ALEX RODRIGUES

GERSON LIBERATO JÚNIOR

MONIQUE SANTOS DE OLIVEIRA

NARA ISA JELLO SHINZATO

RODRIGO BAPTISTA

**SISTEMA PARA CONTROLE DE CAIXA E ESTOQUE APLICÁVEL A MERCADOS DE PEQUENO E MÉDIO PORTE**

São Caetano do Sul/SP

2017

ALEX RODRIGUES

GERSON LIBERATO JÚNIOR

MONIQUE SANTOS DE OLIVEIRA

NARA ISA JELLO SHINZATO

RODRIGO BAPTISTA

**SISTEMA PARA CONTROLE DE CAIXA E ESTOQUE APLICÁVEL A MERCADOS DE PEQUENO E MÉDIO PORTE**

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia de São Caetano do Sul, sob a orientação do Professor MSc. Flávio Viotti, como requisito parcial para obtenção do diploma de Graduação no Curso de Análise e Desenvolvimento de Sistemas.

São Caetano do Sul/SP

2017

**RESUMO**

BAPTISTA, Rodrigo; LIBERATO, Gerson; OLIVEIRA, Monique; RODRIGUES, Alex; SHINZATO, Nara. **SISTEMA PARA CONTROLE DE CAIXA E ESTOQUE APLICÁVEL A MERCADOS DE PEQUENO E MÉDIO PORTE**, xx fls. Trabalho de Conclusão de Curso – Faculdade de Tecnologia de São Caetano do Sul – Antônio Russo, São Caetano do Sul, 2017.

A automatização dos processos de venda tem sido indispensável até mesmo para as menores empresas que atuam no setor de vendas. Este trabalho demonstra as etapas de levantamento de requisitos, projeto e desenvolvimento de um sistema de vendas inicialmente voltado para supermercados. O sistema aqui apresentado engloba o controle de estoque e de caixa, visando reduzir a ocorrência de erros durante a realização de vendas, por exemplo. Inicialmente será feita uma apresentação sobre o domínio da aplicação contendo informações a serem levadas em consideração durante a construção do projeto e os requisitos do sistema. A etapa de projeto é apresentada em seguida, nessa etapa será mostrada a elaboração de diagramas que permitem que a implementação do sistema seja organizada que o sistema seja bem documentado. Ao final do projeto, o sistema será desenvolvido de acordo com os requisitos necessários.

**Palavras-chave**: orientação a objetos; JAVA; UML; desenvolvimento de software; sistema de vendas.

**ABSTRACT**

BAPTISTA, Rodrigo; LIBERATO, Gerson; OLIVEIRA, Monique; RODRIGUES, Alex; SHINZATO, Nara. **SISTEMA PARA CONTROLE DE CAIXA E ESTOQUE APLICÁVEL A MERCADOS DE PEQUENO E MÉDIO PORTE**, xx fls. Trabalho de Conclusão de Curso – Faculdade de Tecnologia de São Caetano do Sul – Antônio Russo, São Caetano do Sul, 2017.

The automation of selling processes have been indispensable even for the smallest companies that operate in the sales sector. This work demonstrates the steps of requirements gathering, project and the development of a sales system initially aimed for supermarkets. The system presented here encompasses inventory and cash control, in order to reduce the occurrence of errors during sales, for example. Initially a presentation will be made on the application domain containing information to be taken into consideration during project construction and the system requirements. The project step is then presented, in this step will be shown the elaboration of diagrams that allow the implementation of the system to be organized and that the system is well documented. At the end of the project, the system will be developed according to the necessary requirements.

