

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE – UERN  
FACULDADE DE CIÊNCIAS EXATAS E NATURAIS – FANAT  
DEPARTAMENTO DE INFORMÁTICA – DI  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

LEONARDO ALVES DA SILVA

**Desenvolvimento de um Sistema de Gestão de Estoque para  
Plataforma Web.**

LEONARDO ALVES DA SILVA

**Desenvolvimento de um Sistema de Gestão de Estoque para  
Plataforma Web.**

Monografia apresentada à Universidade do Estado do Rio Grande do Norte como um dos pré-requisitos para obtenção do grau de bacharel em Ciência da Computação, sob orientação do Prof. Dr. Henrique Jorge Amorim Holanda.

**Catálogo da Publicação na Fonte.**  
**Universidade do Estado do Rio Grande do Norte.**

S586d Silva, Leonardo Alves da  
Desenvolvimento de um Sistema de Gestão de  
Estoque para Plataforma Web.. / Leonardo Alves da Silva.  
- Mossoró RN, 2018.  
83p.

Orientador(a): Prof. Dr. Henrique Jorge Amorim  
Holanda.

Monografia (Graduação em Ciência da Computação).  
Universidade do Estado do Rio Grande do Norte.

1. Gestão de Estoque. 2. Papelaria. 3. Projeção de  
Vendas. I. Holanda, Henrique Jorge Amorim. II.  
Universidade do Estado do Rio Grande do Norte. III.  
Título.

**Leonardo Alves Da Silva**

DESENVOLVIMENTO DE UM SISTEMA DE GESTÃO DE ESTOQUE PARA PLATAFORMA WEB

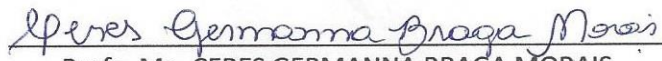
Monografia apresentada como pré-requisito para a obtenção do título de Bacharel em Ciência da Computação da Universidade do Estado do Rio Grande do Norte – UERN, submetida à aprovação da banca examinadora composta pelos seguintes membros:

Aprovada em: 21/06/2018

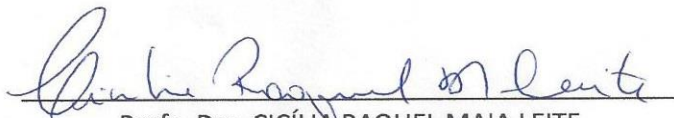
**Banca Examinadora**



Prof. Dr. HENRIQUE JORGE AMORIM HOLANDA  
Universidade do Estado do Rio Grande do Norte - UERN



Profa. Ma. CERES GERMANNA BRAGA MORAIS  
Universidade do Estado do Rio Grande do Norte - UERN



Profa. Dra. CÍCÍLIA RAQUEL MAIA LEITE  
Universidade do Estado do Rio Grande do Norte - UERN

Aos que acreditaram em mim  
desde sempre, meu pai e minha mãe.

## AGRADECIMENTOS

Agradeço primeiramente a Deus, pois confiando nele iluminou meu caminho e fez com que conseguisse chegar até aqui. A minha família por sempre acreditar em mim, principalmente meus pais, João e Rita que sempre me deram força e estiveram o tempo todo comigo, apoiando-me e incentivando-me em todos os momentos. À minha esposa Maria Neuza, pelo carinho, compreensão, companheirismo, e por sua presença nos momentos em que mais necessitei.

A minha filha Maria Luiza, um presente que Deus colocou em minha vida para fazer meus dias melhores, e me dar motivação para seguir em frente com os objetivos. Ao meu orientador, Prof. Dr. Henrique Jorge Amorim Holanda que me auxiliou nesta jornada, compartilhando suas ideias e reflexões, e também por seu apoio e compreensão. A todos os amigos, colegas e conhecidos que de certa forma contribuíram para a minha formação.

Agradeço a todos os professores do DI/UERN: Prof<sup>a</sup>. Círcia Raquel Maia Leite, Prof<sup>a</sup>. Ceres Germanna Braga Morais, Prof. Marcelino Pereira dos Santos Silva, Prof. Antonio Oliveira Filho, Prof. Henrique Jorge Amorim Holanda, Prof<sup>a</sup>. Carla Katarina de Monteiro Marques, Prof. Francisco Chagas de Lima Júnior, Prof. Sebastião Emidio Alves Filho, Prof. Rommel Wladimir de Lima, Prof. André Pedro Fernandes Neto, Prof. Carlos Heitor Pereira Liberalino, Prof<sup>a</sup>. Jessica Neiva de Figueiredo Leite, Prof<sup>a</sup>. Alexandra Ferreira Gomes, Prof. Alysson Mendes de Oliveira, Prof. Dario José Aloise, Prof. Harold Ivan Angulo Bustos e Prof. Pedro Fernandes Ribeiro Neto. Pelos ensinamentos, contribuição para meu desenvolvimento intelectual e pelos conselhos motivacionais. Saibam que vocês professores não formam apenas profissionais para mercado, mas também seres humanos com hábitos formidáveis para o mundo. Por fim, o meu muito obrigado a todos.

Se você não puder se destacar pelo talento, vença pelo esforço.  
Dave Weinbaum

## RESUMO

A gestão de estoque é hoje um fator determinante para o sucesso de uma empresa. Por representar uma parte importante dos ativos é necessário que se tenha um sistema que supra as necessidades de gestão de estoque da empresa. A empresa Papelaria Real, possui uma variedade de produtos em seu estoque, sem nenhum controle rígido. O objetivo deste trabalho é identificar os diferentes fatores que influenciam o estoque, desenvolvendo um sistema de gestão de estoque para a empresa. Um modelo de sistema com controle de entrada e saída de produtos, produtos mais e menos vendidos e projeção de vendas é proposto para resolver o controle dos produtos da empresa. A análise realizada mostrou uma utilização incorreta dos recursos da empresa, tanto de espaço físico, quanto de datas de compras e tipos de produtos a serem comprados em cada momento. Baseados na análise dos dados e da situação são esboçados algumas recomendações, através do desenvolvimento do sistema. Para o desenvolvimento a linguagem JAVA foi utilizada, com gerenciador de banco de dados MySQL e HTML, JavaScript e Bootstrap para interface.

**Palavras-chave:** Gestão de Estoque. Papelaria. Projeção de Vendas.



## **ABSTRACT**

Stock management is now a determining factor for a company's success. Because it represents a significant part of the assets, it is necessary to have a system that meets the inventory management needs of the company. The company Papelaria Real, owns a variety of products in its stock, without any rigid control. The objective of this work is to identify the different factors that influence the inventory, developing an inventory management system for the company. A system model with input and output control of products, more and less sold products and sales projection is proposed to solve the control of the company's products. The analysis showed an incorrect use of the company's resources, both physical space, dates of purchases and types of products to be purchased at any time. Based on data and situation analysis, some recommendations are outlined through the development of the system. For development the JAVA language was used, with MySQL database manager and HTML, JavaScript and Bootstrap for interface.

**Keywords:** Inventory Management. Stationary store. Sales projection.

## LISTA DE ABREVIATURAS E SIGLAS

BD	<i>Banco de Dados</i>
CPF	<i>Cadastro de Pessoa Física</i>
CSS	<i>Cascading Style Sheets</i>
DML	Linguagem de Manipulação de Dados
DDL	Linguagem de Definição de Dados
DTD	<i>Document Type Definition</i>
ECT	<i>Empresa Brasileira de Correios e Telégrafos</i>
HTML	<i>Hyper Text Markup Language</i>
HyTime	<i>Hypermedia/Time-based Document Structuring Language</i>
ISO	International Organization for Standardization
IDE	Integrated Development Environment
JS	<i>JavaScript</i>
JSP	Java Server Pages
MER	<i>Modelo Entidade Relacionamento</i>
PEPS	<i>Primeiro a Entrar, Primeiro a Sair</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	Structured Query Language
SGML	<i>Standart Generalized Markup Language</i>
UML	<i>Unified Modeling Language</i>
UEPS	<i>Último a Entrar, Primeiro a Sair</i>
WEB	<i>World Wide Web</i>

## LISTA DE FIGURAS

Figura 1: Custos dos materiais em uma empresa industrial .....	19
Figura 2: Fórmula para cálculo do estoque mínimo .....	21
Figura 3: Cálculo do custo médio .....	22
Figura 4: Padrões de demanda .....	22
Figura 5: Controle de estoque .....	24
Figura 6: Saída de material.....	25
Figura 7: Diagrama de casos de uso do sistema de validação de senha .....	35
Figura 8: Diagrama de casos de uso: Usuário autorizado .....	36
Figura 9: Diagrama de casos de uso: Usuário .....	36
Figura 10: Diagrama de atividade: Usuário - login.....	37
Figura 11: Diagrama de atividade: Usuário - autenticar.....	38
Figura 12: Diagrama de atividade: Abastecimento de estoque.....	39
Figura 13: Diagrama de sequência: Cadastrar produto .....	39
Figura 14 Diagrama de sequência: Consultar produto .....	40
Figura 15 Diagrama de sequência: Alterar produto .....	40
Figura 16: Diagrama de sequência: Excluir produto .....	41
Figura 17: Diagrama de classes: Cliente .....	42
Figura 18: Diagrama de classes: Visão geral da papelaria.....	43
Figura 19: Diagrama de objetos: Usuário .....	44
Figura 20: Diagrama de estados: Usuário .....	44
Figura 21: Modelo conceitual.....	46
Figura 22: Modelo lógico.....	47
Figura 23: Trigger - inserção de registros na tabela compra de produtos. ....	48
Figura 24: Trigger - atualização de registros na tabela compra de produtos ..	49
Figura 25: Trigger - exclusão de registros na tabela compra de produtos .....	49
Figura 26: Trigger - inserção de registros na tabela Venda de produtos .....	50
Figura 27: Trigger - atualização de registros na tabela venda de produtos ....	50
Figura 28: Trigger - exclusão de registros na tabela venda de produtos .....	51
Figura 29: Procedure - atualização de registros na tabela estoque.....	51
Figura 30: Tela de login .....	52
Figura 31: Tela de login: autenticar .....	53
Figura 32: Código fonte da tela de login: autenticar .....	53

## LISTA DE FIGURAS

Figura 33: Código fonte da tela de login .....	54
Figura 34: Tela inicial.....	54
Figura 35: Código fonte do cabeçalho do sistema.....	55
Figura 36: Menu cadastrar.....	55
Figura 37: Campos para cadastro de usuário.....	56
Figura 38: Código fonte da tela cliente: Acesso a dados.....	56
Figura 39: Menu pesquisar .....	57
Figura 40: Excluir fornecedor.....	57
Figura 41: Código Fonte da tela fornecedor: Acesso a dados.....	58
Figura 42: Código fonte da tela Fornecedor: formulário .....	58
Figura 43: Menu de produto.....	59
Figura 44: Tela excluir produto .....	60
Figura 45: Código fonte da tela produto: Acesso a dados.....	60
Figura 46: Menu de compra.....	61
Figura 47: Código fonte da tela compra: Acesso a dados .....	62
Figura 48: Menu de venda.....	63
Figura 49: Código fonte da tela venda: Acesso a dados .....	63
Figura 50: Código fonte da tela venda: Acesso a dados .....	64
Figura 51: Menu estoque.....	64
Figura 52: Código fonte da tela consultar estoque: Acesso a dados.....	65
Figura 53: Código fonte da tela estoque: Acesso a dados.....	65
Figura 54: Opção sair .....	66

## LISTA DE QUADROS

Quadro 1: Requisito funcional: fazer login.....	72
Quadro 2: Requisito funcional: cadastrar produtos .....	72
Quadro 3: Requisito funcional: alterar produtos .....	72
Quadro 4: Requisito funcional: excluir produtos .....	72
Quadro 5: Requisito funcional: realizar compra.....	72
Quadro 6: Requisito funcional: alterar compra .....	73
Quadro 7: Requisito funcional: excluir compra .....	73
Quadro 8: Requisito funcional: realizar venda.....	73
Quadro 9: Requisito funcional: alterar venda .....	73
Quadro 10: Requisito funcional: excluir venda .....	73
Quadro 11: Requisito funcional: consultar estoque .....	74
Quadro 12: Requisito funcional: cadastrar cliente.....	74
Quadro 13: Requisito funcional: alterar cliente.....	74
Quadro 14: Requisito funcional: excluir cliente.....	74
Quadro 15: Requisito funcional: cadastrar fornecedor .....	74
Quadro 16: Requisito funcional: alterar fornecedor .....	75
Quadro 17: Requisito funcional: excluir fornecedor.....	75
Quadro 18: Requisito funcional: cadastrar funcionário.....	75
Quadro 19: Requisito funcional: alterar funcionário.....	75
Quadro 20: Requisito funcional: excluir funcionário .....	75
Quadro 21: Requisito funcional: consultar dados da empresa .....	76
Quadro 22: Requisito não funcional: usabilidade .....	77
Quadro 23: Requisito não funcional: login.....	77
Quadro 24: Requisito não funcional: funcionalidades do sistema .....	77
Quadro 25: Requisito não funcional: funcionalidades do sistema .....	77
Quadro 26: Requisito não funcional: armazenamento de dados.....	78
Quadro 27: Requisito não funcional: confiabilidade .....	78
Quadro 28: Requisito não funcional: confiabilidade .....	78
Quadro 29: Requisito não funcional: segurança.....	78
Quadro 30: Requisito não funcional: login.....	78
Quadro 31: Requisito não funcional: cadastrar produto .....	79
Quadro 32: Requisito não funcional: cadastrar cliente .....	79

## LISTA DE QUADROS

Quadro 33: Requisito não funcional: cadastrar fornecedor .....	79
Quadro 34: Requisito não funcional: excluir compra .....	79
Quadro 35: Requisito não funcional: excluir venda .....	79
Quadro 36: Requisito não funcional: disponibilidade.....	80
Quadro 37: Requisito não funcional: linguagem de implementação.....	80
Quadro 38: Requisito não funcional: interface.....	80
Quadro 39: Requisito não funcional: banco de dados .....	80
Quadro 40: Requisito não funcional: desempenho.....	80
Quadro 41: Requisito não funcional: segurança.....	80
Quadro 42: Rastreabilidade.....	81
Quadro 43: Dicionário de dados da tabela usuários.....	82
Quadro 44: Dicionário de dados da tabela produtos .....	82
Quadro 45: Dicionário de dados da tabela fornecedores .....	82
Quadro 46: Dicionário de dados da tabela compra .....	82
Quadro 47: Dicionário de dados da tabela estoque .....	83
Quadro 48: Dicionário de dados da tabela clientes .....	83
Quadro 49: Dicionário de dados da tabela vendas.....	83
Quadro 50: Dicionário de dados da tabela Administrador .....	83
Quadro 51: Dicionário de dados da tabela funcionários.....	84
Quadro 52: Dicionário de dados da tabela papelaria .....	84

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	17
1.1	OBJETIVOS DO TRABALHO	18
1.2	ESTRUTURA DO TRABALHO	18
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	19
2.1	CONTROLE DE ESTOQUE	19
2.2	ESTOQUE MÍNIMO	20
2.3	AValiação DOS ESTOQUES	21
2.4	PROJEÇÃO DE VENDAS	22
2.5	SISTEMA ATUAL	23
2.6	TRABALHOS CORRELATOS	24
<b>3</b>	<b>TÉCNICAS E FERRAMENTAS UTILIZADAS</b>	26
3.1	UML	26
3.2	SQL	27
3.3	JAVA	28
3.4	HTML	28
3.5	JavaScript	29
3.6	JSP	30
3.7	BOOTSTRAP	30
3.8	ASTAH	31
3.9	brModelo	32
<b>4</b>	<b>DESENVOLVIMENTO DO SISTEMA</b>	33
4.1	DESCRIÇÃO DA EMPRESA	33
4.2	ESPECIFICAÇÃO DOS REQUISITOS	34
<b>4.3</b>	<b>MODELAGEM DE REQUISITOS</b>	35
4.3.1	DIAGRAMAS DE CASOS DE USO	35
4.3.2	DIAGRAMAS DE ATIVIDADES	37
4.3.3	DIAGRAMAS DE SEQUÊNCIA	39
4.3.4	DIAGRAMAS DE CLASSES	41
4.3.5	DIAGRAMAS DE OBJETOS	43
4.3.6	DIAGRAMAS DE ESTADOS	44
<b>4.4</b>	<b>MODELAGEM DO BANCO DE DADOS</b>	45
4.4.1	MODELO CONCEITUAL	45
4.4.2	MODELO LOGICO	47
4.4.3	TRIGGERS	48
4.4.4	STORED PROCEDURE	51

## SUMÁRIO

<b>4.5 PROJETO DO SISTEMA</b> .....	52
4.5.1 IMPLEMENTAÇÃO (CODIGOS E INTERFACE) .....	52
<b>5 RESULTADOS E DISCUSSÃO</b> .....	67
<b>6 CONSIDERAÇÕES FINAIS</b> .....	68
<b>REFERÊNCIAS</b> .....	70
<b>APÊNDICE A – REQUISITOS FUNCIONAIS</b> .....	72
<b>APÊNDICE B – REQUISITOS NÃO FUNCIONAIS</b> .....	77
<b>APÊNDICE C – TABELAS E LISTAS DE RASTREABILIDADE</b> .....	81
<b>APÊNDICE D – DICIONÁRIO DE DADOS</b> .....	82



## 1 INTRODUÇÃO

O estoque é um dos itens mais importantes para a empresa. Segundo o autor Dias (2009), o estoque deve ser bem administrado e quando não administrado de forma correta, pode causar sérios prejuízos às finanças, colocando em perigo a saúde financeira do empreendimento. É de fundamental importância que as empresas tenham um sistema para realizar o controle das mercadorias em estoque para evitar os problemas apresentados pela má administração.

O principal objetivo do controle de estoque é aperfeiçoar o investimento em estoques, para que se possa aumentar o uso eficiente dos meios internos de uma empresa, e diminuir as necessidades de capital investido em estoque (DIAS, 2009). A realização do controle de estoque de forma eficiente permite diminuir os desperdícios de tempo, dinheiro e esforços da empresa com estoque. Para alcançar estes objetivos, o controle de estoque integra as atividades de registrar, fiscalizar e gerenciar a entrada e saída de produtos dentro da empresa.

Investigando a Papelaria Real, situada no município de Mossoró, Estado do Rio Grande do Norte, foi verificada a necessidade de criar um sistema para realizar o controle do estoque da empresa, pois o método que é utilizado atualmente não é eficaz e nem eficiente, e acaba exibindo o controle de estoque da empresa a vários equívocos. O controle de estoque da papelaria Real não é controlado através de um sistema, mas sim por planilhas elaboradas por funcionário da empresa.

Por não serem uniformes, as informações são de difícil tratamento e baixa utilidade para o acompanhamento gerencial. Além dos fatores já citados, a inexistência de um controle eficaz no estoque acarreta atrasos nos pedidos e dificulta a programação das compras a serem feitas. Principalmente quando há eventos a serem feitos, uma apuração correta das quantidades estocadas é de suma importância.

Também para um melhor conhecimento dos rendimentos dos processos e da característica do consumo de cada área, é importante que a quantidade de produtos comprados, em estoque e consumidos sejam conhecidas. Portanto foi solicitado pela microempresa no ramo da papelaria um sistema para manter o controle de seu estoque e assim tornar o processo mais eficaz e eficiente.

Foi verificado que os controles de estoque que são realizados hoje na empresa é através de planilha de dados, isso acaba trazendo muita insegurança, incerteza e riscos para a empresa, que não pode ter informações inconsistentes. Através dessas informações, baseando-se na falta de controle e no crescente

número de produtos que entram no estoque, é altamente necessária a implantação de um controle que atenda tais necessidades.

Com todas essas necessidades apresentadas pela empresa, este trabalho busca desenvolver o sistema de gestão de estoque para que se possa agilizar o controle de estoque da empresa e tornar o processo mais eficiente, eficaz, confiante, indispensável e que garanta um histórico confiável das entradas e saídas dos produtos.

## 1.1 OBJETIVOS DO TRABALHO

O objetivo geral deste trabalho é realizar a especificação, projeto e implementação de um sistema para controlar todo tipo de movimentação, seja de venda ou compra de produtos, e a medida que for comprando ou vendendo produtos se possa realizar o controle do estoque da empresa em tempo real, assim possibilitando a organização e visando o aumento do lucro da empresa. Procura-se com este trabalho manter sempre os produtos disponibilizados, determinando o que deve permanecer no estoque, quando o estoque deve ser reabastecido, receber, armazenar e atender os materiais estocados.

## 1.2 ESTRUTURA DO TRABALHO

Este trabalho está organizado como segue: Capítulo 1 tem-se a Introdução ao tema principal deste trabalho com a apresentação da justificativa e dos objetivos, Capítulo 2 aborda a fundamentação teórica pesquisada, Capítulo 3 mostra as técnicas e ferramentas utilizadas para desenvolvimento do sistema, Capítulo 4 traz o desenvolvimento do sistema, Capítulo 5 é abordado os resultados e discussão, por último no Capítulo 6 mostra as considerações finais.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta assuntos tais como controle de estoque, estoque mínimo, avaliação dos estoques, projeção de vendas, sistema atual e trabalhos correlatos.

### 2.1 CONTROLE DE ESTOQUE

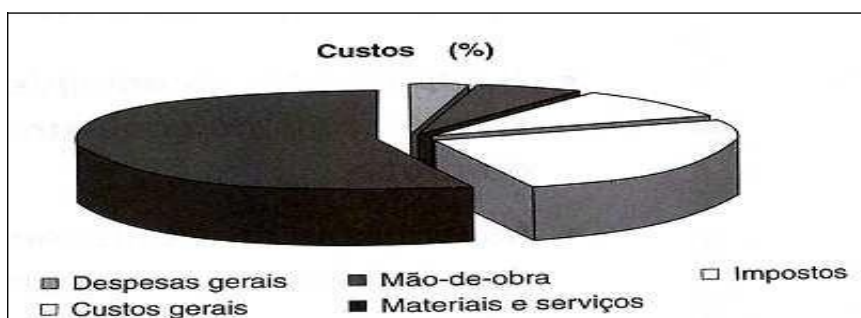
O principal objetivo de uma empresa é dar o máximo retorno que puder sobre o dinheiro que foi investido. Todas as formas de investimento que são realizados dentro da empresa devem ser bem avaliadas para que este dinheiro aplicado não fique parado. Um dos maiores conceitos da administração de materiais é analisar quais são as melhores maneiras de compras de materiais, estocagem, fabricação, vendas e rejeições de produtos e mercadorias em estoque para que os lucros e investimentos da empresa sejam potencializados (DIAS, 2009).

Investimento em estoques é muito importante para a empresa, desde que funcione em conformidade com a área de produção ou, em alguns casos, com a área de compras. Quando o volume e quantidade de produtos em estoque aumenta isso não quer dizer que a quantidade de vendas também irá aumentar.

Pode-se dizer que estoques são certa quantidade de produtos mantidos em disponibilidade para produzir lucros ou serviços. São lucros obtidos através de vendas de produtos a clientes, e essas vendas dão a possibilidade de continuar com o processo produtivo da empresa. (FERNANDES, 1987).

Conforme a Figura 1 apresenta, é possível analisar que mais da metade dos custos de uma empresa industrial são de investimentos em materiais e serviços para a produção.

Figura 1: Custos dos materiais em uma empresa industrial



Fonte: Gonçalves (2007 p.2).

Pode-se dizer que controle de estoque é o mecanismo utilizado para registrar, verificar e ajustar a entrada e saída de mercadorias e produtos seja em indústria ou

no comércio, como é o caso do estudo aplicado na papelaria Real. Deve-se aproveitar o controle de estoque tanto para matéria prima, mercadorias produzidas como para mercadorias vendidas.

Um dos principais diferenciais de uma empresa, quase sempre estará em disponibilizar o produto certo, pelo menor custo possível, na qualidade perfeita, com total crédito ao cliente final. Portanto é necessário que o empresário tenha uma preocupação em relação a “quando” e o “quanto” comprar. Pois não adianta ter a quantidade certa no tempo errado e também não adianta ter a quantidade errada no tempo certo.

Hoje em dia com a grande dificuldade em adquirir recursos financeiros para a empresa, e com o aumento da taxa de juros imposta pelo governo, é inviável colocar em prática algumas das teorias de controle de estoque que um dia já funcionaram. Essa corrente de coisas que acontecem, de planejamento e controle de materiais, chamada de Administração de Materiais, está seguindo uma forma mais ampla de estudo, e está sendo chamada de Logística (DIAS, 2009).

