



**Fundação Educacional do Município de Assis  
Instituto Municipal de Ensino Superior de Assis  
Campus "José Santilli Sobrinho"**

**VITOR HUGO CAETANO COLONHEZE**

**SISTEMA GERENCIADOR DE ESTOQUE COM DEMANDA A  
LOCALIZAÇÃO**

**Assis/SP  
2018**



**Fundação Educacional do Município de Assis  
Instituto Municipal de Ensino Superior de Assis  
Campus "José Santilli Sobrinho"**

**VITOR HUGO CAETANO COLONHEZE**

**SISTEMA GERENCIADOR DE ESTOQUE COM DEMANDA A  
LOCALIZAÇÃO**

Trabalho de Conclusão de Curso apresentado ao curso de Análise e Desenvolvimento de Sistemas do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

**Orientando (a): Vitor Hugo Caetano Colonheze  
Orientador (a): Esp. Domingos de Carvalho Villela  
Junior**

**Assis/SP  
2018**

## FICHA CATALOGRÁFICA

C719s COLONHEZE, Vitor Caetano

Sistema gerenciador de estoque com demanda a localização /  
Vitor Hugo Caetano Colonheze. – Assis, 2018.

28p.

Trabalho de conclusão do curso (Análise e Desenvolvimento de  
Sistemas). – Fundação Educacional do Município de Assis-FEMA

Orientador: Esp. Domingos de Carvalho Villela Júnior

1.JAVA 2. Sistema

CDD 005.133

# SISTEMA GERENCIADOR DE ESTOQUE COM DEMANDA A LOCALIZAÇÃO

VITOR HUGO CAETANO COLONHEZE

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

**Orientador:** \_\_\_\_\_  
Esp. Domingos de Carvalho Villela Júnior

**Examinador:** \_\_\_\_\_  
Dr. Almir Rogério Camolese

## RESUMO

A concorrência pelos clientes no comércio em geral está cada vez mais acirrada, onde as empresas que mais obtêm sucesso são as que estão à frente dos seus concorrentes. Assim sendo, adquirir um sistema para ter mais automação no negócio é uma das principais atitudes adotadas pelos empreendedores, pois com um sistema a venda é realizada de forma mais rápida e com menos espera para o cliente.

**Palavras-chave:** Sistema, Localização, Estoque, Java;

## **ABSTRACT**

Competition by customers in commerce in general is increasingly fierce, where the companies that are most successful are those that are ahead of their competitors. Therefore, acquiring a system to have more automation in the business is one of the main attitudes adopted by the entrepreneurs, since with a system the sale is carried out more quickly and with less waiting for the client.

**Keywords:** System, Location, Stock, Java;

## LISTA DE ILUSTRAÇÕES

Figura 1: EAP .....	15
Figura 2: Estimativa de custos.....	16
Figura 3: UC 01: Geral.....	17
Figura 4: UC 02: Fazer Login.....	18
Figura 5: UC 03: Manter Funcionário.....	18
Figura 6: UC 04: Manter Produto.....	19
Figura 7: UC 05: Manter Marca .....	20
Figura 8: UC 06: Manter Venda .....	21
Figura 9: Diagrama de Classe .....	23
Figura 10: Diagrama de Atividade: Manter Produto .....	24
Figura 11: Diagrama de Atividade: Efetuar Venda .....	25
Figura 12: Cronograma.....	26

## Sumário

<b>1. INTRODUÇÃO</b> .....	9
1.1. OBJETIVO.....	9
1.2. PÚBLICO ALVO.....	9
1.3. JUSTIFICATIVA .....	9
<b>2. LEVANTAMENTO E ANÁLISE DOS REQUISITOS</b> .....	11
<b>3. TECNOLOGIAS UTILIZADAS</b> .....	12
3.1. UML.....	12
3.2. JAVA.....	13
3.3. NETBEANS IDE.....	13
3.4. H2 DATABASES.....	14
<b>4. ANÁLISE DO SISTEMA</b> .....	15
4.1. ESTRUTURA ANÁLITICA DO PROJETO.....	15
4.2. RECURSOS UTILIZADOS PARA A IMPLEMENTAÇÃO DO SISTEMA.....	15
4.3. ESTIMATIVA DE CUSTOS.....	16
<b>5. DIAGRAMAS DE CASO DE USO</b> .....	17
5.1. CASO DE USO GERAL.....	17
5.2. CASO DE USO EFETUAR LOGIN.....	17
5.3. CASO DE USO MANTER FUNCIONARIO.....	18
5.4. CASO DE USO MANTER PRODUTO .....	19
5.5. CASO DE USO MANTER MARCA.....	20
5.6. CASO DE USO EFETUAR VENDA.....	21
<b>6. DIAGRAMA DE CLASSE</b> .....	23
<b>7. DIAGRAMA DE ATIVIDADE</b> .....	24
7.1. MANTER PRODUTO.....	24
7.2. EFETUAR VENDA.....	25
<b>8. CRONOGRAMA</b> .....	26
<b>9. CONSIDERAÇÕES FINAIS</b> .....	27



# **1. INTRODUÇÃO**

O dia-dia nas empresas são extremamente cansativos tanto por parte dos funcionários, no atendimento aos clientes, organização do ambiente de trabalho e também na relação entre os funcionários e os responsáveis pela empresa.

