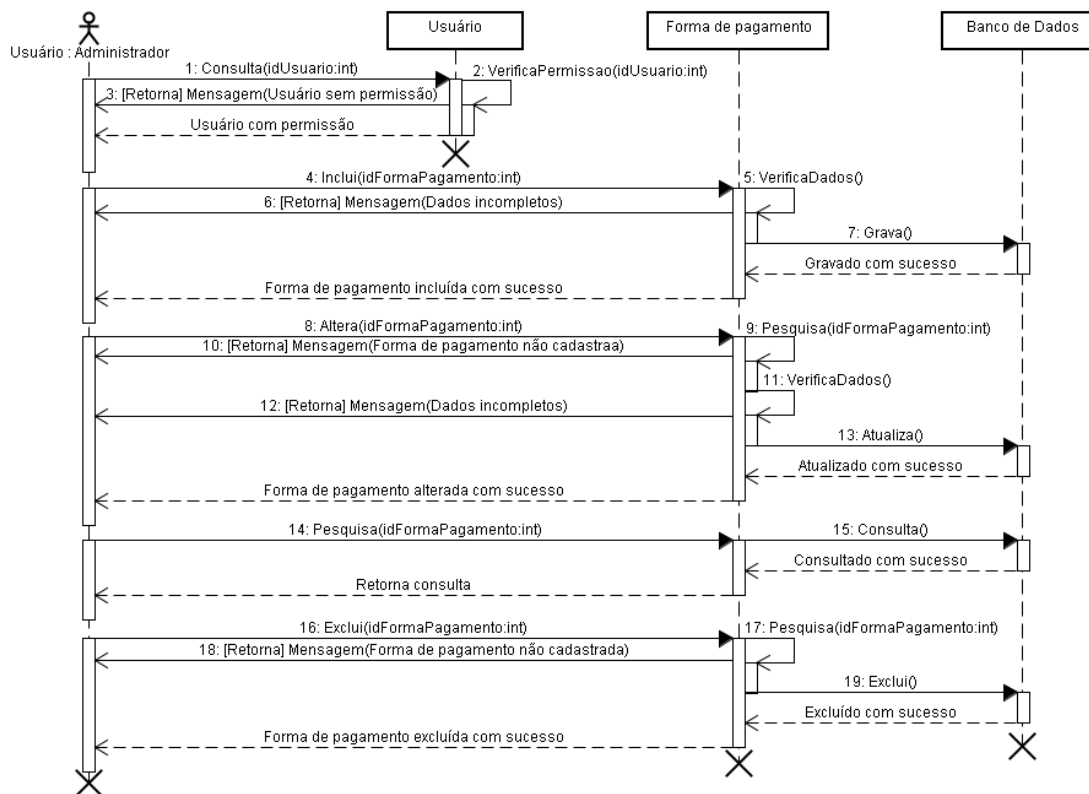


Figura Anexo 28 – Diagrama de Sequência Gerência de forma de pagamento.

Fonte: autor (2010).

UC017 – Gerência de vendedor

Breve descritivo: este caso de uso descreve o processo de gerenciamento de vendedor.

Requisitos funcionais:	REQF16 – [Cadastrar pessoa] REQF17 – [Cadastrar vendedor] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções]
Atores:	Gerente.
Pré-condições:	Ser usuário com permissão de gerente.
Fluxo principal:	O usuário gerente acessa a tela de gerenciamento de vendedor. Os subfluxos são apresentados pelo sistema: Inclusão, Alteração, Consulta e Exclusão.
Subfluxos:	1. Inclusão: 1.1 O usuário seleciona a opção de inclusão; 1.2 O usuário informa dos dados de um novo vendedor: rede, loja, pessoa, apelido, percentual de comissão e situação;

	<p>1.3 O usuário clica no botão/link “Salvar”;</p> <p>1.4 O sistema verifica os dados;</p> <p>1.5 O sistema preenche o código com o próximo disponível</p> <p>1.6 O sistema grava as informações no banco de dados;</p> <p>1.7 O sistema retorna a mensagem “Dados incluídos com sucesso”.</p> <p>2. Alteração:</p> <p>2.1 O usuário seleciona a opção de alteração;</p> <p>2.2 O usuário informa o código do vendedor que deseja alterar;</p> <p>2.3 O sistema preenche os dados do vendedor;</p> <p>2.4 O usuário altera os dados desejados;</p> <p>2.5 O usuário clica no botão/link “Salvar”;</p> <p>2.6 O sistema verifica os dados;</p> <p>2.7 O sistema grava as informações no banco de dados;</p> <p>2.8 O sistema retorna a mensagem “Dados alterados com sucesso”.</p> <p>3. Consulta:</p> <p>3.1 O usuário seleciona a opção de consulta;</p> <p>3.2 O usuário informa o código do vendedor que deseja consultar;</p> <p>3.3 O sistema preenche os dados do vendedor.</p> <p>4. Exclusão:</p> <p>4.1 O usuário seleciona a opção de exclusão;</p> <p>4.2 O usuário informa o código do vendedor que deseja excluir;</p> <p>4.3 O sistema preenche os dados do vendedor;</p> <p>4.4 O usuário clica no botão/link “Excluir”;</p> <p>4.5 O sistema verifica os dados;</p> <p>4.6 O sistema exclui as informações no banco de dados;</p> <p>4.7 O sistema retorna a mensagem “Dados excluídos com sucesso”.</p>
Tratamento de exceções:	<p>1.4.1 O sistema deve verificar se os campos obrigatórios foram preenchidos e retornar a mensagem “Preencha todos os campos obrigatórios.” caso algum não tenha sido preenchido;</p> <p>2.2.1 Se o código informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p> <p>2.6.1 Recebe o mesmo tratamento de exceção do item 1.4.1;</p> <p>3.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.5.1 Recebe mesmo tratamento de exceção do item 1.4.1.</p>
Pós-condições:	<p>1. Um novo vendedor está cadastrado;</p> <p>2. Um vendedor está alterado;</p> <p>3. Um vendedor está consultado;</p>

	4. Um vendedor está excluído.
--	-------------------------------

Quadro Anexo 15 – Caso de Uso Gerência de vendedor.

Fonte: autor (2010).

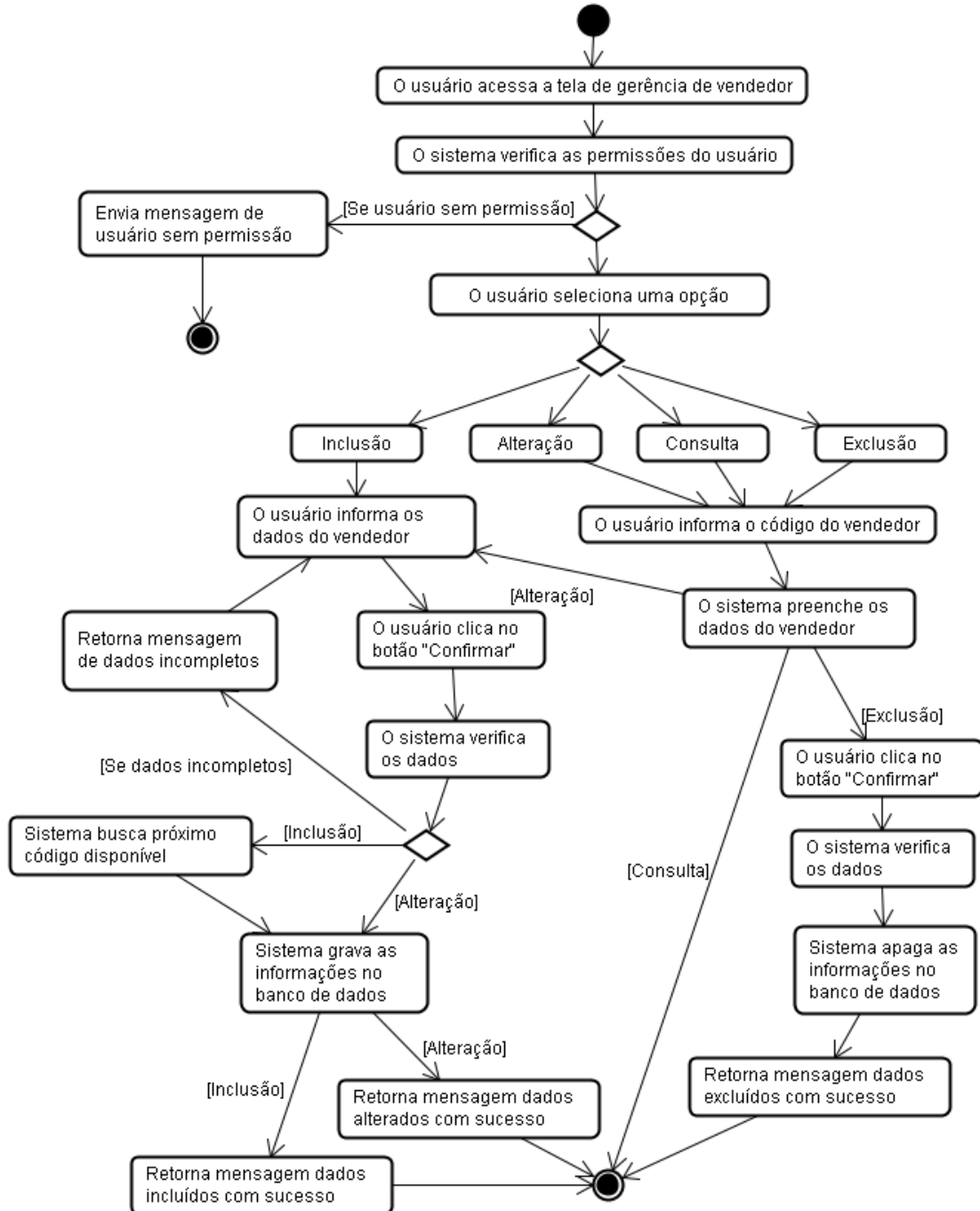


Figura Anexo 29 – Diagrama de Atividade Gerência de vendedor.

