



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

CARLOS EDUARDO GARCIA DE OLIVEIRA

SISTEMA DE GERENCIAMENTO DE LOJAS

Assis
2011

CARLOS EDUARDO GARCIA DE OLIVEIRA

SISTEMA DE GERENCIAMENTO DE LOJAS

Trabalho de Conclusão
de Curso apresentado ao Instituto
Municipal de Ensino Superior de
Assis como requisito para obtenção do grau de tecnólogo
em Processamento de Dados

Orientador: Dr. Almir Rogério Camolesi.

Área de Concentração: Desenvolvimento de Sistemas

Assis
2011

FICHA CATALOGRÁFICA

OLIVEIRA, Carlos Eduardo Garcia

Sistema de Gerenciamento de Lojas/ Carlos Eduardo Garcia de Oliveira. Fundação Educacional do Município de Assis, 2011.

46 p.

Orientador: Dr. Almir Rogério Camolesi

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis – IMESA.

CDD: 001.61
Biblioteca da Fema

SISTEMA DE GERENCIAMENTO DE LOJAS

CARLOS EDUARDO GARCIA DE OLIVEIRA

Trabalho de Conclusão
de Curso apresentado ao Instituto
Municipal de Ensino Superior de
Assis como requisito para obtenção do grau de tecnólogo
em Processamento de Dados

Orientador: Dr. Almir Rogério Camolesi

Analisador: Dr. Alex Sandro Romeo De Souza Poletto

Assis
2011

DEDICATÓRIA

*Dedico este trabalho a Deus,
a meu pai Otail, minha mãe Sonia,
meus irmãos Marina e Theodoro,
aos meus amigos de sala de aula e professores.*

AGRADECIMENTOS

Agradeço primeiramente a Deus por me dar força para cumprir mais uma etapa de minha vida e por dar coragem para enfrentar todos os desafios que foram impostos ao longo desse caminho.

A todos os professores, em especial ao Prof. Almir Rogério Camolesi, pela orientação e estímulo dado a mim durante todo o trabalho.

A todos os meus familiares, meus pais Otail e Sonia pelo amor e incentivo aos estudos, aos meus irmãos Marina e Theodoro pela ajuda e apoio durante este caminho, e as minhas avós Rosalina e Rita que sempre lembraram de mim em suas orações.

Aos meus amigos de classe, Cláudio, Jefferson B., João, Jefferson F., Luis Fernando, Luiz Ângelo, Marcus Keppi, entre outros, que estiveram comigo todos esses anos, enfrentando as mesmas dificuldades e ajudando direta ou indiretamente a superá-las. Agradeço também ao meu amigo Mateus, que muito tem me ajudado nesta conclusão da minha vida acadêmica como no início da minha vida profissional na área de programação Web.

Enfim, agradeço a todos que me ajudaram diretamente ou indiretamente para o desenvolvimento desse trabalho. Muito Obrigado.

RESUMO

Este trabalho tem como objetivo principal apresentar o desenvolvimento de um sistema de gerenciamento de loja. O sistema proposto tem por finalidade controlar entrada e saída de produtos, controle de vendas, controle de usuários, e emissão de diversos relatórios gerenciais. O mesmo deve facilitar a administração da loja, tornando os processos mais ágeis e eficazes e trazendo maiores benefícios ao proprietário. Para a fase de análise foi utilizada a metodologia orientada a objetos juntamente com a linguagem de modelagem UML, sendo que os diagramas foram desenvolvidos na Ferramenta Visual Paradigma. Para o desenvolvimento foram utilizadas as seguintes ferramentas: *Dreamweaver* e - para a programação - a linguagem *ColdFusion*. O *Microsoft SQL Server* foi utilizado para criação e organização do banco de dados.

Palavras Chaves: Metodologia orientada a objetos, *ColdFusion*, *Dreamweaver*, *SQL Server*, UML

ABSTRACT

This work aims present the development of a Store management system. The proposed system aims to control entry and exit of products, sales control, users control, and issuance of various management reports. The same should facilitate the management store, making processes more agile and efficient, and bringing greatest benefits to the owner. For the analysis phase was used the object oriented methodology along with the modeling language UML and the diagrams were developed using Visual Paradigm Tool. For the development phase was used the following tools: Dreamweaver and – for do the programming - ColdFusion language. Microsoft SQL Server was used in oder to create and organize the database.

Keywords: Object Oriented Methodology, ColdFusion, Dreamweaver, SQL Server, UML.

LISTA DE FIGURAS

Figura 1 - Funcionamento do <i>ColdFusion</i> no browser.....	21
Figura 2 – Caso de Uso Geral – Movimentação.....	27
Figura 3 – Caso de Uso Geral – Emitir Relatórios.....	28
Figura 4 – UC – Autenticar Usuário.....	29
Figura 5 – UC – Gerenciar Usuários	30
Figura 6 – UC – Gerenciar Vendas	31
Figura 7 – UC – Gerenciar Cliente	32
Figura 8 – UC – Gerenciar Vendedores	33
Figura 9 – UC – Gerenciar Produtos	34
Figura 10 – UC – Gerenciar Consignação.....	35
Figura 11 – UC – Gerenciar Contas a Receber.....	36
Figura 12 – UC – Gerenciar Contas a Receber.....	37
Figura 13 – UC – Emitir Relatório de Vendas.....	38
Figura 14 – UC – Emitir Relatório de Contas a Pagar.....	39
Figura 15 – UC – Emitir Relatório de Contas a Receber.....	40
Figura 16 – UC – Emitir Relatório de Clientes.....	41
Figura 17 – UC – Emitir Relatório de Usuário	42
Figura 18 – UC – Emitir Relatório de Funcionários	43
Figura 19 – UC – Emitir Relatório de Produtos	44
Figura 20 – UC – Emitir Relatório de Consignação.....	45
Figura 21 - Diagrama de Atividades – Cadastros.....	46
Figura 22 - Diagrama de Atividades – Movimentação.....	47
Figura 23 - Diagrama de Atividades – Relatórios	48
Figura 24 - Diagrama de Classes.....	49
Figura 25 - Diagrama de Atividades – Efetuar Vendas	50
Figura 26 – MER - Geral	51
Figura 27 - WBS (Work Breakdown Structure).....	52
Figura 28 – Organização do Sistema em Camadas.....	55
Figura 29 – <i>Login</i> do Usuário.	57
Figura 30 – Menu Principal.....	58

Figura 31 – Cadastro Cliente.....	59
Figura 32 – Cadastro Cliente.....	60
Figura 33 – Função em JavaScript.....	61
Figura 34 – Div Pessoa Física.....	62
Figura 35 – Div Pessoa Jurídica.....	62
Figura 36 – Exemplos de Cadastro - Tipo Pessoa	63

LISTA DE QUADROS

Quadro 1 - Lista de Eventos.....	26
Quadro 2 – Especificação Do Caso De Uso 01.....	29
Quadro 3 – Especificação do Caso de Uso 02.....	30
Quadro 4 – Especificação do Caso de Uso 03.....	31
Quadro 5 – Especificação do Caso de Uso 04.....	32
Quadro 6 – Especificação do Caso de Uso 05.....	33
Quadro 7 – Especificação Do Caso De Uso 06.....	34
Quadro 8 – Especificação Do Caso De Uso 07.....	35
Quadro 9 – Especificação Do Caso De Uso 08.....	36
Quadro 10 – Especificação Do Caso De Uso 09.....	37
Quadro 11 – Especificação Do Caso De Uso 10.....	38
Quadro 12 – Especificação Do Caso De Uso 11.....	39
Quadro 13 – Especificação Do Caso De Uso 12.....	40
Quadro 14 – Especificação Do Caso De Uso 13.....	41
Quadro 16 – Especificação Do Caso De Uso 15.....	43
Quadro 17 – Especificação Do Caso De Uso 16.....	44
Quadro 18 – Especificação Do Caso De Uso 17.....	45

SUMÁRIO

1 – INTRODUÇÃO	15
2 - FERRAMENTAS DE DESENVOLVIMENTO E ANÁLISE	17
2.1 – COLDFUSION.....	17
2.1.1 – A arquitetura ColdFusion.....	17
2.1.2 – ColdFusion Application Server	18
2.1.3 – Modelos da <i>ColdFusion Markup Language</i> (CFML)	18
2.1.4 – Servidor da <i>web</i>	19
2.1.5 – ColdFusion Administrator	19
2.1.6 – Fontes de Dados	19
2.1.7 – Objetos	20
2.1.8 – Extensões.....	20
2.2 - UML.....	22
2.3 - DREAMWEAVER.....	22
2.4 - MICROSOFT SQL SERVER 2008.....	22
3 – ANÁLISE E ESPECIFICAÇÃO DO SISTEMA.....	23
3.1 – LEVANTAMENTO DE REQUISITOS	23
3.2 - FUNCIONALIDADES DO SISTEMA	24
3.2.1 - Restrições de desenvolvimento.....	25
3.2.2 - Lista de eventos	25
3.3 - VALIDAÇÃO DOS REQUISITOS	26
3.4 - DIAGRAMA DE CASO DE USO GERAL	27
3.4.1 - Movimentação.....	27
3.4.2 - Emissão de Relatório	28
3.5 - DIAGRAMAS DE CASO DE USO ESPECÍFICO	29
3.5.1 - UC 01 – Autenticar Usuário.....	29

3.5.2 - UC 02 – Gerenciar Usuário	30
3.5.3 - UC 03 – Gerenciar Vendas	31
3.5.4 - UC 04 – Gerenciar Cliente	32
3.5.5 - UC 04 – Gerenciar Vendedores	33
3.5.6 - UC 06 – Gerenciar Produtos	34
3.5.7 - UC 07 – Gerenciar Consignação.....	35
3.5.8 - UC 08 – Gerenciar Contas a Receber.....	36
3.5.9 - UC 09 – Gerenciar Contas a Pagar.....	37
3.5.10 - UC 10 – Emitir Relatório de Vendas.....	38
3.5.11 - UC 11 – Emitir Relatório de Contas a Pagar.....	39
3.5.12 - UC 12 – Emitir Relatório de Contas a Receber.....	40
3.5.13 - UC 13 – Emitir Relatório de Clientes.....	41
3.5.14 - UC 14 – Emitir Relatório de Usuário	42
3.5.15 - UC 15 – Emitir Relatório de Funcionários	43
3.5.16 - UC 16 – Emitir Relatório de Produtos	44
3.5.17 - UC 17 – Emitir Relatório de Consignação.....	45
3.6 - DIAGRAMA DE ATIVIDADES	46
3.6.1 - Cadastros	46
3.6.2 - Movimentações	47
3.6.3 Relatórios	48
3.7 – DIAGRAMA DE CLASSE	49
3.8 – DIAGRAMA DE SEQUENCIA	50
3.9 – MODELO DE ENTIDADE RELACIONAMENTO (MER).....	51
4 – PROJETO	52
4.1 - ESTRUTURA ANALÍTICA DO PROJETO (WBS)	52
4.2 - ESPECIFICAÇÃO DOS CUSTOS.....	53
4.2.1 - Recursos necessários para o desenvolvimento	53
4.2.2 - Estimativa de custos	53

4.2.3 - Custos materiais.....	53
5 – IMPLEMENTAÇÃO DO SISTEMA	55
5.1 - ORGANIZAÇÃO DE PACOTE E CLASSES DO SISTEMA	55
5.2 – INTERFACE DO SISTEMA	56
6 – CONCLUSÃO.....	64
6.1 - CRONOGRAMA.....	65
REFERÊNCIAS.....	66

1 – INTRODUÇÃO

Todas as organizações trabalham baseadas em informações que dão sustentação para a tomada de decisões, porém nem todas possuem um sistema de informação bem estruturado. O sistema de informação é a organização das informações da empresa de forma que elas sejam entendidas e utilizadas pelos clientes internos (funcionários e diretores) em prol do desenvolvimento da organização. Tal sistema é baseado em dados e informações da empresa que são transformados em relatórios que auxiliarão na tomada de decisões. Entende-se que os dados são as matérias-primas de um sistema de informação, pois são eles que serão processados e transformados em algo útil para a organização.

“Um sistema de informações processa dados (*input*) e transforma-os em relatórios (*output*). Esses relatórios são as informações destinadas as pessoas que tomam as decisões (usuários)”. (Magalhães e Lunkes, 2000, p. 26)

Stelise Modas é uma loja que está no mercado há vinte e cinco anos. Localizada em Sertanópolis, sua proprietária tem a necessidade de informatizá-la pois não há software para gestão de informações em seu estabelecimento. A criação do sistema de gerenciamento foi baseada nas dificuldades que a loja encontra no seu dia a dia. Um dos principais problemas é o fato das movimentações serem feitas manualmente, como por exemplo, o controle de estoque. Sendo assim, os resultados não são os esperados pela proprietária por razão do controle precário. O sistema de gerenciamento irá ajudar a loja a ter um melhor controle das informações de seus dados, como contas a pagar e receber, cadastro de funcionários, clientes, fornecedores e estoque.

O objetivo do sistema é manter a integridade das informações da empresa, diminuir os custos operacionais e evitar possíveis erros e falhas humanas. Além de manter uma melhor organização das informações referentes à loja, visando auxiliar de uma forma simples e eficiente, gerente e funcionários que utilizarão o sistema.

A proprietária requer um software que torne possível fazer movimentações de venda, cadastro de clientes e controle de estoque de uma forma bem simplificada. Por este motivo surgiu a oportunidade de desenvolver um sistema de gerenciamento de loja via Web. Além de informatizar a Loja, o sistema irá garantir a possibilidade de

ser gerenciado pelo proprietário ou administrador de qualquer local onde haja acesso à Internet.

A análise do sistema de Loja será realizada com base nos conceitos de Análise Orientada à Objetos. Para a modelagem dos diagramas será utilizada a linguagem *Unified Modeling Language* (UML).

O sistema será codificado utilizando a ferramenta Adobe Dreamweaver, para armazenar os dados será utilizado software Microsoft SQL Server 2008, uma ferramenta confiável e robusta. Para uma melhor compreensão, o trabalho foi organizado em outros cinco capítulos além da presente introdução.

No capítulo 2 são apresentadas as tecnologias utilizadas para o desenvolvimento e análise do sistema com uma breve descrição de cada uma. No capítulo 3, serão detalhadas as etapas de Análise e Especificações do Sistema contemplando o Levantamento de Requisitos, Lista de Eventos, o Caso de Uso e suas especificações e Diagramas (Atividades, Classe, Sequência e Entidade Relacionamento). O capítulo 4 mostra a *Work BreakDown Structure (WBS)* e o Orçamento do Sistema. O quinto capítulo apresenta a implementação do sistema com as divisões de pastas, mostrando a interface do software. No sexto e último capítulo serão apresentadas as considerações finais.

2 - FERRAMENTAS DE DESENVOLVIMENTO E ANÁLISE

Neste capítulo serão apresentadas as ferramentas que foram utilizadas para o desenvolvimento do sistema.

2.1 – COLDFUSION

É uma ferramenta de desenvolvimento de aplicações *Web* que permite criar páginas *web* dinâmicas para Internet, Intranets ou Extranets através de integração sofisticada entre elementos como: banco de dados, ambiente *Web* e aplicações de e-mail - além de permitir aplicações Java.

Isto significa que uma vez instalado você terá acesso a sua base de dados, por meio da linguagem SQL de forma dinâmica, sem a necessidade de se criar várias páginas para cada item de pesquisa. Permite ainda a criação de formulários de envio de e-mails, agendas e calendários, etc.

O *ColdFusion* utiliza-se de uma linguagem denominada CFML (*ColdFusion Markup Language*). A CFML é baseada em *tags*, como o HTML, tornando-se uma linguagem de fácil aprendizado (www.adobe.com/devnet/coldfusion.html).