Por fim é verificado que níveis altos de estoques produzem conformidade com o erro e a escassez em estoques pode, por exemplo, prejudicar o andamento de produção e diminuir as vendas.

## 2.2 ESTOQUE MÍNIMO

Segundo Ching (2001, p.44), “o estoque mínimo ajuda a fazer o controle da quantidade adicional do estoque, necessária como proteção contra oscilação na demanda e no tempo de ressuprimento das mercadorias”. Pode-se dizer que estoque mínimo é a mínima quantidade que um produto poderá ficar em estoque. Esse estoque serve de aviso para o controlador do estoque saber quando obter novos produtos, para que assim não falte a mercadoria para os clientes.

Quando o estoque adicional é solicitado em excesso, gerará mais custos em dinheiro e armazenamento para a empresa. Porém se for calculado para menos, a ausência deste produto também gerará custos para a empresa, isto é chamado de custo de ruptura, e afeta diretamente o cliente, deixando-o insatisfeito com a falta do produto e baixa qualidade do atendimento (DIAS, 2009).

Conforme o estudo realizado com a Papelaria Real, foi analisado que o consumo de seus produtos varia e o tempo de reposição junto ao fornecedor permanece inalterado. Se o consumo também fosse constante, o estoque adicional poderia ser igual a zero, portanto para este caso em questão foi constatado que o método mais ideal a ser utilizado é o da porcentagem de consumo que será descrito a seguir.

Para fazer o cálculo do estoque mínimo é levam-se em consideração pelo método da porcentagem, segundo Dias (2009), três elementos:

- a) consumo médio: é a quantidade média de produtos que são consumidos;
- b) consumo máximo: é a quantidade máxima de produtos que são consumidos em um determinado período de tempo;
- c) tempo de reposição: é o tempo necessário que leva para que se possa repor os produtos.

Na Figura 2 é apresentado a fórmula para calcular o estoque mínimo.

Figura 2: Fórmula para cálculo do estoque mínimo

$$E.MÍNIMO = (C.Máximo - C.Médio) \times T.Reposição$$

Fonte: Dias (2009, p.65).

Um exemplo: temo o produto A com consumo máximo de 80 unidades/dia, e o consumo médio de 40 unidades/dia. O tempo de reposição do produto junto ao fornecedor é de 15 dias. Então o estoque mínimo do produto A deverá ser calculado da seguinte forma  $E.MÍNIMO = (80 - 40) \times 15$ , então  $E.MÍNIMO = 600$ . Lembrando que o tempo de reposição do produto tem que ser convertido para a mesma unidade que é utilizada na observação de demanda do produto.

### 2.3 AVALIAÇÃO DOS ESTOQUES

É indispensável que em uma empresa se tenha um controle de preço dos produtos. Conforme Dias (2009 p. 150), a avaliação dos estoques engloba o valor das mercadorias e dos produtos em fabricação.

A avaliação de estoque pode ser realizada através dos seguintes métodos: PEPS (Primeiro a Entrar, Primeiro a sair), UEPS (Último a Entrar, Primeiro a Sair), custo médio e custo de reposição. O custo médio é o único método aceito contabilmente para fins de fiscalização, ele é o mais usado nas empresas, porém no caso em questão da Papelaria Real, é necessário que se tenha como base o método PEPS.

O método PEPS leva em consideração a ordem de entrada dos produtos em estoque e dos preços de entrada. O procedimento de precificação leva em consideração essa ordem e as saídas são contabilizadas de acordo com os históricos das entradas e preços dos produtos (GONÇALVES, 2007).

Por fim é de fundamental importância para a empresa que os primeiros produtos a entrar no estoque sejam os primeiros a sair. No caso do método de custo

médio que obrigatoriamente deve ser utilizado em todas as empresas, é feito a média aritmética do valor de custo dos produtos que estão em estoque, como apresentado na Figura 3 abaixo.

Figura 3: Cálculo do custo médio

Dia	NF	ENTRADAS			SAÍDAS			SALDOS		
		Qte.	Preço Unit.	Total	Qte.	Preço Unit.	Total	Qte.	Total	Médio
7-8	001	500	15	7.500				500	7.500	15
8-8	002	200	20	4.000				700	11.500	16,43
23-9					150	16,43	2.464,50	550	9.035,50	16,43

Fonte: Dias (2009, p.151).

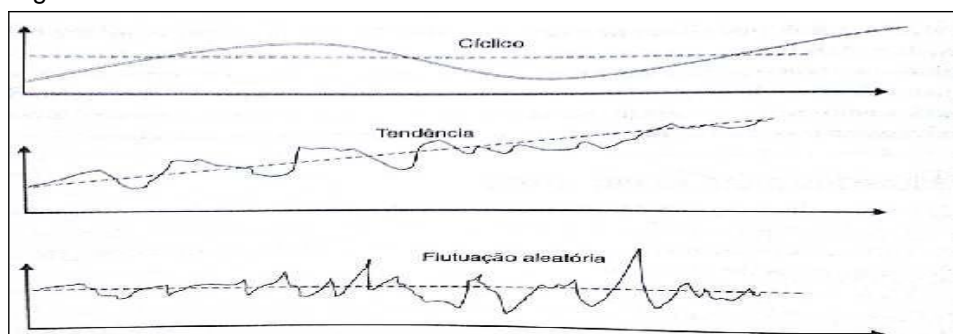
## 2.4 PROJEÇÃO DE VENDAS

Pode-se dizer que uma projeção de venda é uma previsão de consumo de determinado material, e está relacionada diretamente à quantidade de produtos que será vendido e a data que será vendido para o consumidor final.

Pode-se dizer que as falhas nas estimativas de demandas agregadas são bem menores que os resultantes de estimativas de demandas individuais. Além do mais, o erro nas estimativas futuras será maior conforme for aumentando as previsões (GONÇALVES, 2007).

Conforme os gráficos apresentados na Figura 4, é possível verificar que há três padrões de demanda, que possibilitam definir qual o melhor método a ser utilizado. O primeiro gráfico apresentado na Figura 4 mostra o modelo de evolução horizontal, com uma tendência de consumo constante. Já o segundo gráfico apresenta um modelo de evolução sujeito a tendência, e o consumo médio aumenta com o decorrer do tempo. Por fim o terceiro e último gráfico apresenta um modelo de evolução sazonal de consumo, nesse modelo é verificado que as oscilações são regulares.

Figura 4: Padrões de demanda



Fonte: Gonçalves (2007, p.20).

Analisando a Papelaria Real foi capaz de perceber que por tratar-se de produtos de alta tendência em certos períodos do ano, a empresa deve trabalhar com uma previsão utilizando o modelo de evolução sazonal de consumo, pois as oscilações são regulares.

Conforme estudado, o método de previsão de estoque a ser utilizado será o método do último período. Pois foi verificado que este método é apropriado para empresas que tenham tendências de consumo sazonais. O método do último período também é muito usado em empresas que não têm muita rotatividade de produtos. Ao se fazer uma previsão correta de vendas é obtido como resultado diversas vantagens, entre essas vantagens estão o não desperdício de recursos e preparação no atendimento da demanda do mercado consumidor (GONÇALVES, 2007).

## 2.5 SISTEMA ATUAL

Ao analisar a Papelaria Real, foi verificado que atualmente o controle de estoque dos produtos é realizado através de uma planilha eletrônica. Um funcionário da empresa fica responsável por atualizar a planilha e a mesma é protegida por senha onde somente esse funcionário tem acesso. Portanto como o funcionário tem outras tarefas a fazer e há muita entrada de produto, muitas das vezes mercadorias são esquecidas e não são registradas na planilha de dados.

Semanalmente o funcionário da empresa faz a contabilidade de forma manual do estoque para analisar se os dados da planilha estão de acordo com o número real de produtos disponíveis em estoque. Quando chega uma mercadoria na empresa o funcionário faz o controle da forma descrita abaixo:

1. Primeiro o funcionário faz uma comparação com o pedido que foi realizado e após isso realiza o pagamento ao fornecedor do produto;
2. Em seguida verifica se o pedido está de acordo com o solicitado, pois caso não esteja de acordo, é solicitado um novo pedido;
3. Caso esteja certo, os produtos serão encaminhados para um setor em que se possa fazer a contabilidade e registro dos produtos;
4. Por fim, é feito o registro dos produtos na planilha de dados, um por um, em seguida os produtos são encaminhados para o local de armazenamento.

Pelo motivo das informações não serem uniformes, acaba sendo de difícil tratamento e baixa utilidade para o acompanhamento gerencial da empresa. Além do

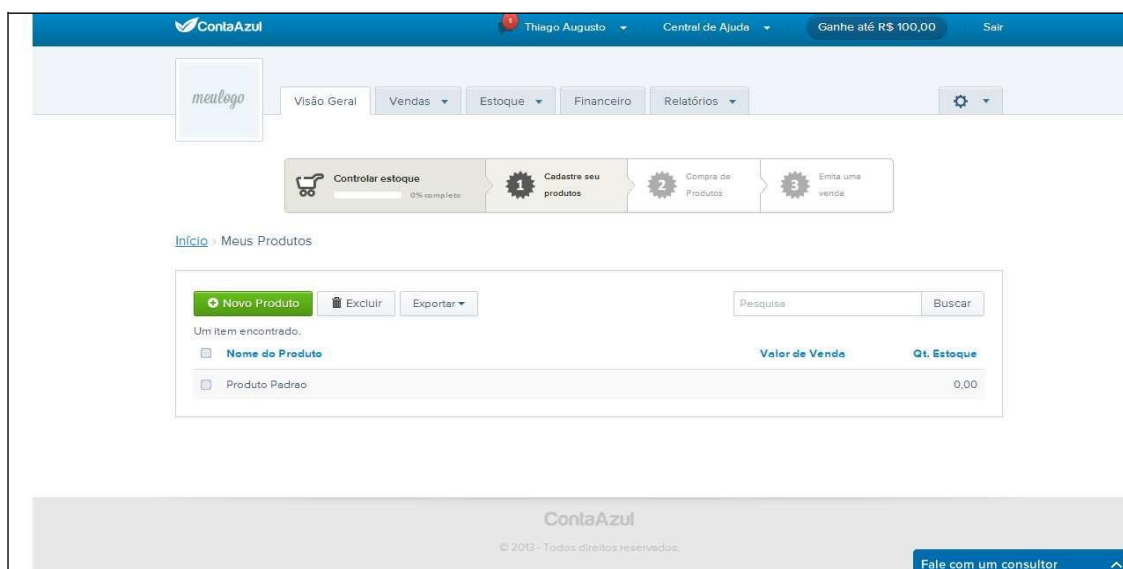
mais a ausência de um controle eficaz no estoque ocasiona atrasos nos pedidos e dificulta a programação das compras a serem realizadas.

Para um melhor conhecimento dos rendimentos dos processos e da característica do consumo, é de fundamental importância que a quantidade de produtos que são comprados, que ficam em estoque e que são consumidos sejam conhecidas. Por este motivo e outros foi solicitado pela microempresa no ramo da papelaria um sistema para manter o controle de seu estoque e assim tornar o processo mais eficaz e eficiente.

## 2.6 TRABALHOS CORRELATOS

A empresa Conta Azul localizada na cidade de Joinville, estado de Santa Catarina, tem o sistema online integrado de gestão Conta Azul, que possibilita fazer o controle do fluxo de caixa, estoque, vendas e clientes. E possui planos que variam de R\$ 24,90 até R\$ 195,90 de acordo com o número de usuários solicitados. É apresentado na Figura 5 a tela principal do sistema, através dela é possível ver como está estruturada a página inicial. O menu de vendas, estoque, financeiro e de relatórios são vistos na barra superior da tela.

Figura 5: Controle de estoque



Fonte: Conta Azul (2013).



Segundo Izidoro (2011), foi analisado que havia a necessidade de aprimorar o processo de gerenciamento dos estoques na Empresa Brasileira de Correios e Telégrafos (ECT) e em seu Trabalho final de Conclusão do Curso de Sistemas de Informação da Universidade Regional de Blumenau (FURB), implementou um sistema para plataforma web com o objetivo de permitir o controle de estoque dos materiais das Agências de Correios.

Para desenvolvimento do sistema foi utilizada a ferramenta Genexus, o banco de dados Microsoft SQL Server 2005, com linguagem de programação C# (C Sharp). É apresentada na Figura 6 a tela para o usuário efetuar a saída de material com as quantidades em estoque, nessa tela também é apresentada de que forma é realizada a saída de material.

Figura 6: Saída de material

WebÁlmax

Usuário logado: Administrador  
Último acesso em 21/11/11 às 21:53:17

Recents: [Principal](#) [Saída Material](#)

Administrador Cadastros Relatórios

### Saída Material

Código Saída

Data de Saída

Destino

Usuário que Cadastrou 1 - Administrador

Usuário que Confirmou 0 -

Data de Confirmação //

Itens Saída			
Código Correios	Saída Descrição	Qty Estoque	Qty Saída
111111111	Lápis	10	5
0		0	0
0		0	0
0		0	0
0		0	0
0		0	0

[Novo registro]

Confirmar Fechar Confirmar Saída Estornar Saída

Fonte: Izidoro (2011).

### 3 TÉCNICAS E FERRAMENTAS UTILIZADAS

Para implementação e desenvolvimento do sistema de gestão de estoque, foi realizado um estudo de quais tecnologias seriam mais apropriadas, levando em consideração alguns aspectos, entre eles servidor, familiaridade do programador com a linguagem de programação e tempo de desenvolvimento. Na modelagem do sistema de gestão de estoque foi utilizada a ferramenta Astah, pois é uma ferramenta que possui versão gratuita e funciona com várias plataformas fornecendo uma transição correta entre os diagramas, tabelas e plataformas.

A ferramenta usada para criar o modelo conceitual e o modelo lógico foi o brModelo. O modelo lógico é o resultado ou produto da conversão de um modelo conceitual para um determinado tipo de banco de dados. É uma descrição de um banco de dados no nível de abstração visto pelo usuário do sistema gerenciador de banco de dados (SGBD). Já o modelo conceitual busca desenhar as relações entre as diversas áreas do sistema para um modelo teórico.

Para criar o Banco de Dados foi escolhido o SGBD Mysql, onde nele foi criado todas as tabelas da papelaria em SQL e usado triggers nas tabelas de compra e venda de produtos para fazer o controle do estoque sempre que for inserido, alterado ou excluído um produto. Os triggers serão ativados sempre que ocorrerem eventos de insert, update e delete na tabela de venda ou compra de produtos. É atualizado o valor na tabela de estoque sempre que um produto é inserido, atualizado ou excluído. Foi utilizado também um stored procedure para manter o estoque sempre atualizado à medida que for sendo realizado venda ou compra de produtos para a papelaria.

O sistema foi implementado em linguagem java, através da IDE Eclipse, utilizando os conceitos de programação orientada a objetos, e o que era tabela no banco de dados foi implementado em classes no java. O sistema foi construído para ser executado em ambiente web. A interface foi implementada em HTML contendo código JavaScript e JSP, utilizando também os recursos avançados do Bootstrap para que o sistema pudesse ficar com uma interface mais amigável.

Na seção 3.1 segue conceitos de ferramentas e tecnologias que foram utilizadas no desenvolvimento do sistema:

#### 3.1 UML

A Unified Modeling Language (UML) é uma linguagem visual para modelar sistemas orientados a objetos, e que define elementos gráficos que podem ser utilizados na modelagem de sistemas. Esses elementos permitem representar os

conceitos do paradigma da orientação a objetos. Através dos elementos gráficos definidos nesta linguagem pode-se construir diagramas que representem diversas perspectivas de um sistema.

Os elementos gráficos da UML possuem uma sintaxe e uma semântica. Na semântica é definido o que significa o elemento e com que objetivo ele deve ser utilizado, Já a sintaxe corresponde à forma de desenhar o elemento. Os criadores da linguagem UML apresentaram um sistema descrito por cinco visões interdependentes (BEZERRA, 2006), os quais são:

1. **Visão de Processo:** essa visão engloba as características de concorrência, sincronização e desempenho do sistema.
2. **Visão de Casos de Uso:** a visão de casos de uso descreve o sistema de um ponto de vista externo como um conjunto de interações entre o sistema e os agentes externos ao sistema, essa visão direciona o desenvolvimento das outras visões.
3. **Visão de Projeto:** visão de projeto enfatiza as características do sistema que dão suporte, tanto comportamental como estrutural, às funcionalidades externamente visíveis do sistema.
4. **Visão de Implementação:** integra o gerenciamento de versões do sistema, produzidas pelo conjunto de módulos e subsistemas.
5. **Visão de Implantação:** a visão de implantação corresponde a distribuição física do sistema em seus subsistemas e à conexão entre essas partes.

### 3.2 SQL

SQL é uma linguagem de banco de dados e foi criada como interface para um sistema de banco de dados relacional experimental. É a linguagem padrão para Sistemas de Gerenciamento de Banco de Dados (SGBD) relacionais (ELMASRI e NAVATHE, 2011).

Abaixo é descrito algumas das partes da linguagem SQL:

1. **Linguagem de Manipulação de Dados (DML):** engloba uma linguagem de consulta baseada na álgebra relacional e no cálculo relacional de tupla. Também engloba comandos para inserir, excluir e modificar tuplas no banco de dados.
2. **Linguagem de Definição de Dados (DDL):** contém os comandos para definir esquemas de relação, excluir relações e modificar esquemas.

3. **Integridade:** A DDL da SQL contém comandos para especificar restrições de integridade aos dados armazenados. As atualizações que violam a integridade são proibidas.
4. **Definição de view:** A DDL da SQL contém comandos para definir views.
5. **Controle de transação:** A SQL contém comandos para especificar o início e fim das transações.
6. **Autorização:** A DDL SQL contém comandos para especificar os direitos de acesso das relações e views.
7. **SQL embutida e SQL dinâmica:** Essas duas definem como as instruções SQL podem ser incorporadas nas linguagens de programação de finalidade geral como: C++ e Java.

### 3.3 JAVA

Java é uma linguagem de programação orientada a objetos, criada por James Gosling na Sun Microsystems e anunciada formalmente em maio de 1995. Um dos principais objetivos do Java é permitir escrever programas que executem em uma grande variedade de sistemas de computador e dispositivos controlados por computador. Essa propriedade é comumente conhecida como “escreva uma vez, execute em qualquer lugar” (DEITEL, 2017).

Conforme os autores Horstmann e Cornell (2010), a linguagem Java tem algumas características, ela é uma linguagem simples de programar, é orientado a objetos, tem compatibilidade com redes, é uma linguagem robusta, segura, portátil e interpretada, sem contar que possui um alto desempenho.

A linguagem Java é a base para praticamente todos os tipos de aplicações em rede e é o padrão global para o desenvolvimento e distribuição de aplicações móveis e incorporadas, jogos, conteúdo baseado na Web e softwares corporativos. Com mais de nove milhões de desenvolvedores em todo o mundo, de forma eficiente, o Java permite que você desenvolva, implante e use aplicações e serviços.

### 3.4 HTML

O Hypertext Markup Language (HTML) significa Linguagem de Marcação de Hipertexto e consiste em uma linguagem para desenvolvimento web. Ela é o resultado da integração da Hypermedia/Time-based Document Structuring Language (HyTime) e da Standart Generalized Markup Language (SGML) (Carvalho, 2004).

1. **Hypermedia/Time-based Document Structuring Language (HyTime)** – é uma linguagem de marcação que é uma aplicação do SGML, ou seja, é o padrão para a apresentação estruturada de hipermídia e conteúdo baseada em tempo. É independente dos padrões de processamento de texto e dar a base para a construção de sistemas de hipertexto padronizados.
2. **Standart Generalized Markup Language (SGML)** – SGML é um padrão da ISO 8879 de formatação de textos, não foi desenvolvido para hipertexto, porém pode ser usado para transformar documentos em hiper objetos e descrever as ligações. Não é aplicado de forma padronizada, todos os produtos têm um sistema próprio para traduzir as tags, para seus formatadores de texto particular.
3. **Document Type Definition (DTD)** – O DTD define regras de formatação para as classes de documentos. Um DTD deve estar contido em qualquer documento de acordo com o padrão SGML. Portanto a linguagem HTML é definida de acordo com um DTD de SGML.

### 3.5 JavaScript

A linguagem JavaScript (JS) foi criada pela Netscape em parceria com a Sun Microsystems, como uma forma de fornecer interatividade as páginas Web. A primeira versão, a JavaScript 1.0 foi lançada no ano de 1995 e implementada no ano de 1996.

Alguns dos problemas que podem ser encontrados pelo JavaScript (JS) são as versões dos navegadores, pois alguns mais antigos não possuem certas funções em seus interpretadores, isso pode ocasionar sérios problemas nas aplicações. Conforme Balduino (2013), com o JavaScript é possível verificar em tempo de execução se certa função existe ou não, com isso é possível escrever um código de verificação e de acordo com o seu retorno escrever uma outra função que resolva ou minimize o problema. De acordo com Silva (2010), algumas das funcionalidades do JavaScript (JS) São:

1. **Interação com outras linguagens:** o JavaScript (JS) pode interagir com outras linguagens de programação para cumprir tarefas complementares.
2. **Manipulação do Browse:** o JavaScript (JS) possibilita realizar o controle do comportamento do browse em vários aspectos. Ele permite

fazer alteração das dimensões do navegador, apresentar mensagens ao usuário e realizar a retirada dos menus.

3. **Interação com formulários:** será capaz de acessar os campos de um formulário HTML e continuar a validação dos dados, realizando cálculos e fornecendo dicas de preenchimento dos campos.
4. **Realizar a Manipulação de conteúdo e apresentação:** permite escrever o HTML e inseri-lo na marcação de um documento já existente. Essa linguagem é capaz de definir, alterar e controlar de forma dinâmica a apresentação de um documento em HTML.

### 3.6 JSP

O JSP é uma linguagem de script criada pela SUN e tem como objetivo principal a geração de conteúdo dinâmico para páginas da Internet. Ao invés de usar a linguagem HTML para o desenvolvimento de páginas Web estáticas, podemos usar o JSP para criar dinamismo. Como a linguagem HTML é estática, o JSP será o responsável por criar dinamismo. O JSP precisa de servidor para que possa funcionar, o usuário não consegue ver a codificação JSP, pois essa codificação é convertida pelo servidor sendo apresentado ao usuário apenas o código HTML.

A página que é criada em JSP possui extensão .jsp e consiste em uma página com codificação HTML e Java, é inserida entre as tag's `<%` e `%>`, e funciona da seguinte forma: o servidor recebe uma requisição para uma página JSP, interpreta esta página gerando a codificação HTML e retorna ao cliente o resultado de sua solicitação. A página JSP que foi interpretada pelo servidor não precisa ser compilada, esta tarefa é realizada em tempo real pelo servidor.

É necessário apenas desenvolver as páginas JSP e disponibilizá-las no Servlet Container (Tomcat, por exemplo). O trabalho restante será realizado pelo servidor que faz a compilação em tempo de uso transformando o jsp em bytecode. Assim, pode-se definir o JSP como uma tecnologia que provê uma maneira simples e prática de desenvolver aplicações dinâmicas baseadas em web, sendo independente de plataforma de sistema operacional.

### 3.7 BOOTSTRAP

O Bootstrap é uma tecnologia de auxílio para os desenvolvedores front-end. É uma coleção de vários elementos e funções personalizáveis para projetos da web de código aberto, empacotados previamente em uma única ferramenta.

É um conjunto de ferramentas para facilitar o desenvolvimento de sistemas

web. Ao projetar um sistema com a tecnologia bootstrap, os desenvolvedores podem escolher quais elementos querem usar, e os elementos escolhidos não conflitarão entre si e serão totalmente responsivos funcionando tanto para desktop como quaisquer outros tamanhos de tela.

Os elementos personalizáveis contidos no bootstrap são uma combinação de HTML5, CSS3 e JavaScript. Por ser um software livre, essa ferramenta é aprimorada constantemente. Abaixo segue algumas vantagens e desvantagens do bootstrap.

#### **Vantagens:**

1. Um dos frameworks mais utilizados no desenvolvimento de portais e sistemas do mundo;
2. Possui uma grande diversidade de temas;
3. Tem documentação detalhada e de fácil entendimento;
4. Possui uma interface bastante amigável e moderna;
5. Integração com qualquer linguagem de programação.