Fonte: autor (2010).

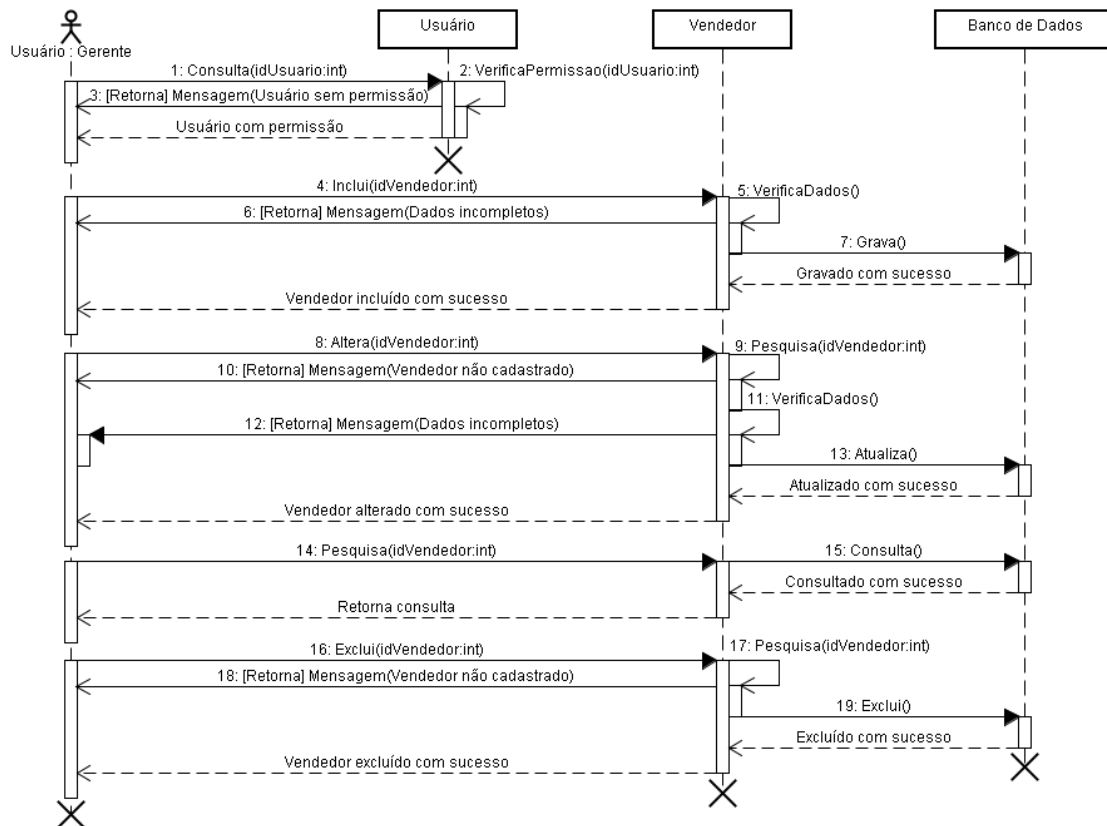


Figura Anexo 30 – Diagrama de Sequência Gerência de vendedor.

Fonte: autor (2010).

UC018 – Gerência de consumidor

Breve descritivo: este caso de uso descreve o processo de gerenciamento de consumidor.

Requisitos funcionais:	REQF16 – [Cadastrar pessoa] REQF18– [Cadastrar consumidor] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções]
Atores:	Gerente, Caixa.
Pré-condições:	Ser usuário com permissão de gerente ou de caixa.
Fluxo principal:	O usuário gerente/caixa acessa a tela de gerenciamento de consumidor. Os subfluxos são apresentados pelo sistema: Inclusão, Alteração, Consulta e Exclusão.
Subfluxos:	1. Inclusão: 1.1 O usuário seleciona a opção de inclusão; 1.2 O usuário informa dos dados de um novo consumidor: rede, loja, pessoa e

	<p>situação;</p> <p>1.3 O usuário clica no botão/link “Salvar”;</p> <p>1.4 O sistema verifica os dados;</p> <p>1.5 O sistema preenche o código com o próximo disponível;</p> <p>1.6 O sistema grava as informações no banco de dados;</p> <p>1.7 O sistema retorna a mensagem “Dados incluídos com sucesso”.</p> <p>2. Alteração:</p> <p>2.1 O usuário seleciona a opção de alteração;</p> <p>2.2 O usuário informa:</p> <p>2.2.1 O código do consumidor que deseja alterar;</p> <p>2.2.2 O CPF do consumidor que deseja alterar;</p> <p>2.3 O sistema preenche os dados do consumidor;</p> <p>2.4 O usuário altera os dados desejados;</p> <p>2.5 O usuário clica no botão/link “Salvar”;</p> <p>2.6 O sistema verifica os dados;</p> <p>2.7 O sistema grava as informações no banco de dados;</p> <p>2.8 O sistema retorna a mensagem “Dados alterados com sucesso”.</p> <p>3. Consulta:</p> <p>3.1 O usuário seleciona a opção de consulta;</p> <p>3.2 O usuário informa:</p> <p>3.2.1 O código do consumidor que deseja consultar;</p> <p>3.2.2 O CPF do consumidor que deseja consultar;</p> <p>3.3 O sistema preenche os dados do consumidor.</p> <p>4. Exclusão:</p> <p>4.1 O usuário seleciona a opção de exclusão;</p> <p>4.2 O usuário informa:</p> <p>4.2.1 O código do consumidor que deseja excluir;</p> <p>4.2.2 O CPF do consumidor que deseja excluir;</p> <p>4.3 O sistema preenche os dados do computador;</p> <p>4.4 O usuário clica no botão/link “Excluir”;</p> <p>4.5 O sistema verifica os dados;</p> <p>4.6 O sistema exclui as informações no banco de dados;</p> <p>4.7 O sistema retorna a mensagem “Dados excluídos com sucesso”.</p>
Tratamento de exceções:	<p>1.4.1 O sistema deve verificar se os campos obrigatórios foram preenchidos e retornar a mensagem “Preencha todos os campos obrigatórios.” caso algum não tenha sido preenchido;</p> <p>2.2.1.1 Se o código informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p>

	<p>2.2.1.2 Se o CPF informado não estiver registrado no banco de dados, o sistema retorna a mensagem “CPF não cadastrado”;</p> <p>2.6.1 Recebe o mesmo tratamento de exceção do item 1.4.1;</p> <p>3.2.1.1 Recebe mesmo tratamento de exceção do item 2.2.1.1;</p> <p>3.2.1.2 Recebe mesmo tratamento de exceção do item 2.2.1.2;</p> <p>4.2.1.1 Recebe mesmo tratamento de exceção do item 2.2.1.1;</p> <p>4.2.1.2 Recebe mesmo tratamento de exceção do item 2.2.1.2;</p> <p>4.5.1 Recebe mesmo tratamento de exceção do item 1.4.1.</p>
Pós-condições:	<ol style="list-style-type: none"> 1. Um novo consumidor está cadastrado; 2. Um consumidor está alterado; 3. Um consumidor está consultado; 4. Um consumidor está excluído.

Quadro Anexo 16 – Caso de Uso Gerência de consumidor.

Fonte: autor (2010).

