

**CESAR HENRIQUE DE OLIVEIRA**

**SISTEMA DE GESTÃO PARA AÇOUGUE**

Assis

2016

## **SISTEMA DE GESTÃO PARA AÇOUGUE**

Trabalho de conclusão de curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação.

**Orientador:** ESP. Domingos de Carvalho Villela Júnior

**Área de Concentração:** Desenvolvimento de Sistemas

Assis

2016

Avenida Getúlio Vargas, 1200 – Vila Nova Santana – Assis – SP - 19807-634

Fone/Fax: (0XX18) 3302 1055 – homepage: [www.fema.edu.br](http://www.fema.edu.br)

## FICHA CATALOGRÁFICA

OLIVEIRA, César Henrique

Sistema de Gestão para Açougue / César Henrique de Oliveira.

Fundação Educacional do Município de Assis, 2016.

X p.

Orientador: ESP. Domingos de Carvalho Villela Júnior

Trabalho de Conclusão de Curso

Instituto Municipal de Ensino de Assis – IMESA.

1. Diagnostico geral, Linguagem de Programação Java,  
3.UML.

CDD: 001.61

Biblioteca da FEMA

## **SISTEMA DE GESTÃO PARA AÇOUGUE**

Trabalho de conclusão de curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora.

**Orientador:** ESP. Domingos de Carvalho Villela Júnior

**Analisador:** Dr<sup>o</sup> Almir Rogério Camolesi

Assis

2016

Avenida Getúlio Vargas, 1200 – Vila Nova Santana – Assis – SP - 19807-634

Fone/Fax: (0XX18) 3302 1055 – homepage: [www.fema.edu.br](http://www.fema.edu.br)

## **DEDICATÓRIA**

Dedico este trabalho á minha família, e a todas as pessoas que acreditaram em min e no meu trabalho apoiando-me para que eu pudesse realizá-lo.

## **AGRADECIMENTOS**

Agradeço primeiramente a deus por me dar força, sabedoria e conhecimento para completar esse projeto.

Agradeço também a minha família, principalmente a minha mãe, Maria Suely Fernandes Oliveira e ao meu pai, Eliel Vieira de Oliveira por me apoiar ao longo de toda a graduação.

E por fim ao meu orientador ESP. Domingos de Carvalho Villela Júnior por me aconselhar e tirar as dúvidas durante todo o projeto.

“A força de vontade deve ser mais forte do que a sua habilidade.”

Muhammad Ali

## RESUMO

Neste trabalho será mostrada a Análise e Desenvolvimento de um sistema para gerenciar um açougue.

O sistema tem por objetivo cadastrar todos os clientes e fornecedores do estabelecimento, também administrar as contas do local.

Com a implementação do sistema o usuário vai ter sua vida facilitada com vários relatórios e todas as informações fáceis de se manipular, o sistema vai participar de todo o processo do açougue dá compra do produto até a venda para o cliente.

Para o desenvolvimento do projeto foram utilizadas as seguintes tecnologias, Linguagem de programação *JAVA*, Banco de Dados *MySQL*, *framework* de persistência *Hibernate*, servidor de aplicação *WildFly*, e o ambiente de desenvolvimento foi o *Eclipse*.

## **ABSTRACT**

This work will show the analysis and development of a system to manage a butcher.

The system aims to register all customers and suppliers of the establishment, also manage local accounts.

With the implementation of the user's system will have your life easier with various reports and all information easy to manipulate, the system will participate in the entire process from the butcher gives product purchase until the sale to the customer.

For the development of the project were used the following technologies, JAVA programming language, database MySql, persistence framework Hibernate, WildFly application server and development environment is Eclipse.

## Lista de Figuras

Figura 1: Visão Geral.....	25
Figura 2: Manter Cliente.....	26
Figura 3: Manter Fornecedor.....	27
Figura 4: Manter Produtos.....	28
Figura 5: Manter Estoque.....	29
Figura 6: Gerenciar Usuários.....	30
Figura 7: Emitir Relatório de Vendas.....	31
Figura 8: Visão do Cadastro de Produto.....	32
Figura 9: Cadastro de Produto.....	33
Figura 10: Remover Produto.....	34
Figura 11: Repor Estoque.....	35
Figura 12: Alterar Informações dos Produtos.....	36
Figura 13: Diagrama de Classe.....	37
Figura 14: Diagrama de Entidade e Relacionamento.....	38
Figura 15: Orçamento - Analista/Programador.....	39
Figura 16: Orçamento - Equipamentos.....	39
Figura 17: Orçamento - Total.....	39
Figura 18: Organização dos Pacotes.....	41
Figura 19: Repositórios Maven 1.....	42
Figura 20: Repositórios Maven 2.....	43
Figura 21: Idiomas - Português.....	44
Figura 22: Idioma - Inglês.....	45
Figura 23: PrimeFaces - Tema.....	46
Figura 24: Telas - Login.....	47
Figura 25: Telas - Cadastro de Usuários.....	48

## Lista de Tabelas

Tabela 1: Manter Cliente.....	26
Tabela 2: Manter Fornecedor.....	27
Tabela 3: Manter Produtos.....	28
Tabela 4: Manter Estoque.....	29
Tabela 5: Gerenciar Usuários.....	30
Tabela 6: Emitir Relatório de Vendas.....	31
Tabela 7: Cadastrar o Produto.....	33
Tabela 8: Remover o Produto.....	34
Tabela 9: Repor Estoque.....	35
Tabela 10: Alterar Informações do Produto.....	36

## Sumário

1. INTRODUÇÃO.....	14
1.1 OBJETIVO.....	14
1.2 PÚBLICO ALVO.....	15
1.3 JUSTIFICAVA.....	15
1.4 ESTRUTURA DO TRABALHO.....	16
2. TECNOLOGIAS E FERRAMENTAS PARA O DESENVOLVIMENTO DO SISTEMA	17
2.1 LINGUAGEM UNIFICADA DE MODELAGEM (UML).....	17
2.1.1 Diagrama de casos de uso (UC).....	18
2.1.2 Diagrama de atividades.....	18
2.1.3 Diagrama de sequência.....	18
2.1.4 Diagrama de classe.....	18
2.1.5 Diagrama de entidade e relacionamento(DER).....	19
2.2 JAVA.....	19
2.3 JAVA EE.....	20
2.4 ECLIPSE.....	20
2.5 JBOSS WildFly.....	20
2.6 COMPUTAÇÃO EM NUVEM.....	21
2.7 MYSQL.....	21
2.8 HIBERNATE.....	22
2.9 MAVEN.....	22
2.10 PRIMEFACES.....	23
2.11 SPRING SECURITY.....	23
3. ESTRUTURA DE DESENVOLVIMENTO DO SISTEMA.....	24
3.1 ANÁLISE, ESPECIFICAÇÃO DO PROJETO.....	25
3.2 Diagrama de Casos de Uso (UC).....	25
3.3 Diagrama de Classe.....	37
3.4 Diagrama de Entidade e Relacionamento.....	38
3.5 Orçamento.....	39
3.6. CRONOGRAMA DA ESTRUTURA DE DESENVOLVIMENTO.....	40

4. DESENVOLVIMENTO DO SISTEMA.....	41
4.1 Organização do Projeto.....	41
4.2 Repositórios Maven.....	42
4.3 Opções de Idiomas.....	44
4.4 Temas - PrimeFaces.....	46
4.5 Telas Do Sistema.....	47
5. Conclusão.....	49
REFERÊNCIAS BIBLIOGRÁFICAS.....	50

# 1. INTRODUÇÃO

O açougue para qual o sistema foi desenvolvido fica no centro da cidade de Assis e já existe a mais de 30 anos, permanece aberto graças a sua freguesia fiel.

No local nunca teve nenhum tipo de sistema computacional, portanto o software é o mais simples e fácil de se usar possível.

O sistema de gestão para açougue foi desenvolvido para sanar e remover quaisquer problemas que existam por conta da falta de automatização. O sistema auxiliara em todas as etapas do trabalho como: compra, venda, validade vencimento de contas e de produtos. Como o sistema ainda não foi implementado, o açougue faz todo o seu processo manualmente e isso prejudica diretamente o serviço e o controle do negócio. Quando há muitos clientes o atendimento fica lento e exaustivo por conta de ter que anotar tudo manualmente, mesmo dos clientes antigos. Também prejudica no controle de produtos por não saber exatamente o que vendeu. As vezes ocorre uma compra de um produto que ainda tem em quantidade razoável, ou o contrário, deixar de comprar sem saber que o produto está em falta. Há muitos problemas na questão da validade pela falta de um sistema que controle tudo isso com facilidade.

## 1.1 OBJETIVO

O sistema vai ajudar a empresa em todas as suas funções, com um controle de estoque mais simples e com inúmeras formas de filtrar o seu produto com um relatório completo e com fácil visualização e entendimento, um modulo de contas a pagar e receber auxiliando nos pagamentos de todas as taxas do local, também deixando o atendimento mais rápido com todos os clientes salvos no banco de dados.

O objetivo da informatização no ramo de açougues é o mesmo de todos os outros locais que atendem o publico com tanta frequência, é a facilidade e automatização que ele traz.

## 1.2 PÚBLICO ALVO

O público alvo do sistema é todo e qualquer açougue que esteja com a necessidade de automatizar o seu atendimento e controle do seu negócio assim facilitando o trabalho.

## 1.3 JUSTIFICAVA

O sistema foi elaborado para auxiliar os funcionários do açougue que explicaram sobre a lentidão no atendimento por conta de anotar tudo em papel e não tem um sistema para gerenciar as mercadorias com facilidade. O objetivo do sistema é que todos os problemas sejam sanados e o trabalho de todos facilitado.

## 1.4 ESTRUTURA DO TRABALHO

O trabalho será dividido em 5 capítulos, sendo eles:

- **Capítulo 1 - Introdução:** O primeiro capítulo apresenta a contextualização e a justificativa para o desenvolvimento da proposta do trabalho.
- **Capítulo 2 - Tecnologia e Ferramentas para o desenvolvimento do sistema:** Neste capítulo foi apresentado as tecnologias e ferramentas utilizadas.
- **Capítulo 3 - Análise, especificações e projeto:** Aqui foi apresentado as ferramentas para modelagem e análise do sistema.
- **Capítulo 4 - Desenvolvimento do sistema:** Neste capítulo foi mostrado as camadas do sistema e como foi desenvolvido, os pacotes e a interface final.
- **Capítulo 4 - Conclusão e Trabalhos futuros:** Este capítulo apresenta a conclusão e trabalhos futuros.
- **Referencias Bibliográficas**

## 2. TECNOLOGIAS E FERRAMENTAS PARA O DESENVOLVIMENTO DO SISTEMA

Para a Análise orientada a objetos, foi utilizado a Linguagem de Modelagem Unificada (UML) e ferramenta de modelagem utilizada foi o *Astah Community*.