**Keywords:** object orientation; JAVA; UML; software development; Sales system.

**LISTA DE ILUSTRAÇÕES**

Figura 1: Diagrama de Contexto 19

Figura 2: Diagrama de Casos de Uso 22

Figura 3: Diagrama Entidade-Relacionamento 46

Figura 4: Diagrama de Classes 54

**LISTA DE QUADROS**

Quadro 1: Requisitos Funcionais 19

Quadro 2: Requisitos Não Funcionais 21

Quadro 3: Documentação do Ator Gerente 23

Quadro 4: Documentação do Ator Estoquista 23

Quadro 5: Documentação do Ator Operador de Caixa 24

Quadro 6: Caso de Uso Efetuar Login 24

Quadro 7: Caso de Uso Autenticar Usuário 26

Quadro 8: Caso de Uso Manter Usuários 27

Quadro 9: Caso de Uso Manter Fornecedores 28

Quadro 10: Caso de Uso Manter Produtos 30

Quadro 11: Caso de Uso Manter Preços de Produtos 32

Quadro 12: Caso de Uso Obter Consulta de Estoque 33

Quadro 13: Caso de Uso Observar Caixas 33

Quadro 14: Caso de Uso Retirar Produtos de Estoque 34

Quadro 15: Caso de Uso Registrar Entrada de Produtos 35

Quadro 16: Caso de Uso Registrar Retirada de Produtos 36

Quadro 17: Caso de Uso Iniciar Caixa 37

Quadro 18: Caso de Uso Adicionar Item à Venda 38

Quadro 19: Caso de Uso Remover Item da Venda 40

Quadro 20: Caso de Uso Cancelar Venda 40

Quadro 21: Caso de Uso Realizar Venda 41

Quadro 22: Caso de Uso Realizar Fechamento de Caixa 43

Quadro 23: Caso de Uso Consultar Vendas 44

Quadro 24: Caso de Uso Obter Relatórios 44

Quadro 25: Tabela Unidades 47

Quadro 26: Tabela Fornecedores 47

Quadro 27: Tabela Endereços Fornecedores 47

Quadro 28: Tabela E-mails Fornecedores 47

Quadro 29: Tabela Telefones Fornecedores 48

Quadro 30: Tabela Funções Usuários 48

Quadro 31: Tabela Categorias Produtos 48

Quadro 32: Tabela Motivos Produtos Retirados 48

Quadro 33: Tabela Usuários 48

Quadro 34: Tabela Caixas 49

Quadro 35: Tabela Sessões 49

Quadro 36: Tabela Sessões Caixas 49

Quadro 37: Tabela Produtos 50

Quadro 38: Tabela Preços de Produtos 50

Quadro 39: Tabela Estoques 51

Quadro 40: Tabela Vendas 51

Quadro 41: Tabela Itens de Vendas 51

Quadro 42: Tabela Pagamentos de Vendas 52

Quadro 43: Tabela Produtos Retirados 52

Quadro 44: Tabela Entradas de Produtos 53

**LISTA DE ABREVIATURAS E SIGLAS**

TI

UML

IDE

SGDB

**SUMÁRIO**

[INTRODUÇÃO 11](#_Toc483842736)

[OBJETIVOS GERAIS 11](#_Toc483842737)

[OBJETIVOS ESPECÍFICOS 12](#_Toc483842738)

[JUSTIFICATIVA 12](#_Toc483842739)

[METODOLOGIA 12](#_Toc483842740)

[RESULTADOS ESPERADOS 13](#_Toc483842741)

[1 DESCRIÇÃO DO SISTEMA 14](#_Toc483842742)

[2 TÉCNICAS E TECNOLOGIAS UTILIZADAS 15](#_Toc483842743)

[2.1 Linguagem de Modelagem Unificada (UML) 15](#_Toc483842744)

[2.2 Git 16](#_Toc483842745)

[2.3 Java 16](#_Toc483842746)

[2.4 MySQL 17](#_Toc483842747)

[3 ESPECIFICAÇÃO DOS REQUISITOS DO SISTEMA 18](#_Toc483842748)

[3.1 O Produto 18](#_Toc483842749)

[3.2 Restrições do Produto 18](#_Toc483842750)

[3.3 Escopo do Projeto 18](#_Toc483842751)

[3.4 Requisitos Funcionais 19](#_Toc483842752)

[3.5 Requisitos Não Funcionais 21](#_Toc483842753)

[4 MODELO DE CASOS DE USO DO SISTEMA 22](#_Toc483842754)

[4.1 Diagrama de Casos de Uso do Sistema 22](#_Toc483842755)

[4.2 Documentação dos Atores 23](#_Toc483842756)

[4.3 Documentação dos Casos de Uso 24](#_Toc483842760)

[5 MODELOS DE ANÁLISE E PROJETO DO SISTEMA 46](#_Toc483842780)

[5.1 Diagrama Entidade-Relacionamento 46](#_Toc483842781)

[5.3 Diagrama de Sequência: Realizar Venda 56](#_Toc483842794)

[5.4 Diagrama de Pacotes 57](#_Toc483842795)

[5.5 Diagrama de Componentes 57](#_Toc483842796)

[5.6 Diagrama de Implantação 57](#_Toc483842797)

[CONSIDERAÇÕES FINAIS 58](#_Toc483842798)

[REFERÊNCIAS 59](#_Toc483842799)

# INTRODUÇÃO

Diversos processos precisam ser efetuados para que uma venda seja realizada. Quando um cliente chega ao caixa de um supermercado para efetuar o pagamento de uma compra, por exemplo, é necessário o cumprimento várias etapas para que a venda seja finalizada. Além disso, as informações sobre as vendas e sobre o estoque de produtos são valiosas para a gerencia do negócio.

A venda torna-se um fenômeno complexo ao considerar as etapas necessárias para que não haja ocorrência de erros e, consequentemente, prejuízos às empresas e para que seja possível extrair informações a partir das vendas para que melhores decisões sejam tomadas a fim de agregar valor ao negócio. Os registros de produtos que entraram no estoque, produtos que foram vendidos, preços de compra e de venda das mercadorias, datas de vencimento e outras informações são indispensáveis para administrar corretamente um estabelecimento comercial.

Baseando-se em registros das vendas realizadas e sobre as movimentações que ocorreram no estoque, como entrada e vencimento de produtos, é possível obter conclusões que tornarão possível a diminuição dos prejuízos e dos custos de mercadorias.

Erros cometidos durante as vendas dificultam a gestão da empresa e a torna mais suscetível a prejuízos. Uma venda pode ser cobrada de indevidamente por um funcionário e, se não houver registros sobre as vendas, isso pode ocorrer inúmeras vezes sem que seja notado. A automatização dos processos de venda e controle de estoque por meio da Tecnologia da Informação é capaz de diminuir a quantidade de falhas e agilizar a realização dessas tarefas. A TI pode ser definida como “um conjunto convergente de tecnologias em microeletrônica, computação (software e hardware), telecomunicações/radiodifusão e optoeletrônica” (Castells, 2000, p.49).

O software desenvolvido tem como enfoque automatizar as tarefas de controle de caixa e de estoque com a finalidade de solucionar os problemas apresentados, possibilitando o armazenamento de informações sobre as atividades do negócio e o aumento da agilidade do trabalho.

## OBJETIVOS GERAIS

O objetivo deste trabalho é criar e documentar um software que visa auxiliar estabelecimentos comerciais por meio da automatização de processos de venda e controle de estoque.