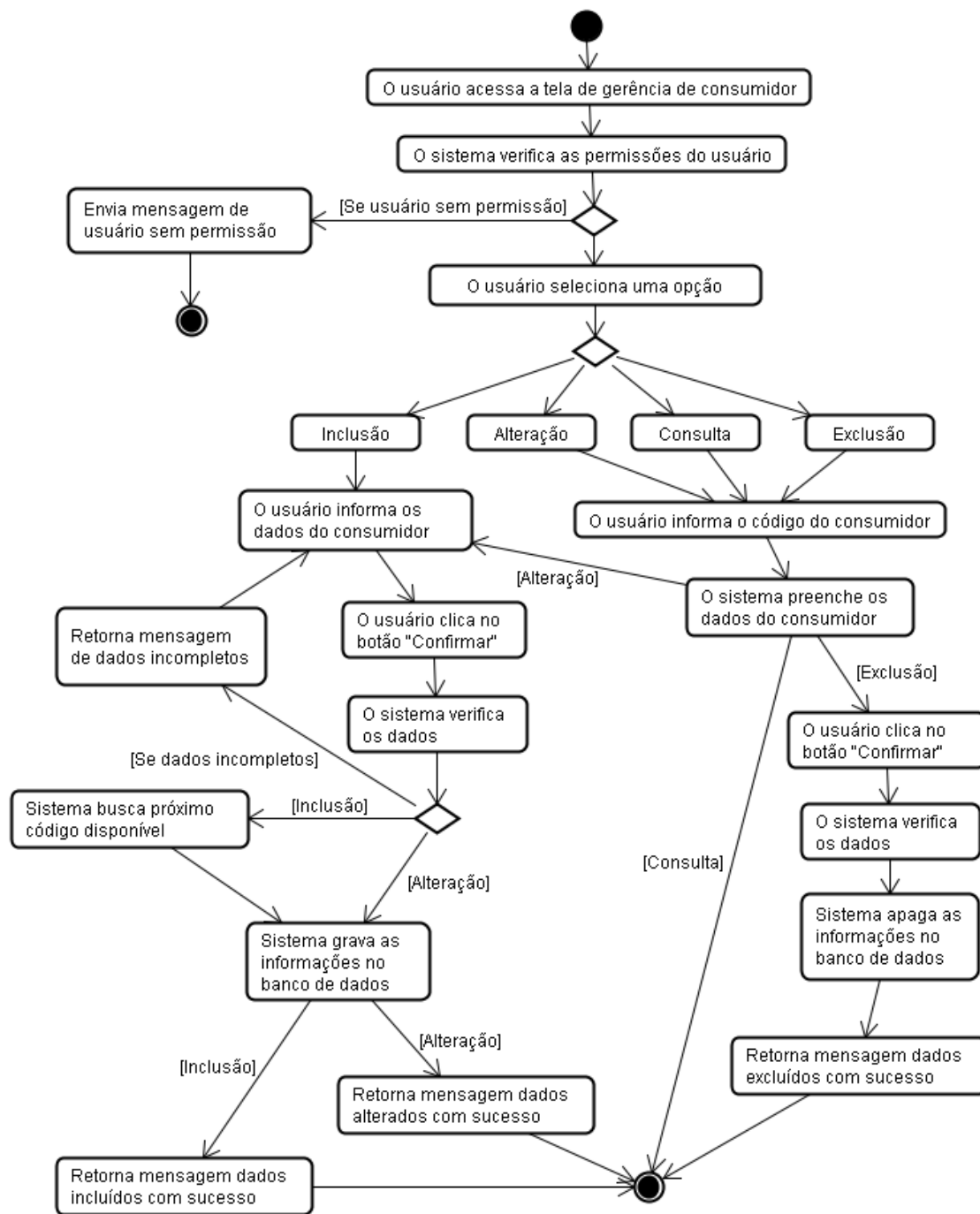


Figura Anexo 31 – Diagrama de Atividade Gerência de consumidor.
 Fonte: autor (2010).

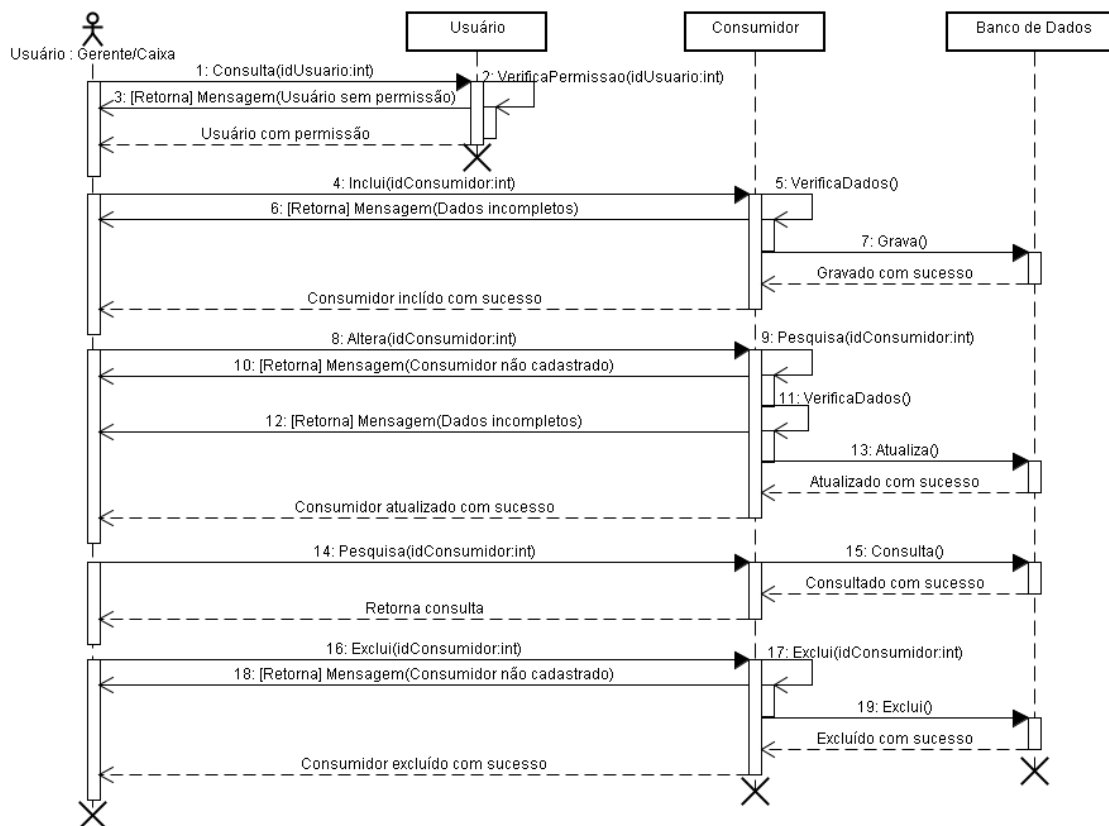


Figura Anexo 32 – Diagrama de Sequência Gerência de consumidor.
Fonte: autor (2010).

UC019 – Gerência de fornecedor

Breve descritivo: este caso de uso descreve o processo de gerenciamento de fornecedor.

Requisitos funcionais:	REQF16 – [Cadastrar pessoa] REQF19 – [Cadastrar fornecedor] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções]
Atores:	Administrador.
Pré-condições:	Ser usuário com permissão de administrador.
Fluxo principal:	O usuário administrador acessa a tela de gerenciamento de fornecedor. Os subfluxos são apresentados pelo sistema: Inclusão, Alteração, Consulta e Exclusão.
Subfluxos:	1. Inclusão: 1.1 O usuário seleciona a opção de inclusão; 1.2 O usuário informa dos dados de um novo fornecedor: razão social, pessoa e

	<p>situação;</p> <p>1.3 O usuário clica no botão/link “Salvar”;</p> <p>1.4 O sistema verifica os dados;</p> <p>1.5 O sistema preenche o código com o próximo disponível</p> <p>1.6 O sistema grava as informações no banco de dados;</p> <p>1.7 O sistema retorna a mensagem “Dados incluídos com sucesso”.</p> <p>2. Alteração:</p> <p>2.1 O usuário seleciona a opção de alteração;</p> <p>2.2 O usuário informa:</p> <p>2.2.1 O código do fornecedor que deseja alterar;</p> <p>2.2.2 O CNPJ do fornecedor que deseja alterar;</p> <p>2.3 O sistema preenche os dados do fornecedor;</p> <p>2.4 O usuário altera os dados desejados;</p> <p>2.5 O usuário clica no botão/link “Salvar”;</p> <p>2.6 O sistema verifica os dados;</p> <p>2.7 O sistema grava as informações no banco de dados;</p> <p>2.8 O sistema retorna a mensagem “Dados alterados com sucesso”.</p> <p>3. Consulta:</p> <p>3.1 O usuário seleciona a opção de consulta;</p> <p>3.2 O usuário informa:</p> <p>3.2.1 O código do fornecedor que deseja consultar;</p> <p>3.2.2 O CNPJ do fornecedor que deseja consultar;</p> <p>3.3 O sistema preenche os dados do fornecedor.</p> <p>4. Exclusão:</p> <p>4.1 O usuário seleciona a opção de exclusão;</p> <p>4.2 O usuário informa:</p> <p>4.2.1 O código do fornecedor que deseja excluir;</p> <p>4.2.2 O CNPJ do fornecedor que deseja excluir;</p> <p>4.3 O sistema preenche os dados do fornecedor;</p> <p>4.4 O usuário clica no botão/link “Excluir”;</p> <p>4.5 O sistema verifica os dados;</p> <p>4.6 O sistema exclui as informações no banco de dados;</p> <p>4.7 O sistema retorna a mensagem “Dados excluídos com sucesso”.</p>
Tratamento de exceções:	<p>1.4.1 O sistema deve verificar se os campos obrigatórios foram preenchidos e retornar a mensagem “Preencha todos os campos obrigatórios” caso algum não tenha sido preenchido;</p> <p>2.2.1.1 Se o código informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p>

	<p>2.2.1.2 Se o CNPJ informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p> <p>2.6.1 Recebe o mesmo tratamento de exceção do item 1.4.1;</p> <p>3.2.1.1 Recebe mesmo tratamento de exceção do item 2.2.1.1;</p> <p>3.2.1.2 Recebe mesmo tratamento de exceção do item 2.2.1.2;</p> <p>4.2.1.1 Recebe mesmo tratamento de exceção do item 2.2.1.1;</p> <p>4.2.1.2 Recebe mesmo tratamento de exceção do item 2.2.1.2;</p> <p>4.5.1 Recebe mesmo tratamento de exceção do item 1.4.1.</p>
Pós-condições:	<ol style="list-style-type: none"> 1. Um novo fornecedor está cadastrado; 2. Um fornecedor está alterado; 3. Um fornecedor está consultado; 4. Um fornecedor está excluído.

Quadro Anexo 17 – Caso de Uso Gerência de fornecedor.

Fonte: autor (2010).