A tecnologia chegou para revolucionar o ambiente de trabalho seja ele em qualquer área de atuação, em relação a organização de estoque, foi pensado em um software que será implementado para facilitar a busca de produtos dentro do próprio estabelecimento. Sendo assim agilizando o processo encontrar um produto, agilizando e facilitando o processo de venda, evitando transtornos para os clientes e funcionários.

Com esse sistema a empresa irá obter mais resultados positivos quanto a agilidade para a localização, disponibilidade e venda do produto. Mesmo não tendo um sistema completo, a agilidade é bem visível em comparação as empresas sem nenhum tipo de tecnologia para acesso a informação.

## **1.1. OBJETIVO**

O objetivo é desenvolver um software denominado Sistema Gerenciador de estoque com demanda a localização, para possa facilitar o processo de localização de um determinado produto na empresa e podendo assim vende-lo. O software será desenvolvido utilizando-se tecnologia Java com banco de dados H2 DataBases que será embarcado juntamente com a aplicação, sem a necessidade de instalação separada, além do Unified Modeling Languagem – UML, que será utilizado para modelagem do sistema.

## **1.2. PÚBLICO ALVO**

Este trabalho visa beneficiar os funcionários e os clientes da empresa “Galpão do Desmanche”, auxiliando na agilização do processo de venda, sendo os usuários finais do sistema: Vendedores.

## **1.3. JUSTIFICATIVA**

Em conversa com proprietários/gerentes de empresas, chegou a um consenso em que o funcionário dedicado a trabalhar no estoque ou até mesmo o vendedor serão incapazes de trabalhar e conviver em um ambiente de trabalho desorganizado que não tenha a noção de onde se encontra algo que deseja no estoque.

Por conta disso foi requisitado um sistema que não apenas faça o trabalho de cadastro de

estoque, mais sim a localização de cada produto no ambiente, ou seja facilitando o processo de busca e agilizando a tarefa a ser realizada evitando engasgos na empresa, após outros encontros foi definido que o sistema necessitava de um controle de vendas.

## **2. LEVANTAMENTO E ANÁLISE DOS REQUISITOS**

Fazer Login;

- O Usuário fará o Login no sistema usando seu Login e Senha.

Manter Funcionário;

- O responsável pelo cadastramento dos Produtos e Marcas é o Funcionário.

- Neste cadastro serão armazenadas as informações referentes ao Funcionário como: ID, Nome, Login, Senha, Telefone, E-mail e Permissão.

Manter Produto;

- Neste cadastro serão armazenadas as informações referentes ao Produto como: ID, Descrição, Preço de Compra, Preço de Venda, Quantidade, Estado, Corredor, Prateleira, Divisória, Nome do Funcionário e Marca.

Manter Marca;

- Neste cadastro serão armazenadas as informações referentes as Marcas como: ID, Descrição e Sigla.

Manter Venda;

- Este cadastro será feito automaticamente, serão armazenadas as informações referentes a Venda como: ID, ID do Produto, Descrição do Produto, Quantidade e Nome do Funcionário que está efetuando a venda.

### **3. TECNOLOGIAS UTILIZADAS**

Este capítulo aponta as tecnologias utilizadas no desenvolvimento deste trabalho.

#### **3.1. UML**

A UML permite que você “desenhe” uma “planta” do seu sistema. A comparação ideal é a de um construtor que vai realizar um projeto sem antes ter toda a planta que defina estrutura a ser construída. A experiência do construtor garante, até certo ponto, o sucesso do projeto. Mas, com certeza, uma vez feito o planejamento, o “cálculo estrutural”, o desenho da planta, a garantia de sucesso antes, durante e depois da efetivação da construção é incomparavelmente maior. O mesmo acontece com um projeto de software. A experiência do desenvolvedor ou analista, não pode substituir a necessidade de um projeto que defina uma “planta” da solução como um todo. Esta “planta” garante, em todas as fases do projeto, seja na definição, desenvolvimento, homologação, distribuição, utilização e manutenção do mesmo, uma maior clareza e objetividade para execução de cada ação, e, com certeza, quanto maior a solução, maior a necessidade de um projeto definido adequadamente. Desta forma, a UML é uma linguagem padrão para visualização, especificação, construção e documentação de um aplicativo ou projeto de software, e objetiva aumentar a produtividade, otimizar as etapas que envolvem o desenvolvimento de um sistema, aumentando assim a qualidade do produto a ser implementado. Ela independe da ferramenta em que o aplicativo será desenvolvido. A ideia é prover uma visão lógica de todo o processo de forma a facilitar a implementação física do mesmo. A UML disponibiliza, através de conceitos, objetos, símbolos e diagramas, uma forma simples, mas objetiva e funcional, de documentação e entendimento de um sistema. Você pode utilizar os diagramas e arquivos que compõem um modelo UML para o desenvolvimento, apresentação, treinamento e manutenção durante todo o ciclo de vida da sua aplicação. Ela é mais completa que outras metodologias empregadas para a modelagem de dados pois, tem em seu conjunto todos os recursos necessários para suprir as necessidades de todas as etapas que compõem um projeto, desde a definição, implementação, criação do modelo de banco de dados, distribuição, enfim, proporcionando sem qualquer outra ferramenta ou metodologia adicional, um total controle do projeto. A UML implementa uma modelagem com uma visão orientada a objetos. Através dela podemos definir as classes que compõem a nossa solução, seus atributos, métodos e como elas interagem entre si. Apesar da UML ter como base a orientação a objetos, não significa que a ferramenta e a linguagem utilizada para a implementação do modelo seja também orientada a objetos, embora seja