## OBJETIVOS ESPECÍFICOS

A construção do software apresentado neste trabalho consiste em descrever os requisitos do sistema, elaborar diagramas para especificar as funcionalidades que serão implementadas, criar um banco de dados relacional e desenvolver o sistema em linguagem de programação orientada a objetos.

## JUSTIFICATIVA

Segundo Bezerra (2015), a informação é um novo tipo de bem econômico e a empresa que dispõe de mais informações sobre seu processo de negócio possui vantagem em relação às outras.

Os sistemas de informação, além de agilizar e tornar o trabalho menos suscetível a erros, são capazes de mostrar ao usuário diversas conclusões com base nas informações obtidas por meio do banco de dados. As conclusões obtidas tornam possível que gestores de estabelecimentos comerciais façam escolhas assertivas ao tomarem decisões para o negócio. Por esses motivos, os sistemas que possibilitam o armazenamento de dados e a automatização de tarefas são indispensáveis para empresas que buscam crescimento.

## METODOLOGIA

Na construção deste trabalho acadêmico, foram realizadas pesquisas em artigos científicos e livros para obtenção de um melhor entendimento a respeito do tema a ser abordado e para adquirir mais conhecimento a respeito das tecnologias utilizadas. Além disso, para o desenvolvimento do software houve levantamento de requisitos do sistema e elaboração de diagramas.

## RESULTADOS ESPERADOS

Ao final do trabalho espera-se ter desenvolvido um sistema coeso, com funcionalidades sólidas e de fácil utilização e que possa gerar valor para os usuários finais e interessados.

# 1 DESCRIÇÃO DO SISTEMA

O sistema *MarketSoft* foi desenvolvido para automatizar as vendas e o controle de estoque de supermercados a fim de reduzir a quantidade de erros cometidos pelos funcionários e de facilitar o acesso às informações sobre vendas, produtos mais vendidos, faturamento e outras informações que podem aumentar a lucratividade e agregar valor ao negócio. Além disso, o sistema agiliza a execução das tarefas pertinentes aos funcionários, o que evita que o trabalho seja atrasado ou que os clientes dos supermercados passem muito tempo em filas, por exemplo.

Três grupos de funcionários serão responsáveis por operar o sistema: operadores de caixa, gerentes e estoquistas. Cada grupo de funcionários tem responsabilidades distintas e executam tarefas diferentes dos demais. Para garantir a integridade e a segurança do sistema, os grupos de usuários não têm acesso às funcionalidades que não fazem parte da rotina do trabalho que executam e é necessária a autenticação de usuário e senha para operar o sistema.

O sistema emite relatórios gerenciais sobre venda de produtos e sobre estoque de produtos. Ademais, o gerente pode obter informações sobre os caixas em tempo real, como por exemplo obter o valor em cada caixa, visualizar quais caixas estão abertos e quais estão ociosos.

Além de fornecer informações de caráter gerencial, é possível cadastrar promoções e descontos em produtos que são um grande atrativo para os clientes do estabelecimento.

# 2 TÉCNICAS E TECNOLOGIAS UTILIZADAS

Na elaboração do projeto foram utilizadas técnicas de Engenharia de Requisitos para estabelecer as funcionalidades do sistema. Os requisitos do sistema são divididos em funcionais, que descrevem as transações que o sistema deve realizar, e não funcionais, que estabelecem regras para qualidade do software.

As etapas de análise e projeto do sistema foram ilustradas por meio de diagramas UML, com a concepção dos modelos de casos de uso, análise e projeto do sistema.

O controle de versões da aplicação foi realizado com o auxílio da ferramenta Git. Para a diagramação e documentação do sistema utilizou-se o Astah Professional, que permite a criação de diversos diagramas UML.

O desenvolvimento do sistema foi realizado utilizando a linguagem Java SE por meio do IDE NetBeans. Para gerenciamento de banco de dados o SGBD escolhido foi o MySQL *Community Server*.

## 2.1 Linguagem de Modelagem Unificada (UML)

A UML é uma linguagem visual para a modelagem de sistemas por meio de diversos diagramas que possibilitam a representação de conceitos do paradigma de orientação a objetos. Durante a definição da notação da UML, procurou-se aproveitar as melhores características de notações que já existiam anteriormente, removendo alguns elementos e adicionando outros a fim de construir uma linguagem mais expressiva.

Segundo Bezerra (2006), pode-se fazer uma analogia da UML com uma caixa de ferramentas: um construtor usa sua caixa de ferramentas para realizar tarefas e, da mesma forma, a UML pode ser vista como uma caixa de ferramentas utilizadas pelos desenvolvedores de sistemas para construir modelos.

Para que seja possível examinar um sistema complexo por diversas perspectivas, é sugerido pela UML que um sistema pode ser descrito por cinco visões distintas: visão de casos de uso, que descreve um conjunto de interações entre o sistema e agentes externos; visão de projeto, que enfatiza as características que dão suporte às funcionalidades externamente visíveis do sistema; visão de implementação, que abrange o gerenciamento de versões do sistema; visão de implantação, que corresponde à distribuição física do sistema e à conexão entre as partes; e visão de processo, que enfatiza as características de concorrência, sincronização e desempenho do sistema.

## 2.2 Git

O Git é uma ferramenta que possui como principais finalidades o controle de versão e a integração de código durante o desenvolvimento de sistemas. O Git permite obter qualquer versão existente do sistema por meio de um repositório, para isso, é necessário que cada alteração realizada no sistema seja rastreada e armazenada no repositório do Git.