2.1.1 – A arquitetura ColdFusion

Existem diversos componentes que formam o ambiente do *ColdFusion* e com os quais você pode desenvolver os seus aplicativos. Um aplicativo do *ColdFusion* é apenas uma coleção de modelos (páginas da *web*) que funcionam juntos permitindo que um usuário execute uma tarefa. Para ter uma ideia melhor de como uma

aplicativo é criado, você precisa entender os componentes que formam o ambiente do *ColdFusion* (BROOKS-BILSON, 2001).

2.1.2 – ColdFusion Application Server

O *ColdFusion Application Server* processa todo o código CFML dos modelos que o servidor *web* passa para ele. Em seguida, ele retorna os dados gerados dinamicamente para o servidor de *web* para que a saída seja enviada ao *browser* do usuário. O *ColdFusion Application Server* se integra a diversos servidores conhecidos para *web* por meio das APIs nativas e também pode ser executado no modo CGI. Após a configuração do *ColdFusion Application Server*, ele funciona silenciosamente em segundo plano (BROOKS-BILSON, 2001).

2.1.3 – Modelos da *ColdFusion Markup Language* (CFML)

A *ColdFusion Markup Language* (CFML) é a linguagem usada para criar os aplicativos do *ColdFusion*, baseada em *tags*, assim como a HTML. Seu uso é em conjunto com a HTML e com outras linguagens no lado do cliente, tais como *JavaScript* e ao *VBScript*, para criar os modelos que formam um aplicativo *ColdFusion*. Além disso, ela é usada para determinar o que será exibido, enquanto o HTML especifica como aquilo será exibida - essa é uma distinção importante (BROOKS-BILSON, 2001).

2.1.4 – Servidor da *web*

O servidor da *web* canaliza as solicitações de modelos da CFML que os browsers fazem usando o *ColdFusion Application Server*. O servidor da *web* também é responsável por passar a saída retornada pelo *ColdFusion Application Server* de volta para o browser (BROOKS-BILSON, 2001).

2.1.5 – *ColdFusion* Administrator

O *ColdFusion Administrator*, na verdade, é um aplicativo para configurar e administrar o *ColdFusion Application Server*. O *ColdFusion Administrator* trata de tudo desde o registro e a configuração da fontes de dados até o *login* e a segurança (BROOKS-BILSON, 2001).

2.1.6 – Fontes de Dados

O *ColdFusion* pode interagir com diversas fontes de dados externas, incluindo banco de dados (por meio da ODBC, do OLE-DB e dos drivers nativos), as coleções *Verity*, os diretórios LDAP, os servidores de correio POP3 e SMTP, os servidores de FTP e outros servidores de HTTP. Isso permite que sejam criados aplicativos do *ColdFusion* que enviam e recebem *e-mail*, transferem arquivos com o FTP, consultam servidores de diretórios e solicitam conteúdo de outros servidores da *Web* (BROOKS-BILSON, 2001).

2.1.7 – Objetos

O *ColdFusion* pode interagir com diversos objetos externos, incluindo os objetos COM/DCOM, CORBA e Java (entre eles os componentes *Enterprise JavaBeans*). Isso permite que seus aplicativos *ColdFusion* interajam com componentes de terceiros bem como com sistemas de *back-end* escritos em outras linguagens, o que torna o *ColdFusion* uma excelente linguagem para agir como uma “cola” que une todo tipo de sistemas diferentes em um aplicativo coeso (BROOKS-BILSON, 2001).

2.1.8 – Extensões

A CFML pode ser estendida por meio das extensões personalizadas escritas em diversas linguagens, tais como C++, Java, Delphi e até mesmo CFML. Ao escrever uma extensão pode-se incluir funcionalidade, que esta disponível nativamente na linguagem de núcleo do *ColdFusion*. Por exemplo, o *ColdFusion* não pode se conectar nativamente a um servidor de notícias NNTP, mas se o programador tiver habilidades em C++, Java ou Delphi, poderá criar facilmente uma extensão CFX para dar suporte a tal conexão (BROOKS-BILSON, 2001).

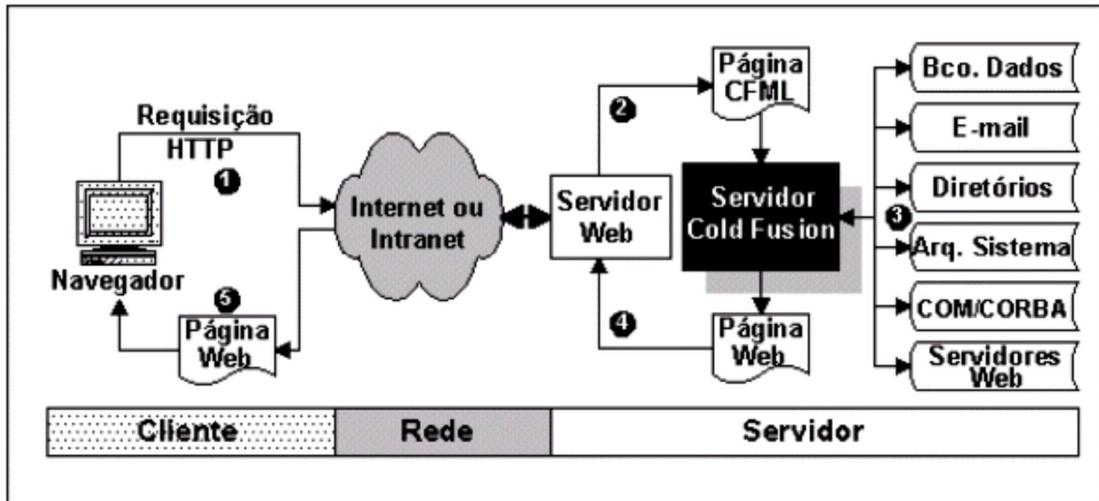


Figura 1 - Funcionamento do *ColdFusion* no browser

A seguir serão descritos os passos realizados durante o funcionamento do ColdFusion no browser, conforme apresentado na Figura 1.

Passo 1: Quando um usuário seleciona o botão “Submit” (Enviar) em um formulário ou um link em uma página, o navegador envia uma requisição HTTP para o servidor Web via Internet ou Intranet.

Passo 2: O servidor Web passa os dados submetidos pelo cliente e a página apropriada para o Servidor *ColdFusion* através da API.

Passo 3: O *ColdFusion* lê os dados do cliente e processa o CFML dentro da página. Baseado no CFML, o servidor interage com servidores de banco de dados, arquivos de sistema, servidores SMTP, e potencialmente outras aplicações e extensões através da API *ColdFusion* ou através de COM/DCOM.

Passo 4: O *ColdFusion* gera dinamicamente uma página HTML que é retornada para o servidor Web.

Passo 5: O servidor Web então retorna a página HTML para o navegador do usuário.

2.2 - UML

UML é uma linguagem visual utilizada para modelar sistemas computacionais por meio do paradigma de Orientação a Objetos. Essa Linguagem tornou-se, nos últimos anos, a linguagem padrão de modelagem de *software* adotada internacionalmente pela indústria de Engenharia de *Softwares* (GUEDES, 2005).

2.3 - DREAMWEAVER

O Adobe® Dreamweaver® CS5.5 é o software de criação e edição na *Web* que fornece recursos visuais e em nível de código para a criação de sites baseados em padrões e *designs* para *desktops*, *smartphones*, *tablets* e outros dispositivos (www.adobe.com/br/products/dreamweaver.html).

2.4 - MICROSOFT SQL SERVER 2008

O Microsoft SQL Server é um Sistema Gerenciador de Banco de Dados (SGBD) completo e integrado de gerenciamento e análise de dados. Permite a gerência de informações de forma confiável (www.microsoft.com).

3 – ANÁLISE E ESPECIFICAÇÃO DO SISTEMA

Neste capítulo são apresentadas as informações das ferramentas utilizadas para a modelagem do sistema proposto.

3.1 – LEVANTAMENTO DE REQUISITOS

Entrevista realizada na loja Stelise Modas em Sertanópolis.

a) Qual é o ramo da Loja?

R: A loja atua no ramo de vendas de roupas e acessórios.

b) Quais são as pessoas que fazem parte da Stelise Modas?

R: A Gerente (Proprietária) e uma Funcionária.

c) Descreva o problema gerado pela falta de um Sistema de Gerenciamento de Lojas?

R: Dificuldade no controle de estoque e gerenciamento de caixa.

d) Quais requisitos são essenciais para que o sistema atenda as necessidades de gerenciamento?

R: O sistema precisa conter um gerenciamento de clientes, de funcionários, de fornecedor, de produtos, de vendas e finanças. Onde seja possível cadastrar, atualizar cadastros e excluir cadastros de clientes, funcionário, fornecedores e produtos, faça o controle de estoque informando sempre quando um produto estiver em falta, gerencie também as vendas realizando todos os cálculos possíveis em

uma venda, como valor total da venda, valores de descontos tanto em valor real como em percentual e controle a parte financeira, onde será gerada as contas a pagar, a receber, salario e adicionais de funcionário, enfim toda entrada e saída de dinheiro.

e) Qual o resultado esperado após a implantação do Sistema de Gerenciamento de Loja?

R: Com a implantação espera-se mais agilidade e praticidade no atendimento, maior controle do estoque e, por meio de relatórios, conseguir informações que possam auxiliar melhor na tomada de decisões.

f) E no que influencia essas informações para o dia a dia da empresa?

R: Com estas informações pode-se prever a falta de algum produto no estoque, controlar o fluxo de caixa.

3.2 - FUNCIONALIDADES DO SISTEMA

A seguir são descritos os principais eventos relacionados ao sistema, destacando os que foram desenvolvidos:

- *Login* de Acesso ao Sistema;
- Cadastros (Produto, Usuário, Funcionário, Fornecedor, Cliente);
- Controle administrativo do sistema;
- Efetuar venda, troca e reposição de estoque;
- Gerar Relatórios.

3.2.1 - Restrições de desenvolvimento

- Microcomputador Intel® Core™i5 @2.53GHz, 4,00 GB RAM;
- O Sistema deverá ser executado no Windows 7 Home Premium ou compatível;
- Os relatórios precisam ser visualizados antes de impressos para a correção de eventuais falhas.

3.2.2 - Lista de eventos

Número	Evento	Caso de Uso
01	O Administrador efetua Autenticação de Usuário (Login)	Autenticar Usuário
02	O Administrador efetua cadastro de Usuário	Gerenciar Usuário
03	O Administrador efetua cadastro de Contas a Pagar	Gerenciar Contas a Pagar
04	O Administrador efetua cadastro de Contas a Receber	Gerenciar Conta a Receber
05	O Administrador ou Usuário efetua cadastro de Cliente	Gerenciar Cliente
06	O Administrador ou Usuário efetua cadastro de Produto	Gerenciar Produto
07	O Administrador ou Usuário efetua cadastro de Vendas	Gerenciar Vendas
08	O Administrador ou Usuário efetua cadastro de Funcionários	Gerenciar Funcionários
09	O Administrador solicita emissão de Relatório de Contas a Pagar	Emitir Relatório de Contas a Pagar
10	O Administrador solicita emissão de Relatório de Contas a Receber	Emitir Relatório de Contas a Receber
11	O Administrador solicita emissão de Relatório de Vendas	Emitir Relatório de Vendas
12	O Administrador ou Usuário solicita emissão de Relatório de Clientes	Emitir Relatório de Clientes

13	O Administrador ou Usuário solicita emissão de Relatório de Usuários	Emitir Relatório de Usuário
14	O Administrador ou Usuário solicita emissão de Relatório de Vendedores	Emitir Relatório de Vendedores
15	O Administrador ou Usuário solicita emissão de Relatório de Produtos	Emitir Relatório de Produtos
16	O Administrador ou Usuário solicita emissão de Relatório de Consignação	Emitir Relatório de Consignação

Quadro 1 - Lista de Eventos

3.3 - VALIDAÇÃO DOS REQUISITOS

a) Os requisitos levantados através de entrevistas estão completos?

SIM (X) NÃO ()

b) As informações levantadas atendem as necessidades do cliente?

SIM (X) NÃO ()

c) Os requisitos são reais?

SIM (X) NÃO ()

d) Houve algum tipo de redundância e falta de clareza dos requisitos?

SIM () NÃO (X)

e) A análise está de acordo com as necessidades solicitadas pelo cliente?

SIM (X) NÃO ()

3.4 - DIAGRAMA DE CASO DE USO GERAL

O diagrama de caso de uso é utilizado para modelar os requisitos do sistema, dando uma visão geral dos relacionamentos entre caso de uso e atores (MELO, 2010).

3.4.1 - Movimentação

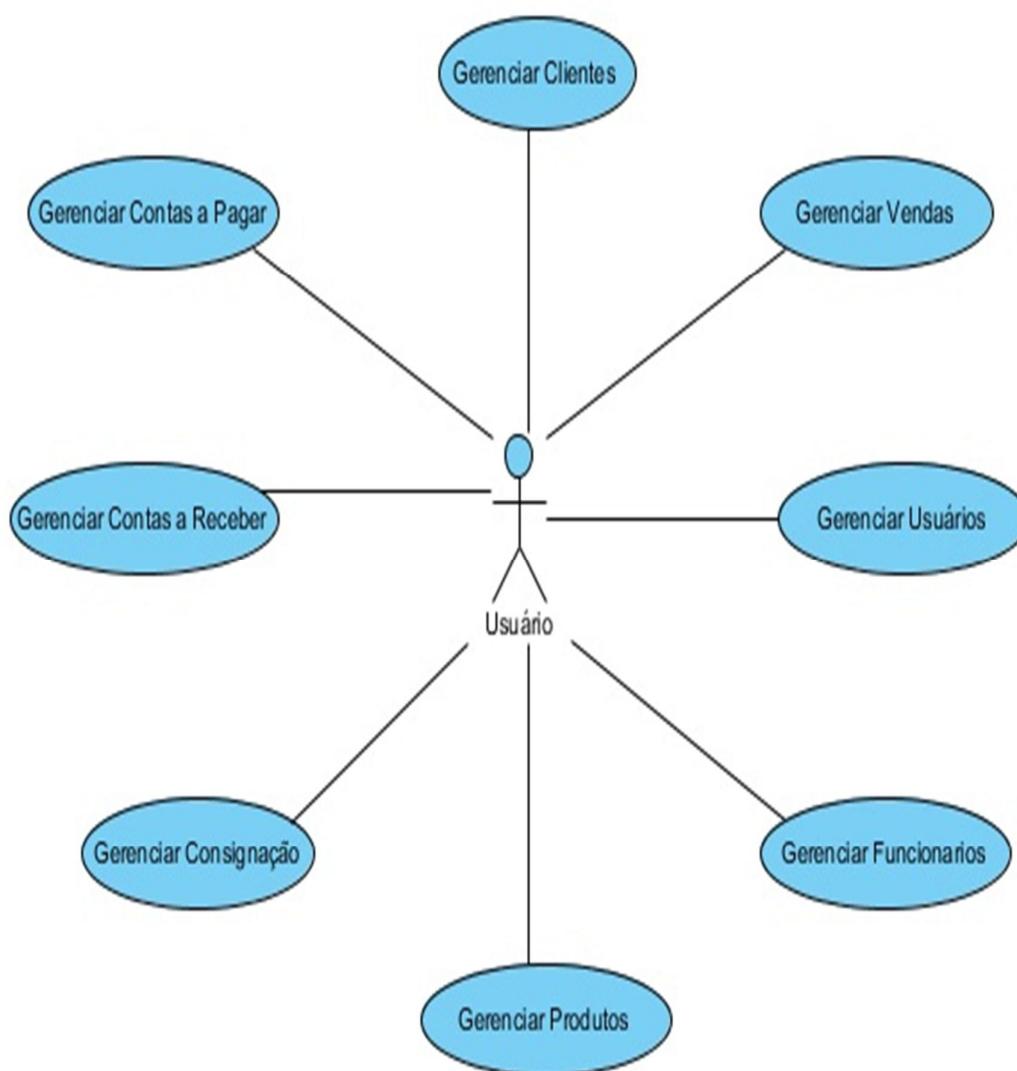
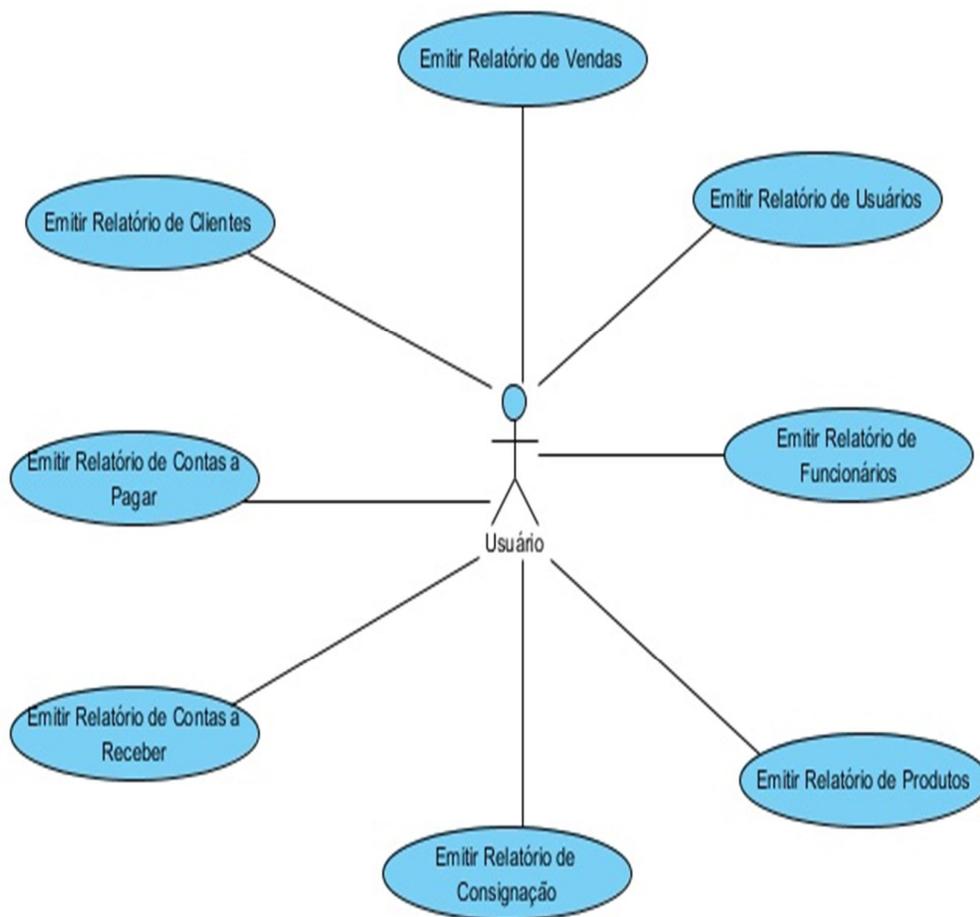