O projeto utilizou como base na tecnologia *Java*, pois é uma linguagem de alto nível, o ambiente de desenvolvimento foi o *Eclipse* e com o banco de dados *MySQL*.

### 2.1 LINGUAGEM UNIFICADA DE MODELAGEM (UML)

A Linguagem Unificada de Modelagem (UML) é desenvolvida para descrever graficamente “*software*”. A UML ajuda no desenvolvimento do sistema do começo ao fim ela dá uma visão geral do “*software*” antes mesmo de começar a programar, ela é essencial também na documentação dos trabalhos.

Trabalhar com diagramas nos ajuda ter uma visão clara das funcionalidades do sistema. Ajuda a apontar inconsistências e omissões. Por fim, possibilita documentar os artefatos do sistema. (GOÉS, 2014, P.16-17).

### 2.1.1 Diagrama de casos de uso (UC)

A principal função do diagrama de casos de uso (UC) é apresentar quais são os módulos que compõe o sistema, quais são os usuários e quais funções cada um vai desempenhar para o funcionamento do sistema. (GÓES, 2014).

### 2.1.2 Diagrama de atividades

O diagrama de atividades é essencialmente um gráfico de fluxo, ele mostra o fluxo de controle de uma atividade, de forma que possa ser ilustrada uma modelagem dos aspectos dinâmicos (GÓES, 2014).

### 2.1.3 Diagrama de sequência

O diagrama de sequência descreve a sequência de comunicação entre os objetos de um sistema de informação. Seus principais objetivos são: documentar casos de uso, mostrar como os objetos do sistema se comunicam validar se todas as operações das classes foram identificadas. (GOÉS, 2014).

### 2.1.4 Diagrama de classe

O diagrama de classe ilustra um conjunto de classes e seus relacionamentos de forma estática de um sistema, eles são importantes não só para documentação, mas também para o desenvolvimento do sistema contribuindo de forma relevante para a consolidação do escopo das funcionalidades do mesmo. (GÓES, 2014).

### 2.1.5 Diagrama de entidade e relacionamento(DER)

O Diagrama Entidade e Relacionamento - DER é o diagrama de documentação do banco de dados relacionais, sendo responsável por mostrar de maneira gráfica os relacionamentos entre as entidades do banco de dados. (HEUSER, 2009).

## 2.2 JAVA

A história do java começou em 1991 em um projeto denominado Projeto *Green* que era para criação de novas tecnologias. A *Sun Microsystems* financiou um projeto de pesquisa corporativa interna (**Projeto Green**) em 1991. Esse projeto resultou em uma linguagem de programação baseada na linguagem C++. O seu criador James Gosling, chefe do projeto, chamou essa linguagem de Oak em homenagem a uma árvore de carvalho vista por sua janela na empresa que trabalhava a *Sun*, porém como alguém já tinha escolhido este nome ele mudou para Java pois era o nome da terra de origem do café (DEITEL, 2010, p. 6).

O *Java* é uma tecnologia usada para desenvolver aplicações que tornam a *Web* mais divertida e útil.

## 2.3 JAVA EE

A *Java EE (Java Platform, Enterprise Edition)* é uma plataforma padrão para Desenvolver aplicações *Java* de grande porte e/ou para a internet.

É funcionalidades para implementar *software Java* distribuído, baseado em componentes modulares que executam em servidores de aplicações e que suportam escalabilidade, segurança, integridade e outros requisitos de aplicações corporativas ou de grande porte (THIAGO F, 2013, P.10).

## 2.4 ECLIPSE

O *Eclipse* é uma IDE(*integrated development environment*). O mesmo é líder do mercado e a sua última versão é a 4.6 chamada *Eclipse Neon*.

O *Eclipse* IDE é uma ferramenta *open source*(código aberto), que inicialmente foi liderada pela IBM (*International Business Machines*). A IBM após terminar a primeira versão da IDE doou-o para a comunidade de *software* Livre.

“A empresa *International Business Machines Corp (IBM)* lançou, com apoio da *Eclipse org.*”(Eclipse Platform Technical Overview, 2006).

## 2.5 JBOSS WildFly

O WildFly ou JBOSS *AS* é um servidor de aplicação *open source*(código aberto), Inicialmente desenvolvido pela JBOSS e agora pela *Red Hat*.

O grande diferencial do WildFly para os seus concorrentes *Apache TomCat* e *GlassFish* é a sua velocidade na inicialização, foram feitos alguns testes e ele chegou a iniciar com 2 segs.

“O grupo *4LINUX* publicou.”(O QUE É *WILDFLY*, 2014).

## 2.6 COMPUTAÇÃO EM NUVEM

A computação em nuvem utiliza de recursos de servidores pela internet, assim você não precisa ter um computador de grande porte para ter um sistema em execução. Não vai precisar ficar se preocupando em trocar de computador por problemas no *software*, e se por uma fatalidade o seu computador estragar ou o seu HD (Disco Rígido) corromper ou algo parecido, você não terá nem um problema pois está tudo na nuvem.

“A editora livro aberto lançou, com apoio da *Federal Institute of Education*.”(COMPUTAÇÃO EM NUVEM, 2010).

## 2.7 MYSQL

O MySQL é um dos bancos de dados com o código aberto mais popular atualmente, ele é um sistema de gerenciamento de banco de dados (SGBD), e utiliza como linguagem o SQL.

O projeto do mysql foi iniciado em 1994 e seus desenvolvedores foram o Michael Widenius e Axmark, a sua primeira versão interna foi lançada em 25 de maio de 1995 e em 8 de janeiro de 1998 foi lançada sua primeira versão para *windows 95* e NT.

(PEDRO M, RUI P F RUAS, 2005,p.21).

## 2.8 HIBERNATE

O *Hibernate* é um *framework* escrito em java, porém também disponível em .NET chamado de *Nhibernete* a função do mesmo é, transformar as classes em *java* em tabelas portanto ele escreve toda a SQL para o desenvolvedor.

O *hibernate* foi criado por vários desenvolvedores através do mundo, mais foi liderado pelo Gavin King, a versão atual do mesmo é 5.2 e é publicado pela *redhat*.

“O grupo Comunidade Livre Paraíba lançou, com apoio do G.U.J.(Grupo de Usuários Java).” (Introdução ao Hibernate 3, p.1).

## 2.9 MAVEN

O *Maven* é uma ferramenta de automação de compilação, ele utiliza como padrão um arquivo XML chamado de “**pom**”, neste arquivo os desenvolvedores colocam os chamados “**repositórios**” que a partir desses repositórios o maven consegue adicionar qualquer *framework*, só basta colocar o repositório do mesmo.

Vou colocar aqui como exemplo o repositório do *PrimeFaces*:

### <dependency>

1. <groupId>org.primefaces</groupId>
2. <artifactId>primefaces</artifactId>
3. <version>6.0</version>
- 4.</dependency>

Depois de Adicionar esse Repositório e Atualizar o *Maven* com a opção *Maven Update*, o seu projeto está apto para ter os temas do *PrimeFaces*, sem precisar baixar nada do site e adicionar no projeto.

(WALTER S FILHO, 2008, p.6).

## 2.10 PRIMEFACES

O *PrimeFaces* é uma biblioteca para *Java* com mais de 200 componentes que lhe permite criar várias interfaces completas no seu sistema web, o *PrimeFaces* é uma ferramenta da *Prime Teknoloji* (empresa da Turquia) e ele oferece componentes ricos para o *JavaServer Faces*.

O *PrimeFaces* tem uma documentação muito completa no seu próprio site na parte *DEMOS* e depois em *SHOWCASE*, ai você vai encontrar vários exemplos.

(THIAGO FARIA, 2013, p.173).

## 2.11 SPRING SECURITY

O Spring Security começou com uma alternativa em segurança para a tradicional especificação do Java EE, e hoje é o *framework* mais utilizado na área de segurança.

A ferramenta é tão utilizada pelo fato dela ser útil em qualquer aplicação *web* que necessite restringir os seus recursos, afinal segurança é um requisito importante nos dos sistemas.

O *framework* foi utilizado no sistema para gerenciar a parte do login e também o controle dos usuários.

“A editora *Java Magazine* edição 69 lançou”. (*Spring Security*, p.1).

### **3. ESTRUTURA DE DESENVOLVIMENTO DO SISTEMA**

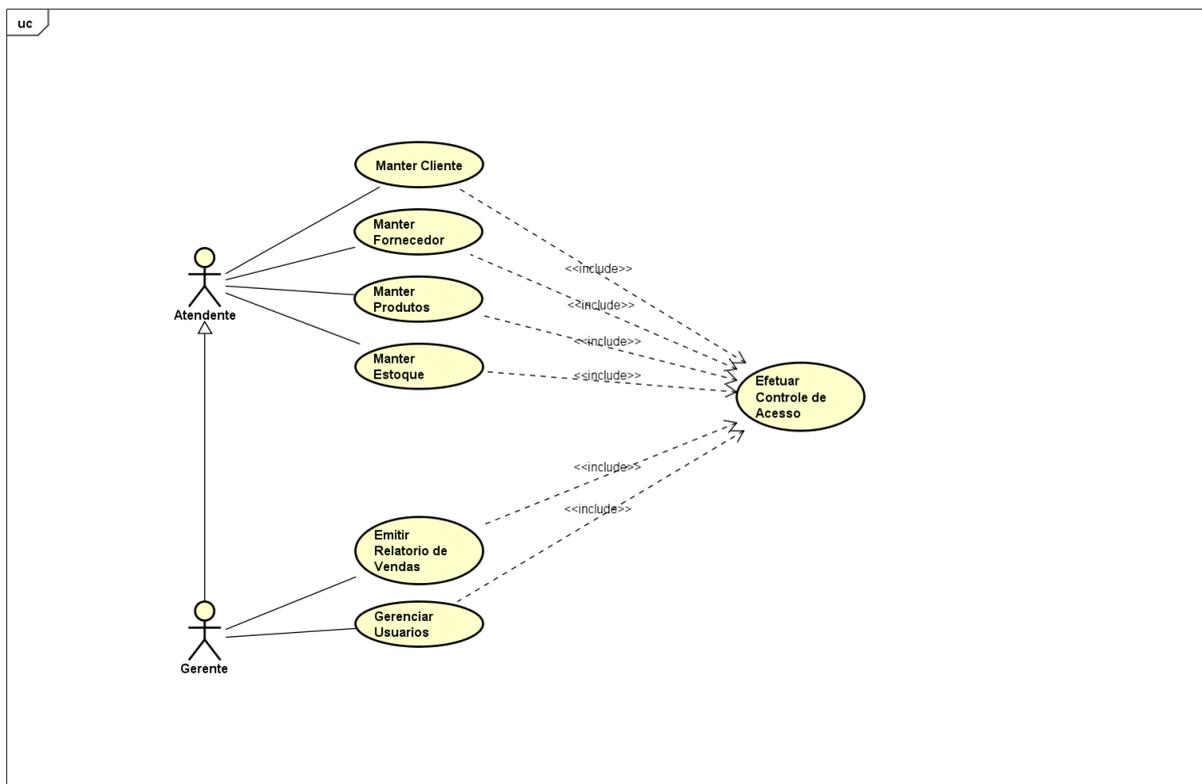
O método pode ser definido como um caminho para se atingir um objetivo.