Membros de equipes de desenvolvimento podem ter acesso às diferentes versões do sistema e às alterações realizadas em cada versão armazenada no repositório. No Git, são mostradas as linhas de código que foram adicionadas ou removidas em cada versão, tal funcionalidade permite que alterações realizadas por desenvolvedores diferentes sejam mescladas de maneira mais fácil e rápida.

Linus Torvalds, que estaria descontente com o BitKeeper (sistema de controle de versão utilizado no desenvolvimento do *kernel* do Linux), criou o Git em 2005. Alguns exemplos de sistemas de controle de versão mais antigos são CVS, ClearCase, SourceSafe e SVN (amplamente utilizado nas empresas).

Outra ferramenta que atua em conjunto com o Git é o GitHub. Criado em 2008, o GitHub é uma aplicação Web que possibilita a hospedagem de repositórios do Git, facilitando assim o acesso remoto aos repositórios. Além disso, o GitHub pode servir como uma rede social para desenvolvedores de software; cada usuário tem um perfil que exibe suas contribuições em projetos, seus repositórios e ainda permite seguir e ser seguido por outros usuários.

## 2.3 Java

A plataforma Java foi lançada em 1995 pela Sun. James Gosling, considerado pai do Java, liderou o desenvolvimento da plataforma que tinha o nome de *Oak* em 1991. Desde o início, a ideia do time liderado por James Gosling era criar um interpretador (máquina virtual) para os pequenos dispositivos a fim de facilitar a reescrita de código para os diferentes aparelhos eletrônicos, com a máquina virtual o software escrito passaria a ter portabilidade, ou seja, a máquina virtual passaria a lidar com as especificidades de cada sistema operacional permitindo que o mesmo código pudesse ser executado em diferentes aparelhos eletrônicos.

A partir do conceito de máquina virtual, foi propagado o slogan “*Write once, run anywhere*”. Na mesma época, o Java lançou os *Applets*, pequenas aplicações embutidas em páginas Web e executadas no navegador do cliente, que foram muito populares.

O Java é uma plataforma completa para desenvolvimento e execução de aplicações composta por três pilares: a máquina virtual Java (JVM), um grande conjunto de APIs e a linguagem de programação Java.

O JDK (*Java Development Kit*) é uma distribuição do Java destinada aos desenvolvedores, fornecendo ferramentas úteis para o desenvolvimento e o compilador. A outra distribuição do Java é o JRE (*Java Runtime Enviroment*) que possibilita apenas a execução de aplicações Java pelo usuário final.

## 2.4 MySQL

O MySQL é um sistema gerenciador de banco de dados relacional distribuído por meio de duas licenças que irão depender do tipo de uso da ferramenta: a primeira é software livre e a segunda é comercial.

Inicialmente, o MySQL foi projetado para trabalhar com aplicações de pequeno e médio porte. Todavia, os limites e capacidades do MySQL foram crescendo no decorrer das versões.

Além de ser altamente confiável, o MySQL é extremamente rápido por armazenar dados em tabelas no modo ISAM (código de baixo nível). Essas características fazem com que o MySQL seja uma ferramenta indicada para uso em aplicações de diversas áreas de negócio, independentemente do tamanho da aplicação.

# 3 ESPECIFICAÇÃO DOS REQUISITOS DO SISTEMA

## 3.1 O Produto

O sistema desenvolvido visa a automatização do das vendas de supermercados além de permitir o gerenciamento de caixas e estoques de um supermercado ou uma rede de supermercados. Os possíveis clientes do sistema incluem gestores de supermercados que buscam obter controle sobre as transações ocorridas durante o funcionamento do estabelecimento, diminuir os erros cometidos pelos funcionários e agilizar a realização de tarefas.

Os usuários do sistema são divididos em três grupos diferentes: gerentes, operadores de caixa e estoquistas. Os gerentes são responsáveis pela manutenção de cadastros e solicitação de relatórios gerenciais; os operadores de caixa realizarão vendas, abertura e fechamento de caixas; os estoquistas serão responsáveis por registrar entradas e possíveis retiradas de produtos do estoque.

## 3.2 Restrições do Produto

O sistema realiza vendas de produtos, informando ao operador de caixa os valores de cada produto e o valor total da venda. Ao final de cada venda realizada, o sistema atualiza o banco de dados sobre a saída de produtos do estoque.