recomendável. Este artigo não irá explorar os conceitos de orientação a objetos, e sim a implementação de um modelo UML simples, para início da documentação de um sistema, utilizando dois diagramas implementados pela UML que são o “Diagrama de Casos e Uso” e o “Diagrama de Classes”. Os diagramas têm como objetivo representar, através de um conjunto de elementos, como o sistema irá funcionar e como cada peça do sistema irá trabalhar e interagir com as outras. Outra vantagem vem da facilidade de leitura dos diagramas que compõe a UML, além da facilidade de confeccioná-los, pois existem inúmeras ferramentas para modelagem de dados orientados a objetos (ferramentas Case), dentre elas o Rational Rose, o Model Maker, e o Poseidom UML. Além dos diagramas citados a UML disponibiliza outros diagramas, dentre os quais podemos citar o Diagrama de Objetos, Diagrama de Sequência, Diagrama de Colaboração, Diagrama de Estado, Diagrama de Atividade e Diagrama de Componentes.

### 3.2. JAVA

Java: É uma linguagem de programação interpretada orientada a objetos desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems. Diferente das linguagens de programação convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um bytecode que é executado por uma máquina virtual. A linguagem de programação Java é a linguagem convencional da plataforma java mas não é a sua única linguagem. A Sun disponibiliza a maioria das distribuições Java gratuitamente e obtém receita com programas mais especializados como o Java Enterprise system. Em 13 de novembro de 2006, a Sun liberou partes do Java como software livre, sob a licença GNU General Public License. A liberação completa do código fonte sob a GPL ocorreu em maio de 2007. Vou utilizá-lo para a programação do meu projeto, pois me adéquo mais seu uso. Ele é um software de uso livre sem necessidade de compra. Neste trabalho utilizarei a tecnologia do Java Swing para a implementação visual.

### 3.3. NETBEANS IDE

A IDE NetBeans é um ambiente de desenvolvimento multiplataforma, uma ferramenta que auxilia programadores a escrever, compilar, debugar e instalar aplicações, foi arquitetada em forma de uma estrutura reutilizável que visa simplificar o desenvolvimento e aumentar a produtividade pois reúne em uma única aplicação todas estas funcionalidades. Totalmente escrita em Java, mas que pode suportar qualquer outra linguagem de programação ou linguagem que desenvolva com Swing, algumas das linguagens que o NetBeans suporta

são o C, C++, Ruby, PHP, XML e linguagens HTML. O NetBeans fornece uma base sólida para a criação de projetos e módulos, possui um grande conjunto de bibliotecas, módulos e API's (Application Program Interface, um conjunto de rotinas, protocolos e ferramentas para a construção de aplicativos de software) além de uma documentação vasta inclusive em português bem organizada, tais ferramentas auxiliam o desenvolvedor de forma a escrever seu software de maneira mais rápida. A distribuição da ferramenta é realizada sob as condições da SPL (Sun Public License), uma variação da MPL (Mozilla Public License), esta licença tem como objetivo garantir a redistribuição de conhecimento à comunidade de desenvolvedores quando novas funcionalidades forem incorporadas à ferramenta. Atualmente está distribuído em diversos idiomas e isto tem o tornado cada vez mais popular, facilitando o acesso a iniciantes em programação e possibilitado o desenvolvimento de aplicativos multilíngue. Como o NetBeans é escrito em Java, é independente de plataforma, funciona em qualquer sistema operacional que suporte a máquina virtual Java (JVM).

#### 3.4. H2 DATABASES

O desenvolvimento do H2 foi iniciado em maio de 2004, mas foi publicado pela primeira vez em 14 de dezembro de 2005. O autor original do H2 é Thomas Mueller, também é o desenvolvedor original do SQL Hypersonic. Em 2001, ele ingressou na PointBase Inc. onde escreveu o PointBase Micro, um banco de dados Java SQL comercial. Nesse ponto ele interrompeu o SQL Hypersonic. O Grupo HSQLDB foi formado para continuar trabalhando na base de código SQL Hypersonic. O nome H2 significa Hypersonic 2, no entanto H2 não compartilha o código do SQL Hypersonic ou do HSQLDB. O H2 é construído do zero.