#### **Desvantagens:**

1. Tema padrão e comum do Bootstrap (caso não faça ajustes visuais, ou troque o tema seu projeto se parecerá com outros que também utilizam o Bootstrap).
2. Seu código terá de seguir os padrões de desenvolvimento Bootstrap.

### 3.8 ASTAH

A Astah é uma ferramenta bastante utilizada para a modelagem de soluções de software fazendo uso da linguagem UML. Essa ferramenta possui uma versão free “community” e de uma versão paga “professional”.

Ela é desenvolvida na plataforma JAVA e permite que sejam modeladas soluções de software fazendo uso de uma linguagem que seja mais próxima do pensamento humano. Ela admite que os modelos criados sejam transformados em códigos, conceito conhecido na computação como engenharia à frente. A ferramenta Astah também aceita que códigos já criados, sejam transformados em modelos UML, através da engenharia reversa.

Abaixo são apresentadas algumas das principais características da ferramenta Astah:

1. **SUPORTE À LINGUAGEM UML:** a ferramenta Astah em sua versão profissional, dispõe de todos os diagramas da UML, entres eles os: Diagrama de Estado, Objeto, Classe, Caso de Uso, Sequencia e

Atividade.

2. **ENGENHARIA REVERSA:** Possibilita selecionar arquivos .java para a área de trabalho da ferramenta, e através das classes Java, gerar o diagrama de classe da UML.
3. **COMPLETUDE:** Contém funcionalidades tais como geração de código, engenharia à frente, e a engenharia reversa. Possui funcionalidades de exportação de diagramas de em diversos formatos e dispõe de funcionalidades para o rastreabilidade de requisitos de caso de uso.
4. **CÓPIA, INSTALAÇÃO E UTILIZAÇÃO DA FERRAMENTA:** A ferramenta Astah possui a versão Astah Community que é uma versão grátis, porém apresenta grandes limitações. As funcionalidades mais avançadas só estão disponíveis na versão paga da ferramenta, ou seja, na versão profissional.

### 3.9 brModelo

O BrModelo é uma ferramenta que possibilita a definição de modelagens conceituais e lógicas para banco de dados. Uma das principais finalidades do desenvolvimento da brModelo foi a de oferecer uma ferramenta voltada para o aprendizado de modelagem de banco de dados relacionais.

Através da ferramenta BrModelo é possível gerar diagramas conceituais e lógicos. O diagrama conceitual é um diagrama Entidade-Relacionamento que contém os requisitos de dados elicidados. Já o diagrama lógico é um pouco mais elaborado que o conceitual, pois contém o mapeamento do modelo conceitual para o modelo relacional. Para o desenvolvimento deste sistema web foi utilizada a versão 3.0 do sistema brModelo. Principais propostas da ferramenta brModelo:

1. Geração de dicionários de dados completos;
2. Realizar o mapeamento da modelagem conceitual para a modelagem lógica, bem como o mapeamento da modelagem lógica para a modelagem física;
3. Basear-se na metodologia completa de projeto de banco de dados relacionais;
4. Ter como foco de implementação a modelagem conceitual, por se tratar de uma importante etapa na modelagem de banco de dados.



## 4 DESENVOLVIMENTO DO SISTEMA

Neste capítulo é apresentado a descrição da empresa, especificação dos requisitos, modelagem de requisitos, modelagem do banco de dados e o projeto do sistema de gestão de estoque.

O sistema web irá controlar a compra e venda de produtos e permitir que um vendedor possa fazer consultas no sistema e verificar a disponibilidade de um determinado produto no estoque, determinando quando o estoque deve ser ajustado, o que deve continuar em estoque, inserir, atualizar e debitar produtos do estoque. Possibilita ainda a organização mais eficiente e eficaz e visa o aumento do lucro da Papelaria Real.

### 4.1 DESCRIÇÃO DA EMPRESA

Foi realizado vários encontros com funcionários e gerente da empresa para que se pudesse entender o seu funcionamento interno.

A Papelaria Real é categorizada como uma microempresa – ME, seu nome fantasia é: Papelaria Real, razão social: Papelaria - ME e CNPJ: 11.222.234/0001-00. Está localizada na Rua Papel, nº0001, bairro Comercio, CEP: 000-380, Mossoró-RN, Brasil. Está classificada no setor de economia terciário, ramo de atividade é comercio e tipo de negócio papelaria. CNAE 000101.

A **papelaria** possui (CNPJ, nome, estado, cidade, bairro, rua, número, telefone, id do usuário, CPF do funcionário e CPF do Administrador). É administrada por um **Administrador** que possui um (CPF, nome, estado, cidade, bairro, rua, número, telefone, data de nascimento, sexo e salário). A papelaria contém um **estoque** de produtos que é caracterizado por um (id, id produto, qtde e valor unitário). O estoque tem **produtos**, os produtos da Papelaria Real são identificados por um (id, status, descrição, estoque mínimo e estoque máximo). O Administrador é o responsável por realizar a **compra** de produtos para a papelaria, a compra possui um (id, id fornecedor, id produto, qtde, valor unitário e data da compra). Os produtos são fornecidos por **fornecedores** que possuem um (id, nome, cidade, bairro, rua, número e telefone). A papelaria contém **usuários** que possuem um (id, nome, login e senha). O usuário é responsável por inserir produtos e cadastrar os clientes no sistema. A papelaria tem vários **clientes** que fazem a solicitação de produtos, e cada cliente é identificado por (id, nome, estado, cidade, bairro, rua, número e telefone). Na papelaria trabalha vários funcionários, cada **funcionário** possui um (CPF, nome, estado, cidade, bairro, rua, número, telefone, data de nascimento, sexo e salário). Os funcionários têm como atividade realizar a **venda** de produtos, as vendas são

caracterizadas por um (id, id cliente, id produto, qtde, data e valor unitário).

## 4.2 ESPECIFICAÇÃO DOS REQUISITOS

Esta etapa corresponde a compreensão do problema a ser aplicado no desenvolvimento de software (Bezerra, 2006). O principal objetivo da especificação de requisitos é proporcionar que usuários e desenvolvedores tenham a mesma visão do problema a ser resolvido. Os desenvolvedores em conjunto com os clientes, buscam levantar e definir as necessidades dos usuários do sistema.

O resultado final desse levantamento é o documento de requisitos. Geralmente esse documento é escrito informalmente, e deve ter os tipos de requisitos do sistema. Segundo Bezerra (2006), as principais seções deste documento são:

1. **Requisitos Funcionais:** definem as funcionalidades do sistema.
2. **Requisitos Não Funcionais:** declaram as características de qualidade que o sistema deve possuir e que são relacionadas às suas funcionalidades. Alguns tipos de requisitos não funcionais são:
  - **Confiabilidade:** corresponde as medidas quantitativas de confiabilidade do sistema. Como: tempo médio entre falhas, recuperação de falhas ou quantidade de erros.
  - **Desempenho:** definem tempos de respostas esperados pelas funcionalidades do Sistema.
  - **Portabilidade:** restrições sobre as plataformas de hardware e software onde sistema será implantado e a facilidade de transporte para outras plataformas.
  - **Segurança:** limitações em relação a acesso não autorizados.
  - **Usabilidade:** requisitos que se relacionam ou afetam a usabilidade do sistema.

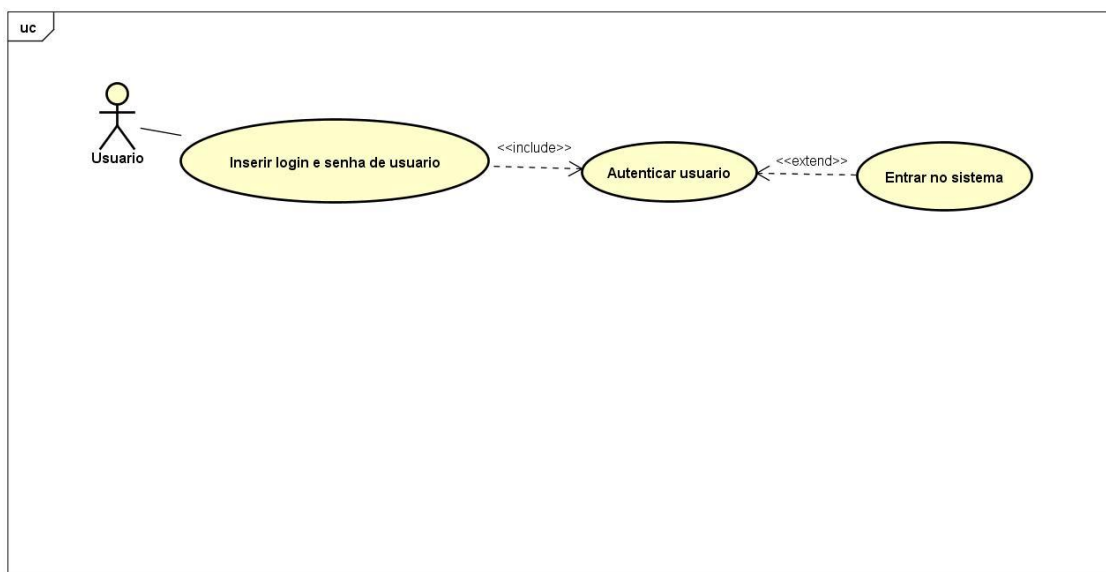
### 4.3 MODELAGEM DE REQUISITOS

A modelagem de requisitos é de fundamental importância, pois possibilita o primeiro contato com a estrutura de funcionamento do sistema, e demonstra de forma teórica e graficamente, através dos diagramas, como será o comportamento de cada elemento que irá compor o sistema. Nesta fase é possível desenvolver diversos diagramas, como o de casos de uso, atividades, sequência, classes, objetos e estados para simular na teoria o comportamento de um sistema.

#### 4.3.1 DIAGRAMAS DE CASOS DE USO

Os casos de uso são definidos através do ponto de vista de um ator. O Ator é um papel que os dispositivos desempenham ao ter interação com o software. A Figura 7 exibe o diagrama de caso de uso, para validação de senha, caso o usuário digite a senha incorreta ou que não está cadastrada no banco de dados será emitido uma mensagem de usuário não encontrado.

**Figura 7.** Diagrama de casos de uso do sistema de validação de senha.

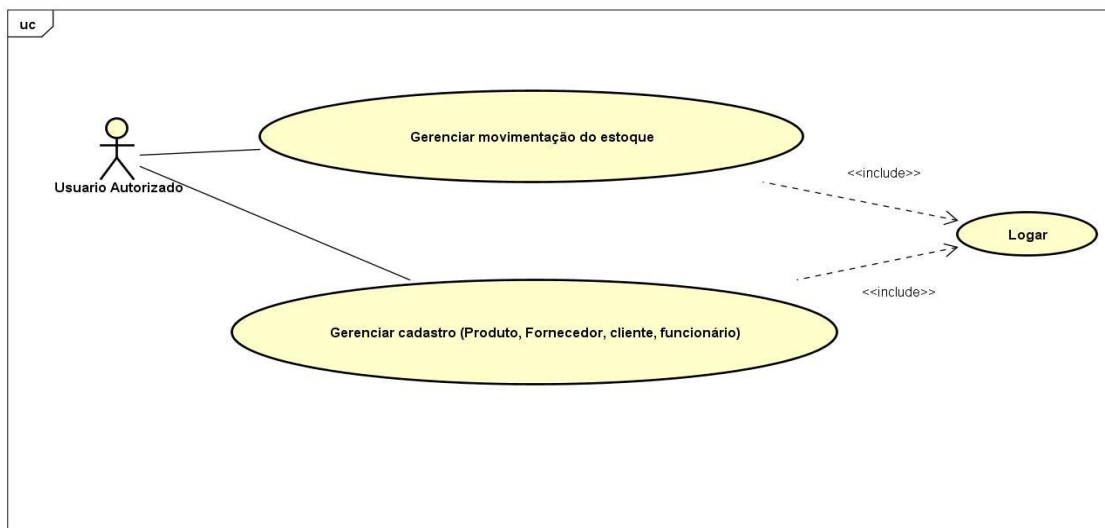


powered by Astah

Fonte: Próprio Autor (2018)

A Figura 8 exibe o diagrama de caso de uso usuário autorizado, onde o mesmo para gerenciar a movimentação do estoque ou realizar o cadastro de produtos, fornecedores, clientes e funcionários obrigatoriamente deve estar logado no sistema.

**Figura 8.** Diagrama de casos de uso: Usuário Autorizado

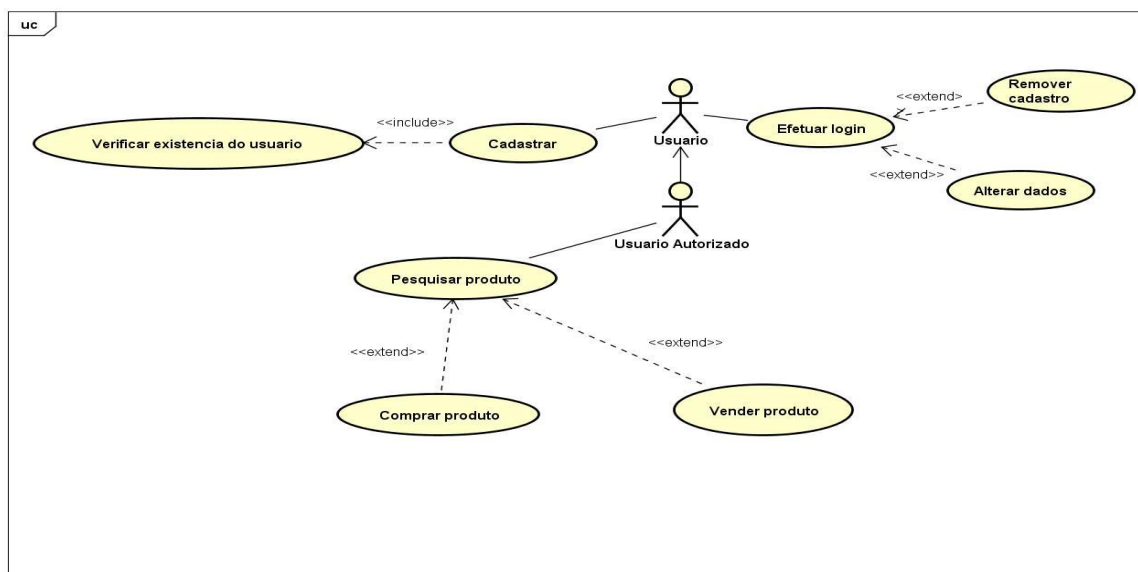


powered by Astah

Fonte: Próprio Autor (2018)

A Figura 9 exibe o diagrama de caso de uso usuário, onde o mesmo poderá efetuar login com seu usuário e senha que foi cadastrado no sistema e a partir daí ter o privilégio de acessar os produtos que tem disponível em estoque, realizar uma compra ou venda de produtos e alterar dados cadastrais.

**Figura 9.** Diagrama de casos de uso: Usuário



powered by Astah

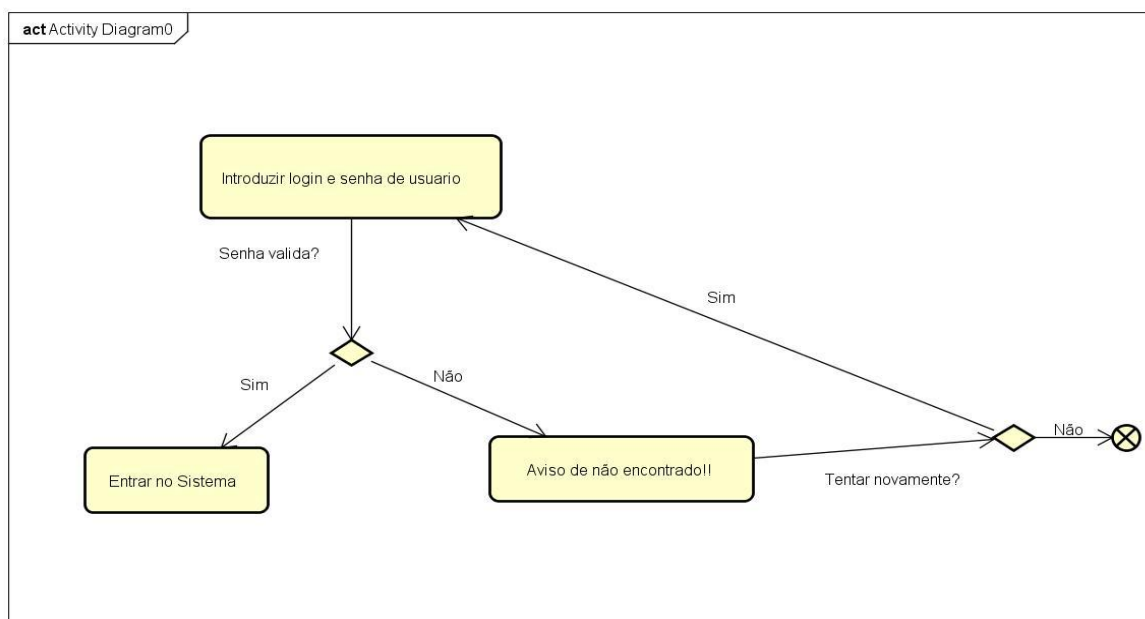
Fonte: Próprio Autor (2018)

### 4.3.2 DIAGRAMAS DE ATIVIDADES

O diagrama de atividades complementa o diagrama de caso de uso por meio de uma representação gráfica do fluxo de interação. Utiliza retângulos para representar determinada função do sistema, setas para representar o fluxo através do sistema e losangos de decisão para representar uma decisão com ramificação.

A Figura 10 exibe o diagrama de atividade usuário para efetuar login, onde o mesmo para conseguir efetuar login obrigatoriamente tem que já estar cadastrado no sistema. Caso o usuário digite uma senha inválida, será emitida uma mensagem de não encontrado, onde o usuário poderá optar por tentar novamente ou não. Caso a senha seja válida, o usuário efetua o login e terá acesso a todas as funcionalidades do sistema que lhes são permitidas.

**Figura 10.** Diagrama de Atividade: Usuário - Login

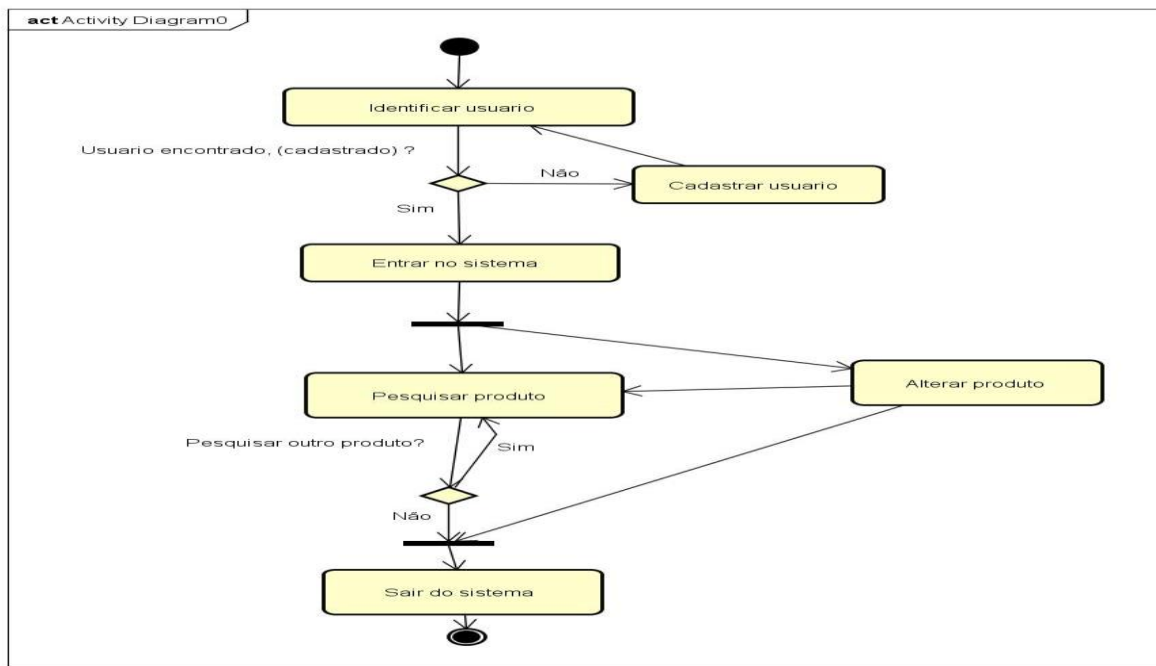


powered by Astah

Fonte: Próprio Autor (2018)

A Figura 11 exibe o diagrama de atividade para autenticar usuário. Caso o usuário tente logar com dados não cadastrados, será emitida uma mensagem e redirecionado para página de cadastro. Após se cadastrar, o usuário autorizado terá o privilégio de entrar no sistema e conseguir pesquisar e alterar um produto ou sair do sistema.

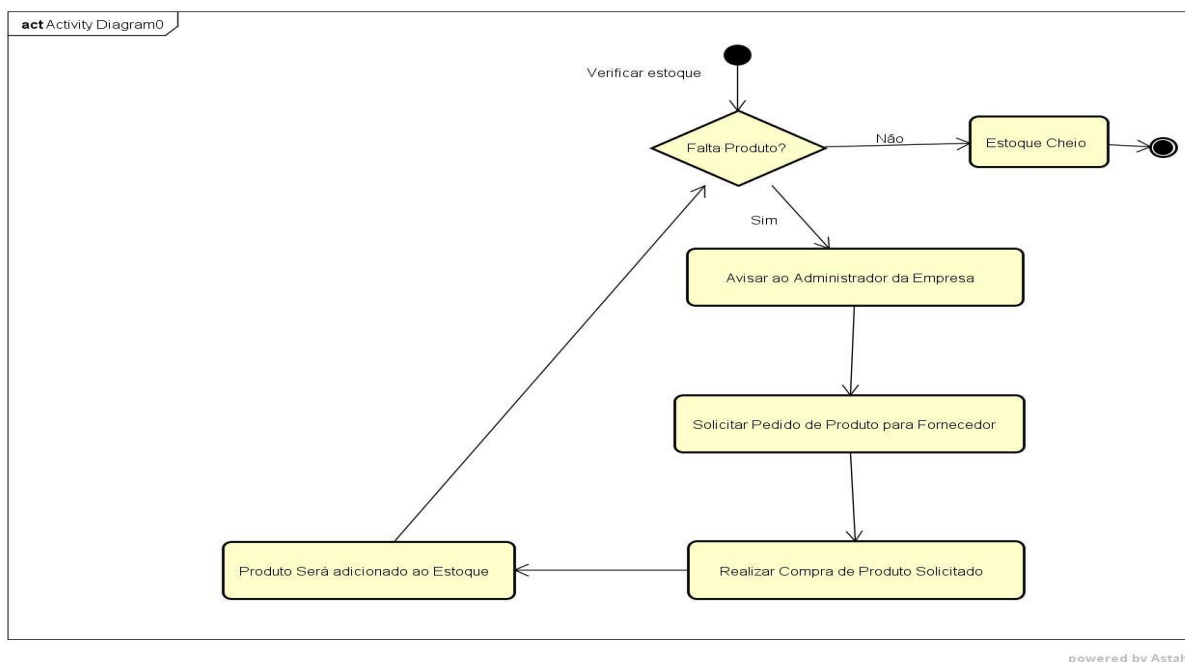
**Figura 11.** Diagrama de Atividade: Usuário - Autenticar



Fonte: Próprio Autor (2018)

A Figura 12 exibe o diagrama de atividade para abastecimento de estoque. O usuário autorizado ao se logar, inicialmente verifica o estoque. Caso não falte produto, ele finaliza como estoque cheio, e caso falte produto ele avisa ao administrador da empresa onde o mesmo faz a solicitação de pedido de produtos para o fornecedor, o usuário realiza a compra dos produtos solicitados e os produtos serão adicionado ao estoque.