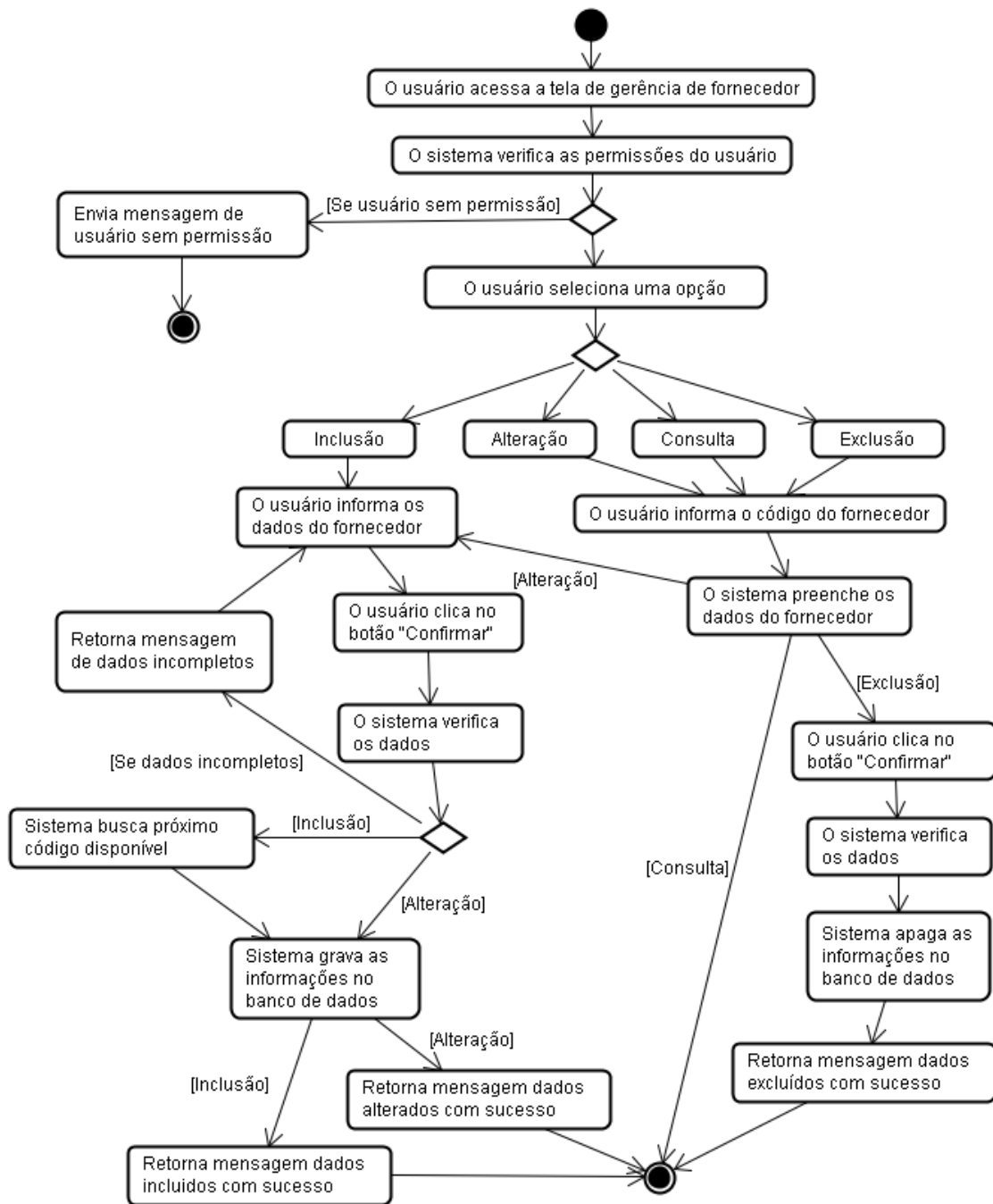


Figura Anexo 33 – Diagrama de Atividade Gerência de fornecedor.
 Fonte: autor (2010).

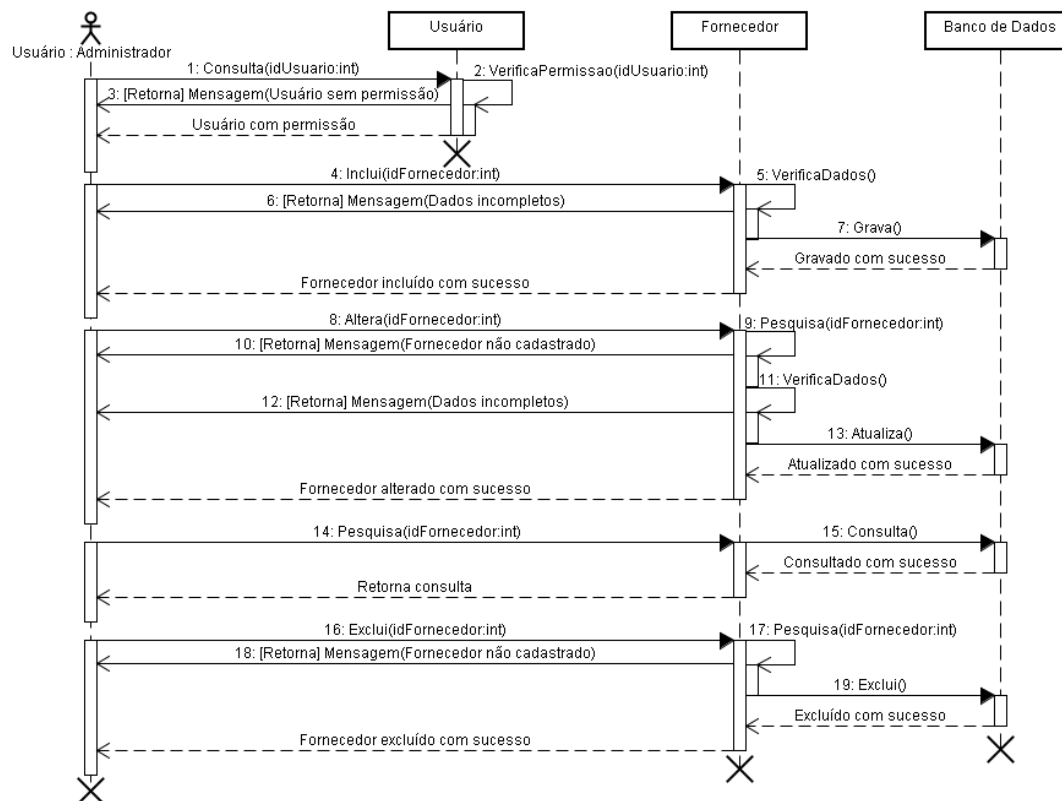


Figura Anexo 34 – Diagrama de Sequência Gerência de fornecedor.

Fonte: autor (2010).

UC020 – Emissão de ordem de compra

Breve descritivo: este caso de uso descreve o processo de gerenciamento de ordem de compra.

Requisitos funcionais:	REQF2 – [Cadastrar rede de lojas] REQF3 – [Cadastrar loja] REQF13 – [Cadastrar produto] REQF16 – [Cadastrar pessoa] REQF19 – [Cadastrar fornecedor] REQF20 – [Emitir ordem de compra] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções]
Atores:	Administrador.
Pré-condições:	Ser usuário com permissão de administrador.
Fluxo principal:	O usuário administrador acessa a tela de gerenciamento de fornecedor. Os subfluxos são apresentados pelo sistema: Inclusão, Alteração, Consulta e

	Exclusão.
Subfluxos:	<p>1. Inclusão:</p> <p>1.1 O usuário seleciona a opção de inclusão;</p> <p>1.2 O usuário informa dos dados de uma nova ordem de compra: rede, loja, fornecedor, data de entrega e produto, quantidade, preço e desconto;</p> <p>1.3 O usuário clica no botão/link “Salvar”;</p> <p>1.4 O sistema verifica os dados;</p> <p>1.5 O sistema preenche o código com o próximo disponível</p> <p>1.6 O sistema grava as informações no banco de dados;</p> <p>1.7 O sistema retorna a mensagem “Dados incluídos com sucesso”.</p> <p>2. Alteração:</p> <p>2.1 O usuário seleciona a opção de alteração;</p> <p>2.2 O usuário informa o código da rede, da loja e da ordem de compra que deseja alterar;</p> <p>2.3 O sistema preenche os dados da ordem de compra;</p> <p>2.4 O usuário altera os dados desejados;</p> <p>2.5 O usuário clica no botão/link “Salvar”;</p> <p>2.6 O sistema verifica os dados;</p> <p>2.7 O sistema grava as informações no banco de dados;</p> <p>2.8 O sistema retorna a mensagem “Dados alterados com sucesso”.</p> <p>3. Consulta:</p> <p>3.1 O usuário seleciona a opção de consulta;</p> <p>3.2 O usuário informa o código da rede, da loja e da ordem de compra que deseja consultar;</p> <p>3.3 O sistema preenche os dados da ordem de compra.</p> <p>4. Exclusão:</p> <p>4.1 O usuário seleciona a opção de exclusão;</p> <p>4.2 O usuário informa o código da rede, da loja e da ordem de compra que deseja excluir;</p> <p>4.3 O sistema preenche os dados da ordem de compra;</p> <p>4.4 O usuário clica no botão/link “Excluir”;</p> <p>4.5 O sistema verifica os dados;</p> <p>4.6 O sistema exclui as informações no banco de dados;</p> <p>4.7 O sistema retorna a mensagem “Dados excluídos com sucesso”.</p>
Tratamento de exceções:	<p>1.4.1 O sistema deve verificar se os campos obrigatórios foram preenchidos e retornar a mensagem “Preencha todos os campos obrigatórios” caso algum não tenha sido preenchido;</p> <p>2.2.1 Se o código informado não estiver registrado no banco de dados, o sistema retorna a mensagem “Código não cadastrado”;</p>

	<p>2.6.1 Recebe o mesmo tratamento de exceção do item 1.4.1;</p> <p>3.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.2.1 Recebe mesmo tratamento de exceção do item 2.2.1;</p> <p>4.5.1 Recebe mesmo tratamento de exceção do item 1.4.1.</p>
Pós-condições:	<p>1. Uma nova ordem de compra está cadastrada;</p> <p>2. Uma ordem de compra está alterada;</p> <p>3. Uma ordem de compra está consultada;</p> <p>4. Uma ordem de compra está excluída.</p>

Quadro Anexo 18 – Caso de Uso Emissão de ordem compra.
Fonte: autor (2010).