Figura 2 – Caso de Uso Geral – Movimentação

3.4.2 - Emissão de Relatório

**Figura 3 – Caso de Uso Geral – Emitir Relatórios**

3.5 - DIAGRAMAS DE CASO DE USO ESPECÍFICO

3.5.1 - UC 01 – Autenticar Usuário

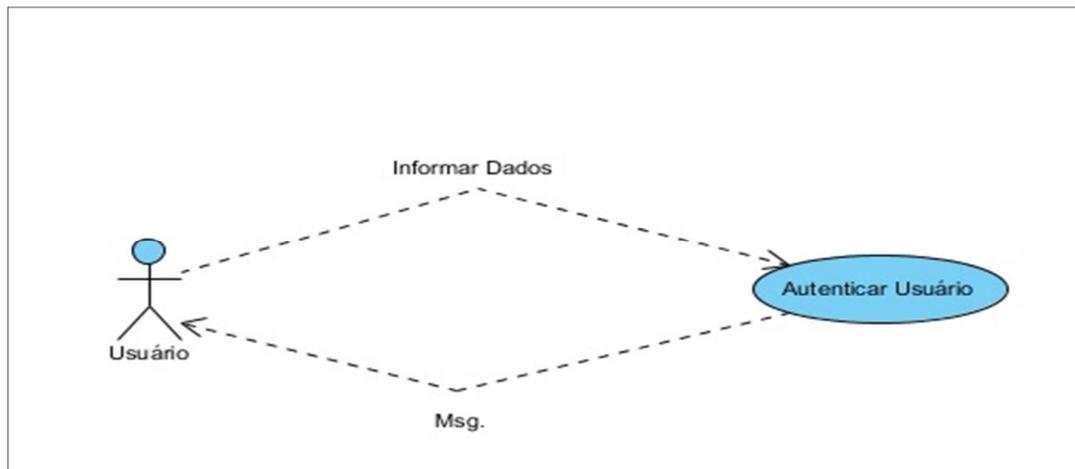


Figura 4 – UC – Autenticar Usuário

Nome do UC 01	Autenticar Usuário
Ator(es)	Usuário
Pré-Condições	Iniciar o Sistema
Cenário Principal	1 – O Sistema solicita usuário e senha 2 – O Usuário informa os dados para acesso 3 – O Usuário seleciona o botão <i>Login</i> 4 – O Sistema acessa a sua tela principal
Cenário Alternativo	O Usuário poderá cancelar Autenticação durante o processo
Casos de Testes	3.1 – O Sistema verifica se Usuário e Senha estão corretos 3.2 – Caso esteja incorreto o Sistema emite a mensagem “ <i>Login</i> e/ou Senha incorreto(s)” 3.3 – O Sistema solicita ao usuário informar a senha novamente.

Quadro 2 – Especificação Do Caso De Uso 01

3.5.2 - UC 02 – Gerenciar Usuário

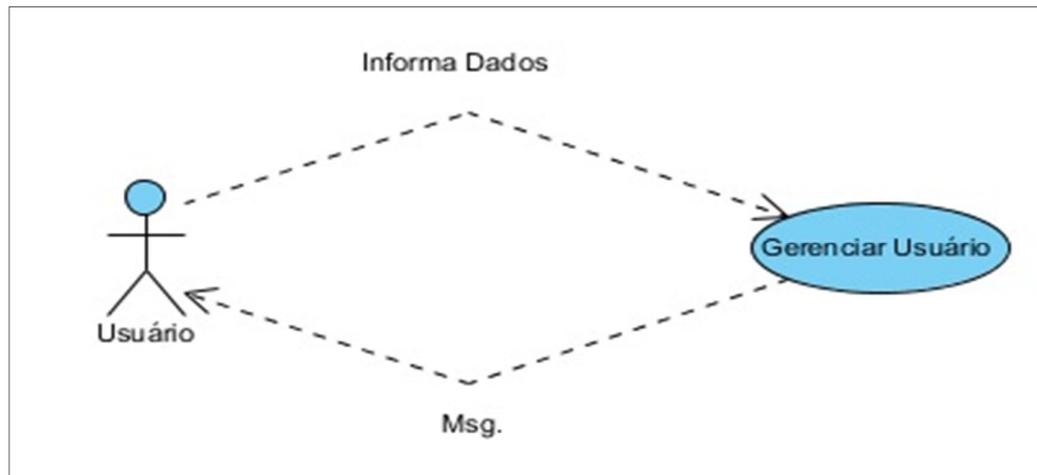


Figura 5 – UC – Gerenciar Usuários

Nome do UC 02	Gerenciar Usuários
Ator(es)	Usuário
Pré-Condições	O Usuário deverá estar autenticado no Sistema
Cenário Principal	1 – O Sistema solicita os dados necessários para o cadastro do Usuário 2 – O Usuário informa os dados necessários para o cadastro 3 – O Usuário seleciona o botão <i>Cadastrar</i> 4 – O Sistema emite uma mensagem “Usuário cadastrado com sucesso” 5 – O Sistema cadastra o Usuário
Cenário Alternativo	O Usuário poderá cancelar o processo durante o cadastro
Casos de Testes	3.1 – O Sistema verifica se os campos foram preenchidos corretamente e se o Usuário já existe 3.2 – Caso o Usuário esteja cadastrado, o sistema emite a mensagem “Usuário já Cadastrado” 3.3 – O Sistema volta para a tela de Cadastrar Usuário

Quadro 3 – Especificação do Caso de Uso 02

3.5.3 - UC 03 – Gerenciar Vendas

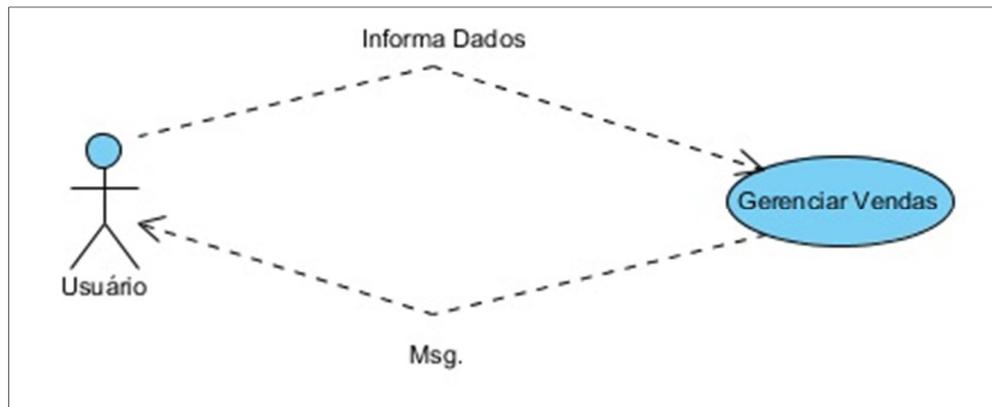


Figura 6 – UC – Gerenciar Vendas

Nome do UC 03	Gerenciar Vendas
Ator(es)	Administrador e Usuário
Pré-Condições	O Usuário deverá estar autenticado no Sistema e autorizado para realizar a Venda
Cenário Principal	1 – O Sistema solicita os dados necessários para a realização da Venda 2 – O Usuário informa os dados necessários para a realização da Venda. 3 – O Usuário seleciona o botão <i>Concluir</i> 4 – O Sistema emite uma mensagem “Venda realizada com sucesso” 5 – O Sistema realiza a Venda
Cenário Alternativo	O Usuário poderá cancelar o processo durante a Venda
Casos de Testes	3.1 – O Sistema verifica se os campos foram preenchidos corretamente 3.2 – Após a conclusão da Venda, o sistema emite a mensagem “Deseja realizar outra Venda?” 3.3 – O Sistema retorna para a interface de Gerenciar Vendas

Quadro 4 – Especificação do Caso de Uso 03

3.5.4 - UC 04 – Gerenciar Cliente

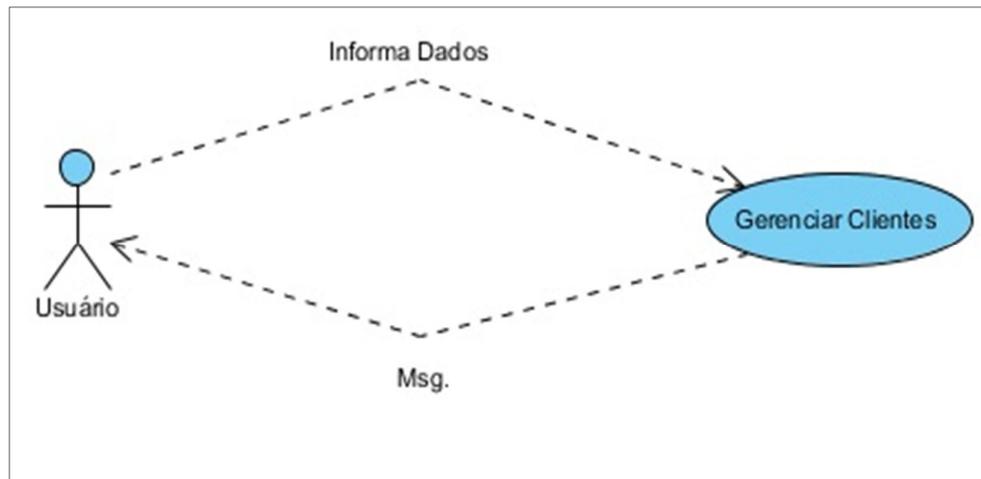


Figura 7 – UC – Gerenciar Cliente

Nome do UC 04	Gerenciar Clientes
Ator(es)	Usuário
Pré-Condições	O Usuário deverá estar autenticado no Sistema
Cenário Principal	1 – O Sistema solicita os dados necessários para o cadastro do Cliente 2 – O Usuário informa os dados necessários para o cadastro 3 – O Usuário seleciona o botão <i>Cadastrar</i> 4 – O Sistema emite uma mensagem “Cliente cadastrado com sucesso” 5 – O Sistema cadastra o Cliente
Cenário Alternativo	O Usuário poderá cancelar o processo durante o Cadastro do Cliente
Casos de Testes	3.1 – O Sistema verifica se os campos foram preenchidos corretamente e se o Cliente já está cadastrado 3.2 – Caso o Cliente esteja cadastrado, o sistema emite a mensagem “Cliente já Cadastrado”. 3.3 – O Sistema retorna para a interface de Gerenciamento de Cliente

Quadro 5 – Especificação do Caso de Uso 04

3.5.5 - UC 04 – Gerenciar Vendedores

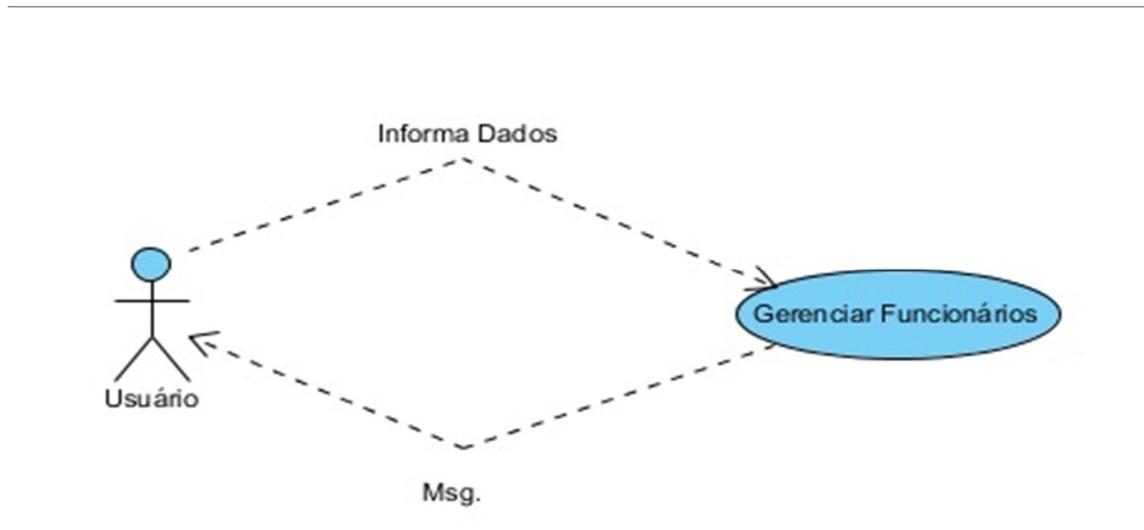


Figura 8 – UC – Gerenciar Vendedores

Nome do UC 05	Gerenciar Vendedores
Ator(es)	Usuário
Pré-Condições	O Usuário deverá estar autenticado no Sistema
Cenário Principal	1 – O Sistema solicita os dados necessários para o cadastro do Vendedor 2 – O Usuário informa os dados necessários para o cadastro 3 – O Usuário seleciona o botão <i>Cadastrar</i> 4 – O Sistema emite uma mensagem “Vendedor cadastrado com sucesso” 5 – O Sistema cadastra o Vendedor
Cenário Alternativo	O Usuário poderá cancelar o processo durante o Cadastro do Vendedor
Casos de Testes	3.1 – O Sistema verifica se os campos foram preenchidos corretamente e se o Vendedor já está cadastrado 3.2 – Caso o Vendedor esteja cadastrado, o sistema emite a mensagem “Vendedor já Cadastrado”. 3.3 – O Sistema retorna para a interface de Gerenciamento de Vendedor