**Figura 12.** Diagrama de Atividade: Abastecimento de Estoque

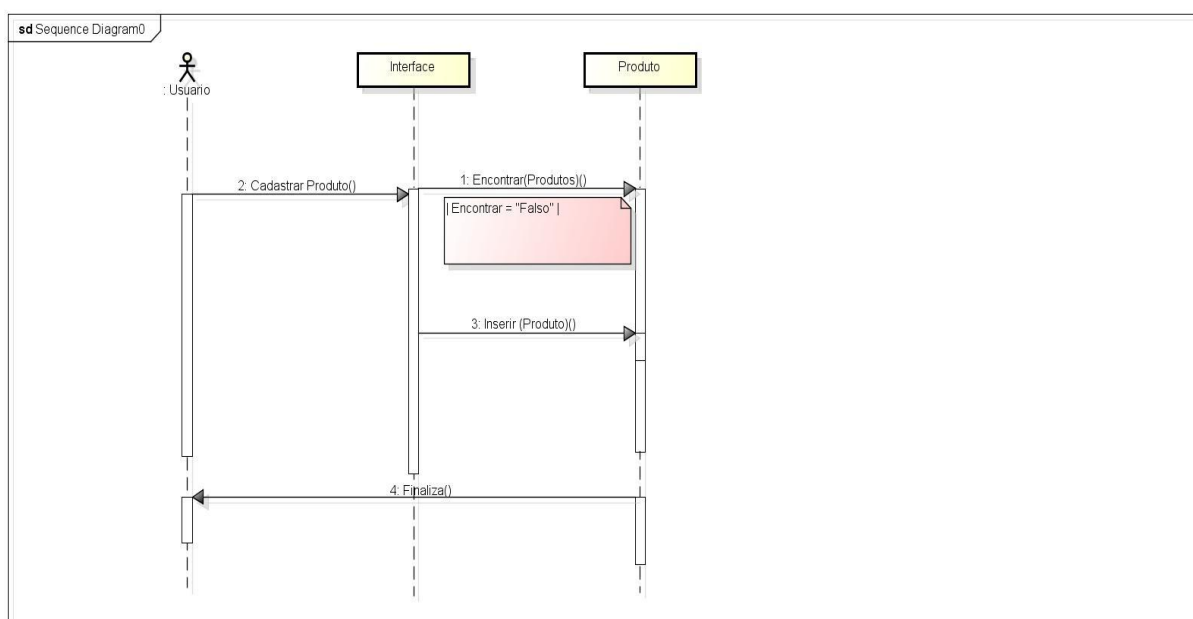


Fonte: Próprio Autor (2018)

### 4.3.3 DIAGRAMAS DE SEQUÊNCIA

O diagrama de sequência é utilizado para indicar as comunicações dinâmicas entre objetos e ele mostra a ordem temporal na qual as mensagens são enviadas entre os objetos para executar determinada tarefa. A Figura 13 exibe o diagrama de sequência para cadastrar produto. O usuário ao entrar na interface do sistema e realizar a pesquisa por um produto, caso não encontre produtos cadastrados ele poderá realizar um novo cadastro de produtos e inserir no estoque.

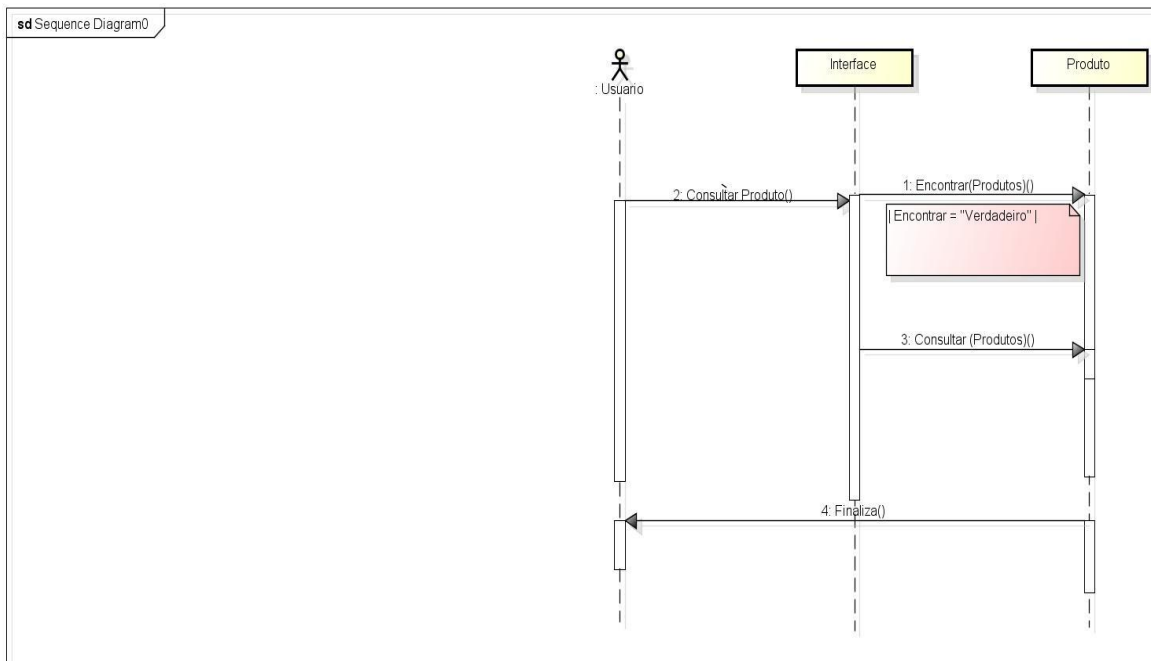
**Figura 13.** Diagrama de Sequência: Cadastrar produto



Fonte: Próprio Autor (2018)

A Figura 14 exibe o diagrama de sequência para consultar produto, o usuário ao entrar na interface do sistema e consultar um produto o mesmo será encontrado apresentado alguns atributos como descrição e valor unitário.

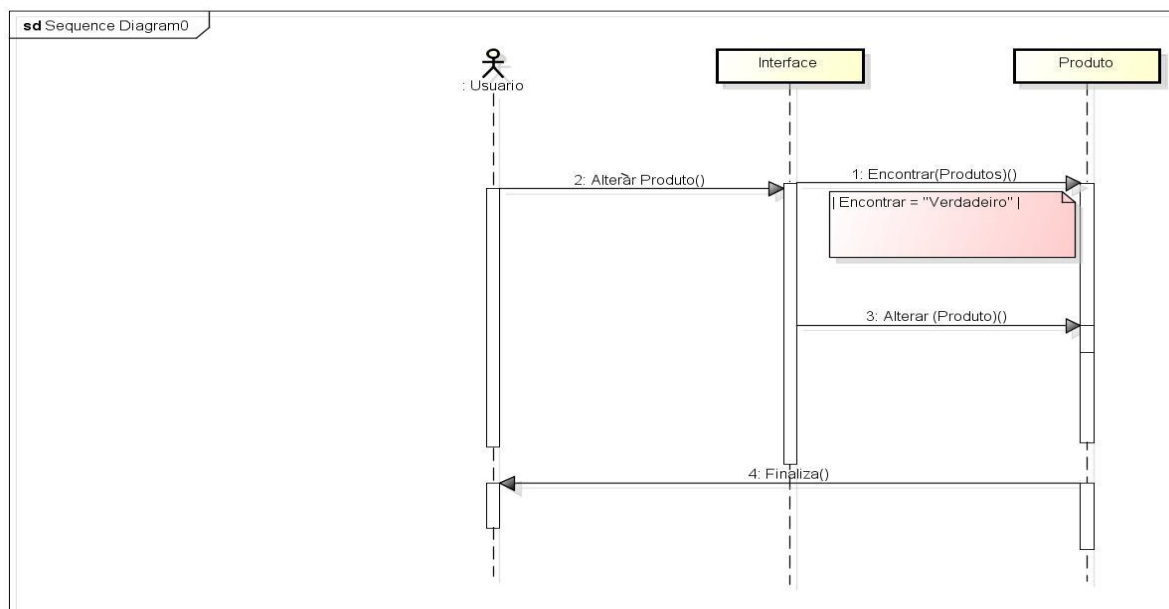
**Figura 14.** Diagrama de Sequência: Consultar produto



Fonte: Próprio Autor (2018)

A Figura 15 exibe o diagrama de sequência para alterar produto, o usuário ao entrar na interface do sistema consegue pesquisar um produto, ao encontrar o produto solicitado o usuário poderá fazer alterações e finalizar o procedimento.

**Figura 15.** Diagrama de Sequência: Alterar produto

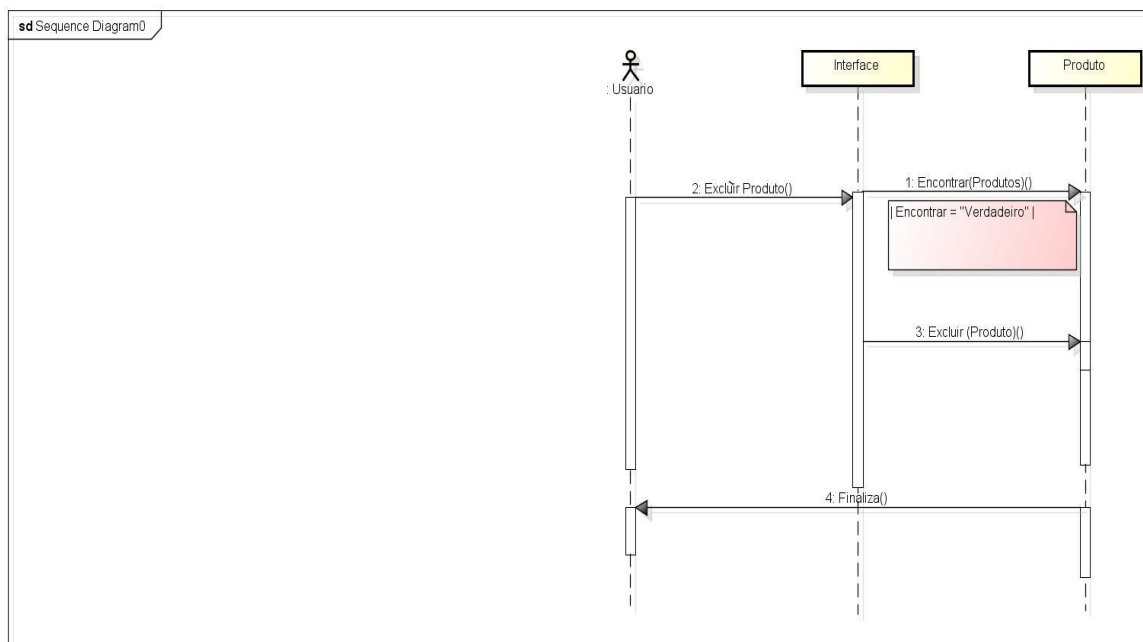


Fonte: Próprio Autor (2018)



A Figura 16 exibe o diagrama de sequência para excluir produto, o usuário ao entrar na interface do sistema consegue pesquisar um produto pelo seu identificador único, ao encontrar o produto solicitado o usuário poderá excluí-lo e finalizar o procedimento.

**Figura 16.** Diagrama de Sequência: Excluir produto

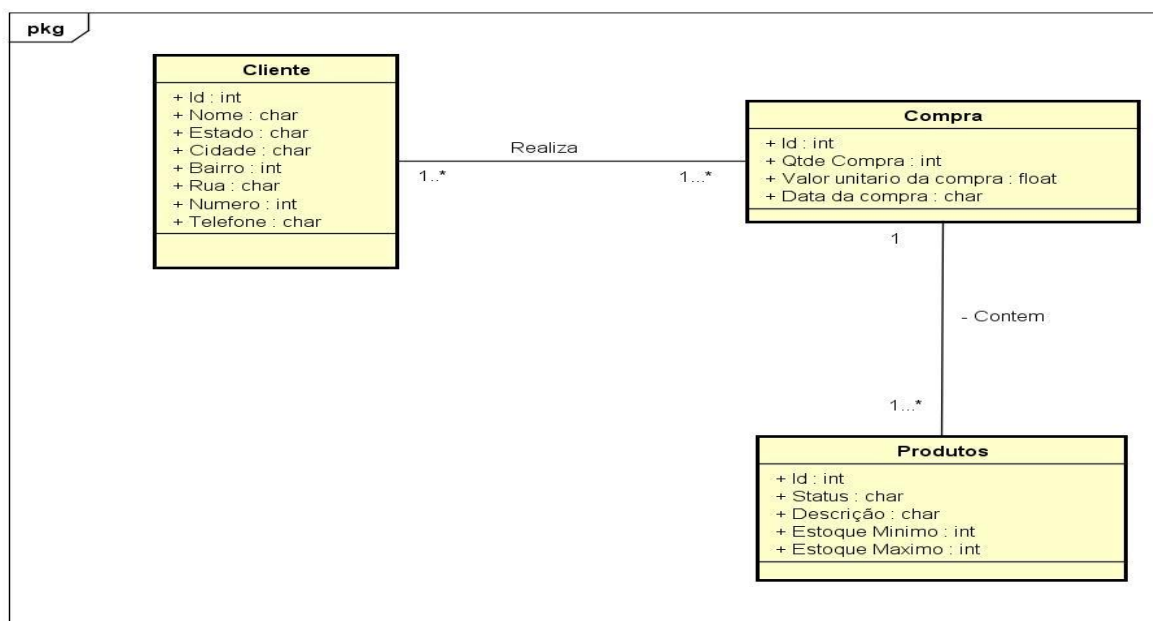


Fonte: Próprio Autor (2018)

#### 4.3.4 DIAGRAMAS DE CLASSES

Os diagramas de classe são utilizados para mostrar as classes de um sistema e as ligações entre essas classes. Uma classe de objeto é uma definição geral de um tipo de objeto do sistema. A Figura 17 exibe o diagrama de classes cliente apresentando atributos como identificador único, nome, cidade e telefone. Um ou vários clientes realizam uma ou várias compras, uma compra possui atributos como identificador único, quantidade de produtos, valor unitário e data da compra. Uma compra contém um ou vários produtos e produtos são compostos por identificador único, descrição, estoque mínimo e estoque máximo.

Figura 17. Diagrama de Classes: Cliente

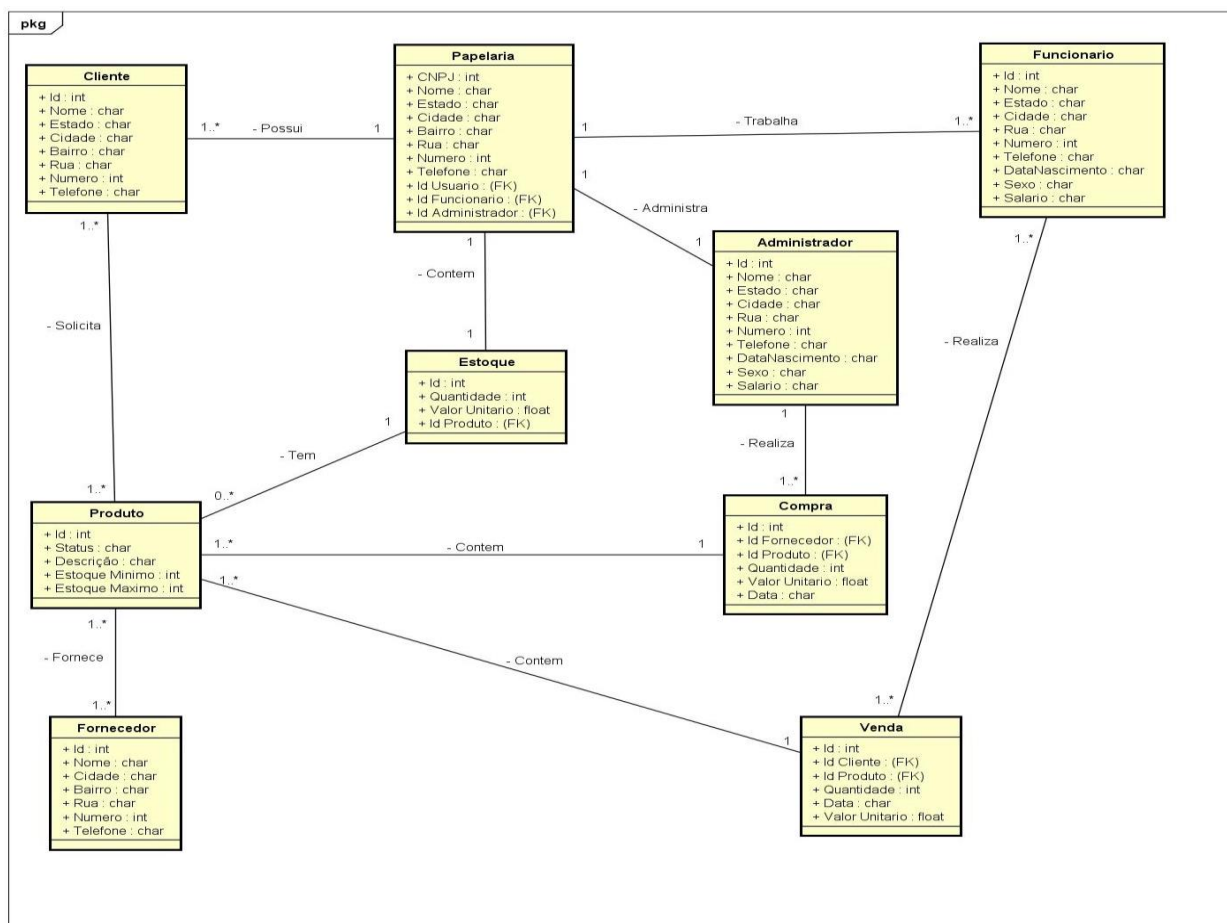


powered by Astah

Fonte: Próprio Autor (2018)

A Figura 18 exibe o diagrama de classes com a visão geral da Papelaria Real, a papelaria possui um ou vários clientes, um ou vários funcionários, um administrador e um estoque que contém um ou vários produtos. Um ou vários clientes poderá solicitar um ou vários produtos. Um ou vários fornecedores fornece um ou vários produtos para a empresa. O administrador realiza uma ou várias compras para a empresa e uma compra contém um ou vários produtos. O funcionário autorizado que trabalha na empresa consegue efetuar uma ou várias vendas de produtos, uma venda contém um ou vários produtos.

Figura 18. Diagrama de Classes: Visão Geral da Papelaria

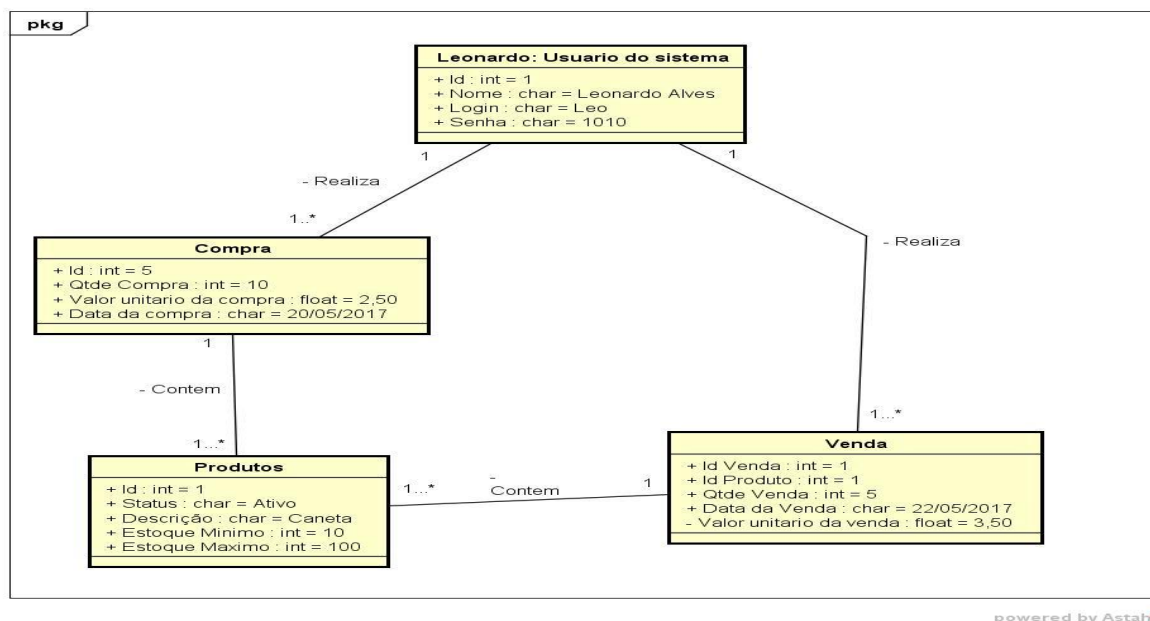


Fonte: Próprio Autor (2018)

#### 4.3.5 DIAGRAMAS DE OBJETOS

Um diagrama de objetos é uma variação do diagrama de classes. A principal diferença é que o diagrama de objetos mostra os objetos que foram instanciados das classes. A Figura 19 exibe o diagrama de objetos para usuário, o usuário possui atributos como identificador único, nome, login e senha. O usuário autorizado ao efetuar login consegue realizar uma ou várias compras de produtos a fornecedores. Também consegue realizar uma ou várias vendas de produtos para clientes. Uma compra é composta pelos atributos identificador único, quantidade de produtos, valor unitário e data da compra. Uma compra contém um ou vários produtos, um produto contém identificador único, descrição, estoque mínimo e estoque máximo. Já uma venda possui identificador da venda, identificador do produto, quantidade de produtos vendidos, data em que foi realizada a venda e valor unitário.

Figura 19. Diagrama de Objetos: Usuário

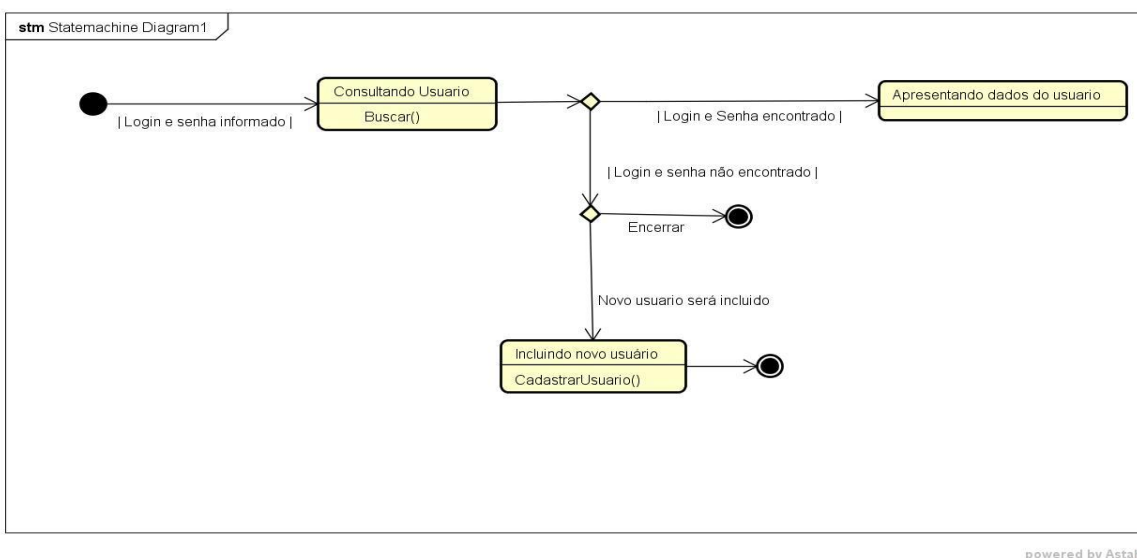


Fonte: Próprio Autor (2018)

#### 4.3.6 DIAGRAMAS DE ESTADOS

O diagrama de estados é utilizado para representar o comportamento de um sistema por meio da representação de seus estados. A Figura 20 exibe o diagrama de estados para o usuário, o usuário informa seu login e senha e o sistema irá realizar uma busca e consultar se o mesmo está cadastrado. Caso encontre o login e senha na base de dados seus dados serão apresentados, porém se não encontrar login e senha o processo será encerrado e o usuário terá a opção de se cadastrar no sistema.

Figura 20. Diagrama de Estados: Usuário



Fonte: Próprio Autor (2018)

## 4.4 MODELAGEM DO BANCO DE DADOS

O modelo de BD apresenta a estrutura lógica de banco de dados e engloba relações e restrições que possibilitam determinar como os dados serão armazenados. A seguir será apresentado o modelo conceitual e o modelo lógico.