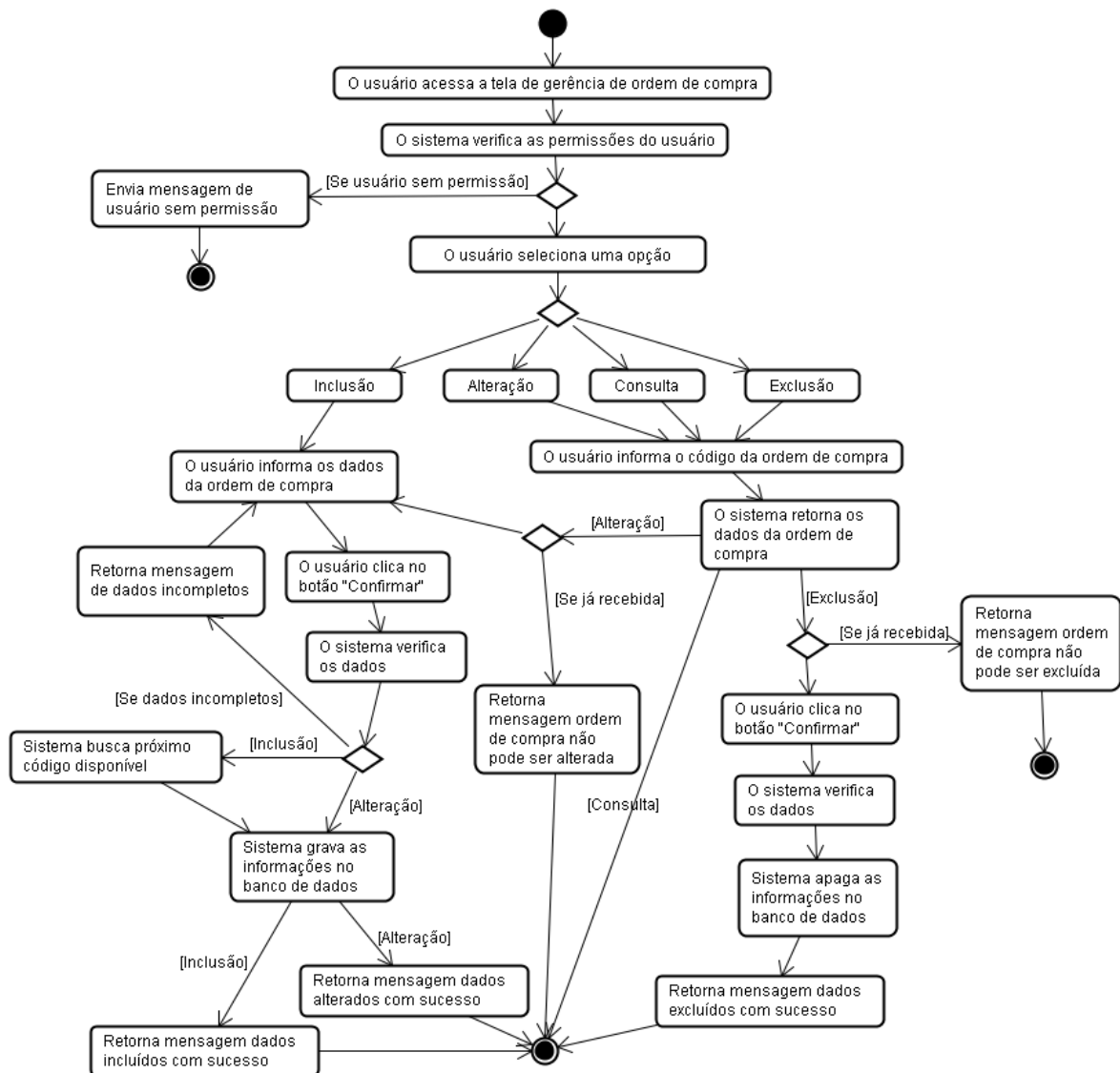


Figura Anexo 35 – Diagrama de Atividade Emissão de ordem de compra.
Fonte: autor (2010).

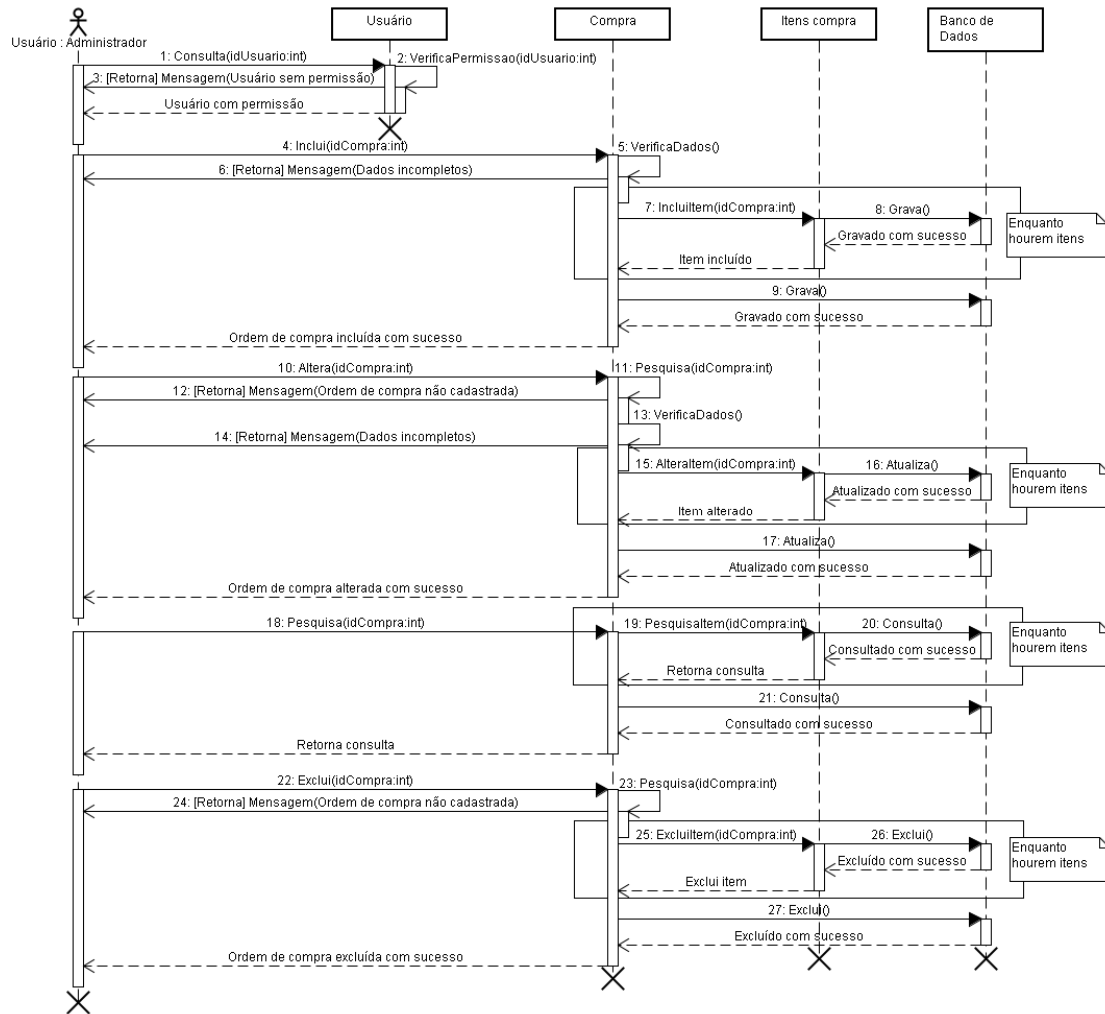


Figura Anexo 36 – Diagrama de Sequência Emissão de ordem de compra.

Fonte: autor (2010).

UC021 – Recebimento de produto

Breve descritivo: este caso de uso descreve o processo de gerenciamento de ordem de compra.

Requisitos funcionais:	REQF20 – [Emitir ordem de compra] REQF21 – [Receber produto] REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções]
Atores:	Administrador.
Pré-condições:	Ser usuário com permissão de administrador.
Fluxo principal:	O usuário administrador acessa a tela de gerenciamento de fornecedor. 1. O usuário informa os dados da ordem de compra: rede, loja e número de ordem

	<p>de compra;</p> <ol style="list-style-type: none"> 2. O sistema verifica os dados; 3. O sistema busca os dados da ordem de compra e preenche na tela; 4. O usuário informa o número da nota fiscal; 5. O sistema verifica os dados; 6. O usuário clica no botão/link “Confirmar recebimento” 7. O sistema grava as informações no banco de dados; 8. O sistema retorna a mensagem “Ordem de compra recebida com sucesso”;
Tratamento de exceções:	<p>1.2.1 O sistema deve verificar se a ordem de compra está cadastrada e retornar a mensagem “Ordem de compra não cadastrada”, caso não esteja cadastrada;</p> <p>1.2.2 O sistema deve verificar se a ordem de compra ainda não tenha sido recebida e retornar a mensagem “Ordem de compra já recebida” caso já tenha sido recebida;</p> <p>1.5.1 O sistema deve verificar se o número da nota fiscal ainda não consta no banco de dados e retornar a mensagem “Nota fiscal já cadastrada”, caso já cadastrada.</p>
Pós-condições:	A ordem de compra está recebida e os produtos constam no estoque.