Quadro 6 – Especificação do Caso de Uso 05

3.5.6 - UC 06 – Gerenciar Produtos

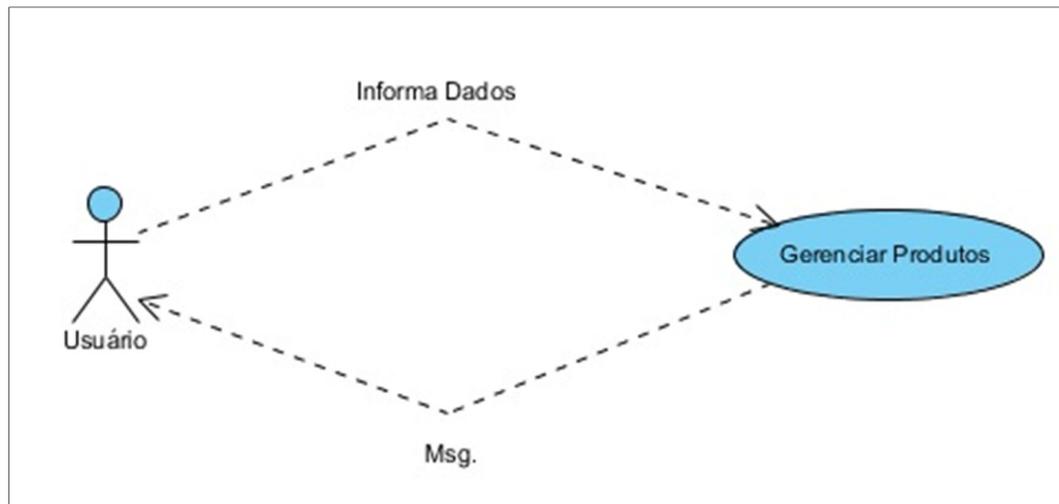


Figura 9 – UC – Gerenciar Produtos

Nome do UC 06	Gerenciar Produtos
Ator(es)	Usuário
Pré-Condições	O Usuário deverá estar autenticado no Sistema
Cenário Principal	1 – O Sistema solicita os dados necessários para o cadastro do Produto 2 – O Usuário informa os dados necessários para o cadastro 3 – O Usuário seleciona o botão <i>Cadastrar</i> 4 – O Sistema emite uma mensagem “Produto cadastrado com sucesso” 5 – O Sistema cadastra o Produto
Cenário Alternativo	O Usuário poderá cancelar o processo durante o Cadastro do Produto
Casos de Testes	3.1 – O Sistema verifica se os campos foram preenchidos corretamente 3.2 – O Sistema retorna para a interface de Gerenciamento de Produto

Quadro 7 – Especificação Do Caso De Uso 06

3.5.7 - UC 07 – Gerenciar Consignação

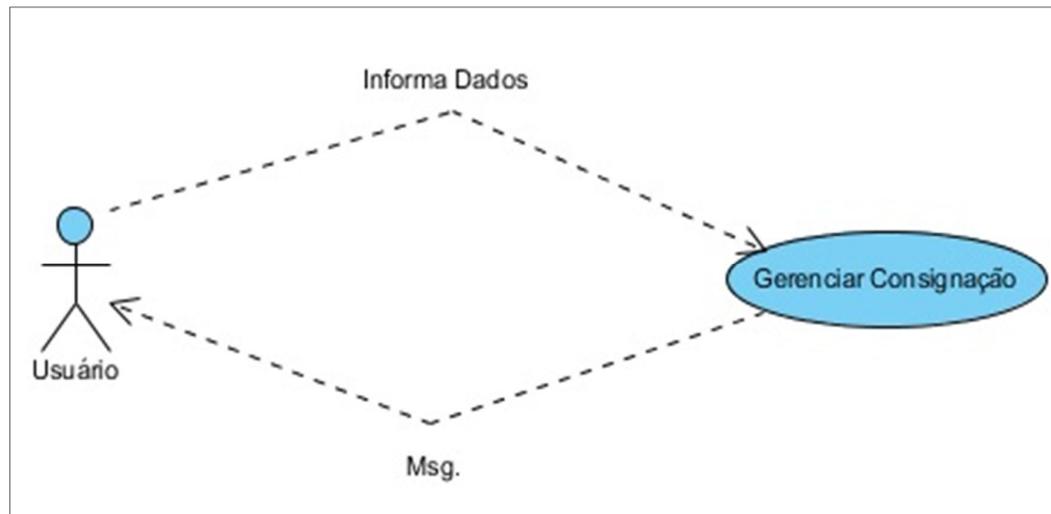


Figura 10 – UC – Gerenciar Consignação

Nome do UC 07	Gerenciar Consignação
Ator(es)	Usuário
Pré-Condições	O Usuário deverá estar autenticado no Sistema
Cenário Principal	1 – O Sistema solicita os dados necessários para o cadastro da Consignação 2 – O Usuário informa os dados necessários para o cadastro 3 – O Usuário seleciona o botão <i>Cadastrar</i> 4 – O Sistema emite uma mensagem “Consignação cadastrada com sucesso” 5 – O Sistema cadastra a Consignação
Cenário Alternativo	O Usuário poderá cancelar o processo durante o Cadastro da Consignação
Casos de Testes	3.1 – O Sistema verifica se os campos foram preenchidos corretamente 3.2 – Caso a Consignação esteja cadastrado, o sistema emite a mensagem “Consignação já Cadastrada” 3.3 – O Sistema retorna para a interface de Gerenciamento de Consignação

Quadro 8 – Especificação Do Caso De Uso 07

3.5.8 - UC 08 – Gerenciar Contas a Receber

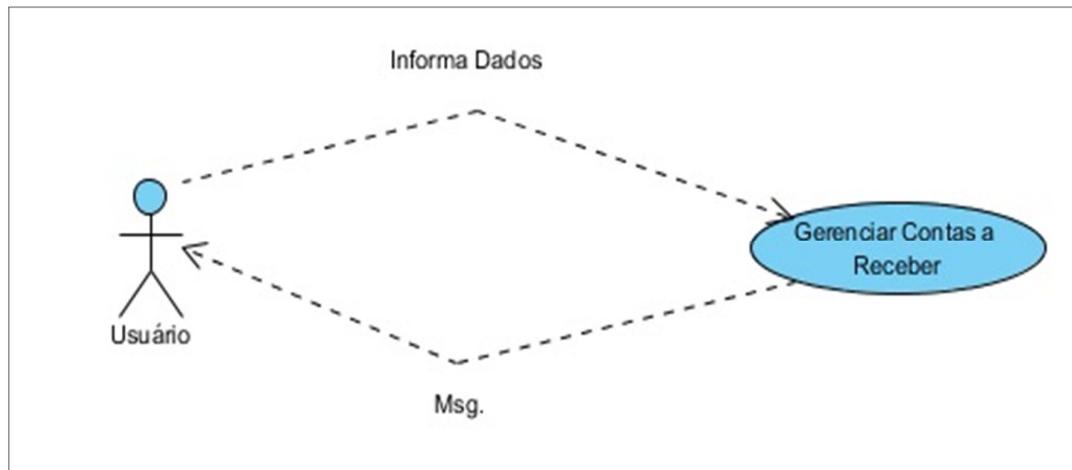


Figura 11 – UC – Gerenciar Contas a Receber

Nome do UC 08	Gerenciar Contas a Receber
Ator(es)	Usuário
Pré-Condições	O Usuário deverá estar autenticado no Sistema
Cenário Principal	1 – O Sistema solicita os dados necessários para o cadastro das Contas a Receber 2 – O Usuário informa os dados necessários para o cadastro 3 – O Usuário seleciona o botão <i>Cadastrar</i> 4 – O Sistema emite uma mensagem “Conta a Receber cadastrada com sucesso” 5 – O Sistema cadastra a Conta a Receber
Cenário Alternativo	O Usuário poderá cancelar o processo durante o Cadastro da Conta a Receber.
Casos de Testes	3.1 – O Sistema verifica se os campos foram preenchidos corretamente 3.2 – Caso a Conta a Receber esteja cadastrada o sistema emite a mensagem “Conta a Receber já Cadastrada” 3.3 – O Sistema retorna para a interface de Gerenciamento de Contas a Receber

Quadro 9 – Especificação Do Caso De Uso 08

3.5.9 - UC 09 – Gerenciar Contas a Pagar

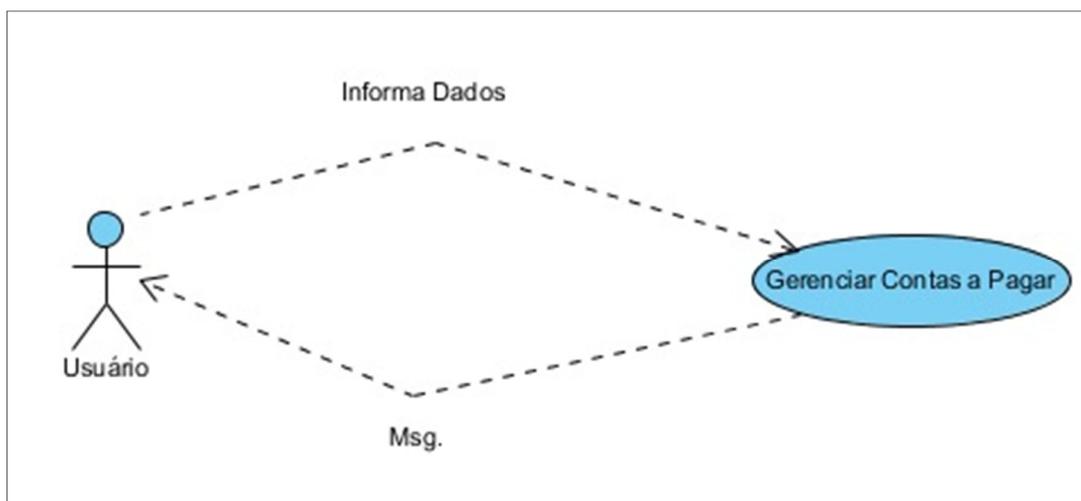


Figura 12 – UC – Gerenciar Contas a Receber

Nome do UC 09	Gerenciar Contas a Pagar
Ator(es)	Usuário
Pré-Condições	O Usuário deverá estar autenticado no Sistema
Cenário Principal	1 – O Sistema solicita os dados necessários para o cadastro das Contas a Pagar 2 – O Usuário informa os dados necessários para o cadastro 3 – O Usuário seleciona o botão <i>Cadastrar</i> 4 – O Sistema emite uma mensagem “Conta a Pagar cadastrada com sucesso” 5 – O Sistema cadastra a Conta a Pagar
Cenário Alternativo	O Usuário poderá cancelar o processo durante o Cadastro da Conta a Pagar
Casos de Testes	3.1 – O Sistema verifica se os campos foram preenchidos corretamente 3.2 – Caso a Conta a Pagar esteja cadastrada, o sistema emite a mensagem “Conta a Pagar já Cadastrada” 3.3 – O Sistema retornar para a interface de Gerenciamento de Contas a Pagar

Quadro 10 – Especificação Do Caso De Uso 09

3.5.10 - UC 10 – Emitir Relatório de Vendas

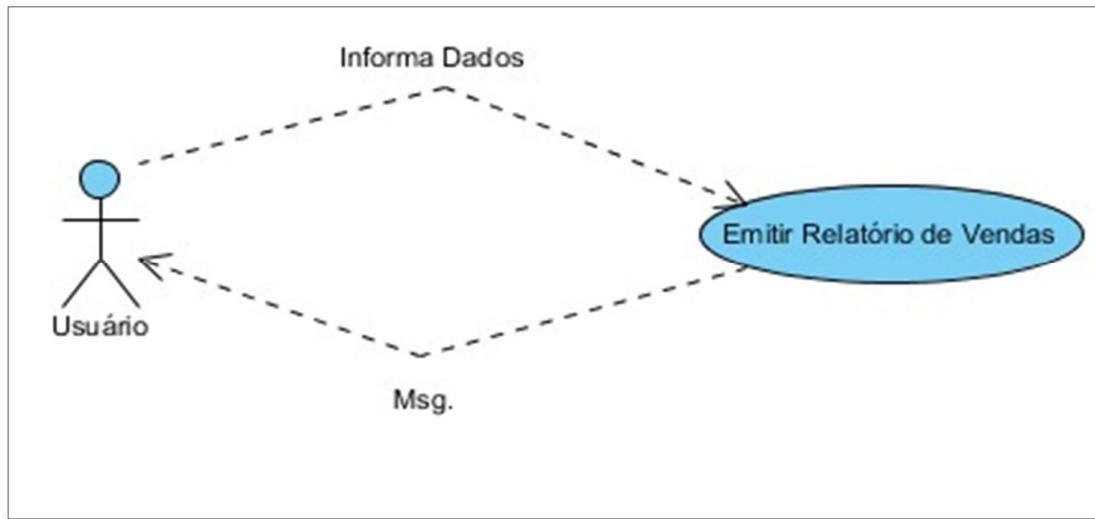


Figura 13 – UC – Emitir Relatório de Vendas

Nome do UC 10	Emitir Relatório de Vendas
Ator(es)	Usuário
Pré-Condições	O Usuário deverá estar autenticado no Sistema
Descrição	Os relatórios deverão ser impressos conforme solicitação
Cenário Principal	1 – O Sistema disponibiliza as informações para o Usuário 2 – O Usuário seleciona o botão Visualizar Relatório 3 – O Sistema exibe o Relatório na tela 4 – O Usuário seleciona o botão <i>Imprimir</i> 5 – O Sistema imprime o Relatório
Cenário Alternativo	O Usuário poderá apenas visualizar o Relatório sem imprimir
Casos de Testes	5.1 – O Sistema cancela a operação

Quadro 11 – Especificação Do Caso De Uso 10

3.5.11 - UC 11 – Emitir Relatório de Contas a Pagar

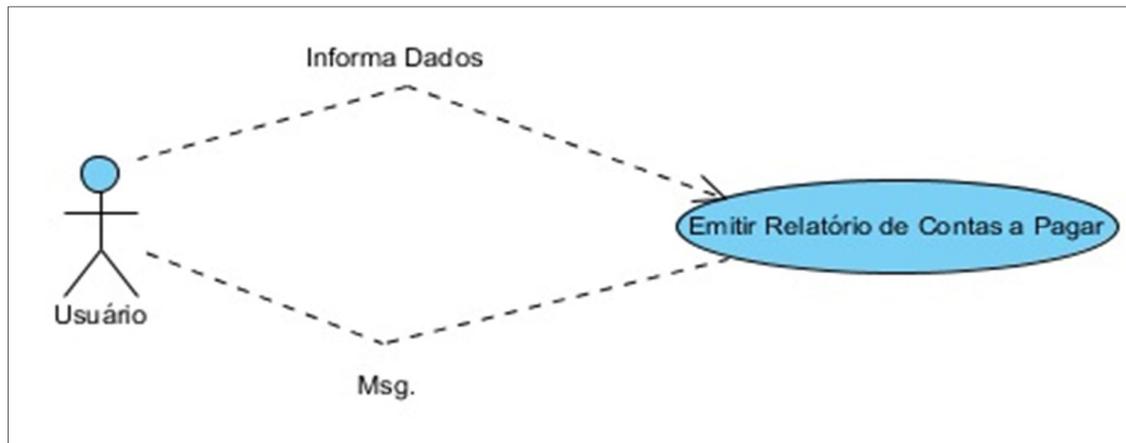


Figura 14 – UC – Emitir Relatório de Contas a Pagar

Nome do UC 11	Emitir Relatório de Contas a Pagar
Ator(es)	Usuário
Pré-Condições	O Usuário deverá estar autenticado no Sistema
Descrição	Os relatórios deverão ser impressos conforme solicitação
Cenário Principal	1 – O Sistema disponibiliza as informações para o Usuário 2 – O Usuário seleciona o botão Visualizar Relatório 3 – O Sistema exibe o Relatório na tela 4 – O Usuário seleciona o botão <i>Imprimir</i> 5 – O Sistema imprime o Relatório
Cenário Alternativo	O Usuário poderá apenas visualizar o Relatório sem imprimir
Casos de Testes	5.1 – O Sistema cancela a operação