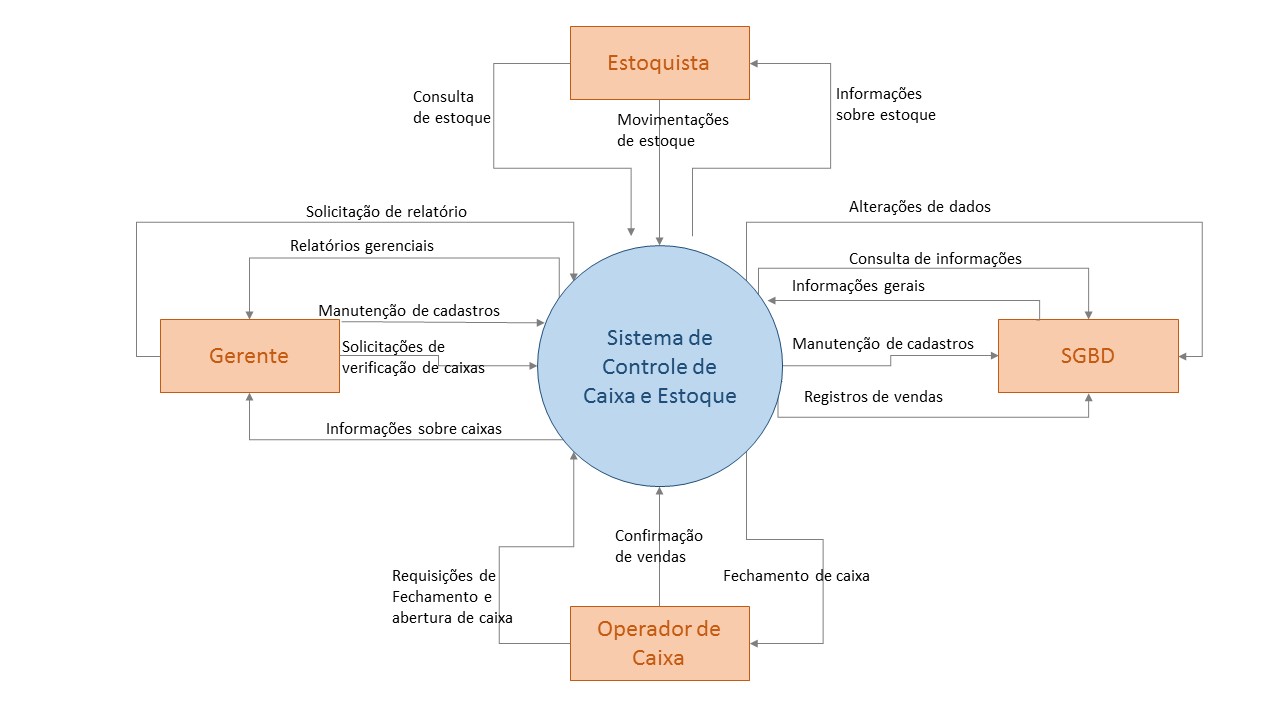
O sistema permite o registro de produtos em estoque, possibilitando controle sobre entrada, venda e saída de produtos do estoque por motivos variados como vencimento ou violação da embalagem de produtos.

O sistema também emite relatório de produtos mais e menos vendidos. Os relatórios podem ser emitidos de acordo com a busca selecionada pelo usuário, como busca por categoria ou por período, por exemplo. Além de relatórios sobre vendas, podem ser gerados relatórios sobre produtos em estoque, fornecendo informações como vencimento e quantidade de produtos disponível para venda.

# 3.3 Escopo do Projeto

O diagrama de contexto apresentado abaixo mostra a interação do sistema com agentes externos. No diagrama é possível visualizar o fluxo de informações movimentadas pelo sistema.

Figura 1: Diagrama de Contexto



Fonte: Figura Nossa

## 3.4 Requisitos Funcionais

Quadro 1: Requisitos Funcionais

|  |  |  |
| --- | --- | --- |
| **ID do Requisito** | **Descrição** | **Caso de Uso** |
| RF001 | O sistema deverá validar usuário e senha antes de permitir o acesso. | UC001 |
| RF002 | O sistema deverá permitir que o caixa seja iniciado a partir do número do caixa e do valor inicial inseridos pelo usuário. | UC012 |
| RF003 | O sistema deverá realizar o fechamento do caixa. | UC017 |
| RF004 | O sistema deverá informar o valor em caixa ao realizar o fechamento. | UC017 |
| RF005 | O sistema deverá realizar venda de produtos cadastrados. | UC016 |
| RF006 | O sistema deverá permitir que itens sejam adicionados à venda. | UC013 |
| RF007 | O sistema deverá permitir que itens sejam removidos da venda. | UC014 |
| RF008 | O sistema deverá permitir que a venda seja cancelada caso não tenha sido finalizada. | UC015 |
| RF009 | O sistema deverá registrar a entrada de produtos em estoque. | UC010 |
| RF010 | O sistema deverá permitir a consulta de produtos em estoque. | UC007 |
| RF011 | O sistema deverá permitir que produtos sejam retirados do estoque sem venda. | UC011 |
| RF012 | O sistema deverá registrar a saída de produtos do estoque sempre que os produtos forem retirados ou vendidos. | UC009 |
| RF013 | O sistema deverá manter o cadastro de produtos. | UC005 |
| RF014 | O sistema deve manter os preços dos produtos. | UC006 |
| RF015 | O sistema deverá manter cadastro de usuários. | UC003 |
| RF016 | O sistema deverá manter cadastro de fornecedores. | UC004 |
| RF017 | O sistema deverá permitir que a situação de cada caixa seja visualizada. | UC008 |
| RF018 | O sistema deverá emitir relatórios gerenciais. | UC019 |
| RF019 | O sistema deverá listar as opções de relatórios. | UC019 |
| RF020 | O sistema deverá atualizar o valor total da venda sempre que um item for adicionado ou removido. | UC013 |
| RF021 | O sistema deverá armazenar a(s) forma(s) de pagamento da venda. | UC016 |
| RF022 | O sistema deverá permitir que o pagamento seja registrado em uma ou mais formas de pagamento. | UC016 |
| RF023 | O sistema deverá permitir que as vendas da data atual sejam consultadas. | UC018 |

Fonte: Quadro Nosso.