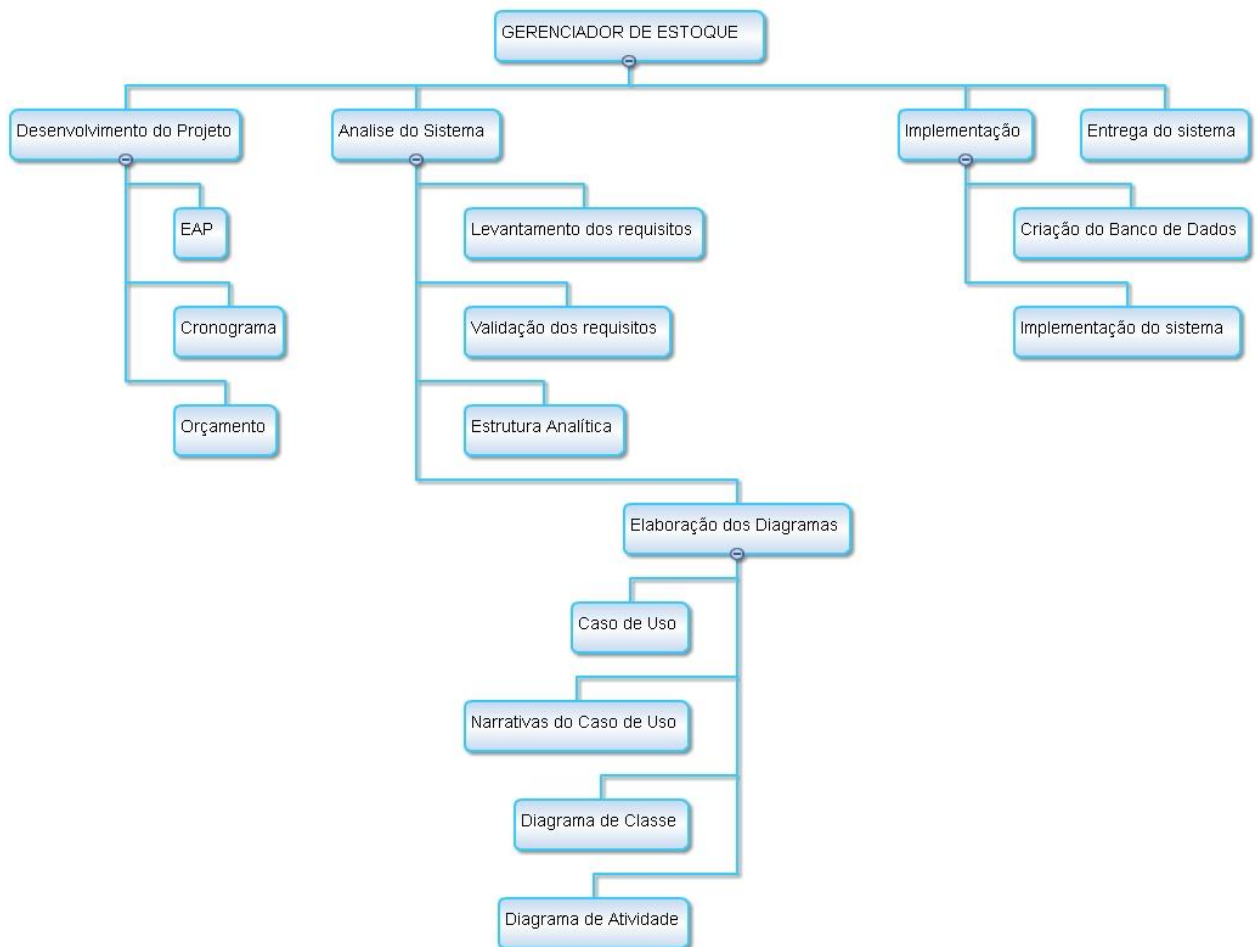
É um banco de dados embarcado, que não precisa ser instalado, pois ele será carregado junto com a aplicação. Este banco de dados é relativamente novo, de uma empresa que dispõe o código livremente. Escrito em Java e muito rápido e pode ser obtido de forma gratuita. Ele é direcionado para Windows, ou seja, pode-se obter mais resultados quando utilizado em um sistema operacional Windows.

## 4. ANÁLISE DO SISTEMA

Neste capítulo se situa os principais diagramas UML utilizados no desenvolvimento desse sistema e também a análise completa do sistema.

### 4.1. ESTRUTURA ANALÍTICA DO PROJETO

Permite que o projeto seja visto de forma mais organizada e simplificada, para facilitar o entendimento do que deverá ser entregue pelo time de desenvolvimento.



www.wbstool.com

Figura 1: EAP

### 4.2. RECURSOS UTILIZADOS PARA A IMPLEMENTAÇÃO DO SISTEMA

01 (um) Notebook;

Sistema Operacional: Windows 7 Professional 64bits;

Banco de Dados: H2 (Embedded);

Software: NetBeans IDE 8.2;

Software: Java 8.0;

### 4.3. ESTIMATIVA DE CUSTOS

Orçamento	
Custo Diário	R\$ 50,00
Total de dias	150
Custo Total	R\$ 7.500,00
Valor Unitário	R\$ 2.000,00
Depreciação (Meses)	24
Custo Mensal	R\$ 83,33
Custo Diário	R\$ 2,78
Custo do Projeto	R\$ 416,67

**Figura 2:** Estimativa de custos



## 5. DIAGRAMAS DE CASO DE USO

Este diagrama UML demonstra de forma simplificada as tarefas que cada usuário do sistema irá executar.