Quadro Anexo 19 – Caso de Uso Recebimento de produto.

Fonte: autor (2010).

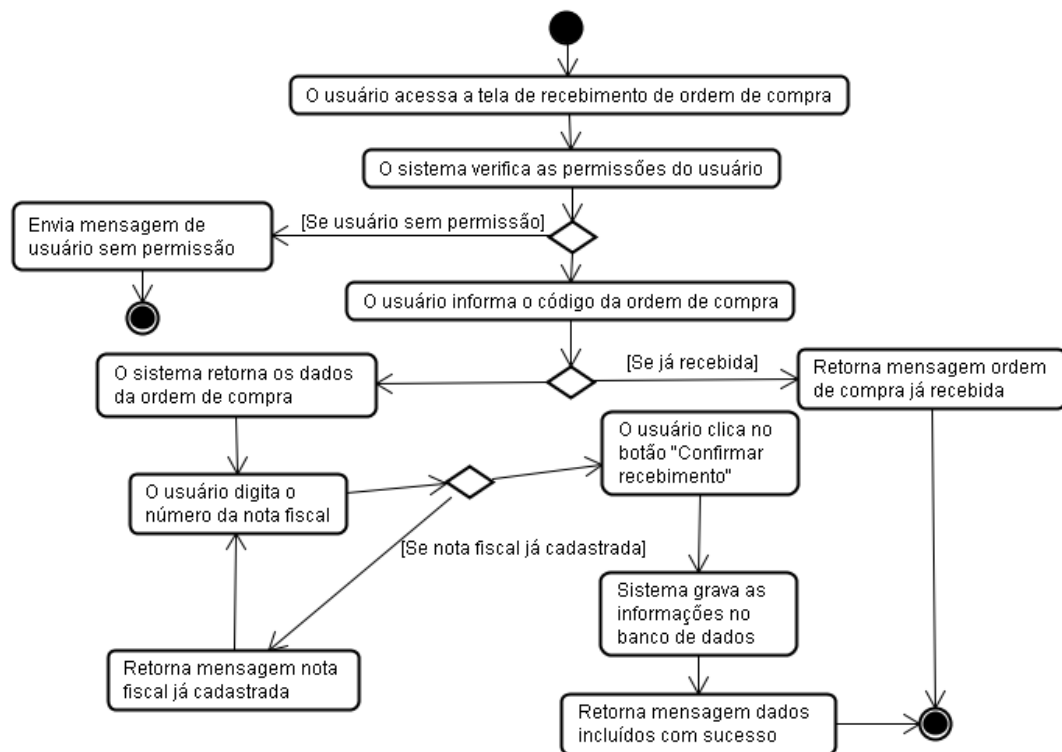


Figura Anexo 37 – Diagrama de Atividade Recebimento de produto.

Fonte: autor (2010).

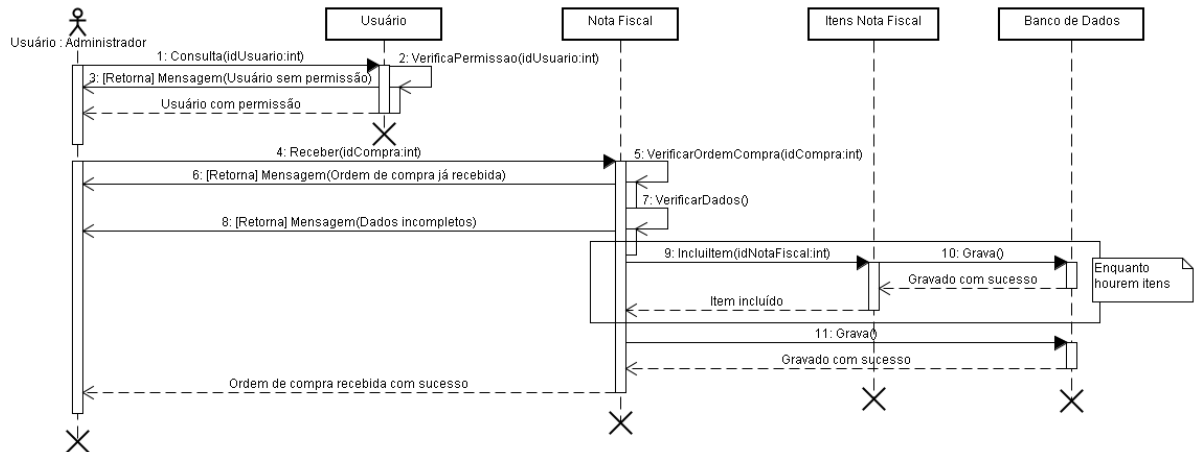


Figura Anexo 38 – Diagrama de Sequência Recebimento de produto.
Fonte: autor (2010).

UC024 – Emissão da leitura da memória da impressora fiscal

Breve descritivo: este caso de uso descreve o processamento de emissão da leitura da memória da impressora.

Requisitos funcionais:	REQF3 – [Cadastrar terminal] REQF24 – [Emitir Leitura da Memória da Impressora] REQF29 – [Controlar acesso ao sistema] REQF30 – [Comunicação com a impressora fiscal]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções] REQNF4 – [Conformidade com a legislação]
Atores:	Gerente, Caixa.
Pré-condições:	Ser usuário com permissão de gerente/caixa;
Fluxo principal:	1. O usuário gerente/caixa acessa a tela de emissão da memória da impressora fiscal; 2. O usuário seleciona o intervalo de data ou o intervalo de Redução Z; 3. O usuário clica no botão “Emitir impressão”; 4. O sistema verifica os dados; 5. O sistema imprime o relatório na impressora fiscal.
Tratamento de exceções:	4.1 O sistema deve verificar se o período digitado é válido. Se for um intervalo inválido, o sistema deve retornar a mensagem “Intervalo inválido”.
Pós-condições:	É impresso o relatório da leitura da memória da impressora fiscal.

Quadro Anexo 20 – Caso de Uso Emissão da leitura da memória da impressora fiscal.
Fonte: autor (2010).

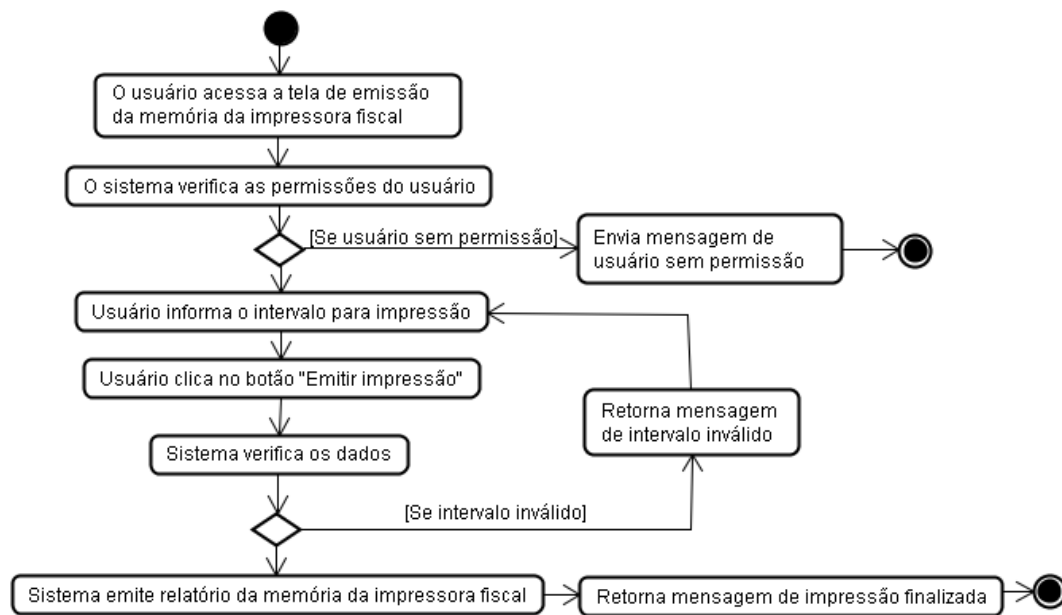


Figura Anexo 39 – Diagrama de Atividade Emissão da leitura da memória da impressora fiscal.

Fonte: autor (2010).

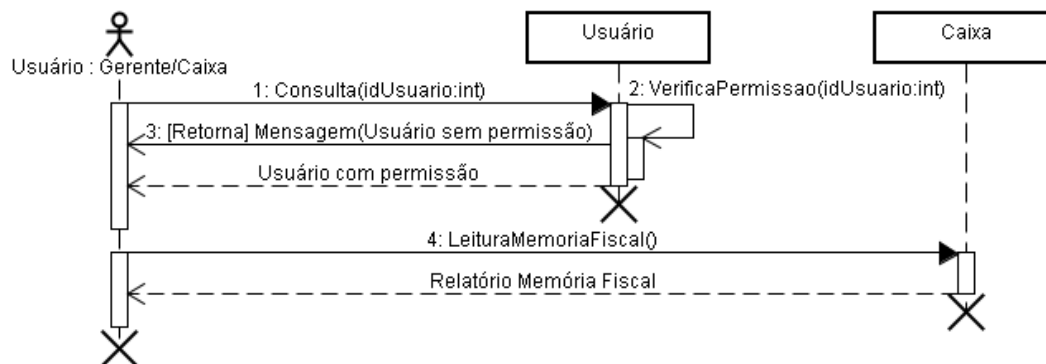


Figura Anexo 40 – Diagrama de Sequência Emissão da leitura da memória da impressora fiscal.

Fonte: autor (2010).