## 3.5 Requisitos Não Funcionais

Quadro 2: Requisitos Não Funcionais

|  |  |  |
| --- | --- | --- |
| **ID do Requisito** | **Descrição** | **Categoria** |
| RNF001 | O sistema deverá possuir interfaces gráficas intuitivas. | Usabilidade |
| RNF002 | O sistema deverá ter a possibilidade de ser operado apenas com interação do teclado. | Usabilidade |
| RNF003 | O sistema deverá conter atalhos com teclas do teclado para as funções mais utilizadas. | Usabilidade |
| RNF004 | O sistema deve permitir acesso às funcionalidades de acordo com o tipo de responsabilidade do usuário. | Segurança |
| RNF005 | O sistema deverá ser executado em sistema operacional Windows7 ou superior. | Implementação |
| RNF006 | O tempo de conclusão de uma venda não deve ser superior a cinco (5) segundos. | Desempenho |
| RNF007 | O sistema deverá permitir que somente os usuários de nível gerencial possam cadastrar outros usuários. | Segurança |
| RNF008 | O sistema deverá ser desenvolvido em linguagem Java SE. | Implementação |
| RNF009 | O sistema não deverá permitir que dados sejam deletados exceto quando uma venda for cancelada. | Confiabilidade |
| RNF010 | O preço cobrado por cada produto durante a venda deverá ser o mesmo registrado do banco de dados. | Confiabilidade |

Fonte: Quadro Nosso.

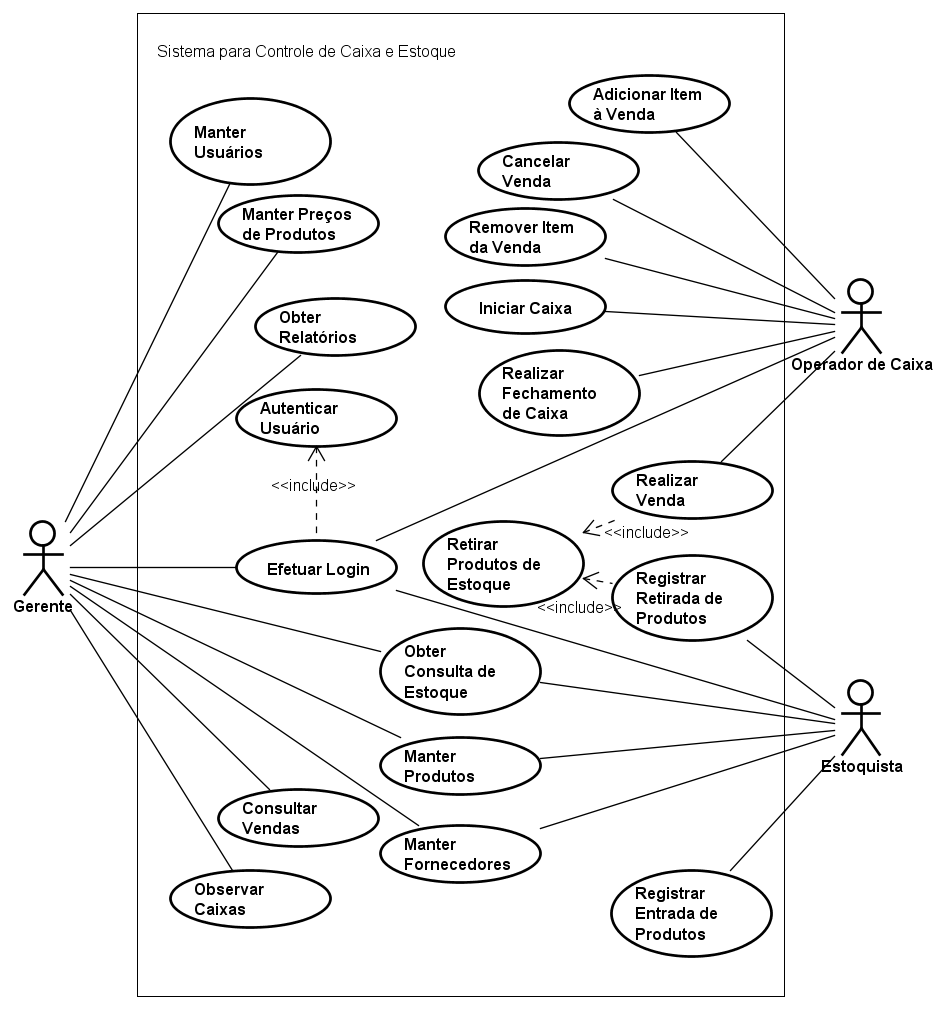
# 4 MODELO DE CASOS DE USO DO SISTEMA

Nesta seção será apresentado o modelo de casos de uso do sistema. Os casos de uso são elaborados a partir dos requisitos funcionais apresentados na seção anterior, por meio deles é possível ter uma visão menos abstrata a respeito das funcionalidades do sistema.

O diagrama de casos de uso apresenta de maneira resumida as responsabilidades de cada ator e as funcionalidades a serem implementadas no sistema.

## 4.1 Diagrama de Casos de Uso do Sistema

Figura 2: Diagrama de Casos de Uso



Fonte: Figura nossa.

## 4.2 Documentação dos Atores

### 4.2.1 Gerente

Quadro 3: Documentação do Ator Gerente

|  |
| --- |
| **Ator:** Gerente |
| **ID:** A1 |
| **Resumo:** O gerente é o ator responsável por manter cadastros de usuários, produtos e outras informações e solicita relatórios gerenciais. |
| **Responsabilidades:**  1. Manter cadastros de funcionários.  2. Manter cadastros de produtos.  3. Manter cadastros de fornecedores.  4. Solicitar emissão de relatórios. |
| **Ambiente Físico**  O gerente permanece dentro do estabelecimento em uma sala específica e circula dentro da unidade quando necessário. |
| **Número e Tipo**  Cada estabelecimento possui apenas um gerente. |

Fonte: Quadro nosso.