- Levantamento de necessidades
- Levantamento dos requisitos
- Análise do requisito
- Gerenciamento dos requisitos
- Diagrama de Casos de Uso (Use Case)
- Diagrama de Atividade
- Diagrama de Sequência
- Diagrama de Classe

### 3.1 ANÁLISE, ESPECIFICAÇÃO DO PROJETO

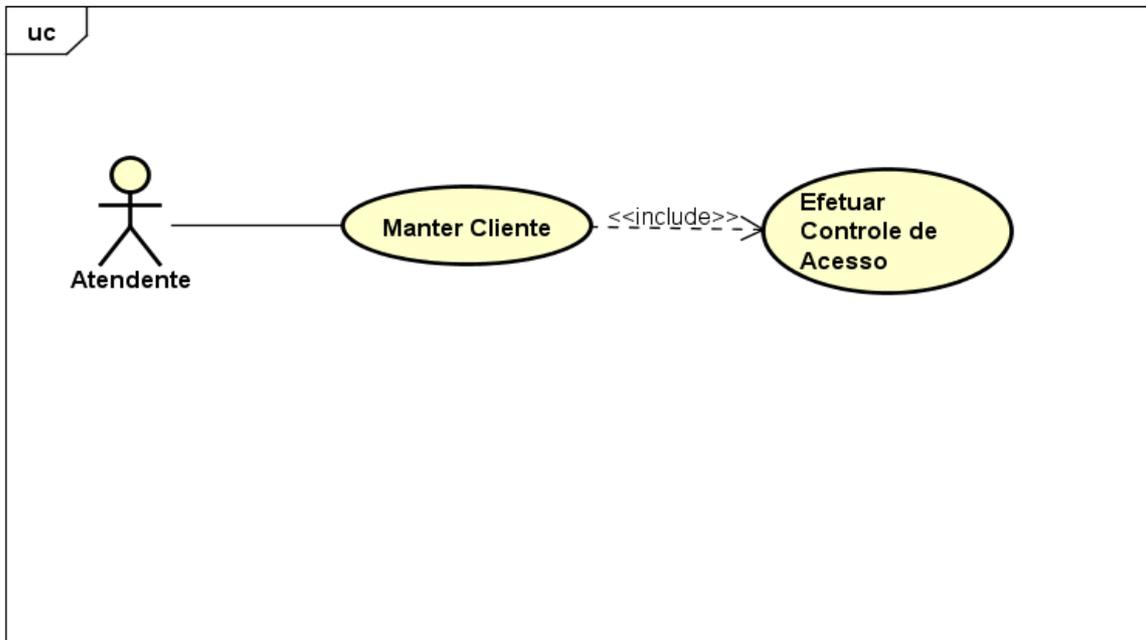
Este capítulo mostra os casos de uso e os principais diagramas da UML.

### 3.2 Diagrama de Casos de Uso (UC)



powered by Astah

Figura 1: Visão Geral

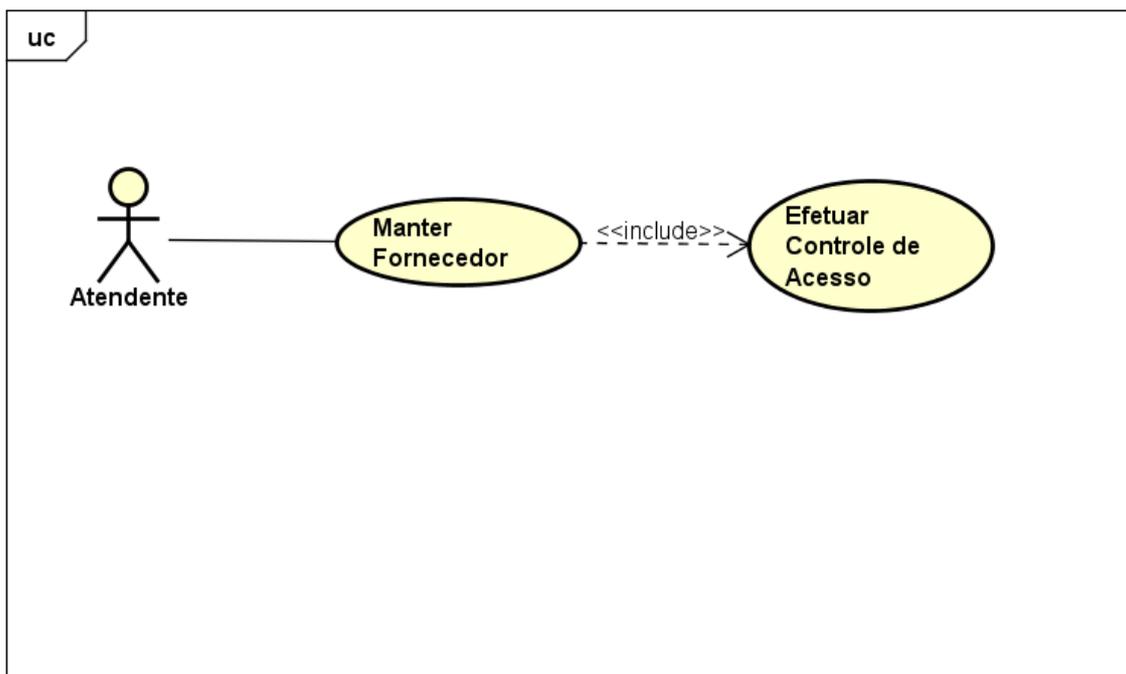


powered by Astah

Figura 2: Manter Cliente

Tabela 1: Manter Cliente

<b>Nome do Caso de Uso</b>	Manter Cliente
<b>Atores</b>	Atendente
<b>Pré-Condições</b>	Efetuar Controle de Acesso
<b>Cenário Principal</b>	1-Atendente informa os dados e em seguida seleciona confirmar. 2-O sistema carrega os dados 3-O atendente finaliza a operação
<b>Cenário Alternativo</b>	Atendente pode cancelar a operação a qualquer momento
<b>Casos de Teste</b>	1-Os campos obrigatórios estando com informações o sistema executa a operação 2-Caso contrário o sistema retorna uma mensagem falando os campos que faltam

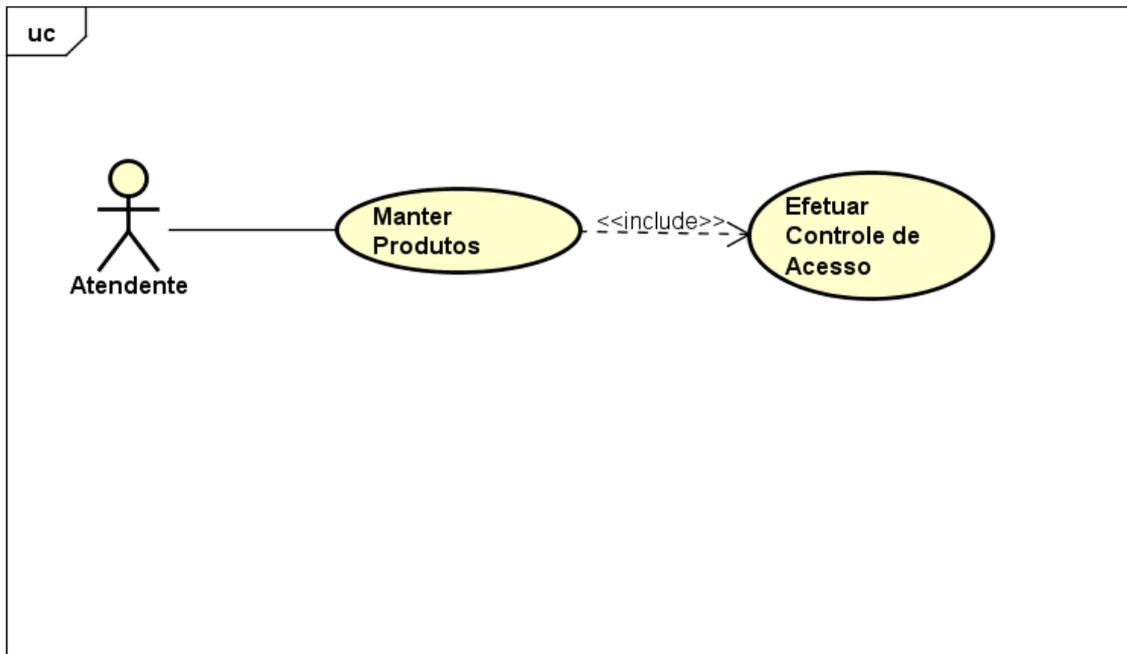


powered by Astah

Figura 3: Manter Fornecedor

Tabela 2: Manter Fornecedor

<b>Nome do Caso de Uso</b>	Manter Fornecedor
<b>Atores</b>	Atendente
<b>Pré-Condições</b>	Efetuar Controle de Acesso
<b>Cenário Principal</b>	1-Atendente informa os dados e em seguida seleciona confirmar. 2-O sistema carrega os dados 3-O atendente finaliza a operação
<b>Cenário Alternativo</b>	Atendente pode cancelar a operação a qualquer momento
<b>Casos de Teste</b>	1-Os campos obrigatórios estando com informações o sistema executa a operação 2-Caso contrário o sistema retorna uma mensagem falando os campos que faltam