### 5.1. CASO DE USO GERAL

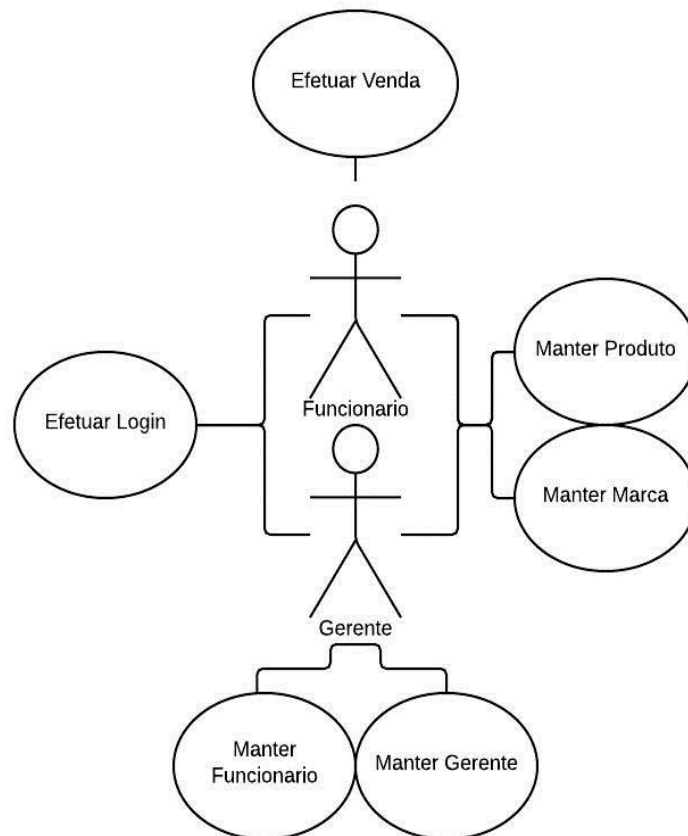
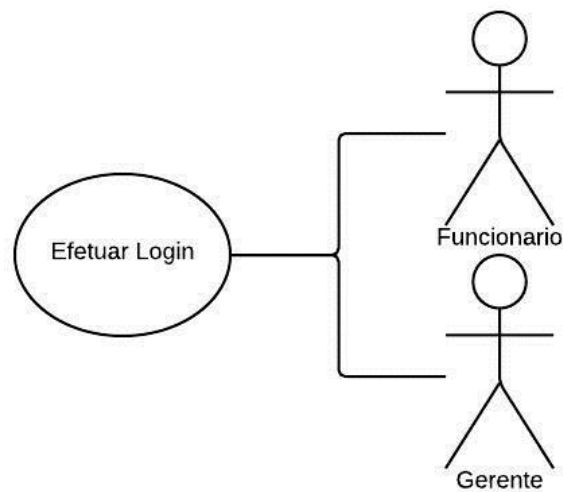


Figura 3: UC 01: Geral

### 5.2. CASO DE USO EFETUAR O LOGIN

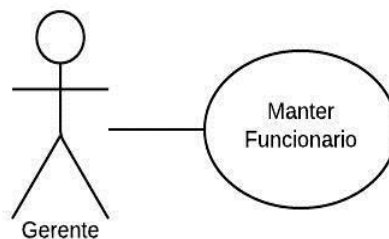


**Figura 4:** UC 02: Fazer Login

Efetuar o Login

- Usuário inicia o caso de uso selecionando o Login;
- sistema oferece as seguintes opções: Login e Senha;
- Usuário informa os dados;
- sistema inicia a aplicação.

### 5.3. CASO DE USO MANTER FUNCIONÁRIO



**Figura 5:** UC 03: Manter Funcionário

Manter Funcionário

- Gerente inicia o caso de uso selecionando a opção desejada [A1, A2, A3];
- sistema dispõe a tela conforme a opção selecionada pelo Gerente;
- Gerente confirma os dados;
- sistema retorna a tela inicial.

Fluxo Alternativo **A1** - Cadastrar

- Sistema solicita os dados;
- Gerente informa os dados;
- Sistema solicita a confirmação;
- Gerente confirma as informações;
- Sistema confirma que o cadastro foi realizado com sucesso e retorna a tela inicial.

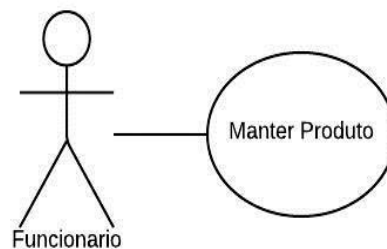
#### Fluxo Alternativo A2 – Editar

- Sistema solicita os dados do Funcionário;
- Gerente informa os dados;
- Sistema busca e retorna os dados;
- Gerente efetua as alterações e confirma as informações;
- Sistema confirma que o cadastro foi realizado com sucesso e retorna a tela inicial.