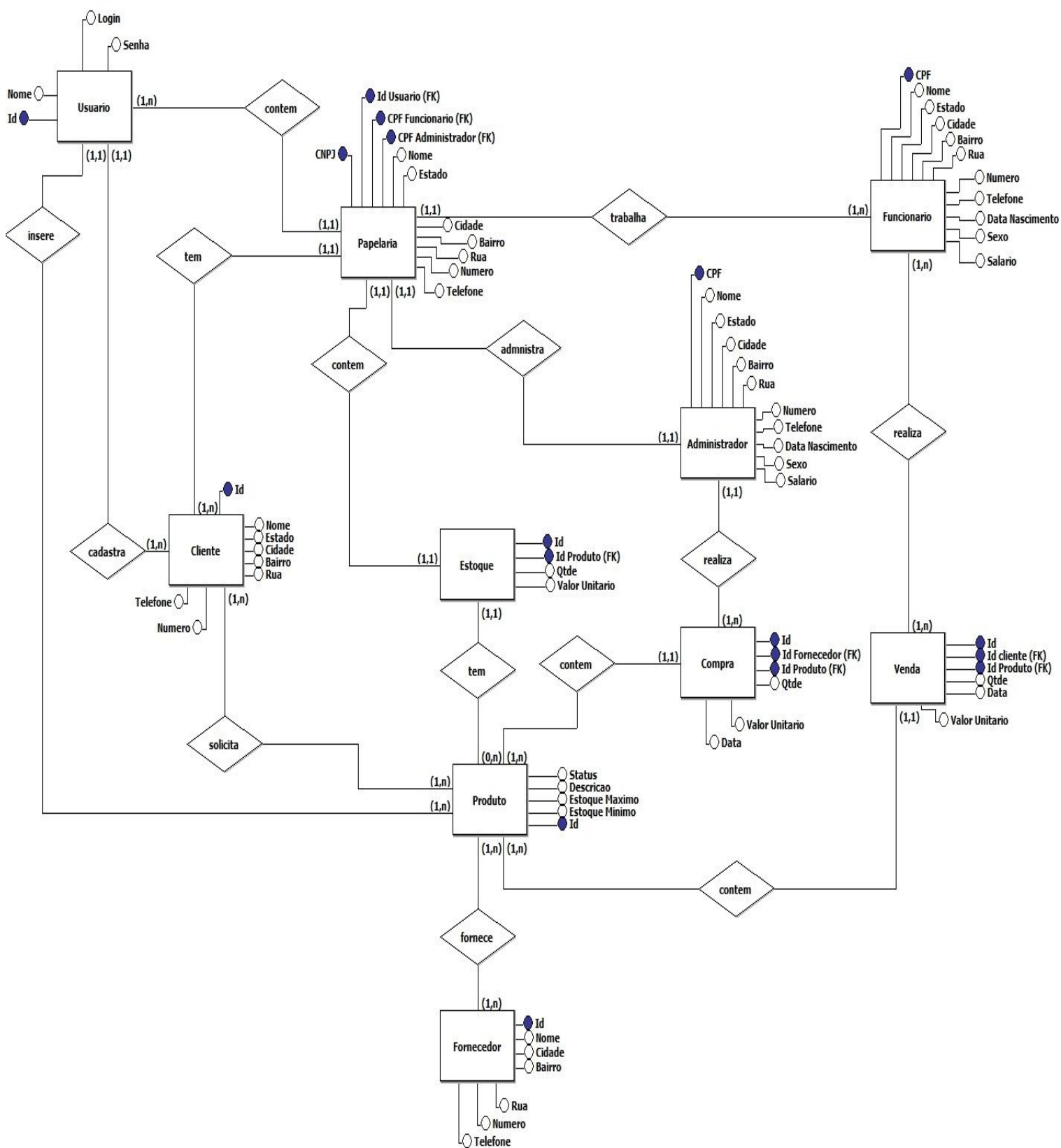
### 4.4.1 MODELO CONCEITUAL

O modelo conceitual procura desenhar as relações entre as diversas áreas ou usuários do sistema para um modelo ainda teórico. Uma das dificuldades para se aprender as técnicas de modelagem em BD é entender um problema do mundo real e convertê-lo em uma solução. Primeiramente é necessário entender o problema e após isso conceituar o que será a solução. Para isso, dois fatores são de fundamental importância:

1. Primeiramente saber ouvir o cliente para entender o que realmente é útil para que se possa fazer a implementação da solução;
2. Ter conhecimento das técnicas de modelagem com o objetivo de fazer uma representação do problema de forma conceitual antes de iniciar o processo de implementação da solução.

A Figura 21 exhibe o modelo conceitual da Papelaria Real, cujas características já foram mencionadas na seção 4.1.

Figura 21. Modelo conceitual

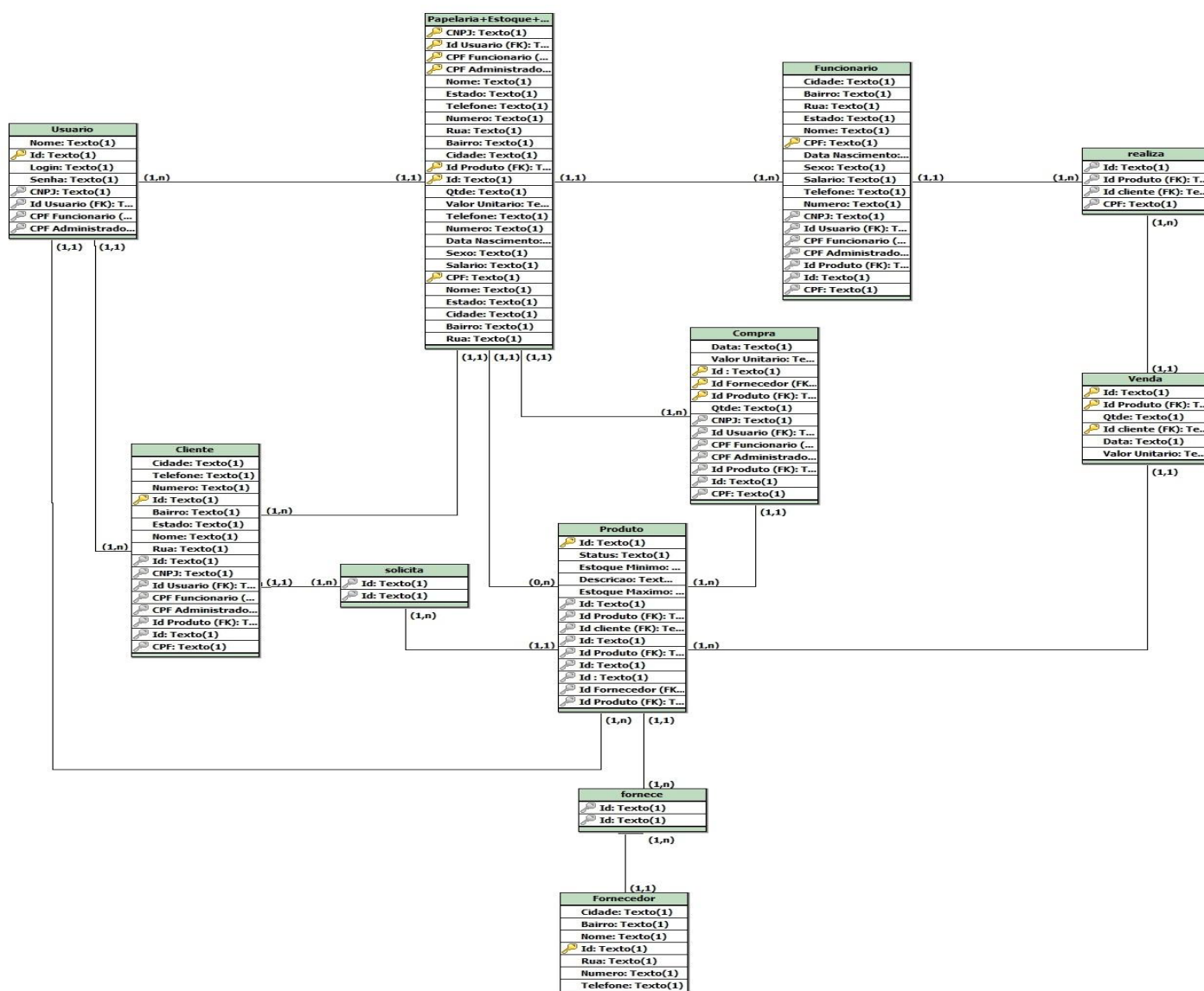


Fonte: Próprio Autor (2018)

## 4.4.2 MODELO LÓGICO

O modelo lógico é o resultado da conversão de um modelo conceitual para um determinado tipo de BD. É uma descrição de um BD no nível de abstração visto pelo usuário do SGBD. Nessa fase algumas regras serão implementadas, como valores únicos que não podem ter duplicatas. Por exemplo, um cadastro de fornecedor ordenado por CPF, como regra não pode ter dois fornecedores com o mesmo Cadastro de Pessoa Física (CPF). Os relacionamentos entre as entidades, chamados de chaves estrangeiras são definidos, como o cadastro de compras, no qual há o registro dos itens comprados. Essa parte é de fundamental importância para a integridade e confiabilidade dos dados. A Figura 22 exibe o modelo lógico contendo todas as tabelas do sistema e suas relações.

Figura 22: Modelo Lógico



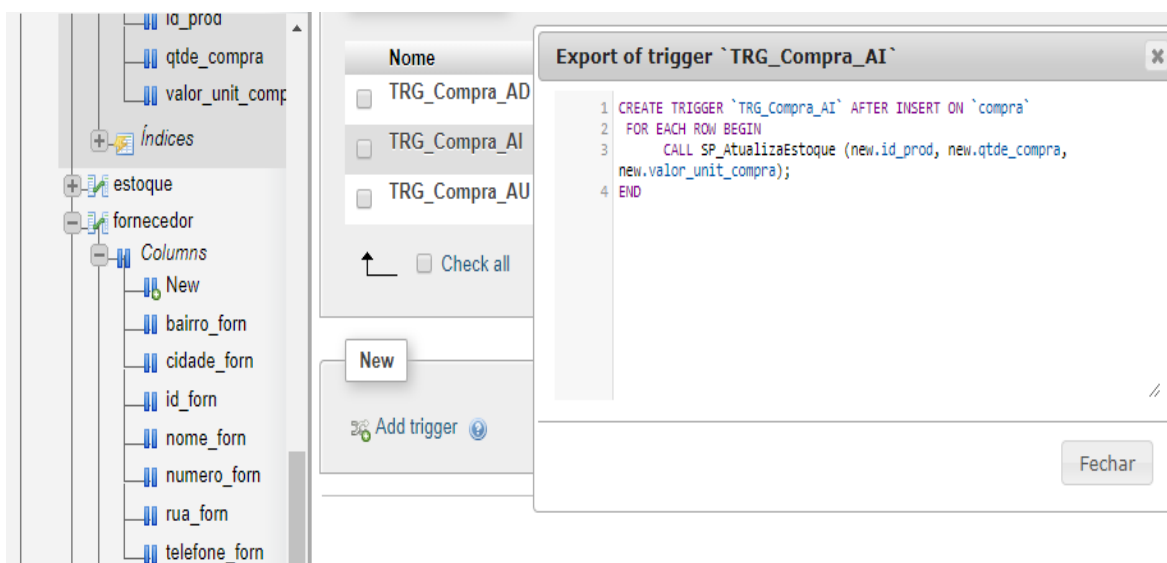
### 4.4.3 TRIGGERS

Triggers são instruções que o sistema executa de forma automática como um efeito de uma modificação realizada no BD. Para que se possa criar um mecanismo de trigger, deve-se cumprir os dois requisitos descritos abaixo:

1. Primeiramente devemos especificar quando um trigger deve ser executado. Isso é definido em um evento que faz com que o trigger seja verificado e uma condição seja satisfeita.
2. Em seguida devemos especificar as ações a serem feitas quando o trigger for executado.

Este tipo de modelo de trigger é o conhecido modelo evento-condição-ação. O BD armazena triggers como se fossem dados, de um modo que sejam persistentes e acessíveis a todas as operações de BD. Quando inserimos um trigger no BD, o sistema de banco de dados faz a execução sempre que o evento especificado ocorrer e a condição for satisfeita. O trigger da Figura 23 será disparado após a inserção de um registro na tabela de compra de produto. Sempre que for inserido um item na tabela compra de produto o trigger irá disparar atualizando a tabela estoque com os novos dados inseridos.

Figura 23: Trigger - inserção de registros na tabela compra de produtos

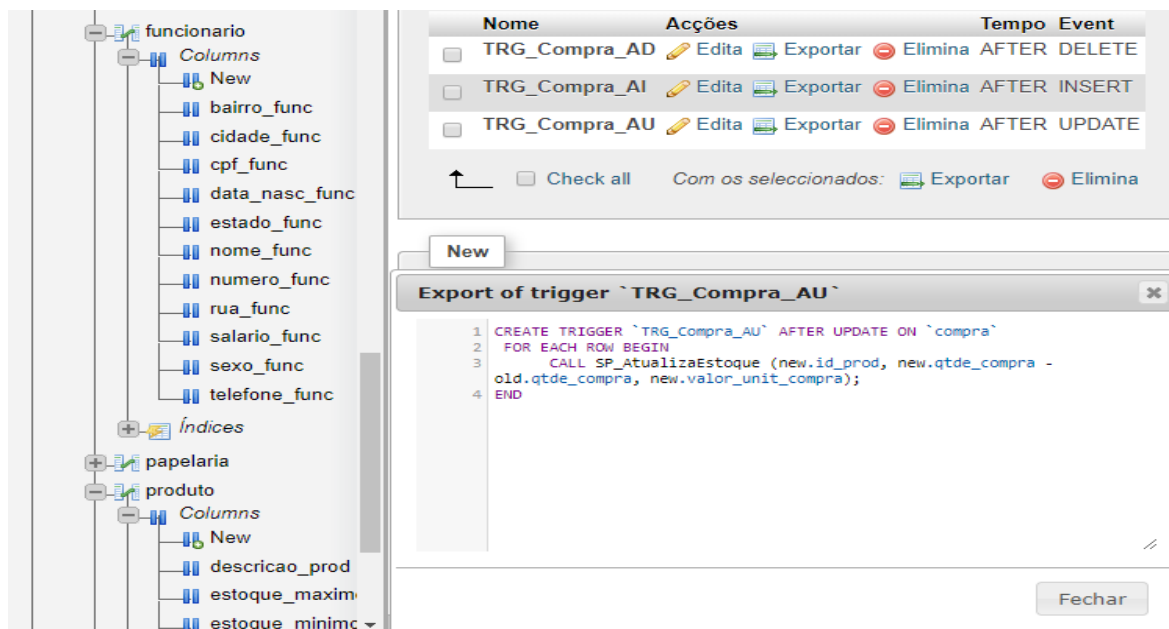


Fonte: Próprio Autor (2018)



O trigger da Figura 24 será disparado após a atualização de um registro na tabela de compra de produto. Sempre que for feito atualização/alteração de um item na tabela compra de produto o trigger irá disparar atualizando a tabela estoque com os novos dados que foram atualizados na tabela compra de produto.

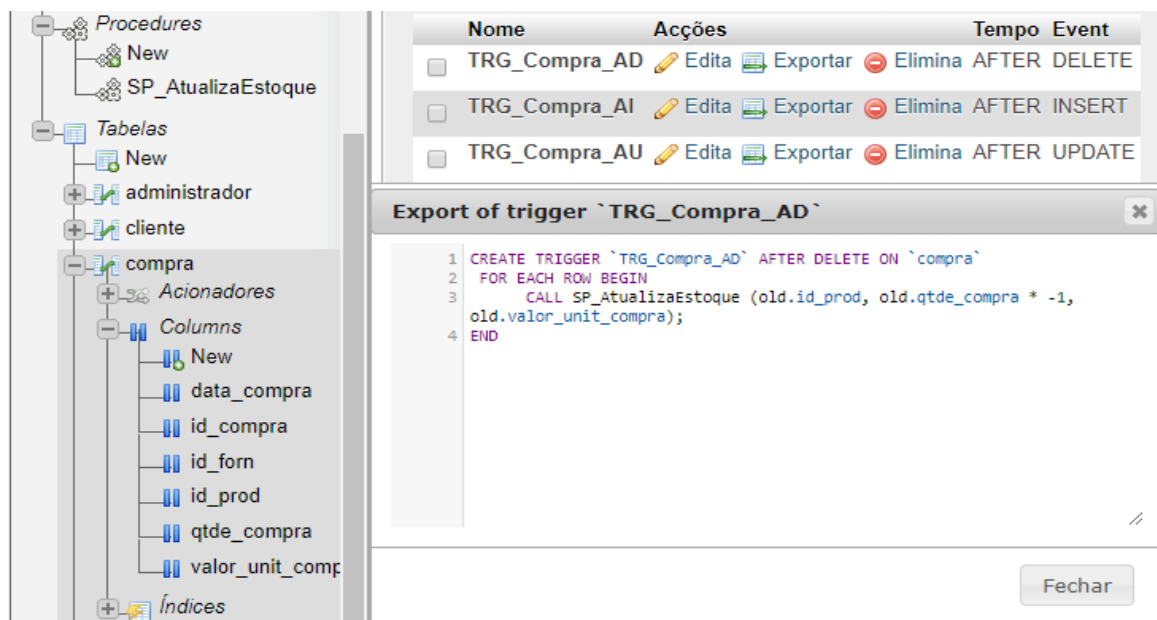
Figura 24: Trigger - atualização de registros na tabela compra de produtos



Fonte: Próprio Autor (2018)

O trigger da Figura 25 será disparado após a exclusão de um registro na tabela de compra de produto. Sempre que for feito exclusão de um item na tabela compra de produto o trigger irá disparar atualizando a tabela estoque sem os dados que foram excluídos da tabela compra de produto.

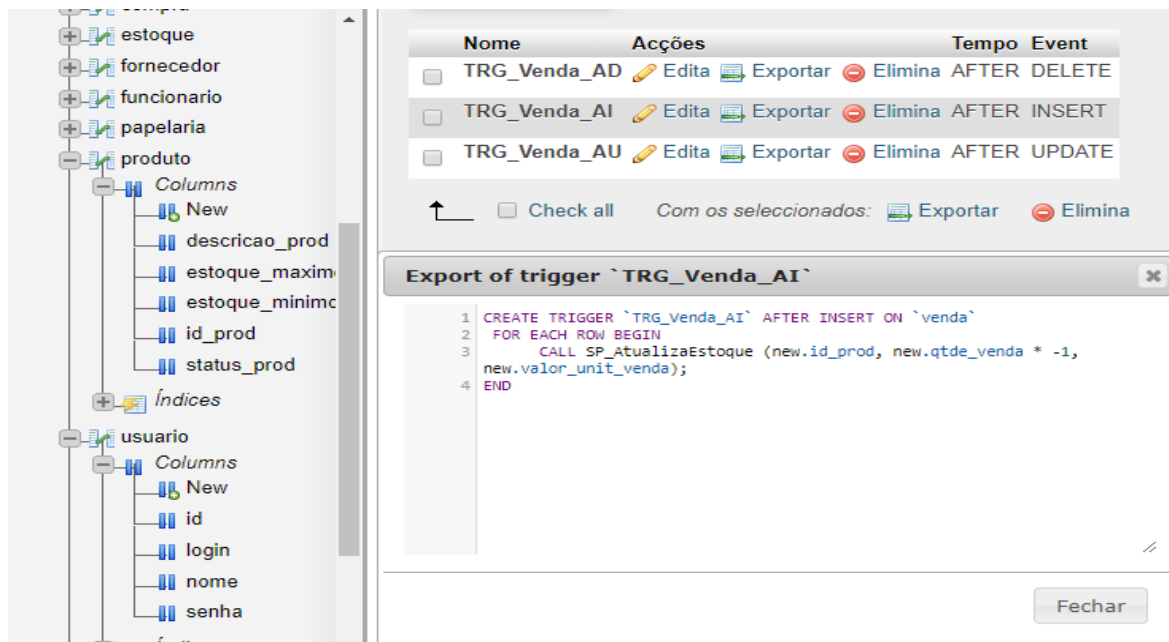
Figura 25: Trigger - exclusão de registros na tabela compra de produtos



Fonte: Próprio Autor (2018)

O trigger da Figura 26 será disparado após a inserção de um registro na tabela de venda de produto. Sempre que for inserido um item de venda de produto o trigger irá disparar atualizando os dados da tabela estoque.

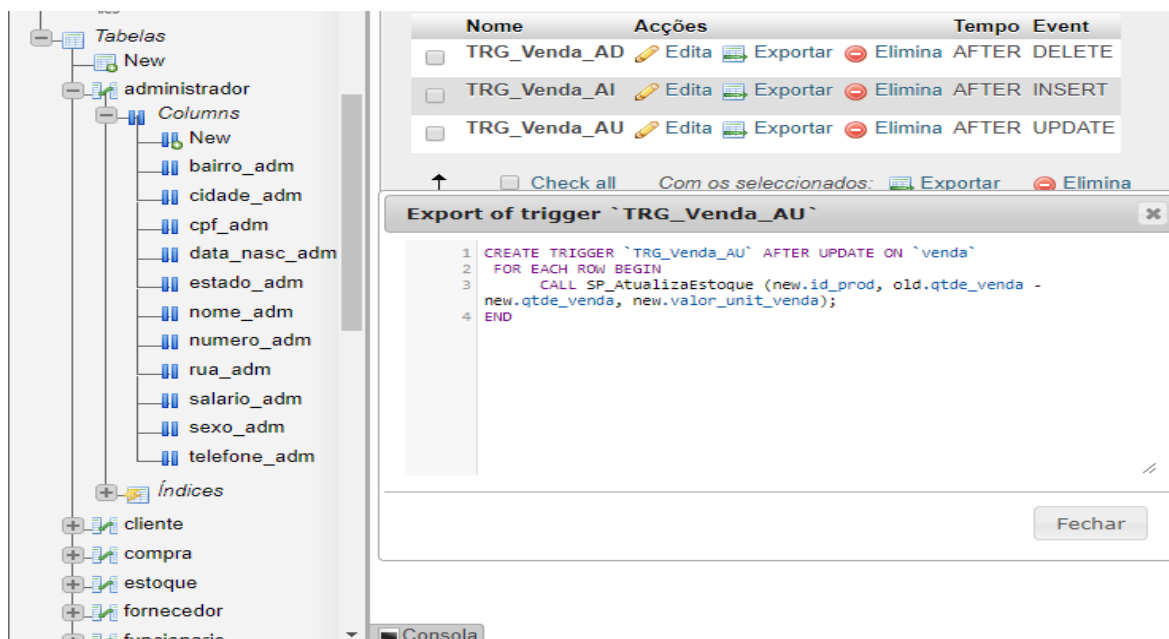
Figura 26: Trigger - inserção de registros na tabela venda de produtos



Fonte: Próprio Autor (2018)

O trigger da Figura 27 será disparado após a atualização de um registro na tabela de venda de produto. Sempre que for atualizado um item de venda de produto o trigger irá disparar atualizando os dados da tabela estoque.

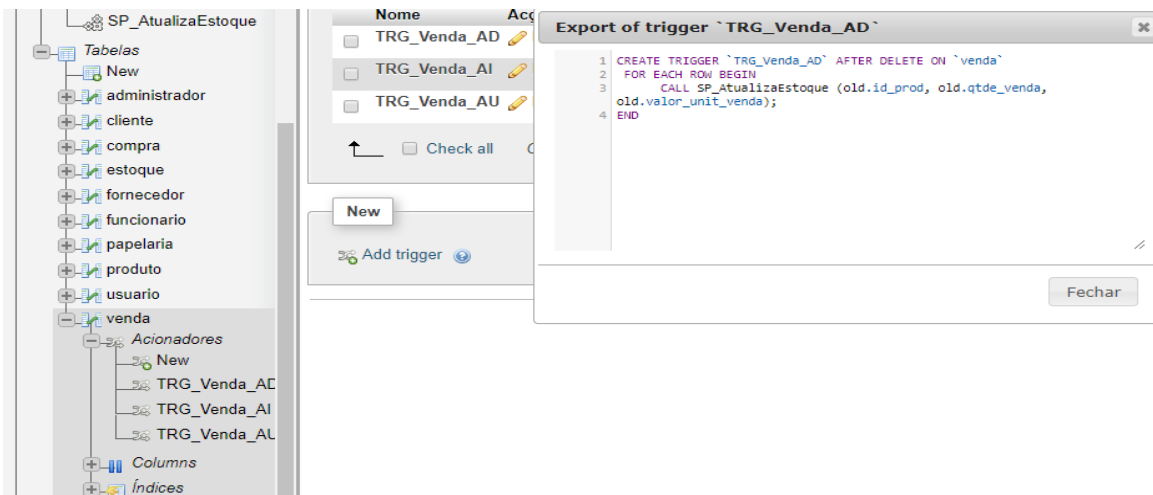
Figura 27: Trigger - atualização de registros na tabela venda de produtos



Fonte: Próprio Autor (2018)

O trigger da Figura 28 será disparado após a exclusão de um registro na tabela de venda de Produto. Sempre que for excluído um item de venda de produto o trigger irá disparar atualizando os dados da tabela estoque.