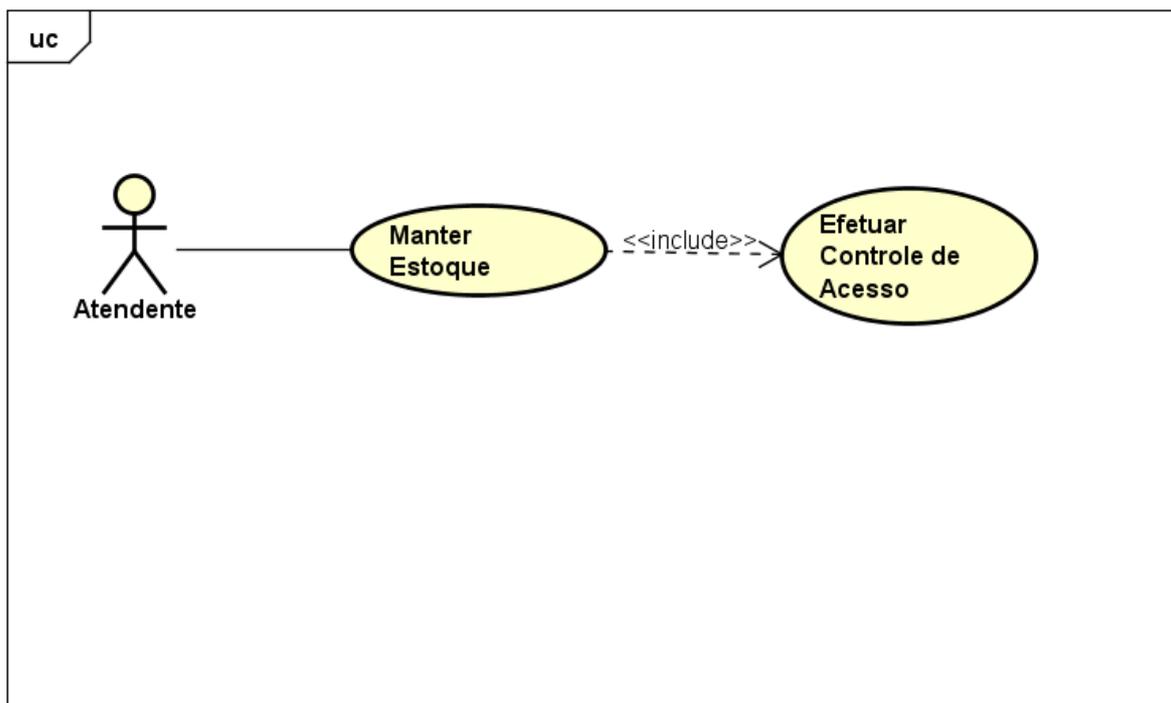


powered by Astah

Figura 4: Manter Produtos

Tabela 3: Manter Produtos

<b>Nome do Caso de Uso</b>	Manter Produtos
<b>Atores</b>	Atendente
<b>Pré-Condições</b>	Efetuar Controle de Acesso
<b>Cenário Principal</b>	1-Atendente informa os dados e em seguida seleciona confirmar. 2-O sistema carrega os dados 3-O atendente finaliza a operação
<b>Cenário Alternativo</b>	Atendente pode cancelar a operação a qualquer momento
<b>Casos de Teste</b>	1-Os campos obrigatórios estando com informações o sistema executa a operação 2-Caso contrário o sistema retorna uma mensagem falando os campos que faltam

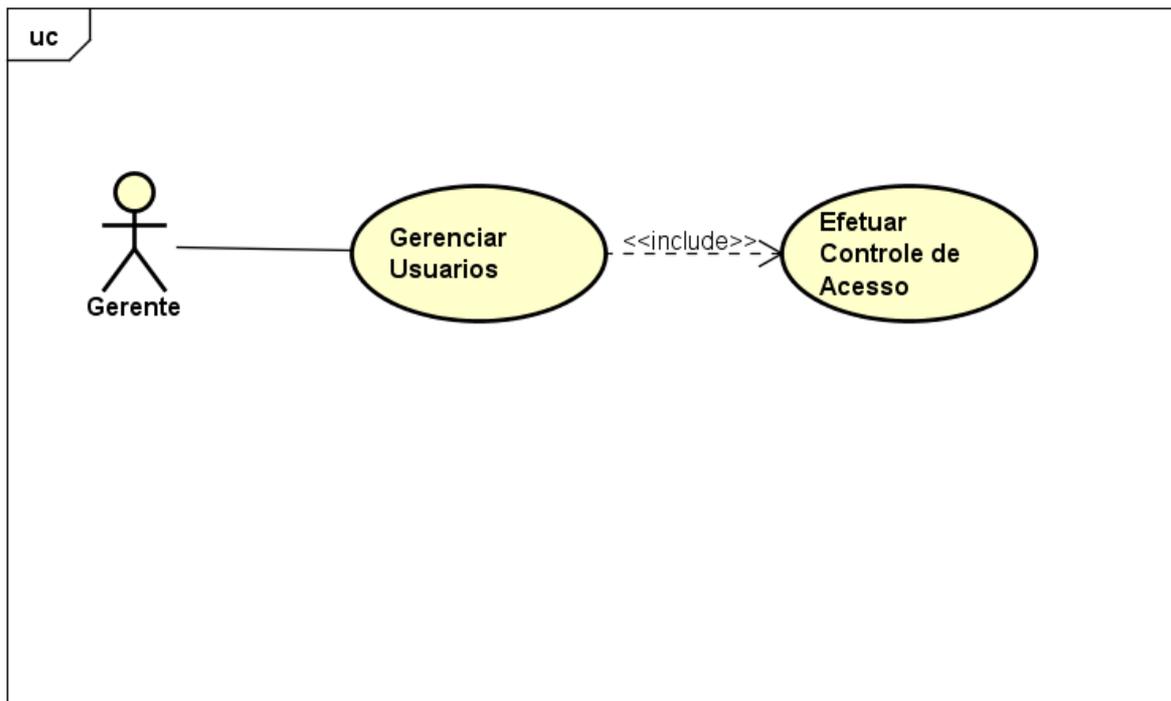


powered by Astah

Figura 5: Manter Estoque

Tabela 4: Manter Estoque

<b>Nome do Caso de Uso</b>	Manter Estoque
<b>Atores</b>	Atendente
<b>Pré-Condições</b>	Efetuar Controle de Acesso
<b>Cenário Principal</b>	1-Atendente informa os dados e em seguida seleciona confirmar. 2-O sistema carrega os dados 3-O atendente finaliza a operação
<b>Cenário Alternativo</b>	Atendente pode cancelar a operação a qualquer momento
<b>Casos de Teste</b>	1-Os campos obrigatórios estando com informações o sistema executa a operação 2-Caso contrário o sistema retorna uma mensagem falando os campos que faltam

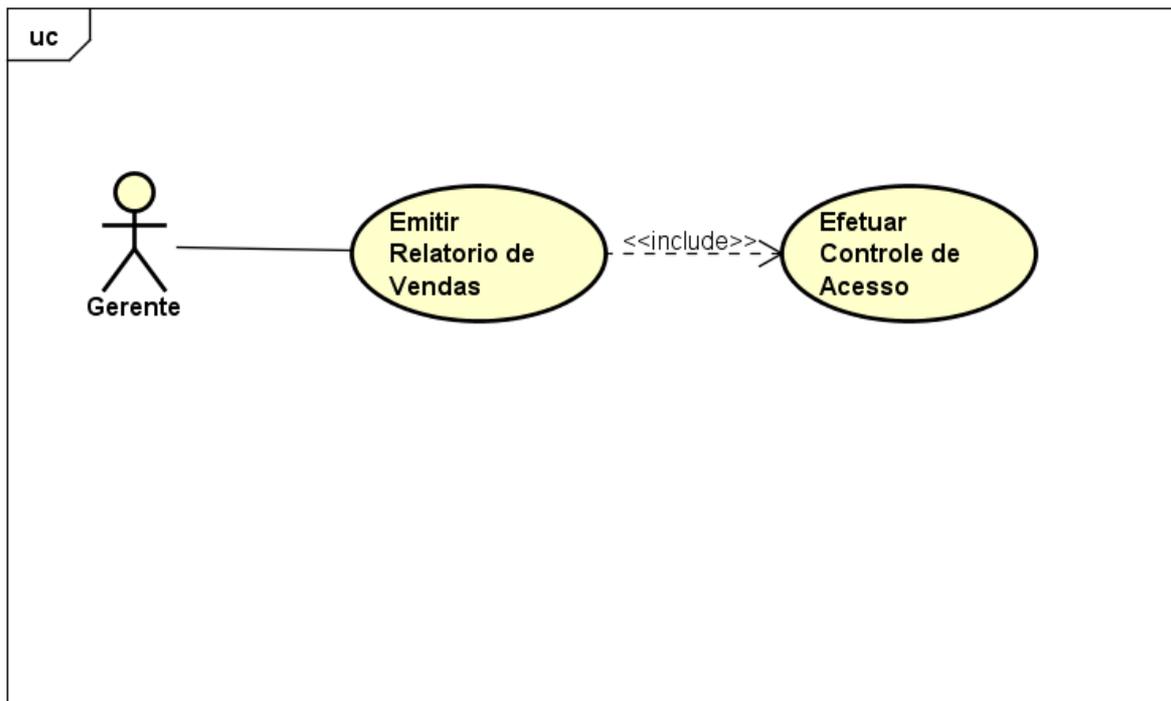


powered by Astah

Figura 6: Gerenciar Usuários

Tabela 5: Gerenciar Usuários

<b>Nome do Caso de Uso</b>	Gerenciar Usuários
<b>Atores</b>	Gerente
<b>Pré-Condições</b>	Efetuar Controle de Acesso
<b>Cenário Principal</b>	1-Atendente informa os dados e em seguida seleciona confirmar. 2-O sistema carrega os dados 3-O atendente finaliza a operação
<b>Cenário Alternativo</b>	Gerente pode cancelar a operação a qualquer momento
<b>Casos de Teste</b>	1-Os campos obrigatórios estando com informações o sistema executa a operação 2-Caso contrário o sistema retorna uma mensagem falando os campos que faltam