#### Fluxo Alternativo A3 – Excluir

- Sistema solicita os dados do Funcionário;
- Gerente informa os dados;
- Sistema busca e retorna solicitando a confirmação;
- Gerente confirma as informações;
- Sistema confirma que o cadastro foi realizado com sucesso e retorna a tela inicial.

## 5.4. CASO DE USO MANTER PRODUTO



**Figura 6:** UC 04: Manter Produto

#### Manter Produto

- Funcionário inicia o caso de uso selecionando a opção desejada [A1, A2, A3];
- sistema dispõe a tela conforme a opção selecionada pelo Funcionário;
- Funcionário confirma os dados;
- sistema retorna a tela inicial.

#### Fluxo Alternativo A1 - Cadastrar

- O Sistema solicita os dados;
- O Funcionário informa os dados;
- O Sistema solicita a confirmação;
- O Funcionário confirma as informações;
- O Sistema confirma que o cadastro foi realizado com sucesso e retorna a tela inicial.

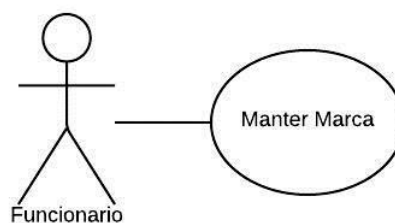
#### Fluxo Alternativo A2 – Editar

- O Sistema solicita os dados do Produto;
- O Funcionário informa os dados;
- O Sistema busca e retorna os dados;
- O Funcionário efetua as alterações e confirma as informações;
- O Sistema confirma que o cadastro foi realizado com sucesso e retorna a tela inicial.

#### Fluxo Alternativo A3 – Excluir

- O Sistema solicita os dados do Produto;
- O Funcionário informa os dados;
- O Sistema busca e retorna solicitando a confirmação;
- O Funcionário confirma as informações;
- O Sistema confirma que o cadastro foi realizado com sucesso e retorna a tela inicial.

## 5.5. CASO DE USO MANTER MARCA



**Figura 7:** UC 05: Manter Marca

#### Manter Marca

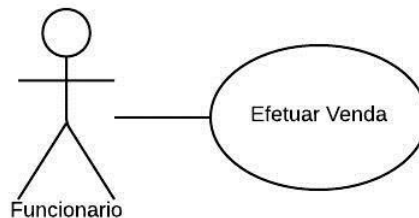
- O Funcionário inicia o caso de uso selecionando a opção desejada [A1];
- O sistema dispõe a tela conforme a opção selecionada pelo Funcionário;

- O Funcionário confirma os dados;
- O sistema retorna a tela inicial.

#### Fluxo Alternativo A1 - Cadastrar

- O Sistema solicita os dados;
- O Funcionário informa os dados;
- O Sistema solicita a confirmação;
- O Funcionário confirma as informações;
- O Sistema confirma que o cadastro foi realizado com sucesso e retorna a tela inicial.

## 5.6. CASO DE USO EFETUAR VENDA



**Figura 8:** UC 06: Efetuar Venda

#### Efetuar Venda

- O Funcionário inicia o caso de uso selecionando o produto desejado e selecionando a quantidade de produtos que serão vendidos;
- O sistema dispõe a tela conforme a opção selecionada pelo Funcionário;
- O Funcionário confirma os dados;
- O sistema retorna a tela de Visualizar Produtos.

#### Fluxo Alternativo A1 - Cadastrar

- O Sistema solicita a quantidade de Produtos que serão vendidos;
- O Funcionário informa os dados;
- O Sistema busca os dados do funcionário;
- O Sistema confere a validade da quantidade de produtos selecionados em relação a quantidade disponível de produtos;
- O Sistema solicita a confirmação;
- O Funcionário confirma as informações;

- O Sistema confirma que a venda foi realizado com sucesso
- O Sistema retorna a tela Visualizar Produtos.

## 6. DIAGRAMA DE CLASSE

O diagrama representa a estrutura do sistema, recorrendo ao conceito de classes. O modelo de classes resulta de um processo de abstração onde são identificados os objetos relevantes do sistema. Um objeto é uma instância da classe.