Figura 28: Trigger - exclusão de registros na tabela venda de produtos

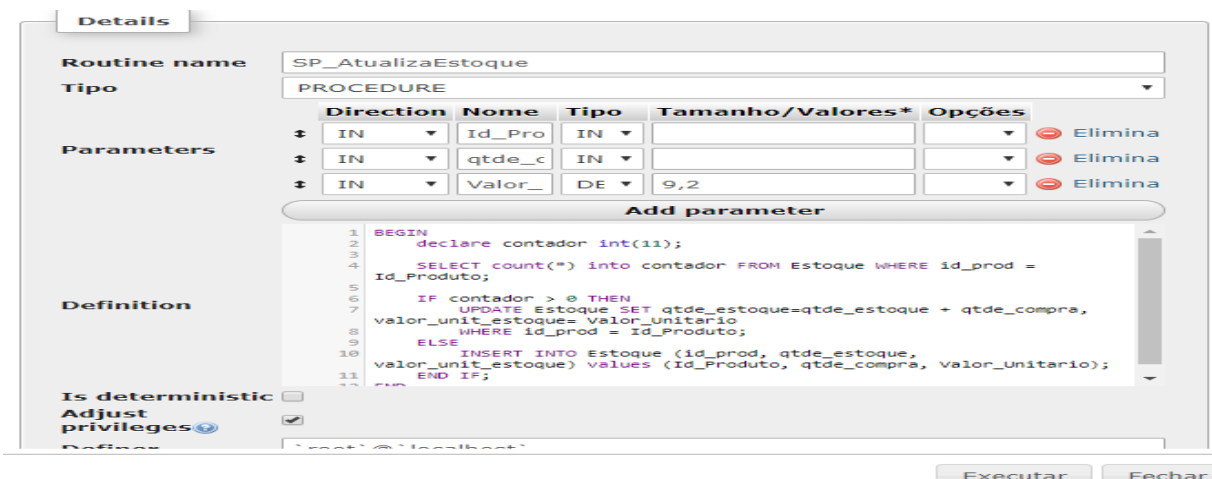


Fonte: Próprio Autor (2018)

#### 4.4.4 STORED PROCEDURE

*Stored Procedure* é uma biblioteca contendo vários comandos em linguagem SQL para utilização em conjunto com o BD. Ela realiza o armazenamento de tarefas repetitivas e aceita parâmetros de entrada para que a tarefa seja efetuada de acordo com a necessidade individual. Um Procedure pode melhorar a performance de um BD, fazer a redução do tráfego na rede e criar tarefas agendadas. Essa tecnologia é de fundamental importância para os DBAs e desenvolvedores. A Stored procedure da Figura 29 irá atualizar os itens na tabela estoque e tem a finalidade de inserir ou debitar produtos.

Figura 29: Procedure - atualização de registros na tabela estoque



Fonte: Próprio Autor (2018)

## 4.5 PROJETO DO SISTEMA

Projetar um sistema é determinar como os requisitos serão implementados em forma de software. Projeto é a forma com a qual se consegue traduzir os requisitos solicitados pelo cliente em um produto de software. Os requisitos são mapeados e as soluções tecnológicas são estruturas.

### 4.5.1 IMPLEMENTAÇÃO (CÓDIGOS E INTERFACE)

A seguir apresentam-se telas do sistema, juntamente com os principais códigos fonte para a compreensão de rotinas, assim como as funcionalidades do sistema. A tela inicial é a tela de login, como pode ser observado na Figura 30.

Ao acessar a página inicial do sistema, o usuário preenche os campos abaixo com login e senha, e em seguida aciona o botão entrar para ter acesso a todas as opções do sistema. Caso o usuário fique inativo por mais de 5 minutos no sistema, sem que acesse as funcionalidades, automaticamente a sessão será expirada e ele será redirecionado para a página inicial para fazer login novamente, caso queira acessar as funcionalidades do sistema novamente.

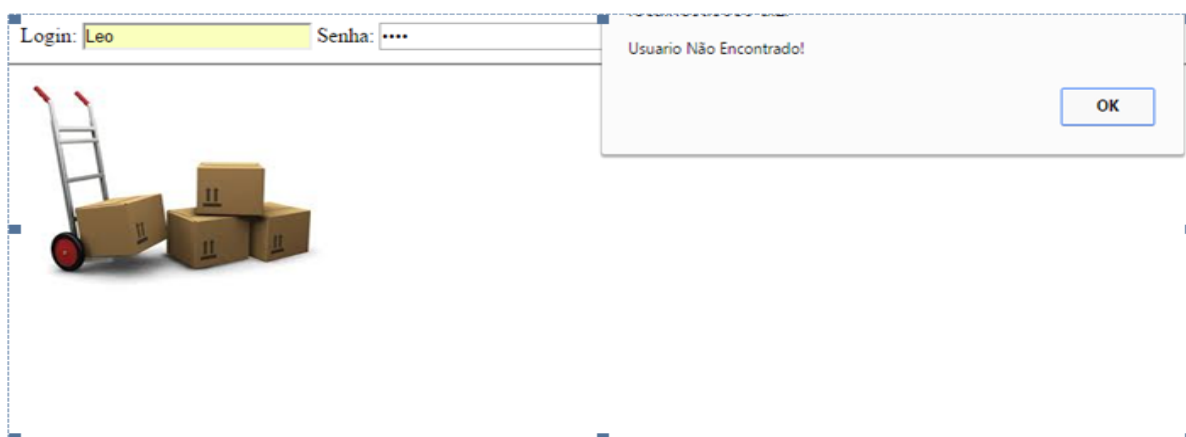
Figura 30: Tela de login



Fonte: Próprio Autor (2018)

Caso o usuário digite login ou senha incorreta, o sistema emite uma mensagem de não encontrado, como descrito na Figura 31.

Figura 31: Tela de login: Autenticar



Fonte: Próprio Autor (2018)

Na Figura 32 tem-se parte do código fonte da tela de login do sistema criado em linguagem JAVA. Mostra o código para autenticação, como é realizada a captura de dados fornecido pelo usuário e como é feito a consulta na base de dados do sistema.

Figura 32: Código fonte da tela de login: Autenticar

```

*AutenticadorController.java
24
25 HttpSession sessao = req.getSession(false);
26
27 if(sessao!=null){
28     sessao.invalidate();
29
30     resp.sendRedirect("login.html");
31
32
33 @Override
34 protected void doPost(HttpServletRequest req, HttpServletResponse resp)
35     throws ServletException, IOException {
36
37     // 1-Capturando dados da tela
38     String login = req.getParameter("login");
39     String senha = req.getParameter("senha");
40
41     //2-Colocando dados em objeto usuario
42     Usuario usuario = new Usuario();
43     usuario.setLogin(login);
44     usuario.setSenha(senha);
45
46     //3-Consultando se usuario existe no banco
47     UsuarioDAO usuarioDAO = new UsuarioDAO();
48     Usuario usuAutenticado = usuarioDAO.autenticar(usuario);
49
50     //4-Verificando se usuario foi encontrado
51     if(usuAutenticado!=null){
52
53         //5-Colocando usuario na sessao
54         HttpSession sessao = req.getSession();
55         sessao.setAttribute("usuAutenticado", usuAutenticado);
56
57         sessao.setMaxInactiveInterval(60*5);
58
59         //6-Redirecionando usuario para tela principal
60         req.getRequestDispatcher("WEB-INF/paginaPrincipal.jsp").forward(req, resp);
61     }else{
62         resp.getWriter().print("<script> window.alert ('Usuario Não Encontrado!'); location.href='login.
  
```

Fonte: Próprio Autor (2018)

Na Figura 33 tem-se parte do código fonte da tela de login do sistema criado em linguagem HTML.

Figura 33: Código fonte da tela de login

```

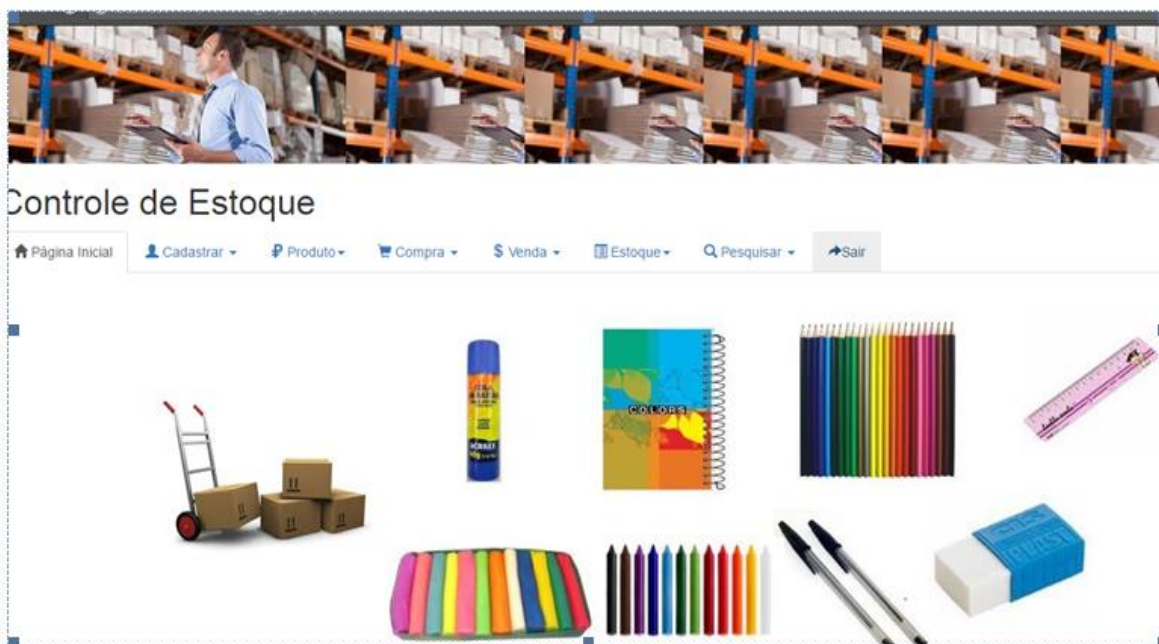
1 <%--
2 Document : login
3 Created on : 10/04/2018, 12:20:03
4 Author : Leonardo Alves
5 --%>
6 <%@page contentType="text/html" pageEncoding="UTF-8"%>
7 <!DOCTYPE html>
8 <html lang="pt-br">
9 <head>
10 <meta charset="ISO-8859-1">
11 <meta http-equiv="X-UA-Compatible" content="IE=edge">
12 <meta name="viewport" content="width=device-width, initial-scale=1">
13 <!-- essas 3 meta tags precisam estar logo após a tag head do documento, qualquer outra meta t
14 <title>Autenticação</title>
15 <!-- link para o arquivo CSS do Bootstrap -->
16 <link href="css/bootstrap.css" rel="stylesheet">
17 <!--[if lt IE 9]>
18 <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
19 <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
20 <![endif]-->
21 </head>
22 <!-- inicio - area de producao -->
23 <body>
24 <div class="container">
25 <form method="POST" action="autenticador.do">
26 <h2 class="form-Login-heading">Entre com seu login e senha:</h2>
27 <div>
28 <label for="inputLogin" class="sr-only">Login</label>
29 <input type="text" name="login" id="inputLogin" class="form-control" placeholder="Logi
30 <label for="inputPassword" class="sr-only">Senha</label>
31 <input type="password" name="senha" id="inputPassword" class="form-control" placeholde
32 </div>
33 <button class="btn btn-Lg btn-primary btn-block" type="submit">Entrar</button>
34 </form>
35 </div>
36 <!-- fim - area de producao -->
37 <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
38 <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
39 <!-- Include all compiled plugins (below), or include individual files as needed -->

```

Fonte: Próprio Autor (2018)

Após efetuar o login e ser validado, o usuário é direcionado diretamente para a tela inicial do sistema. Na tela inicial o usuário possui acesso a todos os módulos, menus do sistema na parte superior. Na Figura 34 tem-se a tela inicial do sistema.

Figura 34: Tela inicial



Fonte: Próprio Autor (2018)



Na Figura 35 tem-se parte do código fonte do cabeçalho do sistema criado em linguagem HTML.

Figura 35: Código fonte do cabeçalho do sistema

```

1 Document : cabecalho
2 Created on : 22/04/2018, 11:15:08
3 Author : Leonardo Alves da Silva
4
5 --%>
6 <%@page contentType="text/html" pageEncoding="UTF-8"%>
7 <!DOCTYPE html>
8 <html>
9 <head>
10 <meta charset="utf-8">
11 <meta http-equiv="X-UA-Compatible" content="IE=edge">
12 <meta name="viewport" content="width=device-width, initial-scale=1">
13 <!-- The above 3 meta tags "must" come first in the head; any other head content must come "after" -->
14 <title>Cabeçalho</title>
15 <!-- Bootstrap -->
16 <link href="css/bootstrap.min.css" rel="stylesheet">
17 <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
18 <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
19 <!--[if lt IE 9]>
20 <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
21 <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
22 <![endif]-->
23 </head>
24 <ul class="nav nav-tabs">
25 <li role="presentation" class="active"> <a href="home.do"> <span class="glyphicon glyphicon-home">
26 <li role="presentation" class="dropdown">
27 <a class="dropdown-toggle" data-toggle="dropdown" href="#" role="button" aria-haspopup="true" aria-
28 <span class="glyphicon glyphicon-user"></span> Cadastrar <span class="caret"> </span>
29 </a>
30 <ul class="dropdown-menu">
31 <li role="presentation" class="active"><a href="usucontroller.do?acao=cad">Usuário</a></li>
32 <li role="presentation"><a href="admcontroller.do?acao=cad">Administrador</a></li>
33 <li role="presentation"><a href="funccontroller.do?acao=cad">Funcionário</a></li>
34 <li role="presentation"><a href="forncontroller.do?acao=cad">Fornecedor</a></li>
35 <li role="presentation"><a href="clientecontroller.do?acao=cad">Cliente</a></li>
36 </ul>
37 </li>
38 <li role="presentation" class="dropdown">
39 <a class="dropdown-toggle" data-toggle="dropdown" href="#" role="button" aria-haspopup="true" aria-
40 <span class="glyphicon glyphicon-ruble"></span> Produto<span class="caret"></span>

```

Fonte: Próprio Autor (2018)

Na parte superior da tela inicial tem-se os menus de acesso as telas e a todas funcionalidades do sistema para cadastros, consultas, alterações e exclusões. Esse menu é visível em qualquer tela do sistema e pode ser acessado a qualquer momento. Cada módulo será explicado separadamente a seguir, começando pelo menu cadastrar como é mostrado na Figura 36.

O cadastro do usuário, administrador, funcionário, fornecedor e cliente é feito através do menu cadastrar. Para realizar um cadastro, o usuário deverá acionar uma das opções do menu cadastrar e em seguida irá aparecer os campos para preenchimento. Após inserir os dados, deverá acionar o botão salvar para finalizar o cadastro.

Figura 36: Menu cadastrar



Fonte: Próprio Autor (2018)

Na Figura 37 tem-se a tela para cadastro de usuário.

Figura 37: Campos para cadastro de usuário

Fonte: Próprio Autor (2018)

Na Figura 38 tem-se parte do código fonte da tela cliente criado em linguagem JAVA. Mostra o código usado para se realizar o cadastro ou alteração de dados de clientes e como esses dados são armazenados no banco de dados.

Figura 38: Código fonte da tela cliente: Acesso a dados

```

14
15 private Connection con = ConexaoFactory.getConnection();
16
17 public void cadastrar(Cliente cliente) {
18     String sql = " insert into cliente (nome_cliente, estado_cliente,cidade_cliente, bairro_cliente,r
19
20     //Criando um preparador - Statment
21     try (PreparedStatement preparador = con.prepareStatement(sql)){
22
23         //substituindo as ? pelos dados do cliente
24         preparador.setString(1, cliente.getNome_cliente());
25         preparador.setString(2, cliente.getEstado_cliente());
26         preparador.setString(3, cliente.getCidade_cliente());
27         preparador.setString(4, cliente.getBairro_cliente());
28         preparador.setString(5, cliente.getRua_cliente());
29         preparador.setInt(6, cliente.getNumero_cliente());
30         preparador.setString(7, cliente.getTelefone_cliente());
31
32         //Executando o comando SQL no BD
33         preparador.execute();
34
35     } catch (SQLException e) {
36         // TODO Auto-generated catch block
37         e.printStackTrace();
38     }
39
40 public void alterar (Cliente cliente){
41     String sql = "UPDATE cliente SET nome_cliente=?, estado_cliente=?,cidade_cliente=?, bairro_cliente
42
43     //Criando um preparador - Statment
44     try {
45         PreparedStatement preparador = con.prepareStatement(sql);
46
47         //substituindo as ? pelos dados do cliente
48         preparador.setString(1, cliente.getNome_cliente());
49         preparador.setString(2, cliente.getEstado_cliente());
50         preparador.setString(3, cliente.getCidade_cliente());
51         preparador.setString(4, cliente.getBairro_cliente());
52         preparador.setString(5, cliente.getRua_cliente());

```

Fonte: Próprio Autor (2018)

Para alterar registros de um usuário, administrador, funcionário, fornecedor, cliente ou dados da empresa, o usuário deverá acessar o menu pesquisar. Em seguida acessa uma das opções do menu, depois clica no botão alterar e irá aparecer os campos com os dados para que se possa fazer alterações. Após isso deverá acionar o botão salvar para que as alterações feitas sejam salvas.

Para excluir um usuário, administrador, funcionário, fornecedor, cliente ou dados da empresa, o usuário deverá acessar o menu pesquisar. Em seguida acessa uma das opções do menu, depois clica no botão excluir. Aparecerá uma mensagem



perguntando se realmente deseja excluir. Para excluir, deve confirmar acionando o botão ok, caso acione o botão cancelar o registro não será removido do sistema.

Para consultar os dados da empresa o usuário deverá acessar o menu pesquisar, em seguida a opção dados da empresa, irá aparecer todos os dados da empresa, como CNPJ, nome, estado, cidade, bairro, rua, número, telefone, identificador do usuário, identificador do funcionário e identificador do administrador. Na Figura 39 tem-se a tela do menu pesquisar.

Figura 39: Menu pesquisar



Fonte: Próprio Autor (2018)

Na Figura 40 tem-se tela para inclusão, alteração ou exclusão de fornecedor.

Figura 40: Excluir fornecedor

Pagina Inicial							
Tem certeza que deseja excluir?							
Id_forn	Nome_forn	Cidade_forn	Bairro_forn	Ru			
1	Papelaria Somatex	Mossoro	Centro	Alberto			
2	Papelaria Potiguar	Mossoro	Costa e Silva	Bela Vis			
3	Papelaria Barbosinha	Mossoro	Centro	Francisco Isodio	59	08433172130	<a href="#">excluir</a> <a href="#">alterar</a>
4	Papelaria Master	Natal	Lagoa Seca	Bernardo Vieira	3775	08433172140	<a href="#">excluir</a> <a href="#">alterar</a>
5	Semear Papelaria	Natal	Nova Pamamirim	Jose Revoredo Neto	229	08433172150	<a href="#">excluir</a> <a href="#">alterar</a>
Novo							

Fonte: Próprio Autor (2018)

Na Figura 41 tem-se parte do código fonte da tela fornecedor, mostrando como foi criado o objeto fornecedor, como realizar a busca desse objeto no banco de dados. Apresenta a implementação para que se possa excluir ou alterar um fornecedor usando apenas o seu identificador único.

Figura 41: Código fonte da tela fornecedor: Acesso a dados

```

40
41 String acao = req.getParameter("acao");
42
43 if(acao.equals("exc")){
44     String id_forn = req.getParameter("id_forn");
45     Fornecedor forn = new Fornecedor();
46
47     if(id_forn!=null){
48         forn.setId_forn(Integer.parseInt(id_forn));
49     }
50     FornecedorDAO fornecedorDAO = new FornecedorDAO();
51     fornecedorDAO.excluir(forn);
52     resp.getWriter().print("<h1>Fornecedor excluido com Sucesso!</h1>");
53
54     //redireciona para mesma pagina
55     resp.sendRedirect("forncontroller.do?acao=lis");
56
57 }else if (acao.equals("lis")){
58     //implementação da lista
59     FornecedorDAO fornecedorDAO = new FornecedorDAO();
60     List<Fornecedor> lista = fornecedorDAO.buscarTodos();
61
62     req.setAttribute("lista", lista);
63
64     RequestDispatcher dispatcher = req.getRequestDispatcher("WEB-INF/listaforn.jsp");
65     dispatcher.forward(req, resp);
66
67 }else if(acao.equals("alt")){
68     //captura parametro da tela
69     String id_forn = req.getParameter("id_forn");
70     FornecedorDAO fornecedorDAO = new FornecedorDAO();
71
72     //busca objeto fornecedor no banco
73     Fornecedor fornecedor = fornecedorDAO.buscarPorId_forn(Integer.parseInt(id_forn));
74
75
76
77
78

```

Fonte: Próprio Autor (2018)

Na Figura 42 tem-se parte do código fonte da tela fornecedor criada em linguagem HTML, apresenta como foi criado o formulário para inserção dos dados do fornecedor.

Figura 42: Código Fonte da tela fornecedor: Formulário

```

2 <html>
3 <head>
4 <meta charset="ISO-8859-1">
5 <title>Insert title here</title>
6 </head>
7 <body>
8
9 <%@include file="menu.jsp" %>
10
11 <%
12 Fornecedor f = (Fornecedor)request.getAttribute("forn");
13 %>
14 <form action="forncontroller.do" method="post">
15
16     Id_forn <input type="number" name="id_forn" value="<%=f.getId_forn()%">
17
18     Nome_forn: <input type="text" name="nome_forn" value="<%=f.getNome_forn()%">/>
19     Cidade_forn: <input type="text" name="cidade_forn" value="<%=f.getCidade_forn()%">/>
20     Bairro_forn: <input type="text" name="bairro_forn" value="<%=f.getBairro_forn()%">/>
21     Rua_forn: <input type="text" name="rua_forn" value="<%=f.getRua_forn()%">/>
22     Numero_forn: <input type="text" name="numero_forn" value="<%=f.getNumero_forn()%">/>
23     Telefone_forn: <input type="text" name="telefone_forn" value="<%=f.getTelefone_forn()%">/>
24     <input type="submit" value="Salvar">
25
26 </form>

```

Fonte: Próprio Autor (2018)

O cadastro de produtos é realizado através do menu de produto. O usuário deverá acionar a opção cadastrar produto, em seguida irá aparecer os campos em que o usuário deverá informar o status, descrição, estoque mínimo e estoque máximo do produto a ser cadastrado. Após inserir os dados do produto, o usuário deverá acionar o botão salvar para cadastrar o produto no sistema.

Para alterar registros de um produto, o usuário deverá acessar o menu de produto. Em seguida, consultar produtos, depois em alterar. Assim aparecerá o status, descrição, estoque mínimo e estoque máximo do produto para que o usuário possa fazer alteração. Em seguida deverá acionar o botão salvar para que as alterações feitas sejam salvas.

Para excluir um produto do sistema, o usuário deverá acessar o menu de produto, em seguida consultar produtos, depois excluir. Aparecerá uma mensagem perguntando se o usuário realmente quer excluir o produto. Para excluir o usuário deve confirmar acionando o botão ok, caso acione o botão cancelar o produto não será removido do sistema. Na Figura 43 tem-se a tela do menu produto.

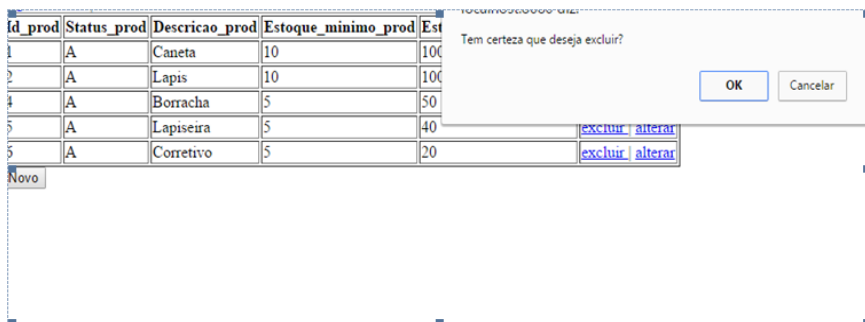
Figura 43: Menu de produto



Fonte: Próprio Autor (2018)

Na Figura 44 tem-se a tela para inclusão, alteração ou exclusão de um produto.