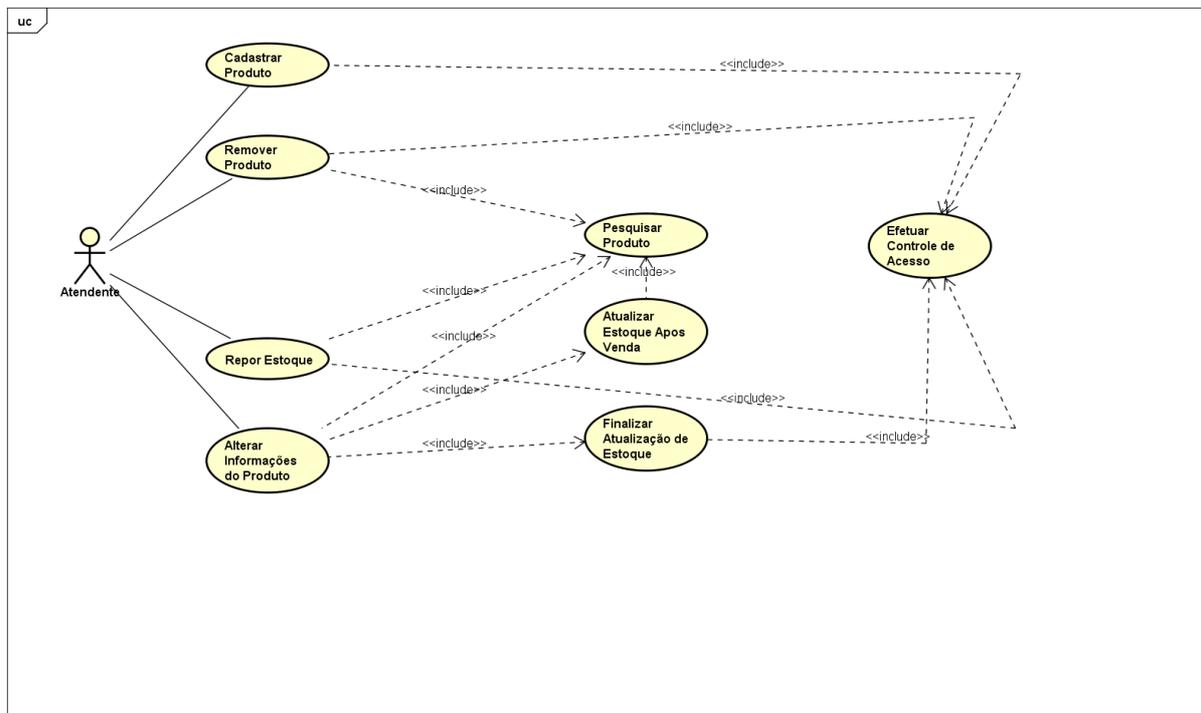
powered by Astah

Figura 7: Emitir Relatório de Vendas

Tabela 6: Emitir Relatório de Vendas

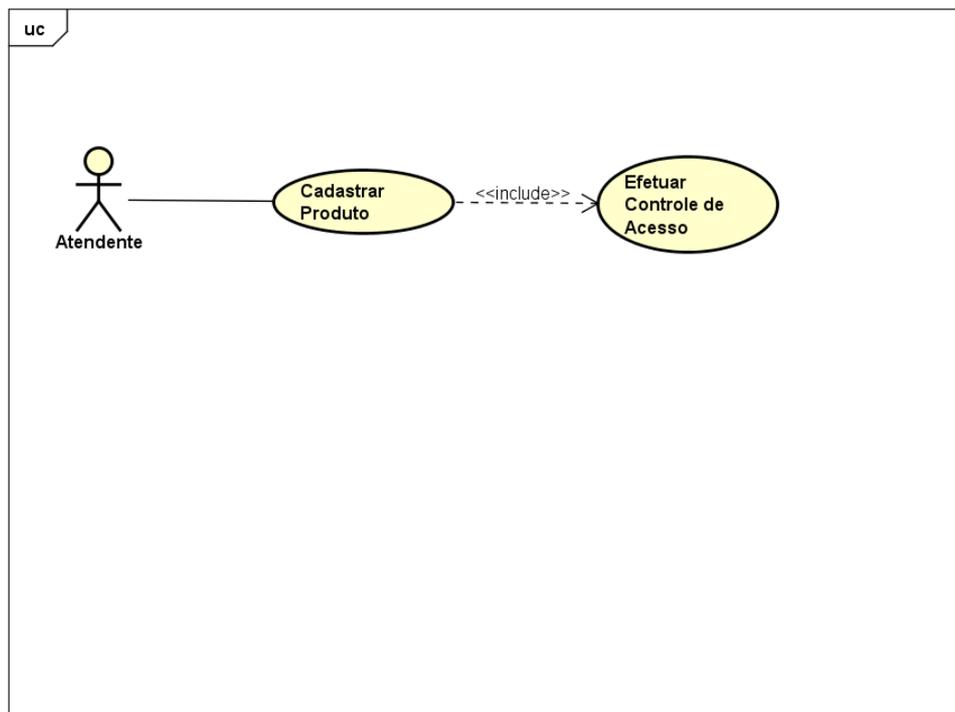
<b>Nome do Caso de Uso</b>	Emitir Relatório de Vendas
<b>Atores</b>	Gerente
<b>Pré-Condições</b>	Efetuar Controle de Acesso
<b>Cenário Principal</b>	1-Gerente consulta os relatórios 2-O sistema carrega os dados 3-O gerente emite os relatórios
<b>Cenário Alternativo</b>	Gerente pode cancelar a operação a qualquer momento
<b>Casos de Teste</b>	1-Os campos obrigatórios estando com informações o sistema executa a operação 2-Caso contrário o sistema retorna uma mensagem falando os campos que faltam

A figura 8 apresenta uma visão geral do cadastro de produto mais especificado.



powered by Astah

Figura 8: Visão do Cadastro de Produto

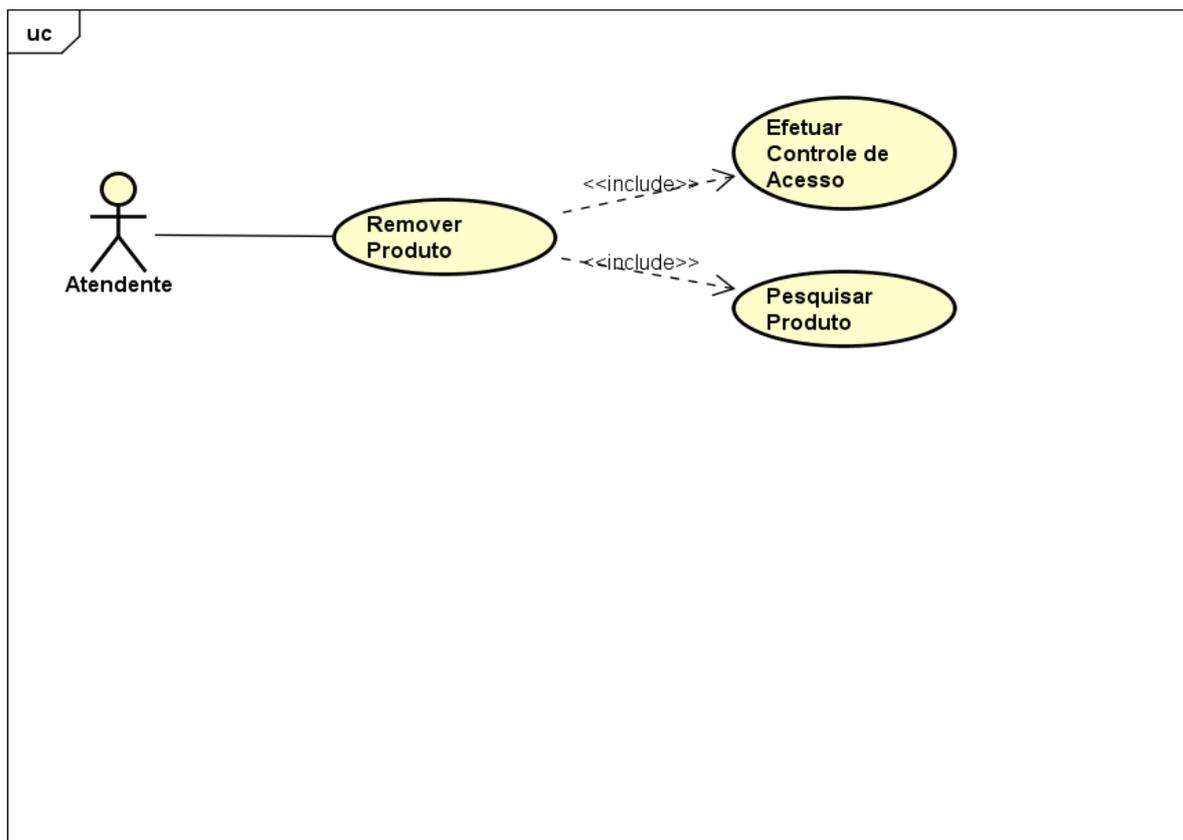


powered by Astah

Figura 9: Cadastro de Produto

Tabela 7: Cadastrar o Produto

<b>Nome do Caso de Uso</b>	Cadastrar Produto
<b>Atores</b>	Atendente
<b>Pré-Condições</b>	Efetuar Controle de Acesso
<b>Cenário Principal</b>	1-Atendente informa os dados e em seguida seleciona confirmar. 2-O sistema carrega os dados 3-O atendente finaliza a operação
<b>Cenário Alternativo</b>	Atendente pode cancelar a operação a qualquer momento
<b>Casos de Teste</b>	1-Os campos obrigatórios estando com informações o sistema executa a operação 2-Caso contrário o sistema retorna uma mensagem falando os campos que faltam