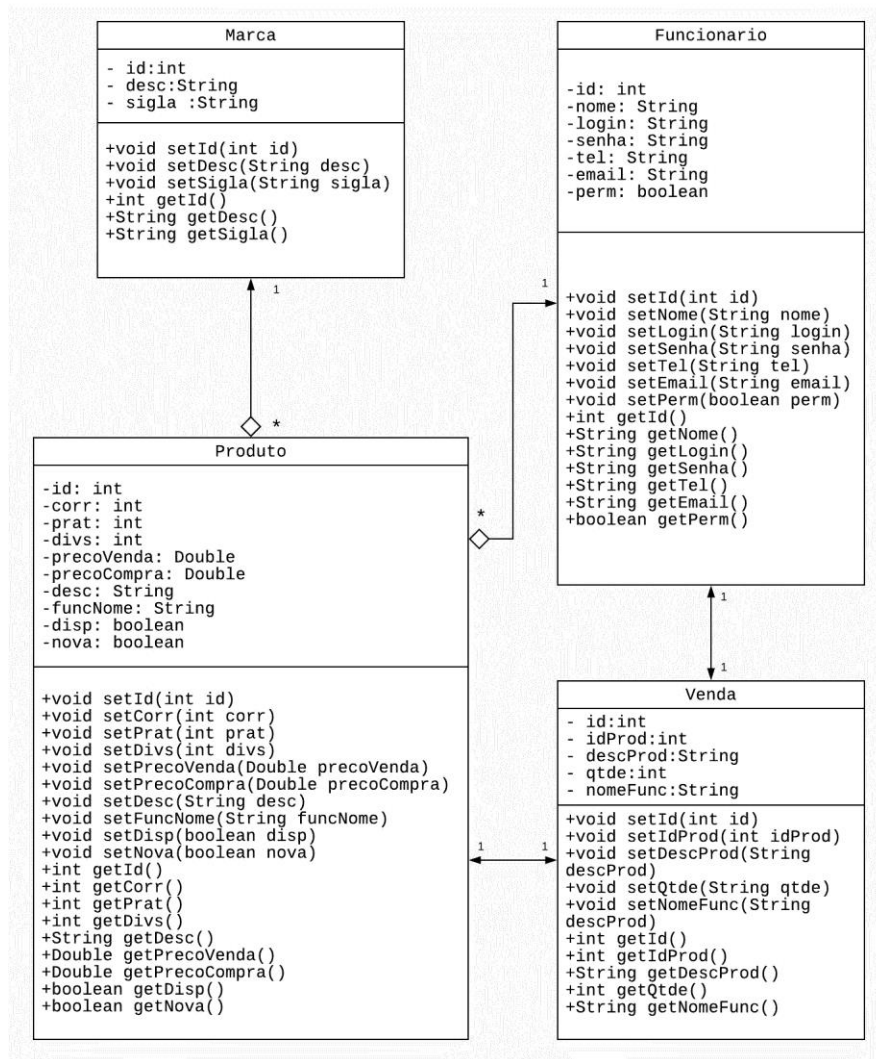


Figura 9: Diagrama de Classe

## 7. DIAGRAMA DE ATIVIDADE

Um diagrama de atividade é essencialmente um gráfico de fluxo, mostrando o fluxo de controle de uma atividade para outra e serão empregados para fazer a modelagem de aspectos dinâmicos do sistema.

### 7.1. MANTER PRODUTO

A imagem abaixo mostra o Diagrama de Atividade Manter Produto, mostrando sua funcionalidade;

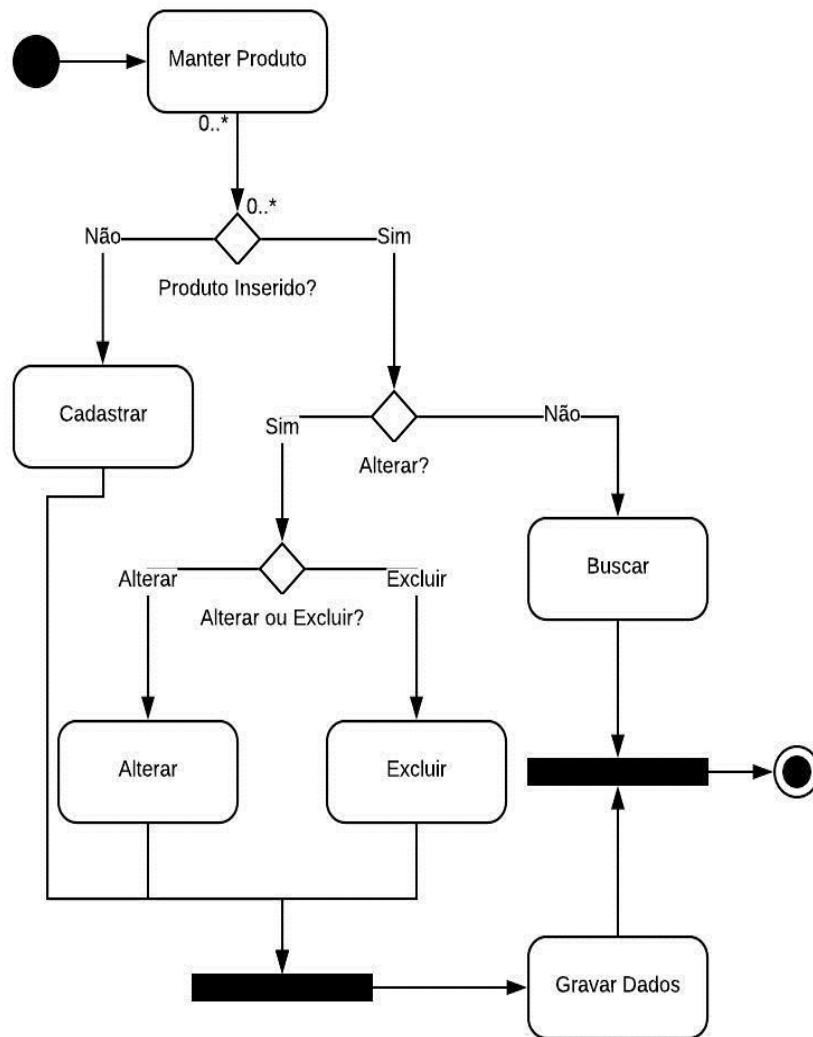
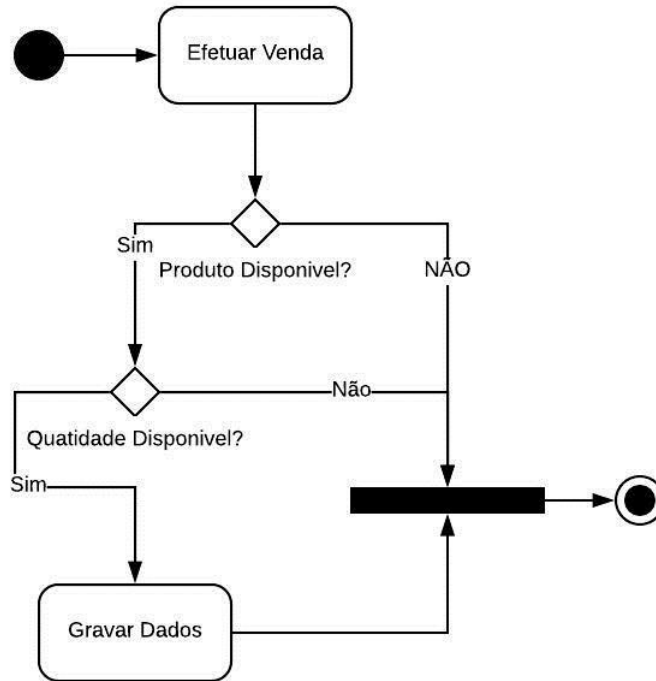