UC026 – Emissão de suprimento de caixa

Breve descritivo: este caso de uso descreve o processamento de emissão de suprimento de caixa (entrada de valor no caixa).

Requisitos funcionais:	REQF3 – [Cadastrar terminal] REQF15 – [Cadastrar forma de pagamento] REQF26 – [Emitir suprimento de caixa] REQF29 – [Controlar acesso ao sistema] REQF31 – [Comunicação com a impressora fiscal]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web]

	REQNF3 – [Links/ícones/botões para acesso as principais funções] REQNF4 – [Conformidade com a legislação]
Atores:	Gerente, Caixa.
Pré-condições:	Ser usuário com permissão de gerente/caixa; Caixa deve estar aberto.
Fluxo principal:	1.O usuário gerente/caixa acessa a tela de emissão de suprimento de caixa; 2.O usuário informa os dados do suprimento de caixa: forma de pagamento e valor; 3.O sistema verifica os dados; 4.O usuário clica no botão “Confirmar”; 5.O sistema imprime o relatório de suprimento na impressora do cupom fiscal; 6.O sistema grava as informações no banco de dados.
Tratamento de exceções:	3.1 O sistema deve verificar se foi selecionada a forma de pagamento. Se não foi selecionada, o sistema deve retornar a mensagem “Selecione o forma de pagamento”; 3.2 O sistema deve verificar se o valor informa é maior que zero. Se não for, o sistema deve retornar a mensagem “Valor inválido”.
Pós-condições:	Um suprimento de caixa é emitido.

Quadro Anexo 21 – Caso de Uso Emissão de suprimento de caixa.

Fonte: autor (2010).

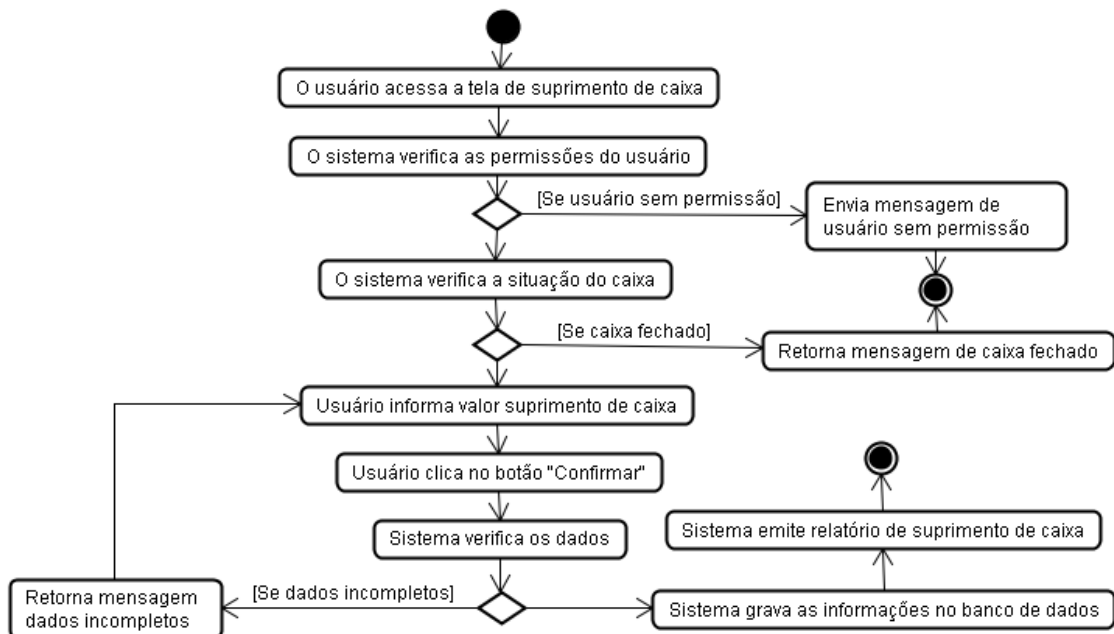


Figura Anexo 41 – Diagrama de Atividade Emissão de suprimento de caixa.

Fonte: autor (2010).

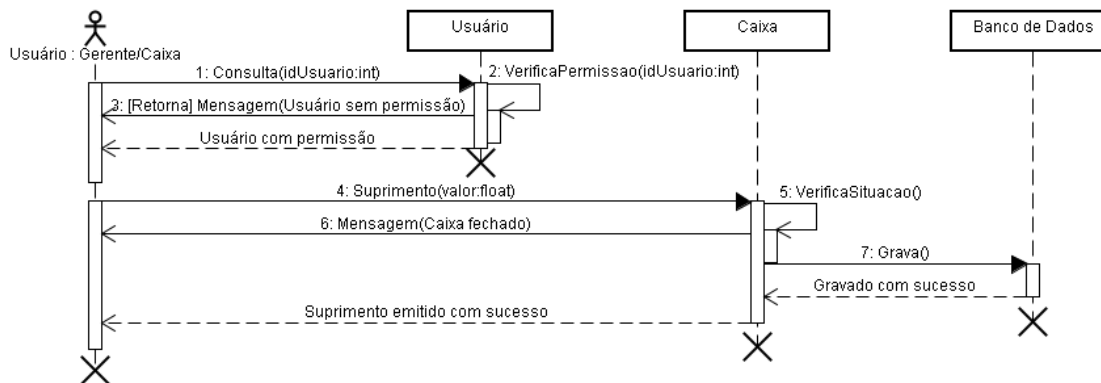


Figura Anexo 42 – Diagrama de Sequência Emissão de suprimento de caixa.

Fonte: autor (2010).

UC027 – Emissão de sangria de caixa

Breve descritivo: este caso de uso descreve o processamento de emissão de sangria de caixa (saída de valor no caixa).

Requisitos funcionais:	REQF3 – [Cadastrar terminal] REQF15 – [Cadastrar forma de pagamento] REQF27 – [Emitir sangria de caixa] REQF29 – [Controlar acesso ao sistema] REQF30 – [Comunicação com a impressora fiscal]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções] REQNF4 – [Conformidade com a legislação]
Atores:	Gerente, Caixa.
Pré-condições:	Ser usuário com permissão de gerente/caixa; Caixa deve estar aberto.
Fluxo principal:	1. O usuário gerente/caixa acessa a tela de emissão de sangria de caixa; 2. O usuário informa os dados da sangria de caixa: forma de pagamento e valor; 3. O sistema verifica os dados; 4. O usuário clica no botão “Confirmar”; 5. O sistema imprime o relatório de sangria na impressora do cupom fiscal; 6. O sistema grava as informações no banco de dados.
Tratamento de exceções:	3.1 O sistema deve verificar se foi selecionada a forma de pagamento. Se não foi selecionada, o sistema deve retornar a mensagem “Selecione o forma de pagamento”. 3.2 O sistema deve verificar se o valor informado é maior que zero. Se não for, o sistema deve retornar a mensagem “Valor inválido”.

Pós-condições:	Uma sangria de caixa é emitida.
----------------	---------------------------------

Quadro Anexo 22 – Caso de Uso Emissão de sangria de caixa.

Fonte: autor (2010).

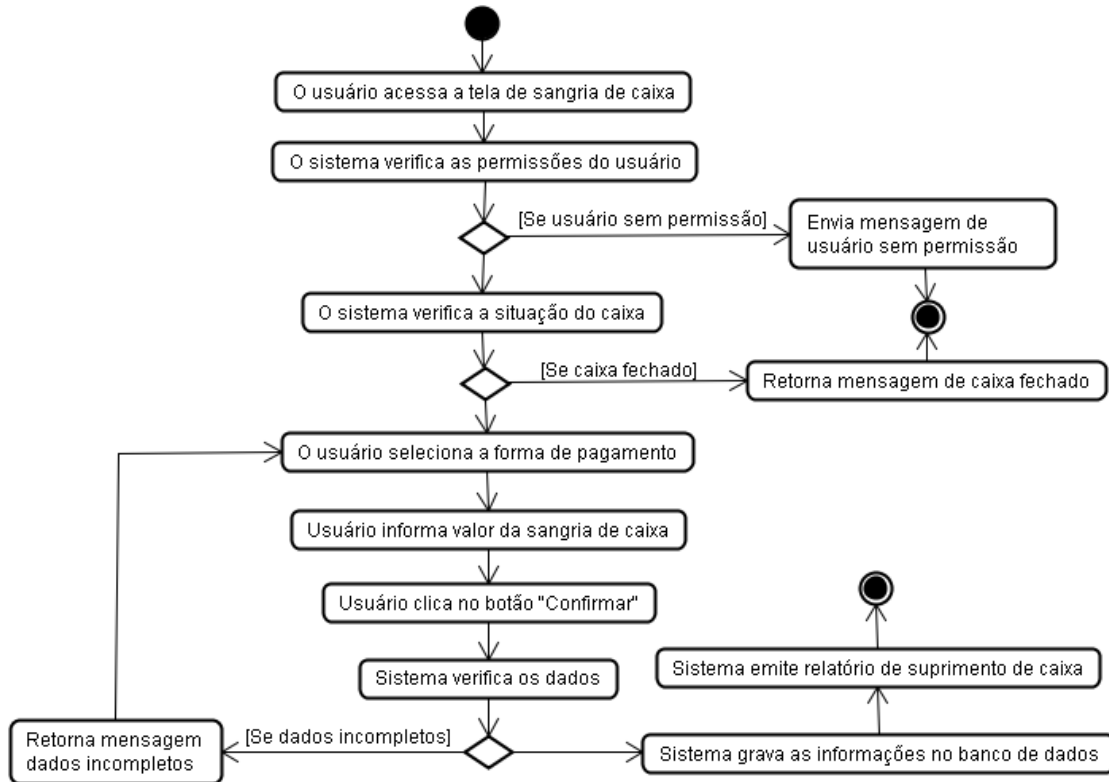


Figura Anexo 43 – Diagrama de Atividade Emissão de sangria de caixa.