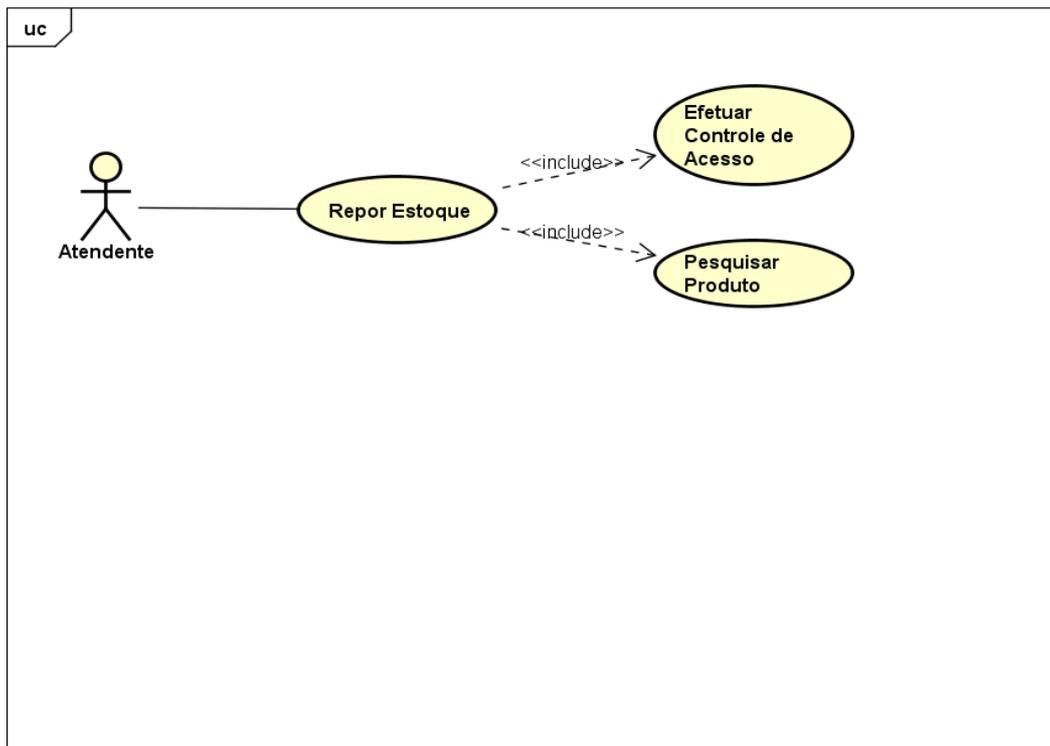


powered by Astah

Figura 10: Remover Produto

Tabela 8: Remover o Produto

<b>Nome do Caso de Uso</b>	Remover Produto
<b>Atores</b>	Atendente
<b>Pré-Condições</b>	Efetuar Controle de Acesso
<b>Cenário Principal</b>	1-Atendente pesquisa o produto 2-O sistema carrega os dados 3-O atendente informa o código 3-O atendente finaliza a operação
<b>Cenário Alternativo</b>	Atendente pode cancelar a operação a qualquer momento.
<b>Casos de Teste</b>	1-Os campos obrigatórios estando com informações o sistema executa a operação. 2-Caso contrário o sistema retorna uma mensagem falando os campos que faltam.

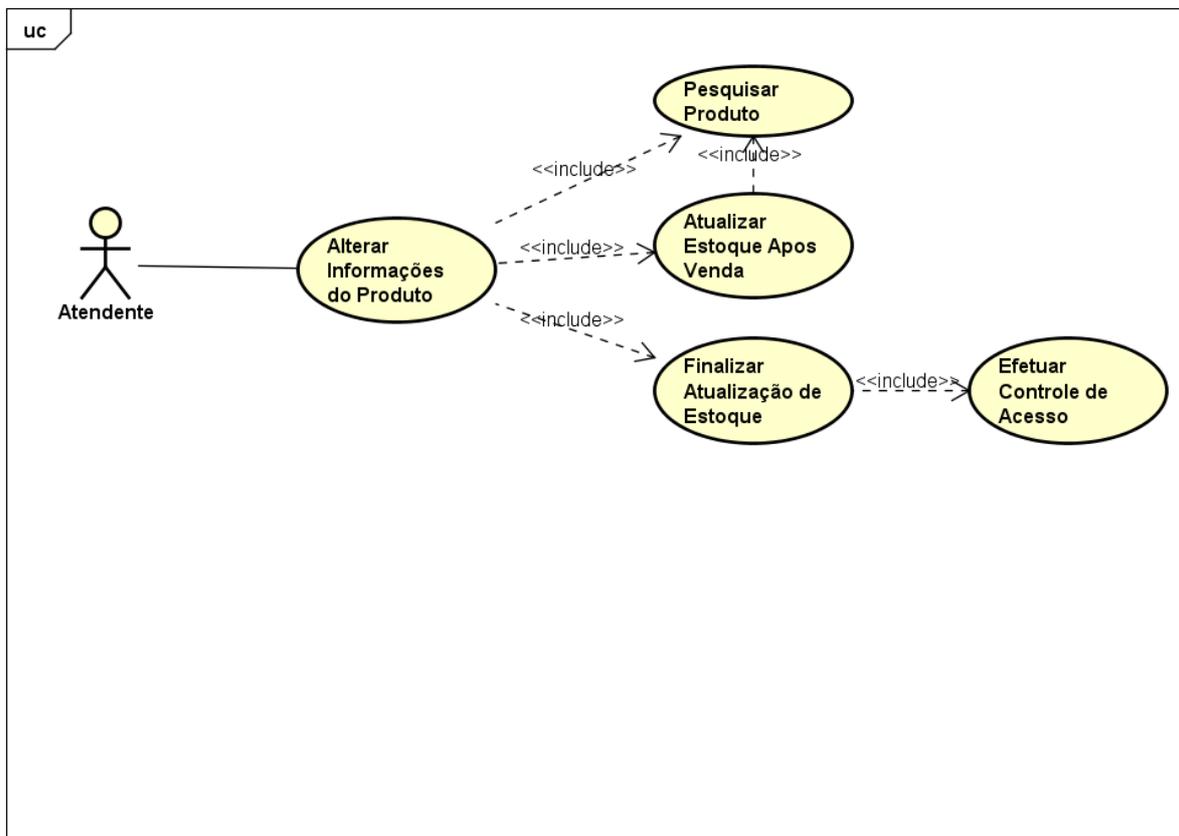


powered by Astah

Figura 11: Repor Estoque

Tabela 9: Repor Estoque

<b>Nome do Caso de Uso</b>	Repor Estoque
<b>Atores</b>	Atendente
<b>Pré-Condições</b>	Efetuar Controle de Acesso
<b>Cenário Principal</b>	1-Atendente pesquisa o produto 2-O sistema carrega os dados 3-O atendente informa os dados 4-O atendente finaliza a operação
<b>Cenário Alternativo</b>	Atendente pode cancelar a operação a qualquer momento
<b>Casos de Teste</b>	1-Os campos obrigatórios estando com informações o sistema executa a operação 2-Caso contrário o sistema retorna uma mensagem falando os campos que faltam



powered by Astah

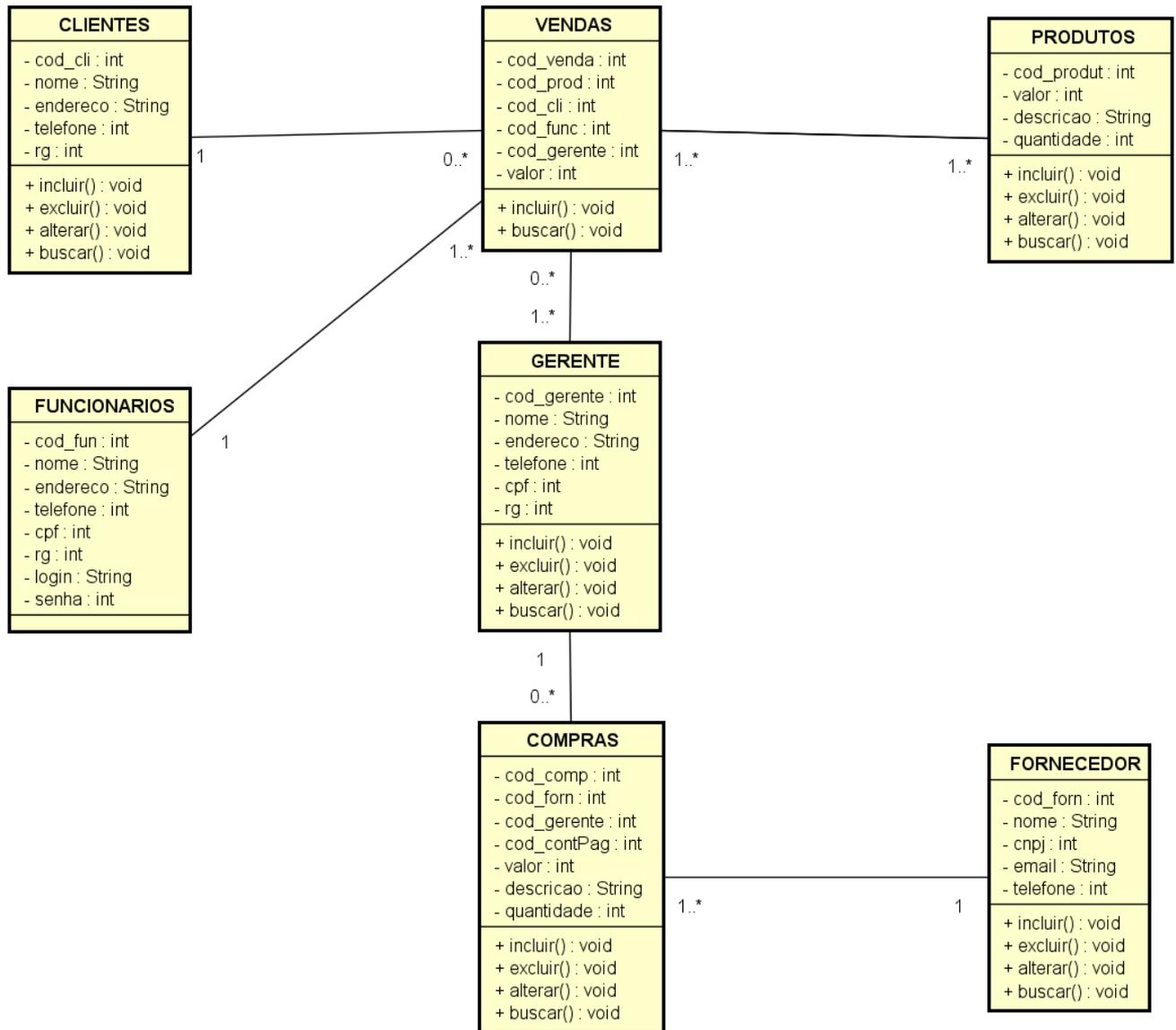
Figura 12: Alterar Informações dos Produtos