Quadro 12 – Especificação Do Caso De Uso 11

3.5.12 - UC 12 – Emitir Relatório de Contas a Receber

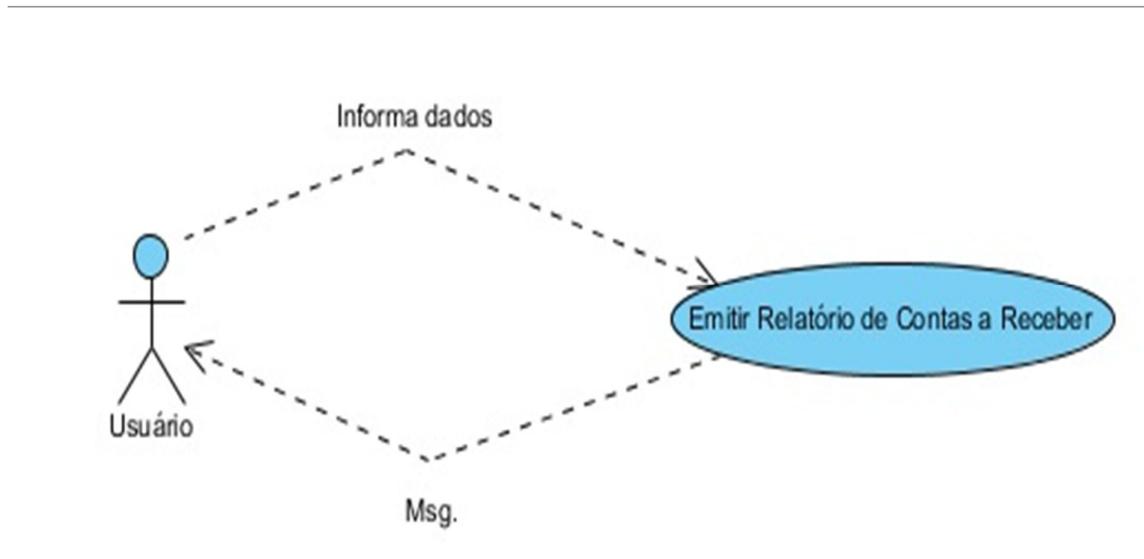


Figura 15 – UC – Emitir Relatório de Contas a Receber

Nome do UC 12	Emitir Relatório de Contas a Receber
Ator(es)	Usuário
Pré-Condições	O Usuário deverá estar autenticado no Sistema
Descrição	Os relatórios deverão ser impressos conforme solicitação
Cenário Principal	1 – O Sistema disponibiliza as informações para o Usuário 2 – O Usuário seleciona o botão Visualizar Relatório 3 – O Sistema exibe o Relatório na tela 4 – O Usuário seleciona o botão <i>Imprimir</i> 5 – O Sistema imprime o Relatório
Cenário Alternativo	O Usuário poderá apenas visualizar o Relatório sem imprimir
Casos de Testes	5.1 – O Sistema cancela a operação

Quadro 13 – Especificação Do Caso De Uso 12

3.5.13 - UC 13 – Emitir Relatório de Clientes

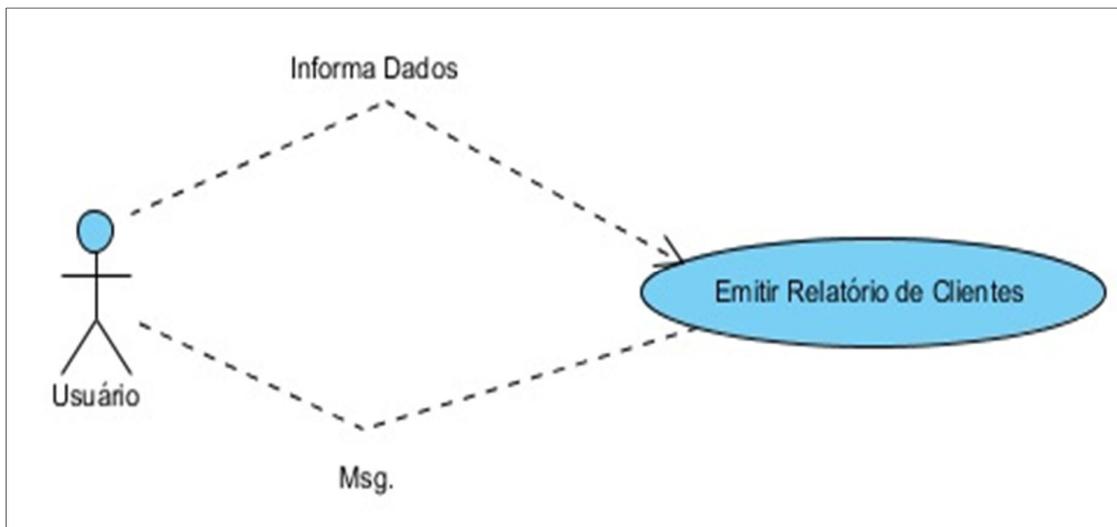


Figura 16 – UC – Emitir Relatório de Clientes

Nome do UC 13	Emitir Relatório de Clientes
Ator(es)	Usuário
Pré-Condições	O Usuário deverá estar autenticado no Sistema
Descrição	Os relatórios deverão ser impressos conforme solicitação
Cenário Principal	1 – O Sistema disponibiliza as informações para o Usuário 2 – O Usuário seleciona o botão Visualizar Relatório 3 – O Sistema exibe o Relatório na tela 4 – O Usuário seleciona o botão <i>Imprimir</i> 5 – O Sistema imprime o Relatório
Cenário Alternativo	O Usuário poderá apenas visualizar o Relatório sem imprimir
Casos de Testes	5.1 – O Sistema cancela a operação

Quadro 14 – Especificação Do Caso De Uso 13

3.5.14 - UC 14 – Emitir Relatório de Usuário

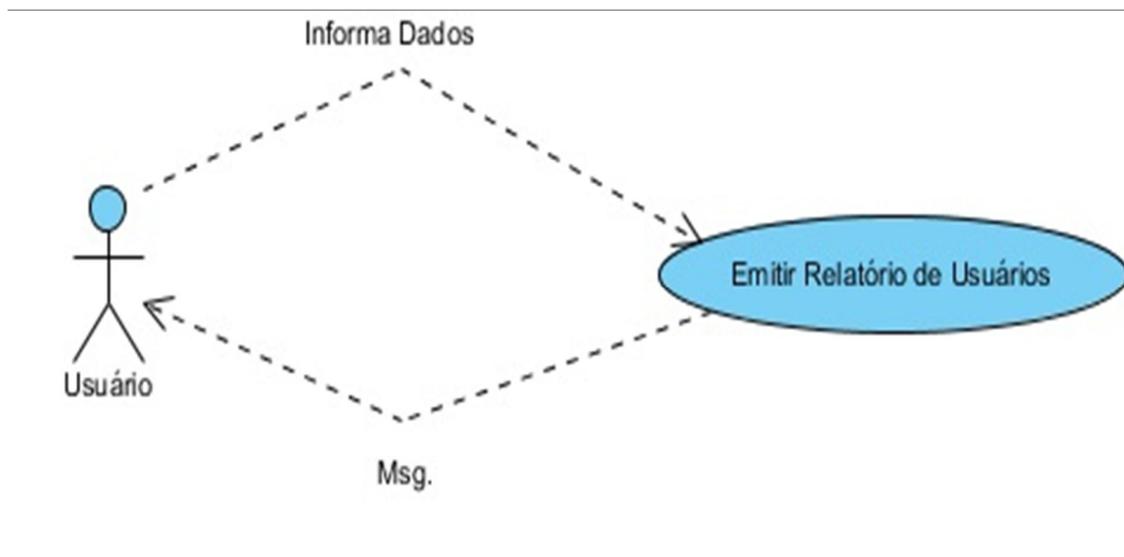


Figura 17 – UC – Emitir Relatório de Usuário

Nome do UC 14	Emitir Relatório de Usuário
Ator(es)	Usuário
Pré-Condições	O Usuário deverá estar autenticado no Sistema
Descrição	Os relatórios deverão ser impressos conforme solicitação
Cenário Principal	1 – O Sistema disponibiliza as informações para o Usuário 2 – O Usuário seleciona o botão Visualizar Relatório 3 – O Sistema exibe o Relatório na tela 4 – O Usuário seleciona o botão <i>Imprimir</i> 5 – O Sistema imprime o Relatório
Cenário Alternativo	O Usuário poderá apenas visualizar o Relatório sem imprimir
Casos de Testes	5.1 – O Sistema cancela a operação

Quadro 15 – Especificação Do Caso De Uso 14

3.5.15 - UC 15 – Emitir Relatório de Funcionários

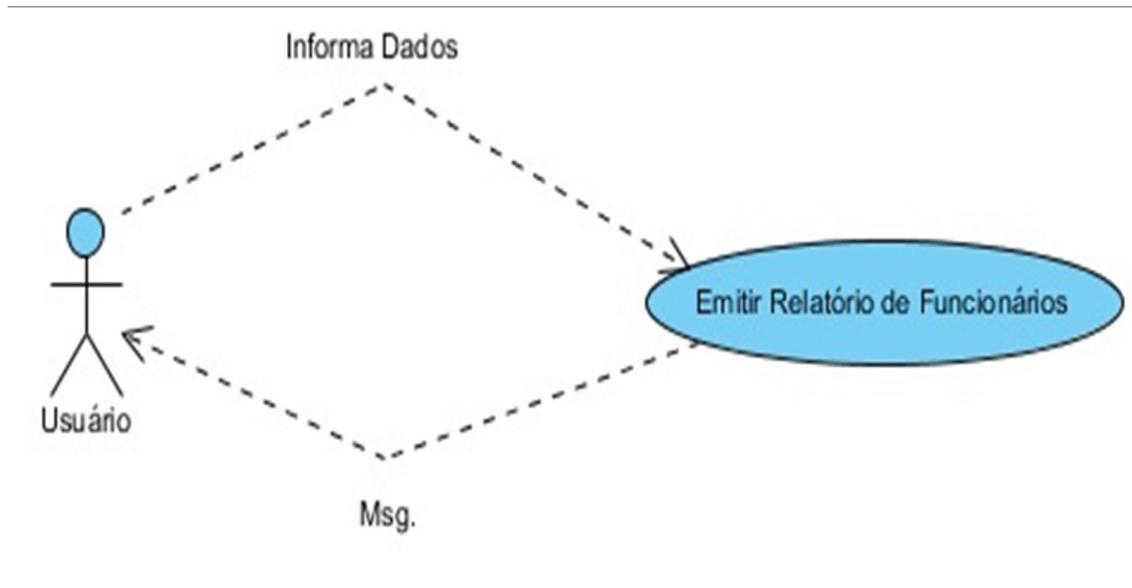


Figura 18 – UC – Emitir Relatório de Funcionários

Nome do UC 15	Emitir Relatório de Funcionários
Ator(es)	Usuário
Pré-Condições	O Usuário deverá estar autenticado no Sistema
Descrição	Os relatórios deverão ser impressos conforme solicitação
Cenário Principal	1 – O Sistema disponibiliza as informações para o Usuário 2 – O Usuário seleciona o botão Visualizar Relatório 3 – O Sistema exibe o Relatório na tela 4 – O Usuário seleciona o botão <i>Imprimir</i> 5 – O Sistema imprime o Relatório
Cenário Alternativo	O Usuário poderá apenas visualizar o Relatório sem imprimir
Casos de Testes	5.1 – O Sistema cancela a operação

Quadro 16 – Especificação Do Caso De Uso 15

3.5.16 - UC 16 – Emitir Relatório de Produtos

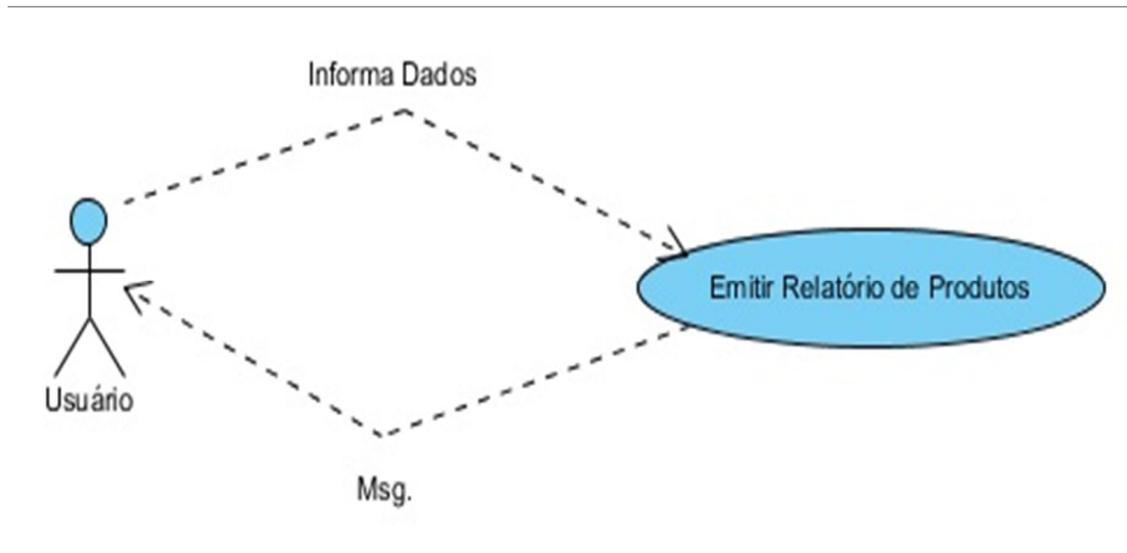


Figura 19 – UC – Emitir Relatório de Produtos

Nome do UC 16	Emitir Relatório de Produtos
Ator(es)	Usuário
Pré-Condições	O Usuário deverá estar autenticado no Sistema
Descrição	Os relatórios deverão ser impressos conforme solicitação
Cenário Principal	1 – O Sistema disponibiliza as informações para o Usuário 2 – O Usuário seleciona o botão Visualizar Relatório 3 – O Sistema exibe o Relatório na tela 4 – O Usuário seleciona o botão <i>Imprimir</i> 5 – O Sistema imprime o Relatório
Cenário Alternativo	O Usuário poderá apenas visualizar o Relatório sem imprimir
Casos de Testes	5.1 – O Sistema cancela a operação