Figura 44: Tela excluir produto



Fonte: Próprio Autor (2018)

Na Figura 45 tem-se parte do código fonte da tela produto criado em linguagem JAVA. Como realizar a busca de um registro no banco de dados pelo seu identificador único e como excluir ou alterar um produto pelo seu identificador.

Figura 45: Código fonte da tela produto: Acesso a dados

```

63
64
65 public void excluir (Produto produto){
66     String sql = "DELETE FROM PRODUTO WHERE id_prod = ?";
67
68     //Criando um preparador - Statment
69     try {
70         PreparedStatement preparador = con.prepareStatement(sql);
71         preparador.setInt(1, produto.getId_prod());
72
73         //Executando o comando SQL no BD
74         preparador.execute();
75         preparador.close();
76     }catch (SQLException ex) {
77         System.out.println("Erro: "+ex.getMessage());
78     }
79 }
80
81 //se o produto ja possuir o id so faz alteração dos dados, se nao cadastra ele no BD
82 public void salvar(Produto produto){
83     if(produto.getId_prod()!=null && produto.getId_prod()!=0){
84         alterar(produto);
85     }else{
86         cadastrar(produto);
87     }
88 }
89
90 /**
91  * Busca de um registro no BD pelo numero do id do produto
92  * @param id É um inteiro que representa o numero do id do produto a ser buscado
93  * @return Retorna um objeto produto quando encontra ou null quando não encontra
94  */
95 public Produto buscarPorId_prod(Integer id_prod){
96     String sql = "Select * from produto where id_prod=?";
97
98     try (PreparedStatement preparador = con.prepareStatement(sql)){
99         preparador.setInt(1, id_prod);
100
101         ResultSet resultado = preparador.executeQuery();

```

Fonte: Próprio Autor (2018)

O cadastro de compras é feito através do menu de compra. O usuário deverá acionar a opção comprar produto, em seguida irá aparecer os campos em que o usuário deverá informar o Identificador do fornecedor, Identificador do produto, quantidade de compra, valor unitário da compra e data da compra a ser cadastrada. Após inserir os dados da compra o usuário deverá acionar o botão salvar para cadastrar a compra no sistema.

Para alterar registros de uma compra o usuário deverá acessar o menu de compra. Em seguida consultar compras, depois clica no botão alterar, irá aparecer o Identificador do fornecedor, Identificador do produto, quantidade de compras, valor unitário e data da compra para que o usuário possa fazer alteração. Em seguida deverá acionar o botão salvar para que as alterações feitas sejam salvas.

Para excluir uma compra do sistema, o usuário deverá acessar o menu de compra, em seguida consultar compras, depois acionar o botão excluir, irá aparecer uma mensagem perguntando se o usuário realmente quer excluir a compra. Para excluir o usuário deve confirmar acionando o botão ok, caso acione o botão cancelar a compra não será removido do sistema. Na Figura 46 tem-se a tela do menu de compra.

Figura 46: Menu de compra



Fonte: Próprio Autor (2018)



Na Figura 47 tem-se parte do código fonte da tela compra criado em linguagem JAVA. Mostra como realizar a busca de todos os registros de compras ou fazer uma busca de uma compra específica utilizando seu identificador único.

Figura 47: Código fonte da tela compra: Acesso a dados

```

97-  /**
98   * Busca de um registro no BD pelo numero do id da compra
99   * @param id É um inteiro que representa o numero do id da compra a ser buscada
100  * @return Retorna um objeto compra quando encontra ou null quando não encontra
101  */
102- public Compra buscarPorId_compra(Integer id_compra){
103  String sql = "Select * from compra where id_compra=?";
104  try (PreparedStatement preparador = con.prepareStatement(sql)){
105  preparador.setInt(1, id_compra);
106  ResultSet resultado = preparador.executeQuery();
107
108  //Posicionando o cursor no primeiro registro
109  if(resultado.next()){
110  Compra compra = new Compra();
111  compra.setId_compra(resultado.getInt("id_compra"));
112  compra.setId_forn(resultado.getInt("Id_forn"));
113  compra.setId_prod(resultado.getInt("Id_prod"));
114  compra.setQtde_compra(resultado.getInt("Qtde_compra"));
115  compra.setValor_unit_compra(resultado.getFloat("Valor_unit_compra"));
116  compra.setData_compra(resultado.getString("Data_compra"));
117
118  return compra;
119  }
120  } catch (SQLException e){
121  // TODO Auto-generated catch block
122  e.printStackTrace();
123  }
124  return null;
125  }
126- /**
127  * Faz a busca de todos os registros da tabela de compra
128  * @return Uma lista de objetos compra contendo 0 elementos quando não tiver registro
129  * ou n elementos quando tiver resultado
130  */
131- public List<Compra> buscarTodos(){
132  String sql = "Select * from compra ";
133  List<Compra> lista = new ArrayList<Compra>();
134  try (PreparedStatement preparador = con.prepareStatement(sql)){
135  ResultSet resultado = preparador.executeQuery();

```

Fonte: Próprio Autor (2018)

O cadastro de vendas é feito através do menu de venda. O usuário deverá acionar a opção vender produto e em seguida irá aparecer os campos em que o usuário deverá informar o identificador do cliente, identificador do produto, quantidade de venda, valor unitário da venda e data da venda a ser cadastrada. Após inserir os dados da venda o usuário deverá acionar o botão salvar para cadastrar a venda no sistema.

Para alterar registros de uma venda o usuário deverá acessar o menu de venda. Em seguida acionar opção consultar vendas, depois o botão alterar, irá aparecer o identificador do cliente, identificador do produto, quantidade de vendas, valor unitário e data da venda para que o usuário possa fazer alteração. Em seguida deverá acionar o botão salvar para que as alterações feitas sejam salvas.

Para excluir uma venda do sistema, o usuário deverá acessar o menu de venda, em seguida acionar a opção consultar vendas, depois o botão excluir. Aparecerá uma mensagem perguntando se o usuário realmente quer excluir a

venda, para excluir o usuário deve confirmar acionando o botão ok, caso acione o botão cancelar a venda não será removida do sistema.

Figura 48: Menu de venda



Fonte: Próprio Autor (2018)

Na Figura 49 tem-se parte do código fonte da tela venda criado em linguagem JAVA. Mostra como é realizado a captura de parâmetro da tela, como realizar a busca de um objeto venda no banco de dados e como encaminhar esse objeto para a tela do sistema.

Figura 49: Código Fonte da tela venda: Acesso a dados

```

*VendaController.java
67 RequestDispatcher dispatcher = req.getRequestDispatcher("WEB-INF/listavenda.jsp");
68 dispatcher.forward(req, resp);
69
70 }else if(acao.equals("alt")){
71
72 //captura parametro da tela
73 String id_venda = req.getParameter("id_venda");
74 VendaDAO vendaDAO = new VendaDAO();
75
76 //busca objeto venda no banco
77 Venda venda = vendaDAO.buscarPorId_venda(Integer.parseInt(id_venda));
78
79 //seta atributos no request com objeto venda
80 req.setAttribute("venda", venda);
81
82 //encaminha objeto venda para tela
83 RequestDispatcher dispatcher = req.getRequestDispatcher("WEB-INF/formvenda.jsp");
84 dispatcher.forward(req, resp);
85
86 }else if(acao.equals("cad")){
87
88 //busca objeto venda no banco
89 Venda venda = new Venda();
90 venda.setId_venda(0);
91 venda.setId_cliente(0);
92 venda.setId_prod(0);
93 venda.setQtde_venda(0);
94 venda.setData_venda("");
95 venda.setvalor_unit_venda(0);
96
97 //seta atributos no request com objeto venda
98 req.setAttribute("venda", venda);
99 //encaminha objeto venda para tela
100 RequestDispatcher dispatcher = req.getRequestDispatcher("WEB-INF/formvenda.jsp");
101 dispatcher.forward(req, resp);
102 }
103
104 }
105

```

Fonte: Próprio Autor (2018)

Na Figura 50 tem-se parte do código fonte da tela venda criado em linguagem JAVA. Mostra como cadastrar, alterar ou excluir uma venda do sistema e como acessar os dados do objeto venda no banco de dados.

Figura 50: Código Fonte da tela venda: Acesso a dados

```

17 public void cadastrar(Venda venda) {
18     String sql = " insert into venda (id_cliente, id_prod, qtde_venda, data_venda,valor_unit_venda) valu
19
20     //Criando um preparador - Statment
21     try (PreparedStatement preparador = con.prepareStatement(sql)){
22
23     //substituindo as ? pelos dados da venda
24     preparador.setInt(1,venda.getId_cliente());
25     preparador.setInt(2,venda.getId_prod());
26     preparador.setInt(3,venda.getQtde_venda());
27     preparador.setString(4,venda.getData_venda());
28     preparador.setFloat(5,venda.getValor_unit_venda());
29
30     //Executando o comando SQL no BD
31     preparador.execute();
32
33     }catch (SQLException e) {
34     // TODO Auto-generated catch block
35     e.printStackTrace();
36     }
37     }
38
39     public void alterar (Venda venda){
40     String sql = "UPDATE venda SET id_cliente=?, id_prod=?, qtde_venda=?, data_venda=?,valor_unit_vende
41
42     //Criando um preparador - Statment
43     try{
44     PreparedStatement preparador = con.prepareStatement(sql);
45
46     //substituindo as ? pelos dados da venda
47     preparador.setInt(1,venda.getId_cliente());
48     preparador.setInt(2,venda.getId_prod());
49     preparador.setInt(3,venda.getQtde_venda());
50     preparador.setString(4,venda.getData_venda());
51     preparador.setFloat(5,venda.getValor_unit_venda());
52     preparador.setInt(6,venda.getId_venda());
53
54     //Executando o comando SQL no BD
55     preparador.execute();
56     preparador.close();

```

Fonte: Próprio Autor (2018)

Para consultar o estoque, o usuário deverá acessar o menu de estoque, em seguida a opção consultar estoque, irá aparecer o identificador do estoque, identificador do produto, quantidade de produtos em estoque e valor unitário.

Figura 51: Menu estoque



Fonte: Próprio Autor (2018)



Na Figura 52 tem-se parte do código fonte da tela estoque criado em linguagem JAVA. Como obter a lista de estoque, encaminhar ao jsp e setar atributos no request com objeto estoque.

Figura 52: Código Fonte da tela consultar estoque: Acesso a dados

```

41
42 String acao = req.getParameter("acao");
43 if(acao != null && acao.equals("lis")){
44     EstoqueDAO estoqueDAO = new EstoqueDAO();
45
46     //1 Obter a lista
47     List<Estoque> lista = estoqueDAO.buscarTodos();
48
49     //Engavetar no request a lista
50     req.setAttribute("lista",lista);
51
52     //Encaminhamento ao JSP
53     RequestDispatcher dispatcher = req.getRequestDispatcher("WEB-INF/listaestoque.jsp");
54     dispatcher.forward(req, resp);
55 }
56
57 if (acao != null && acao.equals("setarEstoque")) {
58     String id_estoque = req.getParameter("id_estoque");
59     Estoque estoque = new Estoque();
60     estoque.setId_estoque(0);
61     estoque.setId_prod(0);
62     estoque.setQtde_estoque(0);
63     estoque.setValor_unit_estoque(0);
64
65     //seta atributos no request com objeto estoque
66     req.setAttribute("Estoque",estoque);
67     EstoqueDAO estoqueDAO = new EstoqueDAO();
68     Estoque estoque1 = estoqueDAO.buscarPorId_estoque(Integer.parseInt(id_estoque));
69     RequestDispatcher dispatcher = req.getRequestDispatcher("WEB-INF/formestoque.jsp");
70     dispatcher.forward(req, resp);
71 }
72
73 @Override
74 protected void doPost(HttpServletRequest req, HttpServletResponse resp)
75     throws ServletException, IOException {
76     String id_estoque = req.getParameter("id_estoque");
77     String id_prod = req.getParameter("id_prod");
78     String qtde_estoque = req.getParameter("qtde_estoque");
79     String valor_unit_estoque = req.getParameter("valor_unit_estoque");
80 }

```

Fonte: Próprio Autor (2018)

Na Figura 53 tem-se parte do código fonte da tela estoque criado em linguagem JAVA. Como obter um registro específico no banco de dados utilizando o identificador único do estoque, ou realizar a busca de todos os registros de uma vez só.

Figura 53: Código fonte da tela estoque: Acesso a dados

```

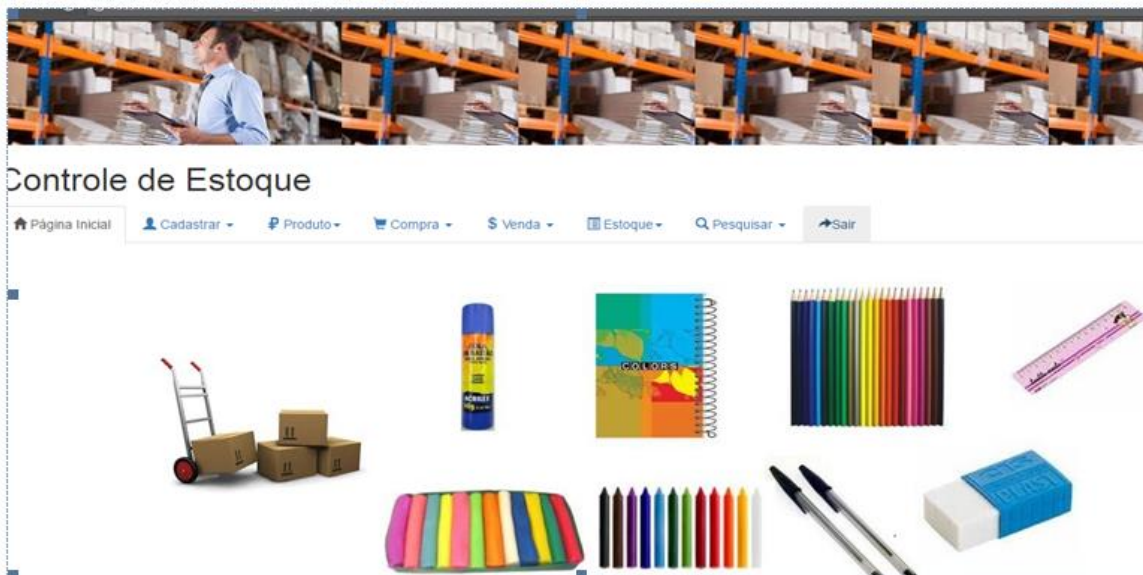
18 /**
19  * Busca de um registro no BD pelo numero do id do estoque
20  * @param id É um inteiro que representa o numero do id do estoque a ser buscado
21  * @return Retorna um objeto estoque quando encontra ou null quando não encontra
22  */
23 public Estoque buscarPorId_estoque(Integer id_estoque){
24     String sql = "Select * from estoque where id_estoque=?";
25     try (PreparedStatement preparador = con.prepareStatement(sql)){
26         preparador.setInt(1, id_estoque);
27         ResultSet resultado = preparador.executeQuery();
28
29         //Posicionando o cursor no primeiro registro
30         if(resultado.next()){
31             Estoque estoque = new Estoque();
32             estoque.setId_estoque(resultado.getInt("id_estoque"));
33             estoque.setId_prod(resultado.getInt("Id_prod"));
34             estoque.setQtde_estoque(resultado.getInt("Qtde_estoque"));
35             estoque.setValor_unit_estoque(resultado.getFloat("Valor_unit_estoque"));
36         }
37         return estoque;
38     }catch (SQLException e){
39         // TODO Auto-generated catch block
40         e.printStackTrace();
41     }return null;
42 }
43
44 /**
45  * Faz a busca de todos os registros da tabela de estoque
46  * @return Uma lista de objetos estoque contendo 0 elementos quando não tiver registro
47  * ou n elementos quando tiver resultado
48  */
49 public List<Estoque> buscarTodos(){
50     String sql = "Select * from estoque ";
51     List<Estoque> lista = new ArrayList<Estoque>();
52     try (PreparedStatement preparador = con.prepareStatement(sql)){
53         ResultSet resultado = preparador.executeQuery();
54
55         //Posicionando o cursor no primeiro registro
56         while(resultado.next()){
57             Estoque estoque = new Estoque();

```

Fonte: Próprio Autor (2018)

Para sair do sistema o usuário deverá acessar a opção sair. Clicando nessa opção ele será redirecionando para página inicial de login. Segue abaixo a Figura 54 com a opção sair do sistema.

Figura 54: Opção sair



Fonte: Próprio Autor (2018)

## 5 RESULTADOS E DISCUSSÃO

O Sistema de Gestão de Estoque para Plataforma Web foi desenvolvido para a Papelaria Real através da linguagem de programação JAVA e do SGBD MySQL. Foi realizada análises para verificar a importância do controle de estoque dentro da empresa, e através disso ficou bastante esclarecido que uma empresa que não realiza o controle do seu estoque está perdendo bastante dinheiro.

A melhoria no fluxo das organizações que tem algum tipo de controle é de fundamental importância para o sucesso de uma empresa. Isto é exercido de forma eficiente e eficaz na empresa Papelaria Real, através deste trabalho desenvolvido. O objetivo principal deste trabalho, de implementar um sistema para automatizar, organizar e controlar o estoque da papelaria Real foi atingido, pois o sistema permite um total controle dos produtos e ainda auxilia a tomada de decisão.

Após finalizar o sistema, o mesmo foi apresentado para o gerente da empresa Papelaria Real, para que o mesmo pudesse analisar e utilizar o sistema desenvolvido. O gerente utilizou localmente, fez os testes, e aprovou o sistema pela facilidade de utilização e pelo retorno financeiro que irá trazer para sua empresa. O mesmo informou que nenhuma dificuldade de utilização foi percebida ao realizar os testes. Através da entrevista informal feita com o gerente, alguns pontos importantes foram destacados:

- Com a utilização do sistema não irá mais ter a necessidade de ir sempre até o estoque verificar a disponibilidade de um produto na hora de realizar uma venda;
- O sistema dar permissão apenas para o gerente preencher os dados, isso é de fundamental importância para que pessoas não autorizadas não possam mais fazer alteração nos dados de estoque;
- O sistema evita a existência de erros corriqueiros no estoque;
- Produtos que de certa forma não trazem benefícios para empresa serão descartados da lista de produtos;
- A projeção de vendas possibilita realizar previsões e preparar o setor financeiro da empresa para futuras compras;
- Logo em breve o sistema estará em uso definitivo na empresa Papelaria Real e irá substituir a planilha eletrônica utilizada.

## 6 CONSIDERAÇÕES FINAIS

Neste trabalho foi desenvolvido um Sistema de Gestão de Estoque para Plataforma Web para automatizar, organizar e controlar o estoque da Papelaria Real de forma eficiente e eficaz. O sistema controla todo tipo de movimentação, de venda ou compra de produtos e à medida que for comprando ou vendendo um produto realiza o controle do estoque da empresa em tempo real, possibilitando a organização e visando aumento do lucro.

Financeiramente falando o estoque de uma empresa é de fundamental importância, pois a partir do momento que a empresa realiza o giro desses estoques, o seu valor se transforma em dinheiro, o que vem a beneficiar o fluxo de caixa trazendo retorno sobre o investimento que foi aplicado. Por este motivo o controle de estoque realizado de forma automatizada tem se tornado uma necessidade competitiva nas empresas.

Empresas que possuem grandes quantidades de estoque correm bastante risco de cometerem erros na hora da tomada de decisão. Isto ocorre principalmente na hora de realizar compra de produtos, se as empresas não possuírem um sistema de informação e suas técnicas de processos não forem automatizados os riscos aumentará. Estoques administrados de forma incorreta são classificados como um dos maiores custos para a empresa.

Este trabalho teve como principal objetivo desenvolver um sistema para controlar o estoque de produtos da empresa Papelaria Real. A empresa realiza o controle de estoque apenas por uma planilha eletrônica, onde o controle é feito manualmente por funcionário, gerando uma grande desconfiança em relação à apresentados dos dados de forma precisa.

A empresa Papelaria Real já testou o sistema e em breve estará com ele em utilização, para lhe auxiliar no controle dos produtos minimizando a possibilidade de erros através de cadastros de forma eficiente e eficaz. O controle de acesso possibilita uma grande confiabilidade ao gerente, onde o mesmo pode saber a qualquer momento o usuário que realizou alterações no sistema de gestão de estoque.

Algumas dificuldades foram encontradas durante o desenvolvimento do trabalho, no entanto o resultado foi positivo e satisfatório. Pois possibilitou o aprendizado de novas tecnologias de programação, engenharia de software e recursos avançados de banco de dados. Regras de negócios e vários conhecimentos específicos na área de estoques foram aprendidos. O presente trabalho possibilitou um grande desenvolvimento tanto profissional como pessoal.

Por fim, a realização desse trabalho faz surgir algumas possibilidades de trabalhos futuros. Abaixo segue sugestões de melhoria que ficam como trabalhos futuros para ser implementado no Sistema de gestão de estoque:

- Implementar um módulo de frente de caixa/financeiro, para que possa eliminar a utilização de dois sistemas diferentes na empresa, assim tornando mais eficiente e eficaz o processo de venda de produtos.
- Módulo financeiro, com custos mensais da empresa, fluxo de caixa e relatórios de faturamento e gastos.
- Fazer a implementação de notas fiscais eletrônicas embutida no módulo de vendas.
- Criar relatórios de projeção de vendas para que através da média dos produtos vendidos possa realizar o cálculo da projeção futura.

## REFERÊNCIAS

Astah. Disponível em: <<http://astah.net/>>. Acesso em: 08 abr. 2018.

BEZERRA, Eduardo. **Princípios de Análise e Projeto de Sistemas com UML**: Um guia prático para modelagem de sistemas orientados a objetos, 2ª ed. Rio de Janeiro, Elsevier Editora Ltda. 2006.

CONTAAZUL. **Software ContaAzul**. Joinville, 2013. Disponível em: <<https://contaazul.com>>. Acesso em: 28 mar. 2018.

CHING, Hong Yuh. **Gestão de Estoques na Cadeia de Logística Integrada**. Supply Chain, 2001.

CARVALHO, José Mexia Crespo de. **Logística**. 3 ed. Lisboa: Sílabo, 2002.

CACHOEIRA, José. **Otimização dos resultados com estoques consignados**. Curso de Ciências Contábeis, Universidade Regional de Blumenau. Blumenau, 2006. Trabalho de Conclusão de Curso (Bacharel em Ciências Contábeis) – Centro de Ciências Sociais Aplicadas, Universidade Regional de Blumenau, Blumenau.

CARVALHO, Flávia Pereira de. **Introdução à linguagem HTML**. Universidade de São Paulo – USP. 2004.

DEITEL, P.; DEITEL, H. **Java Como Programar**. 10. ed. São Paulo: Pearson, 2016.

DIAS, Marco Aurélio P. **Administração de Materiais**. 6 ed. Ed. Compacta. São Paulo: Atlas S.A, 2009.

FERNANDES, José Carlos de F. **Administração de material**: Uma abordagem básica. 3 ed. São Paulo: Atlas S.A, 1987.

GONÇALVES, Paulo Sérgio. **Administração de materiais**. 2 ed. Rio de Janeiro: Elsevier, 2007.

HORSTMANN, Cay S.; CORNELL, Gary. **Core Java, Volume I: Fundamentos**. 8 ed. São Paulo: Pearson Prentice Hall, 2010.

IZIDORO, Bradley. **Sistema para controle de estoque dos almoxarifados de agências de correios do estado de Santa Catarina**. 2011. Trabalho de Conclusão de Curso (Bacharel em Sistemas de Informação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

JAVA. **Obtenha Informações sobre a Tecnologia Java**. 2018. Disponível em: <[https://www.java.com/pt\\_BR/about/](https://www.java.com/pt_BR/about/)> Acesso em 05 abr. 2018.

MARTINS, Petrônio Garcia. **Administração de materiais e recursos patrimoniais**. São Paulo: Saraiva, 2001.

Oracle MySQL. Disponível em: <<https://www.oracle.com/mysql/index.html>>. Acesso em: 10 mai. 2018.

PASCOAL, Janaína Araújo. **Gestão estratégica de recursos materiais: Controle de estoque e armazenagem**. Monografia (Graduação). Curso de Administração. Centro Universitário de João Pessoa – UNIPÉ. João Pessoa, 2008.