Tabela 10: Alterar Informações do Produto

<b>Nome do Caso de Uso</b>	Alterar Informações do Produto
<b>Atores</b>	Atendente
<b>Pré-Condições</b>	Efetuar Controle de Acesso
<b>Cenário Principal</b>	1-Atendente pesquisa o produto 2-O sistema carrega os dados 3-O atendente informa as atualizações 4-O atendente finaliza a operação
<b>Cenário Alternativo</b>	Atendente pode cancelar a operação a qualquer momento
<b>Casos de Teste</b>	1-Os campos obrigatórios estando com informações o sistema executa a operação 2-Caso contrário o sistema retorna uma mensagem falando os campos que faltam

### 3.3 Diagrama de Classe

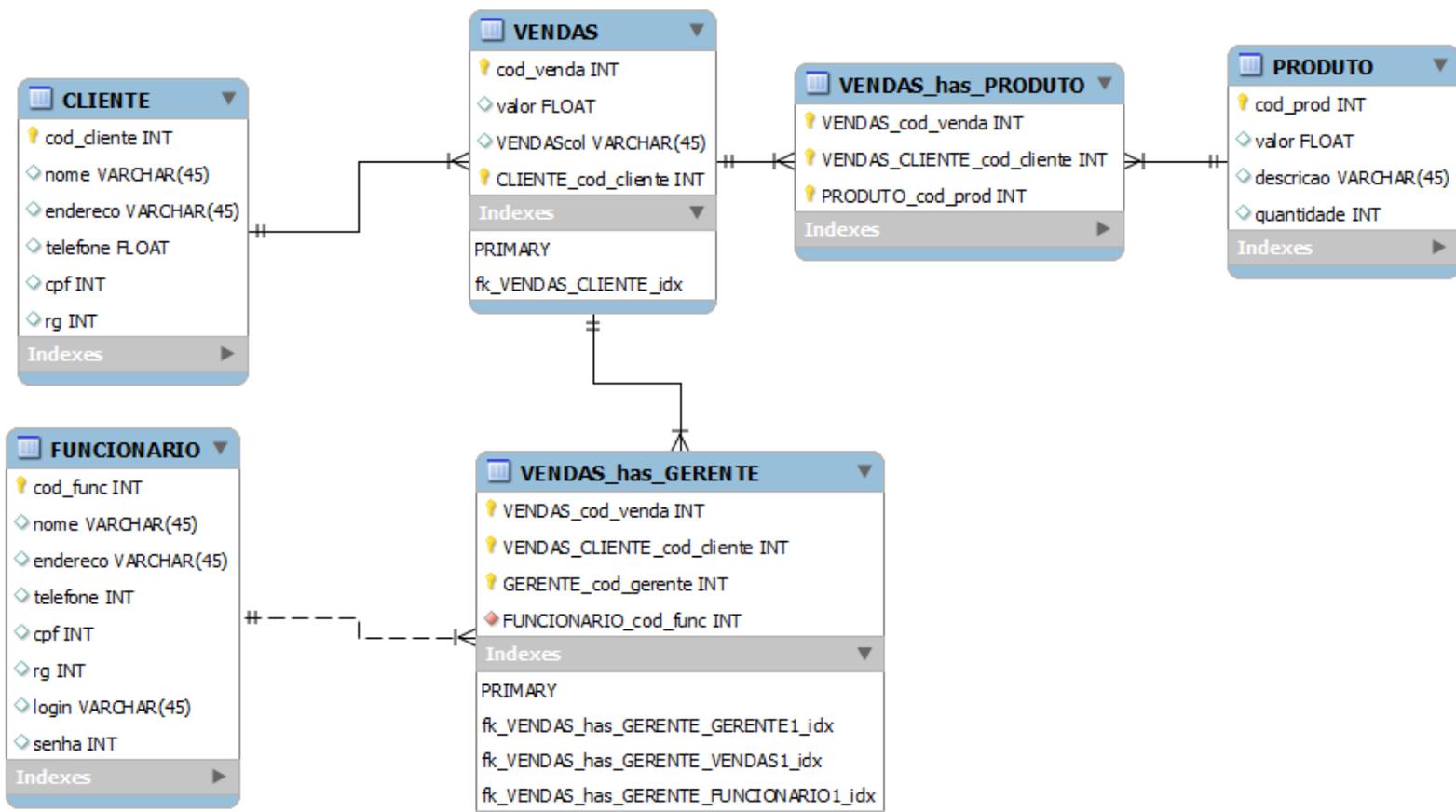
O Diagrama de classe oferece uma notação visual para a modelagem de classe e seus relacionamentos.



### 3.4 Diagrama de Entidade e Relacionamento

É um modelo em rede que descreve a diagramação dos dados armazenados de um sistema em alto nível de abstração. Ele é usado devido à complexidade da estrutura de dados e os relacionamentos.

Portanto é importante especificar e documentar, e o DER é uma ferramenta para isto.



### 3.5 Orçamento

Os recursos necessários para a elaboração da análise e do desenvolvimento do sistema são:

Análise de Sistemas			
Analista	Quantidade de Dias	Custo do Dia	Total
César H de Oliveira	100	R\$ 35,00	R\$ 3,500,00
Programador			
Programador	Quantidade de Dias	Valor Hora	Total
César H de Oliveira	120	R\$ 25,00	R\$ 3,000,00
			<b>Total</b> R\$ 6,500,00

Figura 15: Orçamento - Analista/Programador

Equipamento	Valor	Depreciação(2 Anos)	Total
Notebook	R\$ 2,600,00	R\$ 4,16	R\$ 915,20
			<b>Total</b> R\$ 915,20

Figura 16: Orçamento - Equipamentos

TOTAL	
Mão de Obra	R\$ 6,500,00
Equipamentos	R\$ 915,20
<b>Total</b> R\$ 7,415,20	

Figura 17: Orçamento - Total

### 3.6. CRONOGRAMA DA ESTRUTURA DE DESENVOLVIMENTO

Atividade	Novembro	Dezembro	Janeiro	Fevereiro	Março	Abril	Maiο	Junho	Julho	Agosto	Setembro
Entrega do Pré Projeto	■										
Levantamento das Necessidades		■	■	■							
Levantamento dos Requisitos		■	■	■							
Análise dos Requisitos			■	■							
Gerenciamento dos Requisitos				■							
Diagramas de Casos de Uso					■						
Diagrama de Atividades					■						
Preparação para a Qualificação				■	■						
Apresentação da Qualificação						■					
Diagrama de Classe				■	■						
Diagrama de Sequencia					■		■				
Desenvolvimento do Sistema							■	■	■	■	
Entrega e Apresentação										■	■

■ = Feito

■ = A Fazer

## 4. DESENVOLVIMENTO DO SISTEMA

O sistema utiliza a IDE *Eclipse*, a linguagem de programação *Java* Junto com o banco de dados *MySQL* que é responsável por armazenar os dados.

Mais pode ser reproduzido em qualquer IDE pelo fato do projeto utilizar maven.

### 4.1 Organização do Projeto

Para a melhor organização do projeto, o mesmo foi dividido em pacotes.

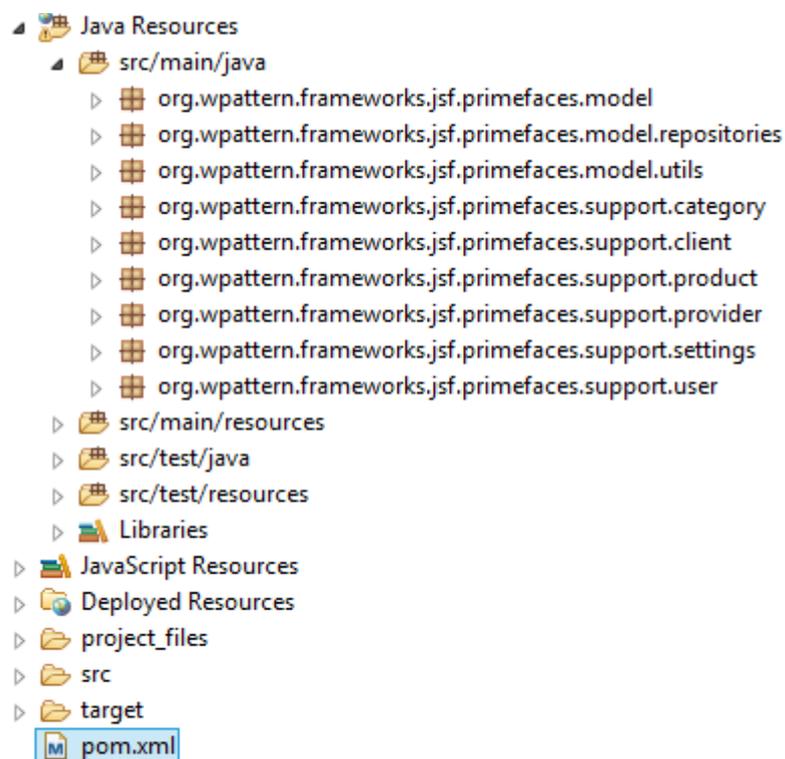


Figura 18: Organização dos Pacotes

## 4.2 Repositórios Maven

A figura abaixo mostra os repositórios que foram utilizados no sistema em conjunto com o Maven, Acredito que seja de grande importância, pelo fato de ter sido a base do desenvolvimento do sistema.

Você pode conseguir cada repositório no site do desenvolvedor de cada *framework*.

```
8  <properties>
9    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
10   <spring.security.version>3.1.2.RELEASE</spring.security.version>
11   <spring.version>4.0.3.RELEASE</spring.version>
12 </properties>
13
14 <repositories>
15   <!-- MAVEN REPOSITORY -->
16   <repository>
17     <id>mvn-repository</id>
18     <url>http://mvnrepository.com/</url>
19   </repository>
20
21   <!-- MAVEN SEARCH REPOSITORY -->
22   <repository>
23     <id>mavensearch</id>
24     <url>http://www.mvnsearch.org/maven2</url>
25   </repository>
26
27   <!-- SPRING REPOSITORY -->
28   <repository>
29     <id>com.springsource.repository.bundles.release</id>
30     <name>EBR Spring Release Repository</name>
31     <url>http://repository.springsource.com/maven/bundles/release </url>
32   </repository>
33
```

Figura 19: Repositórios Maven 1

Esta é a segunda imagem representando a lista de repositórios que foi utilizado em conjunto com o Maven.

```
34      <!-- SPRING REPOSITORY -->
35      <repository>
36          <id>com.springsource.repository.bundles.external</id>
37          <name>EBR External Release Repository</name>
38          <url>http://repository.springsource.com/maven/bundles/external </url>
39      </repository>
40
41      <!-- JBOSS REPOSITORY -->
42      <repository>
43          <id>jboss-repository</id>
44          <url>https://repository.jboss.org/nexus/content/repositories/</url>
45      </repository>
46
47      <!-- JBOSS REPOSITORY -->
48      <repository>
49          <id>JBoss Repo</id>
50          <url>http://repository.jboss.com</url>
51      </repository>
52
53      <!-- GOOGLE REPOSITORY -->
54      <repository>
55          <id>google-repository</id>
56          <url>http://google-maven-repository.googlecode.com/svn/repository/</url>
57      </repository>
58
59      <!-- GLASSFISH REPOSITORY -->
60      <repository>
61          <id>glassfish-repository</id>
62          <url>http://download.java.net/maven/glassfish/</url>
63      </repository>
```

Figura 20: Repositórios Maven 2

### 4.3 Opções de Idiomas

A opção de idioma surgiu no meio do desenvolvimento do sistema, eu estava procurando ajuda pra um erro que estava dando e vi um Tópico falando sobre, então resolvi fazer porque eu achei que ficava mais organizado. Agora o sistema tem as opções de Português e Inglês.