Quadro 17 – Especificação Do Caso De Uso 16

3.5.17 - UC 17 – Emitir Relatório de Consignação

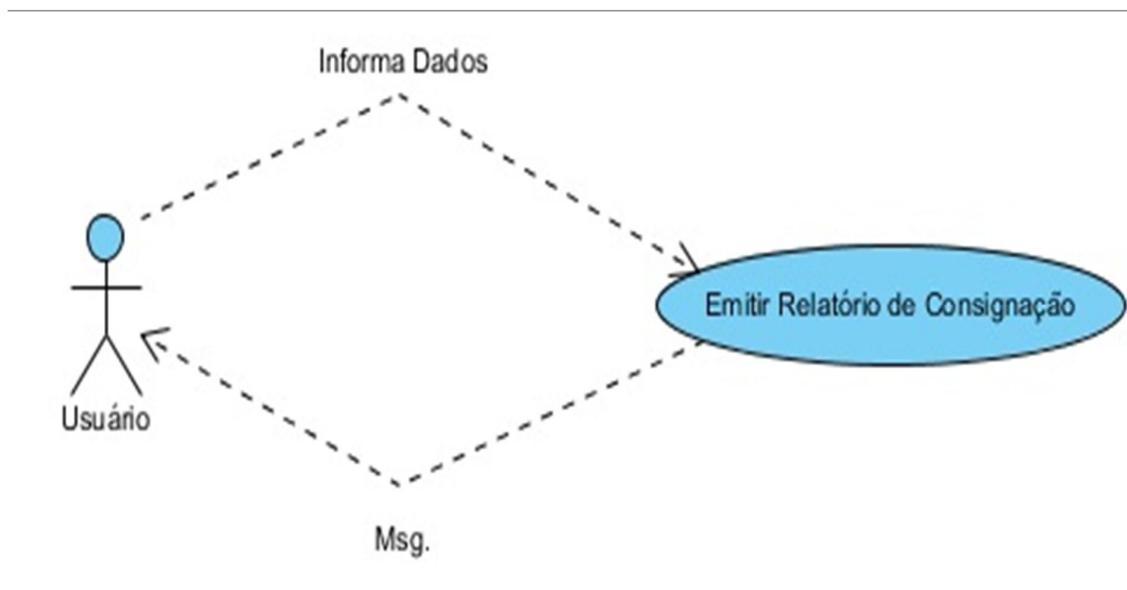


Figura 20 – UC – Emitir Relatório de Consignação

Nome do UC 17	Emitir Relatório de Consignação
Ator(es)	Usuário
Pré-Condições	O Usuário deverá estar autenticado no Sistema
Descrição	Os relatórios deverão ser impressos conforme solicitação
Cenário Principal	1 – O Sistema disponibiliza as informações para o Usuário 2 – O Usuário seleciona o botão Visualizar Relatório 3 – O Sistema exibe o Relatório na tela 4 – O Usuário seleciona o botão <i>Imprimir</i> 5 – O Sistema imprime o Relatório
Cenário Alternativo	O Usuário poderá apenas visualizar o Relatório sem imprimir
Casos de Testes	5.1 – O Sistema cancela a operação

Quadro 18 – Especificação Do Caso De Uso 17

3.6 - DIAGRAMA DE ATIVIDADES

A versão atual do diagrama de atividades está mais coerente com um diagrama de fluxo de dados ou de *workflow*. Este diagrama tem por propósito focalizar um fluxo de atividades que ocorrem internamente em um processamento, dentro de um período de tempo (MELO, 2011).

3.6.1 – Cadastros

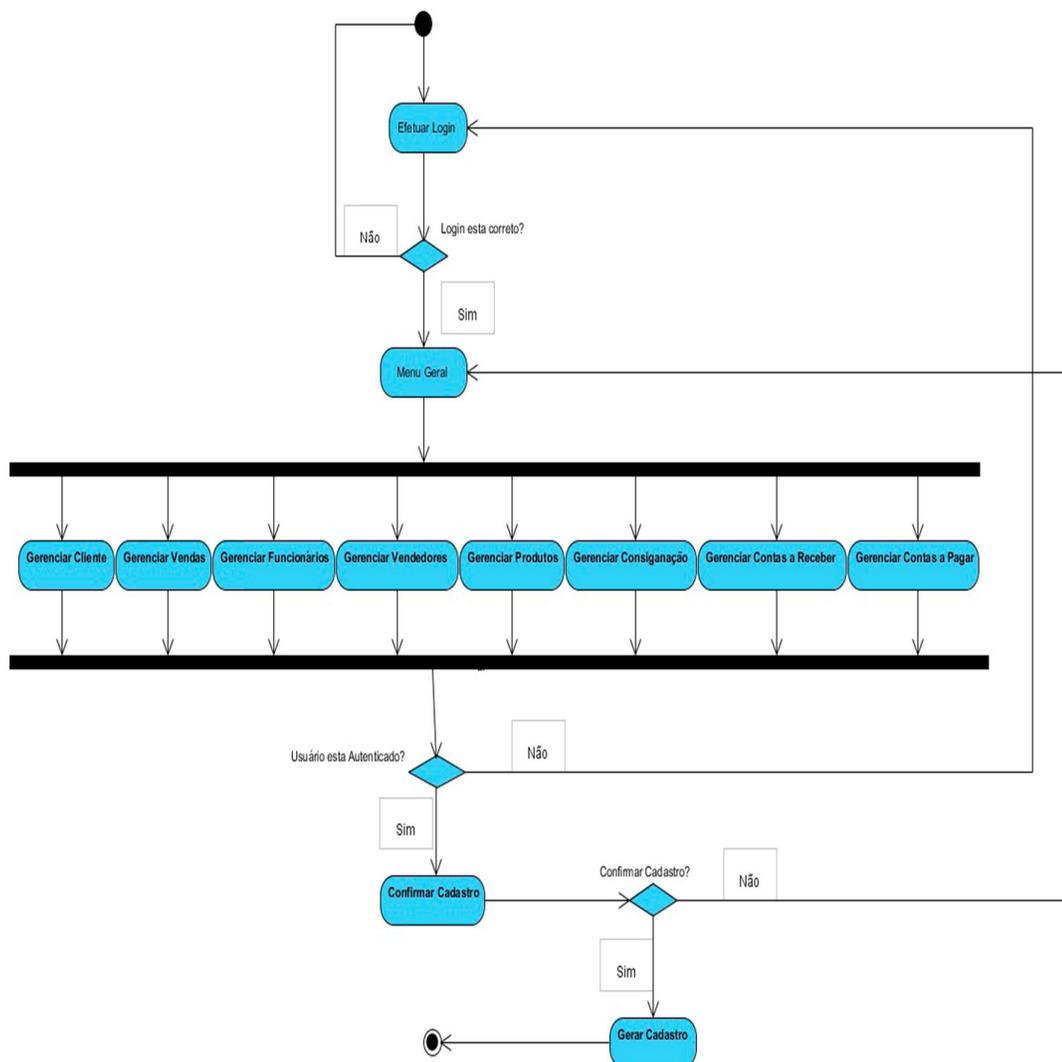


Figura 21 - Diagrama de Atividades – Cadastros

3.6.2 - Movimentações

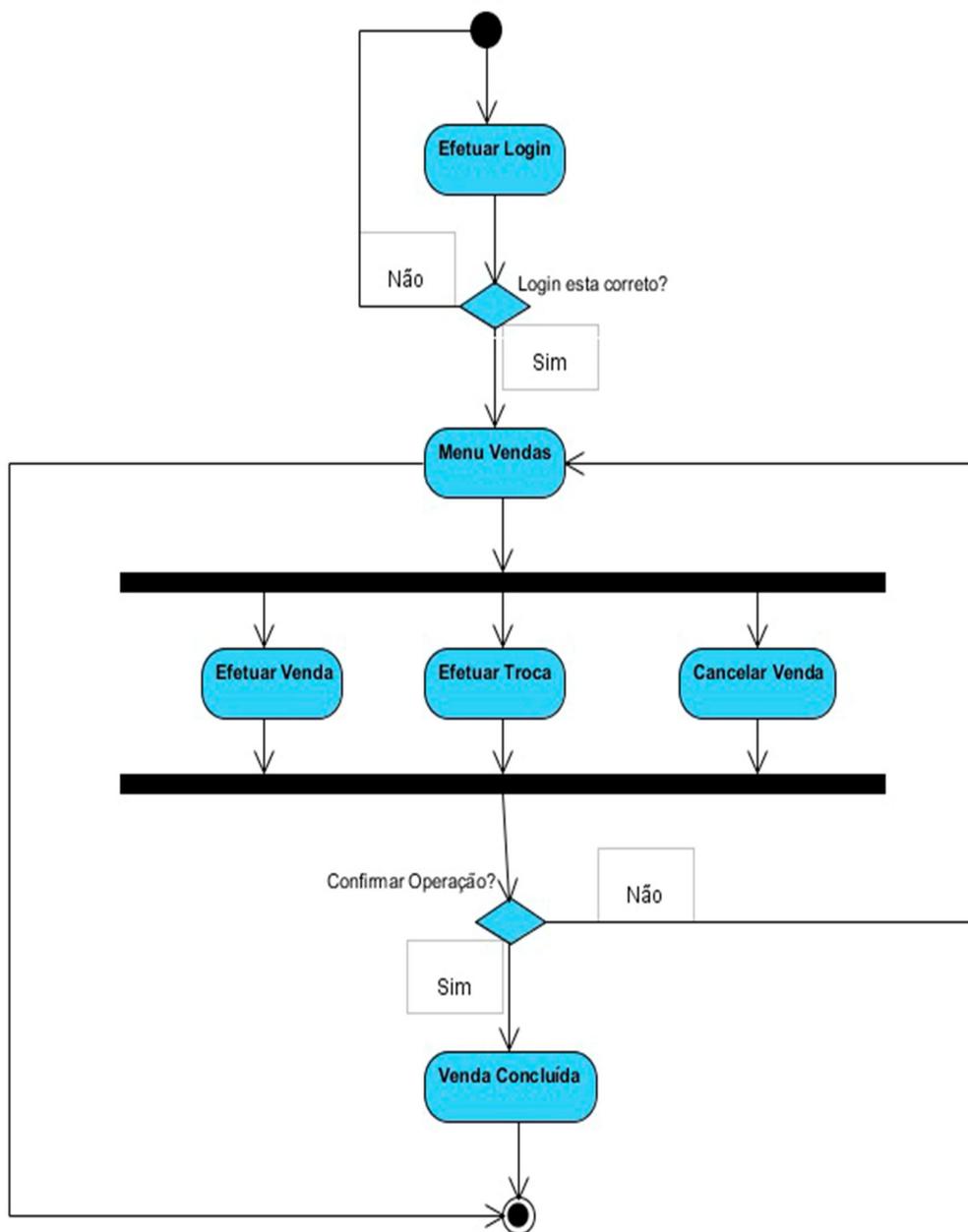


Figura 22 - Diagrama de Atividades – Movimentação

3.6.3 Relatórios

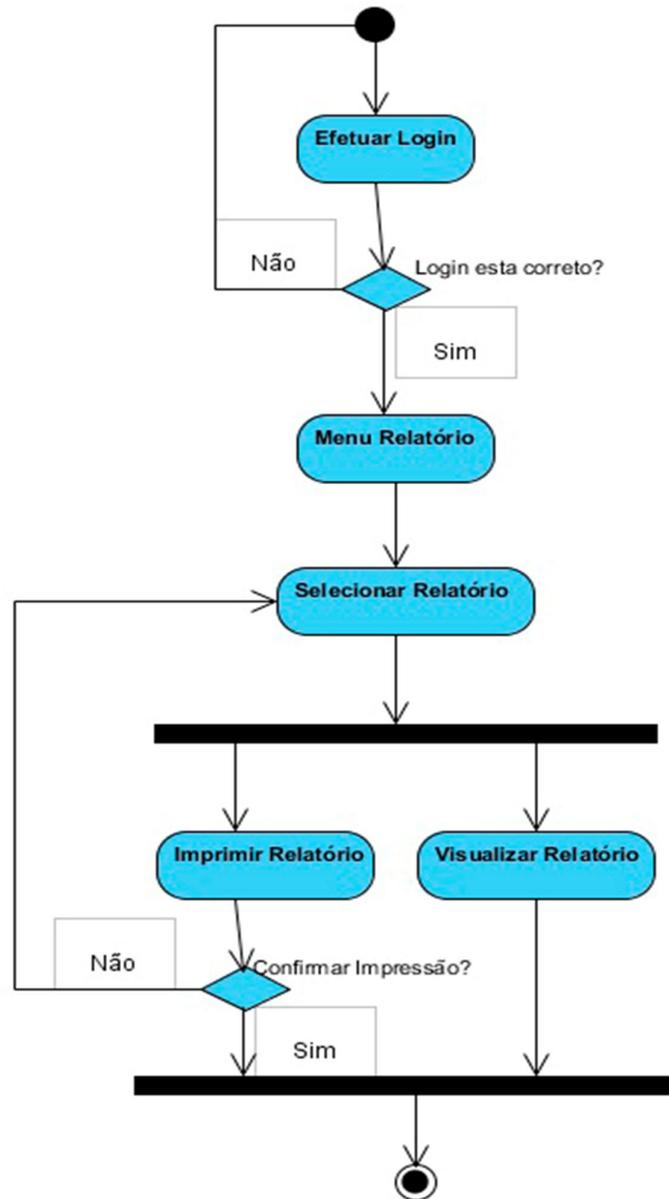


Figura 23 - Diagrama de Atividades – Relatórios

3.7 – DIAGRAMA DE CLASSE

Um diagrama de classes descreve os tipos de objetos presentes no sistema e os vários tipos de relacionamentos existentes entre eles. Eles mostram também as propriedades e as operações de uma classe e as restrições que se aplicam à maneira que os objetos serão conectados (FOWLER, 2005).



Figura 24 - Diagrama de Classes

3.8 – DIAGRAMA DE SEQUENCIA

Um diagrama de sequência mostra as sequências dinâmicas dos cenários em particular e é usado para descrever padrões de comunicação entre objetos (HORSTAMANN, 2006).

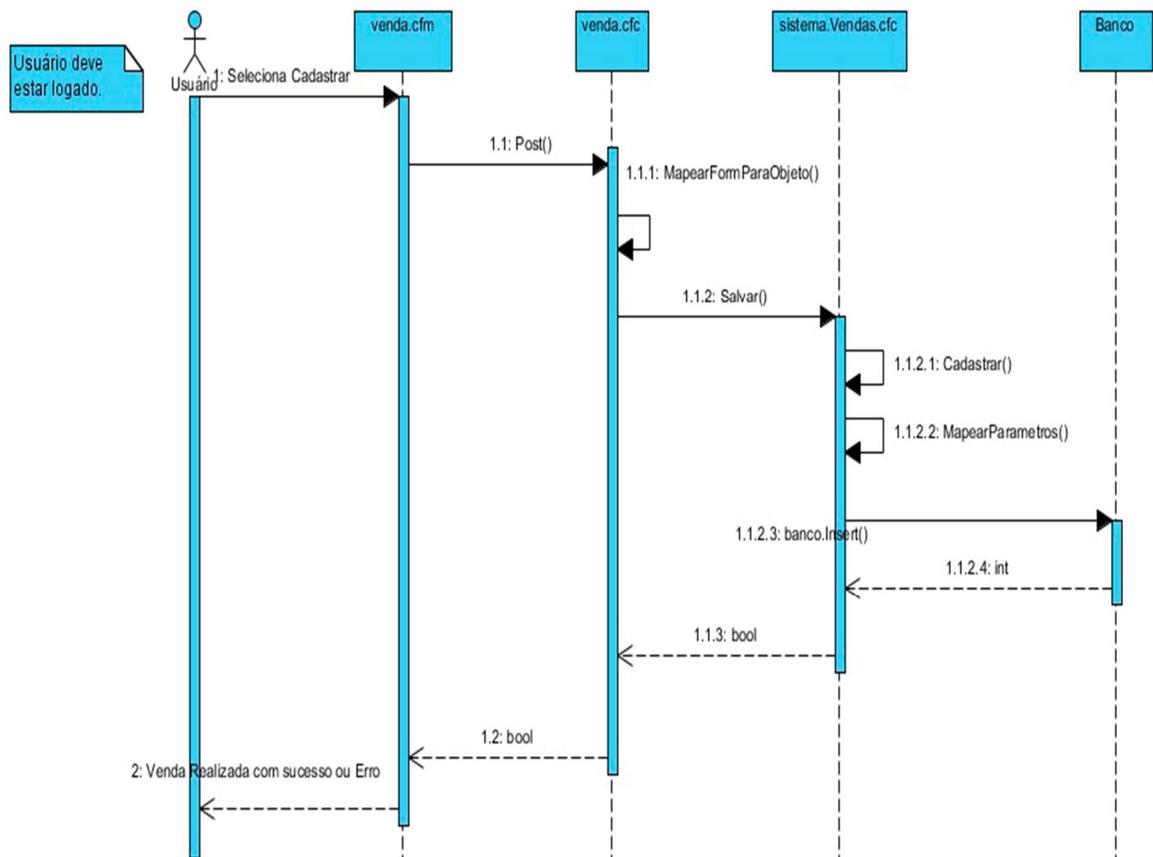


Figura 25 - Diagrama de Sequência – Efetuar Vendas

3.9 – MODELO DE ENTIDADE RELACIONAMENTO (MER)

O modelo de entidade relacionamento baseia-se na percepção do mundo real, que consiste em uma relação entre objetos básicos, chamados entidades, e em relacionamento entre esses objetos (OLIVEIRA, FERNANDES, TAVEIRA, 2000).