### 4.2.2 Estoquista

Quadro 4: Documentação do Ator Estoquista

|  |
| --- |
| **Ator:** Estoquista |
| **ID:** A2 |
| **Resumo:** O estoquista é o ator responsável por manter o estoque de produtos, ele registra a entrada de produtos em estoque e as retiradas de produtos por vencimento, violação da embalagem e outros motivos. |
| **Responsabilidades:**  1. Registrar entrada de produtos em estoque.  2. Registrar retirada de produtos.  3. Consultar produtos em estoque. |
| **Ambiente Físico**  O estoquista permanece no estoque organizando as mercadorias. |
| **Número e Tipo**  Cada estabelecimento pode conter vários estoquistas e todos executam as mesmas funções. |

Fonte: Quadro nosso.

### 4.2.3 Operador de Caixa

Quadro 5: Documentação do Ator Operador de Caixa

|  |
| --- |
| **Ator:** Operador de Caixa |
| **ID:** A3 |
| **Resumo:** O operador de caixa é o ator responsável pela realização de vendas de mercadorias. |
| **Responsabilidades:**  1. Realizar vendas.  2. Realizar fechamento de caixa.  3. Realizar abertura de caixa. |
| **Ambiente Físico**  O operador de caixa permanece no caixa para a realização das vendas. |
| **Número e Tipo**  Cada estabelecimento pode conter de um a muitos operadores de caixa. |

Fonte: Quadro nosso.

## 4.3 Documentação dos Casos de Uso

### 4.3.1 Caso de Uso Efetuar Login

Quadro 6: Caso de Uso Efetuar Login

|  |
| --- |
| **Caso de uso:** Efetuar Login |
| **ID:** UC001 |
| **Resumo:**  Processo pelo qual um usuário efetua seu login no sistema para utilizar as funções relacionadas com seu nível de usuário. |
| **Atores primários:**  1. Gerente  2. Estoquista  3. Operador de Caixa |
| **Atores secundários:**  Nenhum |
| **Pré-condições:**  1. O usuário deve possuir um usuário e senha cadastrados no sistema. |
| **Fluxo de evento principal:**  1. O usuário fornece suas informações de login (nome de usuário e senha).  2. O usuário clica no botão “Entrar”.  **{Executar Caso de Uso Autenticar Usuário}**  **{Verificar Nível de Acesso}**  3. O sistema insere os dados do início da sessão no banco de dados.  4. O sistema exibe o menu principal. |
| **Pós-condições:**  1. O usuário está logado no sistema.  2. O sistema está no menu principal aguardando a ação do usuário. |
| **Fluxo de eventos alternativo:**  Em **{Executar Caso de Uso Autenticar Usuário}**, se as informações fornecidas pelo usuário não forem compatíveis com as cadastradas no banco de dados:  1. O sistema exibe uma mensagem de erro informando ao usuário que as informações fornecidas estão incorretas.  2. O sistema aguarda a ação do usuário.  Em **{Verificar Nível de Acesso}**, se o usuário logado for do tipo “Operador(a) de Caixa”:  1. O sistema permitirá que o usuário tenha acesso somente às funcionalidades de venda, abertura e fechamento de caixa.  Em **{Verificar Nível de Acesso}**, se o usuário logado for do tipo “Estoquista”:  1. O sistema permitirá que o usuário tenha acesso somente às funcionalidades de estoque e consulta de produtos e preços. |

Fonte: Quadro nosso.

### 4.3.2 Caso de Uso Autenticar Usuário

Quadro 7: Caso de Uso Autenticar Usuário

|  |
| --- |
| **Caso de uso:** Autenticar Usuário |
| **ID:** UC002 |
| **Resumo:**  Processo pelo qual o sistema recebe os dados fornecidos pelo usuário e realiza a validação dos dados. |
| **Atores primários:**  1. Gerente  2. Estoquista  3. Operador de Caixa |
| **Atores secundários:**  Nenhum |
| **Pré-condições:**  1. O usuário deve possuir um usuário e senha cadastrados no sistema. |
| **Fluxo de evento principal:**  1. O sistema recebe os dados digitados pelo usuário.  2. O sistema pesquisa, no banco de dados, o usuário com o login que foi recebido.  **{Verificar Pesquisa}**  3. O sistema verifica se a senha digitada pelo usuário é igual à senha cadastrada no banco de dados.  **{Validar Senha}** |
| **Pós-condições:**  1. O usuário está logado no sistema.  2. O sistema está no menu principal aguardando a ação do usuário. |
| **Fluxo de eventos alternativo:**  Em **{Verificar Pesquisa}**, se a pesquisa de usuário for nula:  1. O sistema exibe uma mensagem informando que o usuário informado não existe.  2. O sistema aguarda a ação do usuário.  Em **{Validar Senha}**, se o resultado retornado for false:  1. O sistema exibe uma mensagem informando que o usuário informado não existe.  2. O sistema aguarda a ação do usuário. |

Fonte: Quadro nosso.