O código abaixo mostra a forma que foi atribuído cada texto do sistema a uma variável, existe dois arquivos desse tipo um com os nomes em português e o outro em inglês.

```
1 # Language
2 language=Idioma
3
4 # Login
5 login_username=Usuario
6 login_password=Senha
7 login_go=Faca o Login
8 login_action_go=Acessar
9 login_action_logout=Sair
10 login_invalid=Usuario e/ou senha invalidos.
11
12 # Logout
13 logout=Sair
14
15 # Menu
16 menu_name=Opcoes
17 menu_dataset=Dados
18 menu_category=Categorias
19 menu_product=Produtos
20 menu_user=Usuarios
21 menu_settings=Configuracoes
22 menu_language=Idioma
23 menu_client=Clientes
24 menu_provider=Fornecedores
25
26 # Button
27 button_add=Adicionar
28 button_delete=Deletar
29 button_update=Atualizar
30 button_save=Salvar
31 button_cancel=Cancelar
32
```

Figura 21: Idiomas - Português

Esta é a segunda imagem representando as opções de idiomas do sistema, na foto a baixo é mostrado o arquivo com o idioma na língua inglesa.

```
1 # Language
2 language=Language
3
4 # Login
5 login_username=Username
6 login_password>Password
7 login_go>Login
8 login_action_go>Login
9 login_action_logout=Logout
10 login_invalid=Invalid user or password.
11
12 # Logout
13 logout=Logout
14
15 # Menu
16 menu_name=Options
17 menu_dataset=Data
18 menu_category=Categories
19 menu_product=Products
20 menu_user=Users
21 menu_settings=Settings
22 menu_language=Language
23 menu_client=Client
24 menu_provider=Provider
25
26 # Button
27 button_add=Add
28 button_delete>Delete
29 button_update=Update
30 button_save=Save
31 button_cancel=Cancel
32
```

Figura 22: Idioma - Inglês

#### 4.4 Temas - PrimeFaces

Aqui será explicado a configuração do tema dos temas do *primefaces*, que é algo muito fácil e que muda totalmente a aparência do sistema.

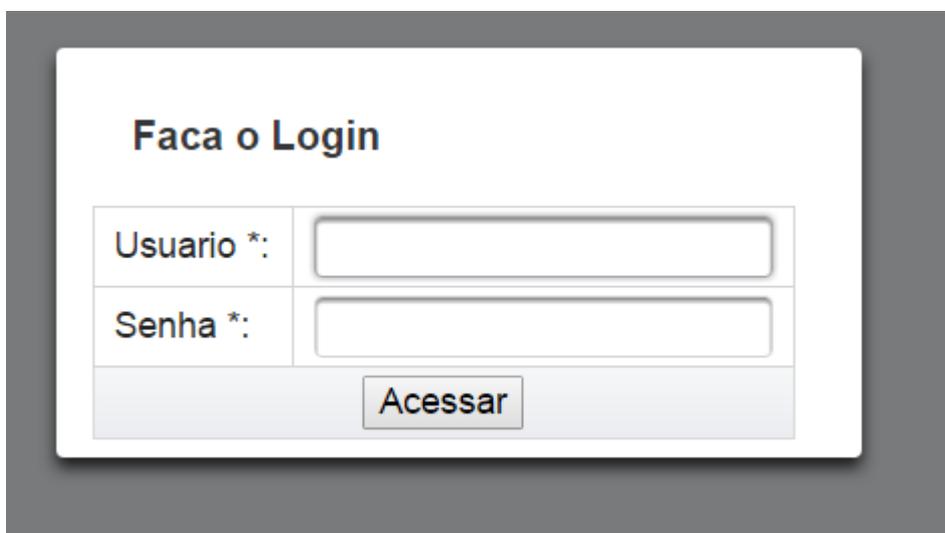
```
<!-- PRIMEFACES THEME CONFIG -->  
<context-param>  
  <param-name>primefaces.THEME</param-name>  
  <param-value>delta</param-value>  
</context-param>
```

Figura 23: PrimeFaces - Tema

Neste caso o sistema está com o tema **DELTA** caso eu queira ou precise mudar é simples, só precisa trocar o nome do mesmo. Por exemplo no lugar de *delta* colocaria o tema *glass-x*. Esta configuração fica em *web.xml*.

#### 4.5 Telas Do Sistema

Nesta imagem é apresentada a tela de *login* do sistema, essa é a primeira tela que o usuário vai ver quando abrir o sistema.



A imagem mostra uma interface de usuário para login. No topo, o título "Faca o Login" está centralizado. Abaixo dele, há um formulário com dois campos de entrada: "Usuario \*" e "Senha \*". Cada campo é precedido por um ícone de lupa, indicando uma função de busca ou sugestão. Abaixo dos campos, há um botão "Acessar" centralizado. O formulário está contido em uma caixa branca com uma sombra, sobre um fundo cinza escuro.

Figura 24: Telas - Login

Após ele digitar o usuário e a senha correta já é direcionado para a tela principal do sistema.

Esta tela é a de cadastro de usuários, ai serão cadastrados todos os usuários que vão utilizar o sistema, nela você tem os campos: Nome, *Login*, Senha e Acesso que é o tanto de privilegio que o seu usuário vai ter, no caso *ROLE\_USER* tem acesso total no sistema.

## Sistema de Gestao

The screenshot displays a web application interface for user management. On the left, a sidebar titled 'Opcoes' contains two menu items: 'Dados' and 'Configuracoes'. The main content area is titled 'Usuario' and features three buttons: 'Adicionar', 'Deletar', and 'Atualizar'. Below these buttons is a table with the following data:

Id	Nome	Login	Senha	Acesso
1	Cesar H de Oliveira	cesar	1010	ROLE_USER

At the bottom of the page, there is a small copyright notice: 'Copyright 2016-2017 Aconque do Milton All rights reserved.'

Figura 25: Telas - Cadastro de Usuários

## 5. Conclusão

Para o desenvolvimento do sistema foi preciso conhecer a rotina de um açougue pra ver as principais necessidades que eles enfrentavam, depois de todo o levantamento começou a parte da modelagem.

O UML ajudou bastante na hora do desenvolvimento, com todo ele pronto foi fácil o desenvolvimento das classes e do banco de dados.

Já no desenvolvimento do projeto tive algumas dificuldades no começo do projeto, que foram resolvidas com o *Maven*, e também depois de criar o primeiro CRUD(cadastro ao banco de dados). Ficou tudo mais simples, ai foi só questão de acertar alguns erros e mudar algumas coisas.

Do que eu imaginei do começo do projeto para o final, eu mudei poucas coisas. Algumas eu tirei outras eu só mudei, e teve alguns que eu não tinha ideia que ia fazer e acabei colocando que é o caso do suporte para linguagem, porém eu acho que é isso mesmo desenvolver um projeto nem sempre é tudo como esperado e nem todo erro é ruim.

Por fim eu gostei muito de fazer o sistema e, sem dúvida, nem uma aprendi muito com o mesmo, sempre que dava algo certo só me animava mais para continuar.

## REFERÊNCIAS BIBLIOGRÁFICAS

**Artigo Java Magazine 69.** Dica de Leitura. Disponível em: <<http://www.devmedia.com.br/artigo-java-magazine-69-spring-security/12883>>. Acesso em: 21 de junho. 2016.

**Computação em Nuvem.** Dica de Leitura. Disponível em: <<http://www.opus-software.com.br/livro-computacao-em-nuvem-landing/>>. Acesso em: 12 de maio. 2016.

DEITEL, Paul et al... **Java como programar.** 8ª Edição. Pearson Prentice Hall, 2010. 1104 p.

**Eclipse Platform Technical Overview.** Dica de Leitura. Disponível em: <<https://eclipse.org/articles/Whitepaper-Platform-3.1/eclipse-platform-whitepaper.pdf>>. Acesso em: 20 de junho. 2016.

GILLIARD CORD. **Aplicações Java para web com JSF e JPA.** Casa do Código, 296 p.

GÓES, Wilson Moraes. **Aprenda UML por meio de estudos de caso.** São Paulo: Novatec, 2014. 286 p.

HÉBERT COELHO. **JSF Eficaz As melhores práticas para o desenvolvedor web Java.** Casa do Código, 207 p.

**Introdução ao Hibernate 3.** Dica de Leitura. Disponível em: <[http://www.cin.ufpe.br/~rdma/documentos/intruducaao\\_hibernate3\\_guj.pdf](http://www.cin.ufpe.br/~rdma/documentos/intruducaao_hibernate3_guj.pdf)>. Acesso em: 28 de junho. 2016.

*O que é WildFly*. Dica de Leitura. Disponível em: <<https://www.4linux.com.br/o-que-e-wildfly>>. Acesso em: 10 de maio. 2016.

PEDRO N, RUI RUAS **O guia prático do MySQL**. Centro Atlântico, 400 p.

*PrimeFaces in the Enterprise*. Dica de Leitura. Disponível em: <<http://www.oracle.com/technetwork/articles/java/java-primefaces-2191907.html>>. e <<http://www.primefaces.org/documentation>>. Acesso em: 16 de maio. 2016.

RICARDO R. GUDWIN. **Introdução a UML**. DCA-FEEC-UNICAMP, 22p.

THIAGO FARIA. **Java EE 7 com JSF, PrimeFaces e CDI**. AlgaWorks, 2013. 199 p.

WALTER S. FILHO **Introdução ao Apache Maven**. ETG Tecnologia de Informação, 2008. 81p.