Fonte: autor (2010).

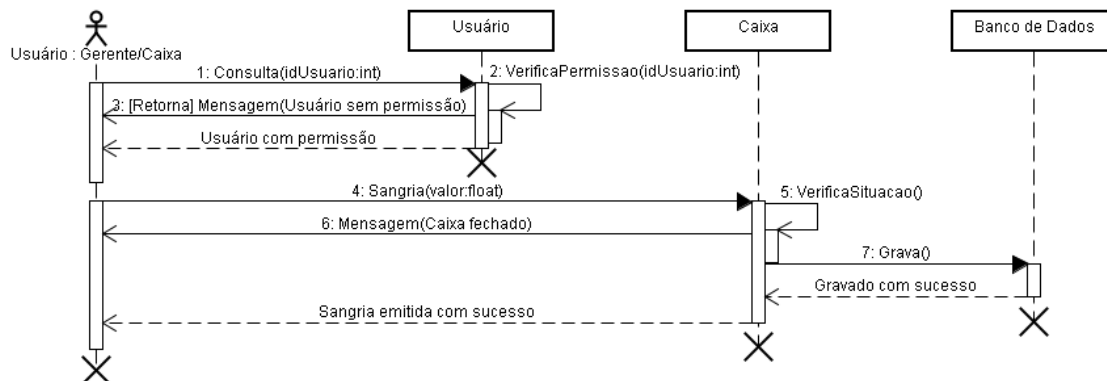


Figura Anexo 44 – Diagrama de Sequência Emissão de sangria de caixa.

Fonte: autor (2010).

UC028 – Emissão de relatórios

Breve descritivo: este caso de uso descreve o processamento de emissão de relatórios.

Requisitos funcionais:	REQF29 – [Controlar acesso ao sistema]
Requisitos não funcionais:	REQNF1 – [Utilização de banco de dados relacional] REQNF2 – [Utilização de servidor web] REQNF3 – [Links/ícones/botões para acesso as principais funções]
Atores:	Administrador, Caixa.
Pré-condições:	Ser usuário com permissão de administrador/gerente.
Fluxo principal:	O usuário administrador/gerente acessa a tela de gerenciamento de fornecedor. Os subfluxos são apresentados pelo sistema. 1. Compras 2. Vendas 3. Estoque
Subfluxos:	1. Compras 1.1 O usuário seleciona os dados para o filtro do relatório: período de compra, fornecedor, divisão, marca, modelo, material, cor e tamanho; 1.2 O usuário clica no botão “Gerar relatório”; 1.3 O sistema busca as informações no banco de dados; 1.4 O sistema lista os dados na tela. 2. Vendas 2.1 O usuário seleciona os dados para o filtro do relatório: período de venda, vendedor, consumidor, divisão, marca, modelo, material, cor e tamanho; 2.2 O usuário clica no botão “Gerar relatório”; 2.3 O sistema busca as informações no banco de dados; 2.4 O sistema lista os dados na tela. 3. Estoque 3.1 O usuário seleciona os dados para o filtro do relatório: divisão, marca, modelo, material, cor e tamanho; 3.2 O usuário clica no botão “Gerar relatório”; 3.3 O sistema busca as informações no banco de dados; 3.4 O sistema lista os dados na tela.
Tratamento de exceções:	1.3.1 O sistema deve verificar se existem informações no banco de dados de acordo com os filtros selecionados. Se não houver, o sistema deve retornar a mensagem “Não há dados para os filtros selecionados”; 2.3.1 Recebe mesmo tratamento de exceção do item 1.3.1; 3.3.1 Recebe mesmo tratamento de exceção do item 1.3.1;
Pós-condições:	Um relatório gerencial é emitido.

Quadro Anexo 23 – Caso de Uso Emissão de relatórios.
Fonte: autor (2010).

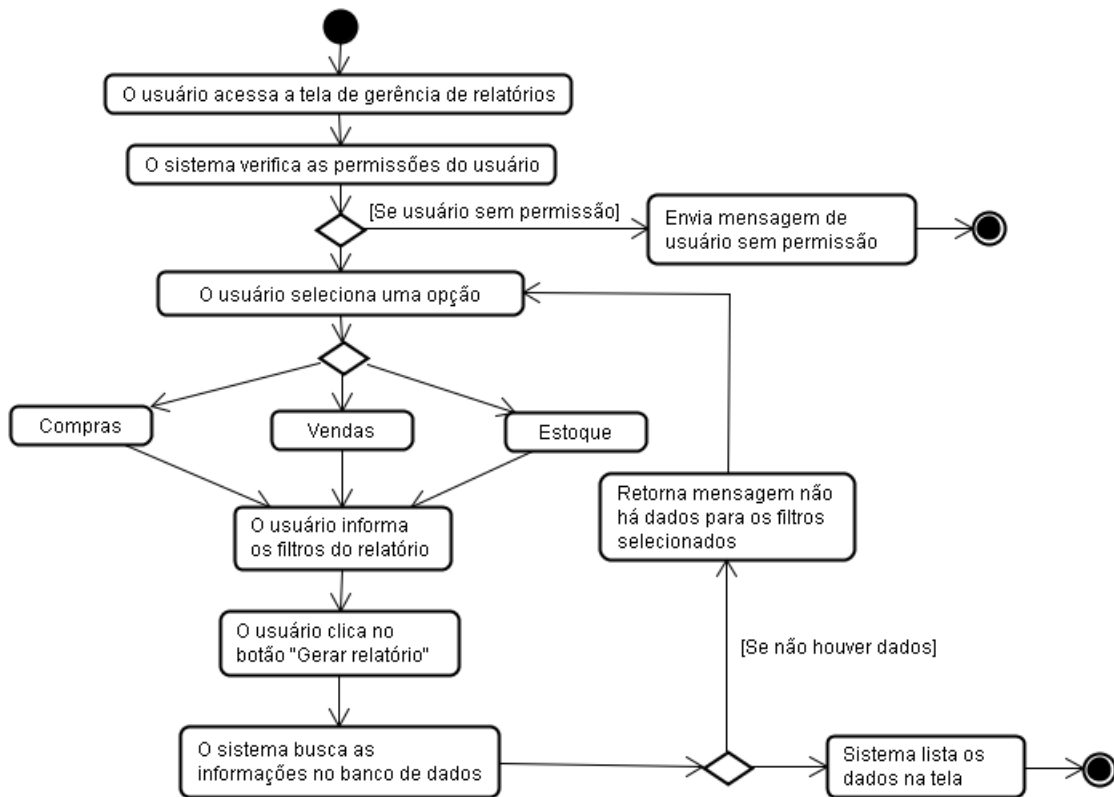


Figura Anexo 45 – Diagrama de Atividade Emissão de relatórios.
 Fonte: autor (2010).

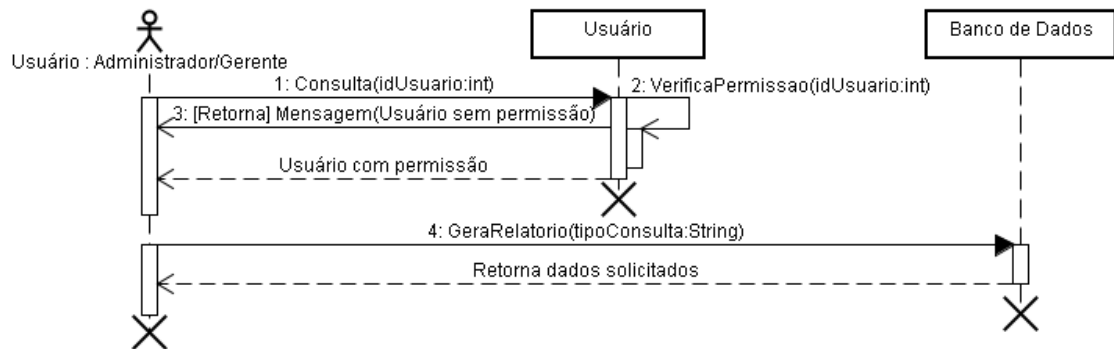


Figura Anexo 46 – Diagrama de Sequência Emissão de relatórios.
 Fonte: autor (2010).

Tabela Anexo 1 – Questionário de validação do sistema.

Questionário de validação do sistema

Data: _____

Assinale abaixo, com relação ao sistema atual, para cada uma das questões, como você avalia o sistema apresentado, considerando que a escala de pontuação negativa julga que o sistema é pior e a escala de pontuação positiva considera melhoria.

1. O sistema lhe é amigável?

-5	-4	-3	-2	-1	0	1	2	3	4	5

2. As funções do sistema são apresentadas de forma clara?

-5	-4	-3	-2	-1	0	1	2	3	4	5

3. O sistema é fácil de operar?

-5	-4	-3	-2	-1	0	1	2	3	4	5

4. A sua interação com o sistema é clara e entendível?

-5	-4	-3	-2	-1	0	1	2	3	4	5

5. O tempo de resposta com a impressora fiscal é eficiente?

-5	-4	-3	-2	-1	0	1	2	3	4	5

6. O ritmo de operação é satisfatório?

-5	-4	-3	-2	-1	0	1	2	3	4	5

7. As informações apresentadas nos relatórios são compreensíveis?

-5	-4	-3	-2	-1	0	1	2	3	4	5

8. O sistema é eficaz, no sentido de atender os objetivos que se propõe?

-5	-4	-3	-2	-1	0	1	2	3	4	5