PRESSMAN, Roger S. **Engenharia de Software: uma abordagem profissional**. 7. ed. São Paulo: Pearson Makron Books, 2011.

POZO, Hamilton. **Administração de recursos materiais e patrimoniais: Uma abordagem logística**. São Paulo: Atlas, 2001.

SILBERSCHATZ, Abraham; KORTH, Henry F.; S. SUDARSHAN. **Sistema de Banco de Dados**. 5. ed. Rio de Janeiro: Elsevier, 2006.

SOMMERVILLE, Ian. **Engenharia de Software**. 8. ed. São Paulo: Pearson Addison Wesley, 2007.

SILVA, Maurício Samy. **JavaScript**. 1ª ed. São Paulo, Novatec Editora Ltda. 2010.

## APÊNDICE A - REQUISITOS FUNCIONAIS

Descrevem funcionalidades e serviços do sistema, documenta como o sistema deve reagir a entradas específicas, e como deve se comportar em determinados momento. Segue alguns requisitos funcionais:

Quadro 1 – Requisito funcional: fazer login

Identificador	Nome
<b>RF-01.</b>	<b>Fazer login</b>
<b>Descrição</b>	
O sistema deve permitir que usuário possa entrar no sistema informando o login e senha que foi cadastrado.	

Fonte: Próprio Autor (2018)

Quadro 2 - Requisito funcional: cadastrar produtos

Identificador	Nome
<b>RF-02.</b>	<b>Cadastrar Produtos</b>
<b>Descrição</b>	
O sistema deve permitir a manutenção no cadastro de produtos. Um usuário com perfil de administrador deverá ser capaz de inserir e visualizar os produtos cadastrados.	

Fonte: Próprio Autor (2018)

Quadro 3 - Requisito funcional: alterar produtos

Identificador	Nome
<b>RF-03.</b>	<b>Alterar Produtos</b>
<b>Descrição</b>	
O sistema deve permitir que o usuário com perfil de administrador possa fazer alteração nos registros dos produtos cadastrados.	

Fonte: Próprio Autor (2018)

Quadro 4 - Requisito funcional: excluir produtos

Identificador	Nome
<b>RF-04.</b>	<b>Excluir Produtos</b>
<b>Descrição</b>	
O sistema deve permitir que o usuário com perfil de administrador possa fazer exclusão de produtos pelo seu identificador único.	

Fonte: Próprio Autor (2018)

Quadro 5 – Requisito funcional: realizar compra

Identificador	Nome
<b>RF-05.</b>	<b>Realizar Compra de Produtos</b>
<b>Descrição</b>	
O sistema deve permitir que o usuário com perfil de administrador possa realizar a compra de produtos.	

Fonte: Próprio Autor (2018)



Quadro 6 - Requisito funcional: alterar compra

Identificador	Nome
<b>RF-06.</b>	<b>Alterar Compra de Produtos</b>
<b>Descrição</b>	
O sistema deve permitir que o usuário com perfil de administrador possa alterar uma compra, informando a quantidade desejada de produtos a ser comprados, e que possa também fazer alteração do fornecedor para o produto especificado.	

Fonte: Próprio Autor (2018)

Quadro 7 – Requisito funcional: excluir compra

Identificador	Nome
<b>RF-07.</b>	<b>Excluir Compra de Produtos</b>
<b>Descrição</b>	
O sistema deve permitir que o usuário com perfil de administrador possa excluir uma compra, informando o identificador da compra a ser excluída.	

Fonte: Próprio Autor (2018)

Quadro 8 – Requisito funcional: realizar venda

Identificador	Nome
<b>RF-08.</b>	<b>Realizar Venda de Produtos</b>
<b>Descrição</b>	
O sistema deve permitir que funcionários possam realizar a venda de produtos para os clientes.	

Fonte: Próprio Autor (2018)

Quadro 9 - Requisito funcional: alterar venda

Identificador	Nome
<b>RF-09.</b>	<b>Alterar Venda de Produtos</b>
<b>Descrição</b>	
O sistema deve permitir que funcionários possam alterar uma venda, informando a quantidade desejada de produtos a ser vendidos conforme desejado pelo cliente.	

Fonte: Próprio Autor (2018)

Quadro 10 - Requisito funcional: excluir venda

Identificador	Nome
<b>RF-10.</b>	<b>Excluir Venda de Produtos</b>
<b>Descrição</b>	
O sistema deve permitir que funcionários possam excluir uma venda, informando o identificador da venda a ser excluída. Nos casos em que o cliente deseje não comprar mais o produto e solicita o cancelamento da venda.	

Fonte: Próprio Autor (2018)

Quadro 11 – Requisito funcional: consultar estoque

Identificador	Nome
<b>RF-11.</b>	<b>Consultar Estoque de Produtos</b>
<b>Descrição</b>	
O sistema deve permitir que usuário com perfil de administrador possa entrar no sistema e verificar a quantidade de produtos disponíveis no estoque e quais os produtos que estão próximos do estoque mínimo.	

Fonte: Próprio Autor (2018)

Quadro 12 – Requisito funcional: cadastrar cliente

Identificador	Nome
<b>RF-12.</b>	<b>Cadastrar Clientes</b>
<b>Descrição</b>	
O sistema deve permitir que usuário com perfil de administrador, possa cadastrar os clientes desta empresa, como dados básicos de nome, endereço, sexo e telefone.	

Fonte: Próprio Autor (2018)

Quadro 13 - Requisito funcional: alterar cliente

Identificador	Nome
<b>RF-13.</b>	<b>Alterar Clientes</b>
<b>Descrição</b>	
O sistema deve permitir que usuário com perfil de administrador, possa fazer alteração nos registros dos clientes desta empresa, como dados básicos de nome, endereço, sexo e telefone. Informando o identificador do cliente a ser alterado.	

Fonte: Próprio Autor (2018)

Quadro 14 – Requisito funcional: excluir cliente

Identificador	Nome
<b>RF-14.</b>	<b>Excluir Clientes</b>
<b>Descrição</b>	
O sistema deve permitir que usuário com perfil de administrador, possa fazer exclusão de um cliente sistema. Informando o identificador do cliente a ser excluído.	

Fonte: Próprio Autor (2018)

Quadro 15 – Requisito funcional: cadastrar fornecedor

Identificador	Nome
<b>RF-15.</b>	<b>Cadastrar Fornecedores</b>
<b>Descrição</b>	
O sistema deve permitir que usuário com perfil de administrador, possa cadastrar os fornecedores desta empresa, como dados básicos de nome, endereço, sexo e telefone.	

Fonte: Próprio Autor (2018)

Quadro 16 - Requisito funcional: alterar fornecedor

Identificador	Nome
<b>RF-16.</b>	<b>Alterar Fornecedores</b>
<b>Descrição</b>	
O sistema deve permitir que usuário com perfil de administrador, possa fazer alteração nos registros dos fornecedores desta empresa, como dados básicos de nome, endereço, sexo e telefone. Informando o id do fornecedor a ser alterado.	

Fonte: Próprio Autor (2018)

Quadro 17 – Requisito funcional: excluir fornecedor

Identificador	Nome
<b>RF-17.</b>	<b>Excluir Fornecedores</b>
<b>Descrição</b>	
O sistema deve permitir que usuário com perfil de Administrador, possa fazer exclusão de um fornecedor no sistema. Informando o identificador do fornecedor a ser excluído.	

Fonte: Próprio Autor (2018)

Quadro 18 – Requisito funcional: cadastrar funcionário

Identificador	Nome
<b>RF-18.</b>	<b>Cadastrar Funcionários</b>
<b>Descrição</b>	
O sistema deve permitir que usuário com perfil de administrador, possa cadastrar os funcionários desta empresa, como dados básicos de nome, endereço, sexo e telefone.	

Fonte: Próprio Autor (2018)

Quadro 19 - Requisito funcional: alterar funcionário

Identificador	Nome
<b>RF-19.</b>	<b>Alterar Funcionários</b>
<b>Descrição</b>	
O sistema deve permitir que usuário com perfil de administrador, possa fazer alteração nos registros dos funcionários desta empresa, como dados básicos de nome, endereço, sexo e telefone. Informando o id do funcionário a ser alterado.	

Fonte: Próprio Autor (2018)

Quadro 20 – Requisito funcional: excluir funcionário

Identificador	Nome
<b>RF-20.</b>	<b>Excluir Funcionários</b>
<b>Descrição</b>	
O sistema deve permitir que usuário com perfil de administrador, possa fazer exclusão de um funcionário no sistema. Informando o identificador do funcionário a ser excluído.	

Fonte: Próprio Autor (2018)

Quadro 21 – Requisito funcional: consultar dados da empresa

Identificador	Nome
<b>RF-21.</b>	<b>Consultar dados da empresa</b>
<b>Descrição</b>	
O sistema deve permitir que usuário com perfil de administrador, possa consultar os dados da empresa. Informando o seu CNPJ.	

Fonte: Próprio Autor (2018)

## APÊNDICE B - REQUISITOS NÃO FUNCIONAIS

Requisitos não funcionais definem propriedades e restrições do sistema como um todo ou de partes do sistema e podem ser mais críticos que requisitos funcionais.

Segue alguns requisitos não funcionais:

Quadro 22 - Requisito não funcional: usabilidade

Identificador	Nome
<b>RNF-01.</b>	<b>Usabilidade</b>
<b>Descrição</b>	
O sistema deve ser simples e intuitivo para que pessoas com pouco conhecimento em informática consigam operá-lo sem problemas.	

Fonte: Próprio Autor (2018)

Quadro 23 - Requisito não funcional: login

Identificador	Nome
<b>RNF-02.</b>	<b>Login no Sistema</b>
<b>Descrição</b>	
Para entrar no sistema o usuário deverá obrigatoriamente informar o login e senha que foi cadastrado. Caso digite login ou senha incorretamente será emitida uma mensagem de que o usuário não foi encontrado na base de dados.	

Fonte: Próprio Autor (2018)

Quadro 24 – Requisito não funcional: funcionalidades do sistema

Identificador	Nome
<b>RNF-03.</b>	<b>Acesso as funcionalidades do sistema</b>
<b>Descrição</b>	
Todo usuário autorizado, deverá efetuar login para realizar as funções pertinentes ao mesmo.	

Fonte: Próprio Autor (2018)

Quadro 25 – Requisito não funcional: funcionalidades do sistema

Identificador	Nome
<b>RNF-04.</b>	<b>Acesso as funcionalidades do sistema</b>
<b>Descrição</b>	
Quando o usuário deixar de acessar as funcionalidades do sistema por mais de 5 minutos a sessão expira e o usuário é redirecionado para página inicial do sistema para que possa fazer login novamente, caso queira ter acesso as funcionalidades do sistema.	

Fonte: Próprio Autor (2018)

Quadro 26 – Requisito não funcional: armazenamento de dados

Identificador	Nome
<b>RNF-05.</b>	<b>Armazenamento dos dados</b>
<b>Descrição</b>	
Todos os dados devem ser armazenados em um banco de dados, apenas terão acesso a esses dados usuários cadastrados no sistema, mediante um login com a solicitação de uma senha de acesso.	

Fonte: Próprio Autor (2018)

Quadro 27 - Requisito não funcional: confiabilidade

Identificador	Nome
<b>RNF-06.</b>	<b>Confiabilidade</b>
<b>Descrição</b>	
O sistema deve informar ao usuário quando ele tentar fazer uma operação ilegal ou quando ele está prestes a realizar uma operação que pode ser perigosa.	

Fonte: Próprio Autor (2018)

Quadro 28 - Requisito não funcional: confiabilidade

Identificador	Nome
<b>RNF-07.</b>	<b>Confiabilidade</b>
<b>Descrição</b>	
O sistema deve possuir mecanismos que garantam que o usuário não perca informações. O sistema precisa oferecer recursos que possibilitem que o usuário recupere o conteúdo do controle de estoque caso ocorra, como por exemplo, erro de execução do sistema, queda de energia, etc.	

Fonte: Próprio Autor (2018)

Quadro 29 - Requisito não funcional: segurança

Identificador	Nome
<b>RNF-08.</b>	<b>Segurança</b>
<b>Descrição</b>	
O sistema deve ser seguro, para evitar ao máximo invasões, pois estes podem causar prejuízos a empresa.	

Fonte: Próprio Autor (2018)

Quadro 30 - Requisito não funcional: login

Identificador	Nome
<b>RNF-09.</b>	<b>Logar no sistema</b>
<b>Descrição</b>	
Cada usuário só poderá estar logado em um único sistema, em uma única máquina. Não pode permitir o mesmo usuário logar duas vezes ao mesmo tempo.	

Fonte: Próprio Autor (2018)

Quadro 31 – Requisito não funcional: cadastrar produto

Identificador	Nome
<b>RNF-010.</b>	<b>Cadastro de Produtos</b>
<b>Descrição</b>	
O sistema não deve permitir duplicidade no cadastro do produto. Para isso, cada produto terá um código único (identificador).	

Fonte: Próprio Autor (2018)

Quadro 32 – Requisito não funcional: cadastrar cliente

Identificador	Nome
<b>RNF-011.</b>	<b>Cadastro de Clientes</b>
<b>Descrição</b>	
O sistema não deve permitir duplicidade no cadastro de clientes. Para isso, cada cliente terá um código único (identificador).	

Fonte: Próprio Autor (2018)

Quadro 33 – Requisito não funcional: cadastrar fornecedor

Identificador	Nome
<b>RNF-012.</b>	<b>Cadastro de Fornecedores</b>
<b>Descrição</b>	
O sistema não deve permitir duplicidade no cadastro de fornecedores. Para isso, cada fornecedor terá um código único (identificador).	

Fonte: Próprio Autor (2018)

Quadro 34 – Requisito não funcional: excluir compra

Identificador	Nome
<b>RNF-013.</b>	<b>Excluir Compra</b>
<b>Descrição</b>	
Quando um usuário tentar excluir uma compra, o sistema deverá emitir uma mensagem de confirmação perguntando se ele deseja realmente excluir, a compra só será excluída caso o usuário confirme a exclusão.	

Fonte: Próprio Autor (2018)

Quadro 35 – Requisito não funcional: excluir venda

Identificador	Nome
<b>RNF-014.</b>	<b>Excluir Venda</b>
<b>Descrição</b>	
Quando um usuário tentar excluir uma venda, o sistema deverá emitir uma mensagem de confirmação perguntando se ele deseja realmente excluir, a venda só será excluída caso o usuário confirme a exclusão.	

Fonte: Próprio Autor (2018)

Quadro 36 - Requisito não funcional: disponibilidade

Identificador	Nome
<b>RNF-015.</b>	<b>Disponibilidade</b>
<b>Descrição</b>	
O sistema deve estar disponível 24 horas por dia, sete dias por semana. Caso seja necessária alguma manutenção no sistema, fazê-lo em uma versão para testes, não disponível para os usuários, e então aplicar as mudanças na versão disponível para os usuários.	

Fonte: Próprio Autor (2018)

Quadro 37 – Requisito não funcional: linguagem de implementação

Identificador	Nome
<b>RNF-016.</b>	<b>Linguagem de Implementação</b>
<b>Descrição</b>	
O sistema deverá ser implementado em Java, utilizando os conceitos de programação orientada a objetos, utilizando a IDE Eclipse.	

Fonte: Próprio Autor (2018)

Quadro 38 – Requisito não funcional: interface

Identificador	Nome
<b>RNF-017.</b>	<b>Interface</b>
<b>Descrição</b>	
O sistema de gestão de estoque será construído para rodar em ambiente web. A interface será implementada em HTML contendo código Java Script e JSP.	

Fonte: Próprio Autor (2018)

Quadro 39 – Requisito não funcional: banco de dados

Identificador	Nome
<b>RNF-018.</b>	<b>Banco de dados</b>
<b>Descrição</b>	
O banco de dados utilizado será o MySQL e será utilizado 6 triggers e um procedures para fazer o controle do estoque sempre que for feito inserts, updates e delete nas tabelas compras e vendas de produtos.	

Fonte: Próprio Autor (2018)

Quadro 40 - Requisito não funcional: desempenho

Identificador	Nome
<b>RNF-019.</b>	<b>Desempenho</b>
<b>Descrição</b>	
Ao registrar um item sendo vendido, a descrição e preço devem aparecer em, no máximo 2 segundos.	

Fonte: Próprio Autor (2018)

Quadro 41 - Requisito não funcional: segurança

Identificador	Nome
<b>RNF-020.</b>	<b>Segurança</b>
<b>Descrição</b>	
O sistema deve garantir que os dados estão protegidos de acessos não autorizados.	

Fonte: Próprio Autor (2018)



## APÊNDICE C - TABELAS E LISTAS DE RASTREABILIDADE

A rastreabilidade de requisitos pode ser implementada por várias ligações entre requisitos. Ela é muito utilizada para fornecer relacionamentos entre requisitos e implementação final do sistema e possibilita uma adequada compreensão dos relacionamentos entre os requisitos. No Quadro 42 tem-se tabelas e listas de rastreabilidade com requisitos funcionais do sistema.

Quadro 42 - Rastreabilidade

<b>Tabelas e Listas de Rastreabilidade</b>	<b>Versão</b>	<b>1.0</b>
--------------------------------------------	---------------	------------

Requisito	Depende de
RF02	RF01
RF03	RF01e RF02
RF04	RF01e RF02
RF05	RF01
RF06	RF01, RF02 e RF03
RF07	RF01e RF05
RF08	RF01, RF02 e RF05
RF09	RF01 e RF05
RF10	RF01, RF02 e RF05
RF11	RF01
RF12	RF01
RF13	RF01 e RF12
RF14	RF01 e RF12
RF15	RF01
RF16	RF01 e RF15
RF17	RF01 e RF15
RF18	RF01
RF19	RF01 e RF18
RF20	RF01 e RF18
RF21	RF01

Fonte: Próprio Autor (2018)

## APÊNDICE D – DICIONÁRIO DE DADOS

Nos quadros 43 a 52 é apresentado o dicionário de dados com as tabelas do sistema.

Quadro 43 – Dicionário de dados da tabela usuários

<b>Usuários:</b> Armazena informações de usuários do sistema			
<b>Campo</b>	<b>Tipo</b>	<b>Tamanho</b>	<b>Descrição</b>
Id	Int	11	Código do usuário (PK)
Nome	Varchar	50	Nome do usuário
Login	Varchar	30	Login do usuário
Senha	Varchar	30	Senha do usuário

Fonte: Próprio Autor (2018)

Quadro 44 – Dicionário de dados da tabela produtos

<b>Produtos:</b> Armazena informações dos produtos			
<b>Campo</b>	<b>Tipo</b>	<b>Tamanho</b>	<b>Descrição</b>
Id	Int	11	Código do produto(PK)
Status	Char	1	Status do produto
Descrição	Varchar	50	Descrição do produto
Estoque mínimo	Int	11	Estoque mínimo do produto
Estoque máximo	Int	11	Estoque máximo do produto

Fonte: Próprio Autor (2018)

Quadro 45 – Dicionário de dados da tabela fornecedores

<b>Fornecedores:</b> Armazena informações dos fornecedores			
<b>Campo</b>	<b>Tipo</b>	<b>Tamanho</b>	<b>Descrição</b>
Id	Int	11	Código do fornecedor(PK)
Nome	Varchar	50	Nome do fornecedor
Cidade	Varchar	30	Cidade do fornecedor
Bairro	Varchar	50	Bairro do fornecedor
Rua	Varchar	40	Rua do fornecedor
Numero	Integer	-	Número do endereço do fornecedor
Telefone	Varchar	30	Cep do cliente

Fonte: Próprio Autor (2018)

Quadro 46 – Dicionário de dados da tabela compra

<b>Compra:</b> Armazena informações das compras			
<b>Campo</b>	<b>Tipo</b>	<b>Tamanho</b>	<b>Descrição</b>
Id da compra	Int	11	Código compra(PK)
Id do fornecedor	Integer	-	Código do fornecedor(FK)
Id do produto	Int	11	Código do produto(FK)
Quantidade	Int	11	Quantidade de compras
Valor unitário	Decimal	(9,2)	Valor unitário da compra
Data	Varchar	30	Data da compra

Fonte: Próprio Autor (2018)

Quadro 47 – Dicionário de dados da tabela estoque

<b>Estoque:</b> Armazena informações do estoque de produtos			
<b>Campo</b>	<b>Tipo</b>	<b>Tamanho</b>	<b>Descrição</b>
Id do estoque	Int	11	Código do estoque(PK)
Id do produto	Int	11	Código do produto(FK)
Quantidade	Int	11	Quantidade de estoque
Valor unitário	Decimal	(9,2)	Valor unitário do estoque

Fonte: Próprio Autor (2018)

Quadro 48 – Dicionário de dados da tabela clientes

<b>Clientes:</b> Armazena informações dos clientes			
<b>Campo</b>	<b>Tipo</b>	<b>Tamanho</b>	<b>Descrição</b>
Id	Int	11	Código do cliente(PK)
Nome	Varchar	50	Nome do cliente
Estado	Char	3	Estado do cliente
Cidade	Varchar	30	Cidade do cliente
Bairro	Varchar	50	Bairro do cliente
Rua	Varchar	40	Rua do cliente
Numero	Integer	-	Número da residência do cliente
Telefone	Varchar	30	Telefone do cliente

Fonte: Próprio Autor (2018)

Quadro 49 – Dicionário de dados da tabela vendas

<b>Vendas:</b> Armazena informações de vendas dos produtos			
<b>Campo</b>	<b>Tipo</b>	<b>Tamanho</b>	<b>Descrição</b>
Id da venda	Int	11	Código da venda(PK)
Id do cliente	Integer	-	Código do cliente(FK)
Id do produto	Int	11	Código do produto(FK)
Quantidade	Int	11	Quantidade de vendas
Data	Varchar	30	Data da venda
Valor unitário	Decimal	(9,2)	Valor unitário da venda

Fonte: Próprio Autor (2018)

Quadro 50 – Dicionário de dados da tabela Administrador

<b>Administrador:</b> Armazena informações do administrador			
<b>Campo</b>	<b>Tipo</b>	<b>Tamanho</b>	<b>Descrição</b>
Id	Int	11	Código do administrador(PK)
Nome	Varchar	50	Nome do administrador
Estado	Char	3	Estado do administrador
Cidade	Varchar	30	Cidade do administrador
Bairro	Varchar	50	Bairro do administrador
Rua	Varchar	40	Rua do administrador
Numero	Int	11	Número da residência do administrador
Telefone	Varchar	30	Telefone do administrador
Data de	Varchar	30	Data de nascimento do

nascimento			administrador
Sexo	Varchar	10	Sexo do administrador
Salário	Varchar	30	Salário do administrador

Fonte: Próprio Autor (2018)

Quadro 51 – Dicionário de dados da tabela funcionários

<b>Funcionários:</b> Armazena informações dos funcionários			
<b>Campo</b>	<b>Tipo</b>	<b>Tamanho</b>	<b>Descrição</b>
Id	Int	11	Código do funcionário(PK)
Nome	Varchar	50	Nome do funcionário
Estado	Char	3	Estado do funcionário
Cidade	Varchar	30	Cidade do funcionário
Bairro	Varchar	50	Bairro do funcionário
Rua	Varchar	40	Rua do funcionário
Numero	Int	11	Número da residência do funcionário
Telefone	Varchar	30	Telefone do funcionário
Data de nascimento	Varchar	30	Data de nascimento do funcionário
Sexo	Varchar	10	Sexo do funcionário
Salário	Varchar	30	Salário do funcionário

Fonte: Próprio Autor (2018)

Quadro 52 – Dicionário de dados da tabela papelaria

<b>Papelaria:</b> Armazena informações da papelaria			
<b>Campo</b>	<b>Tipo</b>	<b>Tamanho</b>	<b>Descrição</b>
CNPJ	Varchar	30	Código da papelaria(PK)
Nome	Varchar	50	Nome da papelaria
Estado	Char	3	Estado da papelaria
Cidade	Varchar	30	Cidade da papelaria
Bairro	Varchar	50	Bairro da papelaria
Rua	Varchar	40	Rua da papelaria
Numero	Integer	-	Número da papelaria
Telefone	Varchar	30	Telefone da papelaria
Id do usuário	Integer	-	Código do usuário(FK)
Id do funcionário	Integer	-	Código do funcionário(FK)
Id do administrador	Integer	-	Código do administrador(FK)

Fonte: Próprio Autor (2018)