Figura 10: Diagrama de Atividade: Manter Produto



## 7.2. EFETUAR VENDA

A imagem abaixo mostra o Diagrama de Atividade Efetuar Venda, mostrando sua funcionalidade;



**Figura 11:** Diagrama de Atividade: Efetuar Venda

## 8. CRONOGRAMA

Atividade/Período	Nov.	Dez.	Jan.	Fev.	Mar.	Abr.	Mai.	Jun.	Jul.	Ago.
Desenvolvimento do projeto										
Levantamento dos requisitos										
Análise dos requisitos										
Elaboração dos diagramas										
Implementação do banco de dados										
Implementação do Sistema										
Implementação Gráfica										
Fase de teste										
Reparos de falhas										
Fase de teste										
Reparos de falhas										
Entrega do Sistema										

Figura 12: Cronograma

## **9. CONSIDERAÇÕES FINAIS**

Este trabalho tem como objetivo desenvolver uma aplicação para ajudar no atendimento e organização de um galpão de desmanche de veículos, visando a gestão e controle do estoque de produtos, para auxiliar a rotina de trabalho.

Após algumas visitas no local da implantação do sistema, recolhi relatos dos funcionários, dos responsáveis pela empresa e de alguns clientes, que dizem ter havido melhorias no atendimento, na agilidade da busca por produtos e na organização do estoque.

## REFERÊNCIAS BIBLIOGRÁFICAS

**“O que é a Tecnologia Java e porque preciso dela?”**. Disponível em:

<[https://www.java.com/pt\\_BR/download/faq/whatis\\_java.xml](https://www.java.com/pt_BR/download/faq/whatis_java.xml)>. Acesso em 02 de dezembro de 2016.

**DUARTE, Diego**. Disponível em: < <http://www.purainfo.com.br/artigos/uml-diagrama-de-atividades>>. Acesso em 01 de janeiro de 2017.

**FILHO, WILSON DE PÁDUA PAULA**. Engenharia de software: fundamentos, métodos e padrões; Editora LTC; 2001.

**GRADY BOOCH, J. R.& IVAR J.** UML – Guia do Usuário, Editora: Campus, 2000.

Microsoft SQL Server Management Studio 2008.

**GUEDES, GILLEANES T. A.** UML 2 – UMA ABORDAGEM PRÁTICA, Editora: NOVATEC, 2008.

**H2 Database Engine**. Disponível em: <<http://www.h2database.com/h2.pdf>>. Acesso em 16 de novembro de 2016.

**OFICINA, Redação**. Disponível em:

<[http://www.oficonadanet.com.br/artigo/1061/o\\_que\\_e\\_o\\_netbeans.amp](http://www.oficonadanet.com.br/artigo/1061/o_que_e_o_netbeans.amp)>. Acesso em 02 de março de 2018.

**RODRIGUES, Joel**. Disponível em: <<http://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332>>. Acesso em 02 de dezembro de 2016.

**SIERRA, KATHY**. USE A CABEÇA JAVA, Editora: ALTA BOOKS, 2005.

**VIEGAS, Gustavo**. Disponível em: < <https://www.devmedia.com.br/introducao-a-uml/6928>>. Acesso em 05 de janeiro de 2018.