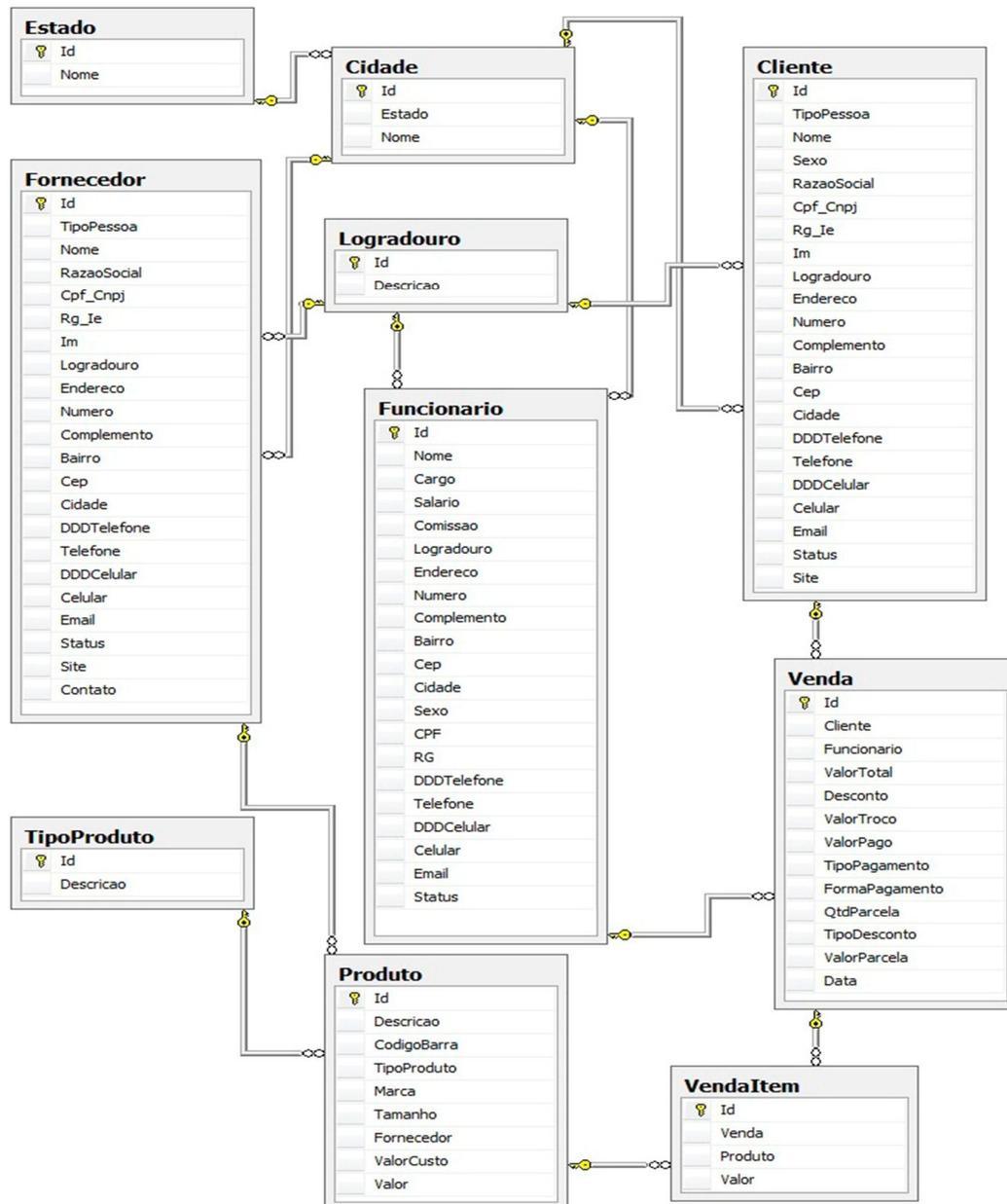


Figura 26 – MER - Geral

4 – PROJETO

Neste capítulo será discorrido sobre a WBS (*Work BreakDown Structure*) e a especificação dos custos.

4.1 - ESTRUTURA ANALÍTICA DO PROJETO (WBS)

A WBS (*Work BreakDown Structure*), tem como objetivo apresentar de forma gráfica (como organograma – Forma de uma árvore) ou como tabela (Lista indentada), identificando as frentes de trabalho a serem desenvolvidas na condução do projeto. É uma representação gráfica e detalhada do escopo do projeto (www.scribd.com).

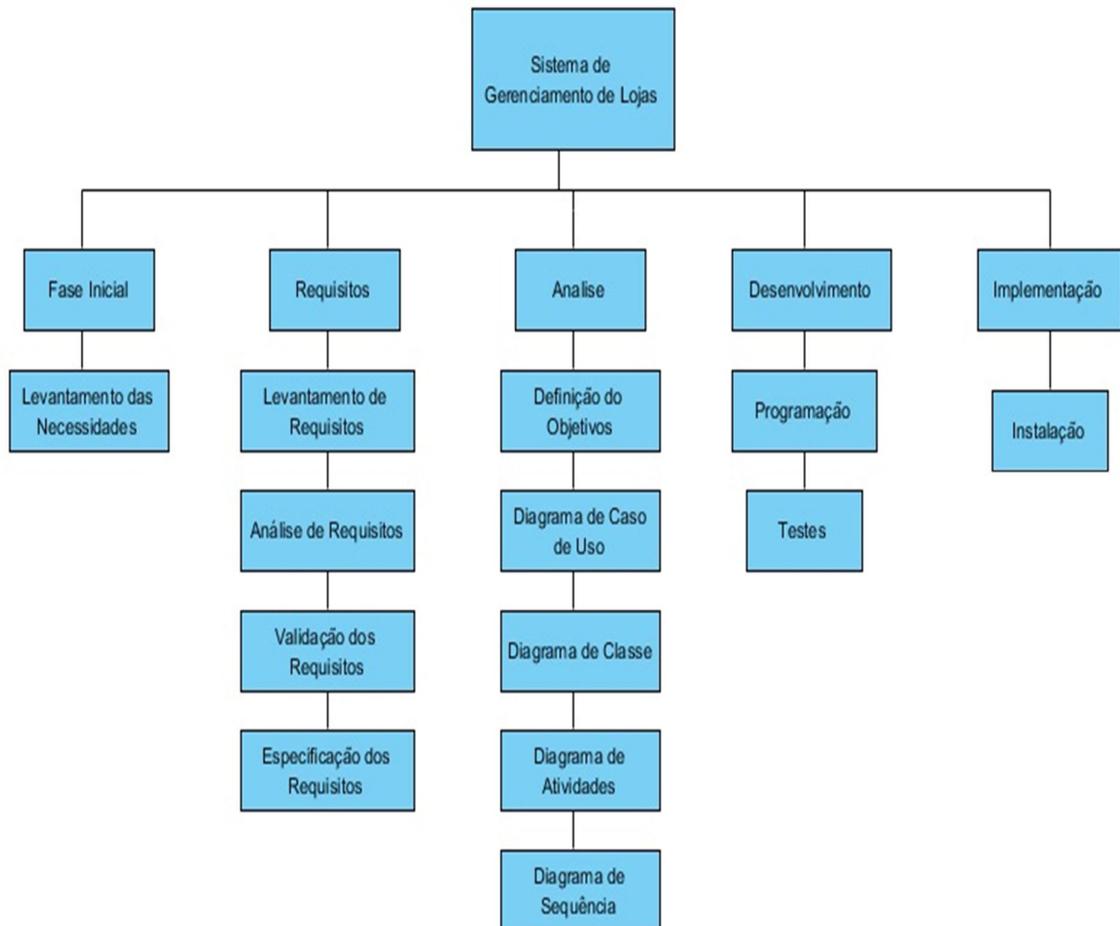


Figura 27 - WBS (Work Breakdown Structure)

4.2 - ESPECIFICAÇÃO DOS CUSTOS

4.2.1 - Recursos necessários para o desenvolvimento

- 01 Analista - Programador
- 01 *Notebook Intel® Core™ i5 CPU M460 @ 2.53GHz, 4,00 GB RAM;*
- 01 Impressora Multifuncional HP
- Sistema Operacional *Windows 7 Home Premium;*
- *Adobe Dreamweaver CS5;*
- *Microsoft SQL Server 2008;*
- *Visual Paradigm for UML 8.2.*

4.2.2 - Estimativa de custos

Custo Analista – Programador

Custo Diário: R\$50,00 (Cinquenta Reais)

Total de Dias: 135

Custo Total: R\$6.750,00 (Seis mil setecentos e cinquenta reais).

4.2.3 - Custos materiais

Notebook: R\$ 1.700,00 (Mil e setecentos reais);

Depreciação 3 anos: $R\$ 1.700,00 / 36$ (meses) = R\$ 47,22 (Quarenta e sete centavos e vinte e dois centavos) por mês;

Custo de um dia: $R\$ 47,22 / 30$ (dias) = R\$ 1,57 (um real e cinquenta e sete centavos) por dia;

Custo de 167 dias: $R\$ 1,57 * 167 = R\$ 262,19$ (duzentos e sessenta e dois reais e dezenove centavos).

Impressora: R\$ 349,00 (Trezentos e quarenta e nove reais)

Depreciação 3 anos: $R\$349,00 / 36$ (meses) = R\$8,88 (Oito reais e oitenta e oito centavos) por mês;

Custo de um dia: $R\$8,88 / 30$ (dias) = R\$ 0,29 (Vinte e nove centavos) por dia;

Custo de 167 dias: $R\$0,29*167 = R\$49,48$ (Quarenta e nove reais e quarenta e oito centavos).

Sistema Operacional Windows 7 Home Premium: R\$329,00 (Trezentos e vinte e nove reais);

Depreciação 2 anos: $R\$329,00 / 24$ (meses) = R\$13,78 (Treze reais e setenta e oito centavos) por mês;

Custo de um dia: $R\$13,78 / 30$ (dias) = R\$0,45 (Quarenta e cinco centavos) por dia;

Custo de 167 dias: $R\$0,45*167 = R\$76,31$ (Setenta e seis reais e trinta e um centavos).

5 – IMPLEMENTAÇÃO DO SISTEMA

Para a implementação do sistema foi utilizado o ambiente de desenvolvimento *Dreamweaver CS5* com a linguagem de programação *ColdFusion 9*.

5.1 - ORGANIZAÇÃO DE PACOTE E CLASSES DO SISTEMA

Para um melhor entendimento o sistema foi organizado em camadas. A Figura 28 apresenta dentro da pasta *Site – Stelise*, contém as 4 pastas (*application*, *classes*, *core*, *imagens*) além dos arquivos que foram utilizados para a criação da interface e controle de usuários via *Login*, sendo eles *_footer.cfm*, *_header.cfm*, *_login_post.cfm*, *_logout.cfm*, *_menu.cfm*, *_title.cfm*, *Application.cfc*, *home.cfm*, *index.cfm*, *limpartabelas.cfm*, *login.cfm*.

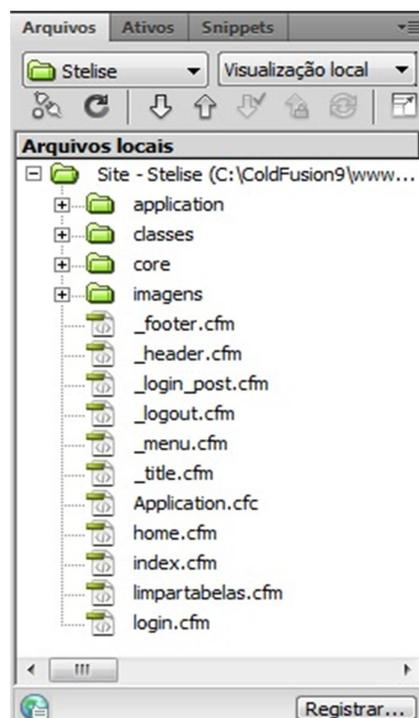


Figura 28 – Organização do Sistema em Camadas

Pasta Site – Stelise: Onde contém as 4 pastas que compõem o sistema.

Pasta *application*: É onde se encontra as subpastas como clientes, funcionários, fornecedores, etc. que armazenam as respectivas páginas para a criação do *CRUD* (*Create, Read, Update e Delete* – Criação, Leitura, Atualização e Exclusão) e movimentações do sistema.

Pasta *classes*: Nesta pasta estão definidas todas as classes responsáveis pela conexão com o banco de dados e também pelas operações de consultas, funções de cadastros, atualização e exclusão de dados, etc.

Pasta *core*: Nesta pasta estão armazenados os componentes utilizados no sistema, formando uma biblioteca que facilita a implementação. Nela também se encontram toda a parte de *CSS* (*Cascading Style Sheets* – Estilos de Folha em Cascata) responsável pela estética das páginas.

Pasta *imagem*: Onde se encontram as imagens utilizadas para a criação dos ícones e o logo da empresa do cliente solicitante.

5.2 – INTERFACE DO SISTEMA

Ao acessar a página inicial o usuário se depara com o *login* do sistema, como mostra a Figura 29.

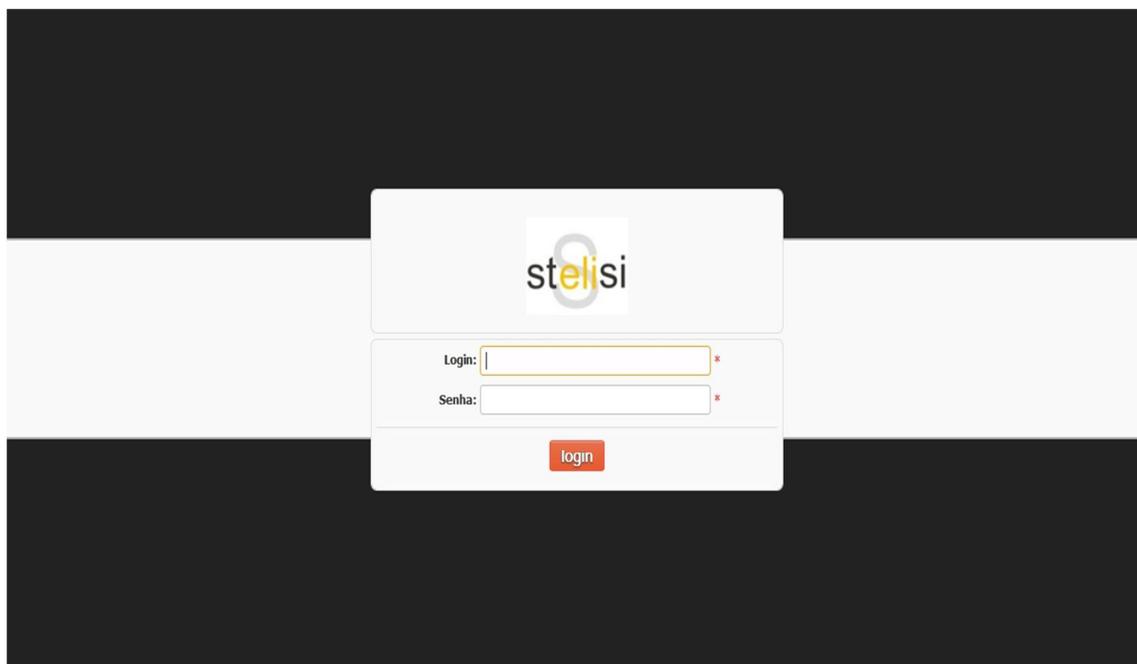


Figura 29 – Login do Usuário.

Após efetuar o *login* o usuário terá acesso ao menu principal, que conterà todas as funcionalidades do sistema: cadastros, movimentações, relatórios, controle de usuários como mostra a Figura 30. Porém cada usuário só terá acesso às funções permitidas a ele.

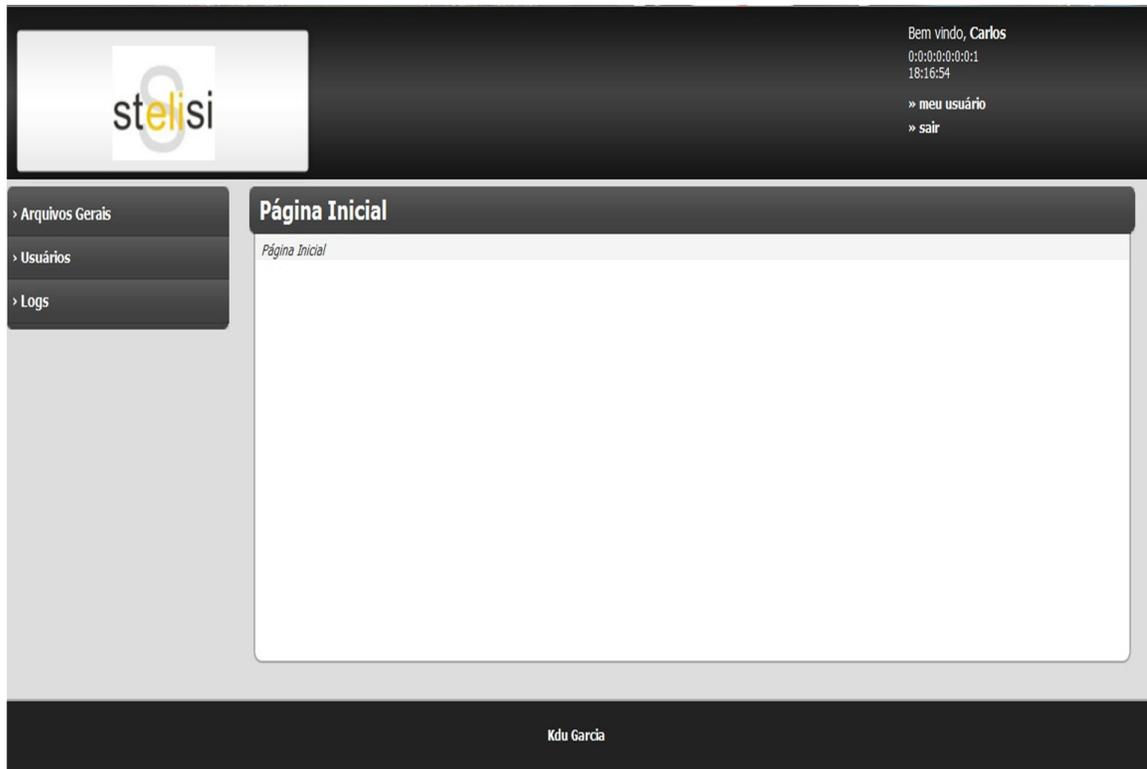


Figura 30 – Menu Principal

No menu Arquivos Gerais serão encontrados os sub menus de cadastro e gerenciamento de Usuários, Fornecedores, Funcionários, Produtos, Tipos de Produto, etc. Toda a parte responsável pelos cadastros do sistema, seguindo um mesmo padrão, uma tela dividida em abas, uma destinada a dados pessoais e outra a dados de contato como ilustra a Figura 31.

> Arquivos Gerais

> Clientes

- > Cadastrar Cliente
- > Gerenciar Cliente

> Funcionarios

> Fornecedores

> Produtos

> Tipos de Produto

> Usuários

> Logs

Cadastrar Cliente

Página Inicial » Arquivos Gerais » Clientes » Cadastrar Cliente

Dados Pessoais | Dados de Contato

Tipo Pessoa: Pessoa Física

Nome: *

Sexo: - Seleccione -

CPF:

RG:

Status: Ativo

cadastrar

Figura 31– Cadastro Cliente

A Figura 32 traz o código utilizado para a criação da interface de cadastro e atualização dos dados.

O código da linha 1 apresenta a importação da biblioteca *components* criada com linguagem *ColdFusion* em */stelise/core/componentes/cf/*, e denominado seu prefixo como *isbox*. Por este motivo todas as *tags ColdFusion* do sistema se iniciam com o *isbox* ou *cf* que é padrão da linguagem.

```

1 <cfimport prefix="isbox" taglib="/stelise/core/components/cf/">
2
3 <isbox:permission routine="#IsDefined('URL.Id') ? Application.Enum.getRotina().Clientes.GerenciarClientes.AtualizarCliente.Id : App:
4 <cfinclude template="cliente_script.cfm">
5 <cfparam name="URL.Id" default="">
6
7 <isbox:form name="Form" method="ajax" action="application/arquivosgerais/clientes/cliente.cfc" returnType="json">
8
9     <cfoutput><input type="hidden" name="Id" value="#URL.Id#"></cfoutput>
10
11 <div class="tab">
12     <div class="tab_header">
13         <div class="selected">Dados Pessoais</div>
14         <div class="selected">Dados de Contato</div>
15     </div>
16     <div class="tab_content">
17         <!--Dados Pessoais-->
18         <div>
19             <div class="row">
20                 <isbox:select name="TipoPessoa" label="Tipo Pessoa:" value="1" style="width:150px;" onchange="change_pessoa()"
21                     array="#Application.Enum.Array('TipoPessoa')#">
22                     <option value="[Id]">[Descricao]</option>
23                 </isbox:select>
24             </div>
25

```

Figura 32 – Cadastro Cliente.

Na linha 20, apresenta-se o comando *select* criado para determinar o tipo de pessoa (física ou jurídica).

Como pode ser visto a *tag* se inicia da seguinte forma:

<isbox: inicia componente a ser implementado **name=**"nome do componente" **label=**"descrição do campo" **value=**"valor do componente" **style=**"determina o tamanho do campo" **onchange=**"chama a função criada em JavaScript para dar o efeito de esconder os campos" **array=**"determina os valores a serem mostrados">

<option value="[valor que determinará a ordem a ser mostrada]">[variável]</option>

</isbox: finaliza componente

A Figura 33 retrata o código da função *change_pessoa()*, utilizada na *tag* exemplificada. Esta função tem por objetivo mostrar e ocultar os campos de pessoa física e pessoa jurídica, da seguinte forma, se o valor da *select* com o *name=TipoPessoa* tiver o valor igual a 1, mostrar todas as *input* e *select* que estão dentro da *tag div* com o *id=pessoa_fisica* e esconder os *input* dentro com da *tag div*

com o `id=pessoa_juridica`. Senão mostrar os input com a tag `div` com o `id=pessoa_juridica` e esconder todos os `input` e `select` que estão dentro da tag `div` com o `id=pessoa_fisica`.

Lembrando que `id` de uma tag em *ColdFusion* é representado pelos símbolo `#` em *JavaScript*.

```
19 function change_pessoa() {
20     if($("#select[name=TipoPessoa]").val() == 1) {
21
22         $("#pessoa_fisica").show().find("input").removeAttr("disabled");
23         $("#pessoa_fisica").show().find("select").removeAttr("disabled");
24         $("#pessoa_juridica").hide().find("input").attr("disabled", "true");
25
26     }
27     else {
28
29         $("#pessoa_juridica").show().find("input").removeAttr("disabled");
30         $("#pessoa_fisica").hide().find("input").attr("disabled", "true");
31         $("#pessoa_fisica").hide().find("select").attr("disabled", "true");
32     }
33 }
```

Figura 33 – Função em JavaScript.

As Figuras 34 e 35 mostram os códigos que são chamados de acordo com o resultado do teste determinado na função `change_pessoa ()`.

```

26 <div id="pessoa_fisica">
27   <div class="row">
28     <isbox:input name="Nome" label="Nome:" style="width:350px;" required="true" requiredMessage="Preencha o campo nome." maxLength="200" />
29   </div>
30
31   <div class="row">
32     <isbox:select name="Sexo" label="Sexo:" firstOptionLabel="- Selecione -" array="#Application.Enum.Array('Sexo')#">
33       <option value="[Id]">[Descricao]</option>
34     </isbox:select>
35   </div>
36
37   <div class="row">
38     <isbox:input name="Cpf_Cnpj" label="CPF:" style="width:90px;" validate="cpf" validateMessage="Preencha um cpf válido." mask="cpf" />
39   </div>
40
41   <div class="row">
42     <isbox:input name="Rg_Ie" label="RG:" style="width:130px;" />
43   </div>
44 </div>

```

Figura 34 – Div Pessoa Física

```

46 <div id="pessoa_juridica">
47   <div class="row">
48     <isbox:input name="Nome" label="Nome Fantasia:" style="width:350px;" required="true" requiredMessage="Preencha o campo nome fantasia." maxLength="200" />
49   </div>
50
51   <div class="row">
52     <isbox:input name="RazaoSocial" label="Razão Social:" style="width:350px;" maxLength="200" />
53   </div>
54
55   <div class="row">
56     <isbox:input name="Cpf_Cnpj" label="CNPJ:" style="width:120px;" validate="cnpj" validateMessage="Preencha um cnpj válido." mask="cnpj" />
57   </div>
58
59   <div class="row">
60     <isbox:input name="Rg_Ie" label="IE:" style="width:130px;" />
61   </div>
62
63   <div class="row">
64     <isbox:input name="Im" label="IM:" style="width:130px;" />
65   </div>
66 </div>

```

Figura 35 – Div Pessoa Jurídica

A Figura 36 demonstra os resultados possíveis no Cadastro de Clientes na aba Dados Pessoais.

The figure displays three screenshots of the 'Cadastrar Cliente' (Register Client) form, specifically the 'Dados Pessoais' (Personal Data) tab. The top screenshot shows the 'Tipo Pessoa' (Person Type) dropdown menu open, with 'Pessoa Física' (Natural Person) selected. The 'Nome' (Name) field is empty and marked with a red asterisk. Below it are fields for 'Sexo' (Sex), 'CPF', 'RG', and 'Status' (set to 'Ativo').

The bottom-left screenshot shows the form for a 'Pessoa Física' (Natural Person). The 'Tipo Pessoa' dropdown is set to 'Pessoa Física'. The 'Nome' field is empty and marked with a red asterisk. Other fields include 'Sexo', 'CPF', 'RG', and 'Status' (set to 'Ativo').

The bottom-right screenshot shows the form for a 'Pessoa Jurídica' (Legal Person). The 'Tipo Pessoa' dropdown is set to 'Pessoa Jurídica'. The 'Nome Fantasia' (Trade Name) field is empty and marked with a red asterisk. Other fields include 'Razão Social' (Social Reason), 'CNPJ', 'IE', 'IM', and 'Status' (set to 'Ativo').

Figura 36— Exemplos de Cadastro - Tipo Pessoa

6 – CONCLUSÃO

O software oferecerá um melhor funcionamento para a loja, facilitando o atendimento aos clientes e melhorando o controle das movimentações das vendas, produtos em estoque, entre outros. Além disso, trará facilidades ao acesso das informações importantes para o auxílio na administração da mesma.

Como a informática hoje é essencial em qualquer ramo de atividade, o software irá suprir toda necessidade de organização, fornecendo relatórios gerenciais que são de extrema importância para uma eventual tomada de decisão, garantindo assim a qualidade do serviço prestado.

O levantamento de requisitos e funcionalidades para o sistema foi realizado através de entrevista; Após, essa etapa foram feitas as modelagens do sistema, como: caso de uso, diagrama de classe, diagrama de sequência, diagrama de atividades e DER.

Os diagramas da UML foram muito importantes para o desenvolvimento do sistema, pois permitiram ter uma visão completa sobre o mesmo. Sem a modelagem seria difícil dar continuidade no desenvolvimento da aplicação.

A maior dificuldade na execução desse projeto foi o fato de possuir pouca experiência com a linguagem *ColdFusion*, escolhida para a implementação. Por outro lado foi um trabalho de suma importância, pois auxiliou em meu aprendizado além de me auxiliar na empresa pela qual fui contratado, uma vez que trabalho com tal linguagem.

REFERÊNCIAS

BROOKS-BILSON, Rob; Programação ColdFusion: criando aplicações dinâmicas para web/Rob Brooks-Bilson; tradução de Katia Roque.- Rio de Janeiro, 2001. Disponível em:

http://books.google.com.br/books?id=tn1zJH0ocMMC&printsec=frontcover&dq=coldfusion&hl=pt-BR&ei=OPL_TYqTKcbv0gGCprSwDg&sa=X&oi=book_result&ct=result&resnum=2&ved=0CDMQ6AEwAQ#v=onepage&q&f=false> Acesso em 10 de abril de 2011.

Documentação completa de jQuery, utilizado na construção do sistema apresentado. Disponível em: http://docs.jquery.com/Main_Page

Documentação da linguagem de programação ColdFusion 9, utilizada para a construção do sistema apresentado. Disponível em: http://help.adobe.com/en_US/ColdFusion/9.0/Developing/index.html

FOWLER, Martin; UML Essencial: um breve guia para a linguagem-padrão de modelagem de objeto / Martin Fowler; trad. João Tortello.- 3.ed.- Porto Alegre: Bookman,2005. Disponível em: http://books.google.com.br/books?id=1rWK_0ginbcC&printsec=frontcover&source=gs_ge_summary_r&cad=0#v=onepage&q&f=false > Acesso em 20 de Setembro de 2011.

GOFFORD, Mat; Object-Oriented Programming in ColdFusion: break free from procedural programming and learn how to optimize your applications and enhance your skills using objects and design patterns – Packt Publishing,2009.

GUEDES, G. T. (2005). *UML 2 Guia de Consulta Rápida*. São Paulo: Novatec.

HORSTAMANN, Cay; Padrões e Projeto Orientados a Objetos: segunda edição – Bookman,2006.

<http://www.adobe.com/br/products/dreamweaver.html>> Acesso em 25 de maio de 2011.

<http://www.adobe.com/devnet/coldfusion.html?view=gettingstarted>> Acesso em 2 de abril de 2011.

<http://www.microsoft.com/brasil/servidores/sql/2005/prodinfo/overview/default.aspx>> Acesso em 25 de maio de 2011.

<http://www.scribd.com/doc/22483180/Artigo-WBS>> Acesso em 7 de Setembro de 2011.

MAGALHÃES, Antonio de Deus F., LUNKES, Irtes Cristina. Sistemas contábeis: o valor informacional da contabilidade nas organizações. São Paulo: Atlas, 2000.

MELO, Ana Cristina; Desenvolvendo Aplicações com UML 2.2: do conceito de implementação / Ana Cristina Melo.- 3.ed.- Rio de Janeiro: Brasport,2010. Disponível em:

<http://books.google.com.br/books?id=BPVHsG17bAYC&printsec=frontcover#v=onepage&q&f=false> >Acesso em 25 de Setembro de 2011.

OLIVEIRA, Antonio Ricardo de; FERNANDES, Leila Maria Pinheiro; TAVEIRA, Gilda Aché; Modelagem de Dados, Rio de Janeiro: Ed. SENAC Nacional,2000.