### 4.3.3 Caso de Uso Manter Usuários

Quadro 8: Caso de Uso Manter Usuários

|  |
| --- |
| **Caso de uso:** Manter Usuários |
| **ID:** UC003 |
| **Resumo:**  Processo pelo qual um gerente insere, altera e pesquisa cadastros de usuários. |
| **Atores primários:**  1. Gerente |
| **Atores secundários:**  1. Operador de caixa  2. Estoquista |
| **Pré-condições:**  1. O usuário deve estar logado no sistema.  2. O usuário deve possuir permissão para cadastrar usuários. |
| **Fluxo de evento principal:**  1. O usuário clica no menu “Cadastro”.  2. O usuário clica no item de menu “Cadastro de Usuários”.  3. O sistema exibe a tela de cadastro de usuários.  **{Verificar Ação}**  3.O sistema exibe a página principal ao usuário. |
| **Pós-condições:**  1. O usuário está logado no sistema.  2. O sistema está no menu principal aguardando a ação do usuário. |
| **Fluxo de eventos alternativo:**  Em **{Verificar Opção}**, se o usuário clicar no botão “Inserir”:  1. O sistema verifica se todos os campos foram preenchidos.  2. O sistema valida os dados digitados.  3. O sistema cadastra o usuário com as informações fornecidas no banco de dados.  4. O sistema exibe uma mensagem de confirmação ao usuário e verifica se o usuário deseja continuar na tela de cadastro.  5. Caso a resposta do usuário seja positiva o sistema retorna ao início do caso de uso.  Em **{Verificar Opção}**, se o usuário clicar no botão “Alterar”:  1. O sistema verifica se todos os campos foram preenchidos.  2. O sistema valida os dados digitados.  3. O sistema altera o cadastro com as informações fornecidas no banco de dados.  4. O sistema exibe uma mensagem de confirmação ao usuário e verifica se o usuário deseja continuar na tela de cadastro.  5. Caso a resposta do usuário seja positiva o sistema retorna ao início do caso de uso.  Em **{Verificar Opção}**, se o usuário clicar no botão “Pesquisar”:  1. O sistema realiza a pesquisa no banco de dados de acordo com o ID inserido.  2. Se o cadastro for localizado, o sistema exibe os dados na tela permite que sejam realizadas alterações. Se não, o sistema informa ao usuário que não foram encontrados registros de acordo com o ID digitado. |

Fonte: Quadro nosso.

### 4.3.4 Caso de Uso Manter Fornecedores

Quadro 9: Caso de Uso Manter Fornecedores

|  |
| --- |
| **Caso de uso:** Manter Fornecedores |
| **ID:** UC004 |
| **Resumo:**  Processo pelo qual um usuário insere, altera e pesquisa cadastros de fornecedores. |
| **Atores primários:**  1. Gerente  2. Estoquista |
| **Atores secundários:**  Nenhum |
| **Pré-condições:**  1. O usuário deve estar logado no sistema.  2. O usuário deve possuir permissão para cadastrar fornecedores. |
| **Fluxo de evento principal:**  1. O usuário clica no menu “Cadastro”.  2. O usuário clica no item de menu “Cadastro de Fornecedores”.  **{Verificar Opção}**  3.O sistema exibe a página principal ao usuário. |
| **Pós-condições:**  1. O usuário está logado no sistema.  2. O sistema está no menu principal aguardando a ação do usuário. |
| **Fluxo de eventos alternativo:**  Em **{Verificar Opção}**, se o usuário clicar no botão “Inserir”:  1. O sistema verifica se todos os campos foram preenchidos.  2. O sistema valida os dados digitados.  3. O sistema cadastra o fornecedor com as informações fornecidas no banco de dados.  4. O sistema exibe uma mensagem de confirmação ao usuário e verifica se o usuário deseja continuar na tela de cadastro.  5. Caso a resposta do usuário seja positiva o sistema retorna ao início do caso de uso.  Em **{Verificar Opção}**, se o usuário clicar no botão “Alterar”:  1. O sistema verifica se todos os campos foram preenchidos.  2. O sistema valida os dados digitados.  3. O sistema altera o cadastro com as informações fornecidas no banco de dados.  4. O sistema exibe uma mensagem de confirmação ao usuário e verifica se o usuário deseja continuar na tela de cadastro.  5. Caso a resposta do usuário seja positiva o sistema retorna ao início do caso de uso.  Em **{Verificar Opção}**, se o usuário clicar no botão “Pesquisar”:  1. O sistema realiza a pesquisa no banco de dados de acordo com o ID inserido.  2. Se o cadastro for localizado, o sistema exibe os dados na tela permite que sejam realizadas alterações. Se não, o sistema informa ao usuário que não foram encontrados registros de acordo com o ID digitado. |

Fonte: Quadro nosso.

### 4.3.5 Caso de Uso Manter Produtos

Quadro 10: Caso de Uso Manter Produtos

|  |
| --- |
| **Caso de uso:** Manter Produtos |
| **ID:** UC005 |
| **Resumo:**  Processo pelo qual um usuário insere, altera e pesquisa cadastros de produtos. |
| **Atores primários:**  1. Gerente  2. Estoquista |
| **Atores secundários:**  Nenhum |
| **Pré-condições:**  1. O usuário deve estar logado no sistema.  2. O usuário deve possuir permissão para cadastrar produtos. |
| **Fluxo de evento principal:**  1. O usuário clica no menu “Cadastro”.  2. O usuário clica no item de menu “Cadastro de Produtos”.  **{Verificar Opção}**  3.O sistema exibe a página principal ao usuário. |
| **Pós-condições:**  1. O usuário está logado no sistema.  2. O sistema está no menu principal aguardando a ação do usuário. |
| **Fluxo de eventos alternativo:**  Em **{Verificar Opção}**, se o usuário clicar no botão “Inserir”:  1. O sistema verifica se todos os campos foram preenchidos.  2. O sistema valida os dados digitados.  3. O sistema cadastra o fornecedor com as informações fornecidas no banco de dados.  4. O sistema exibe uma mensagem de confirmação ao usuário e verifica se o usuário deseja continuar na tela de cadastro.  5. Caso a resposta do usuário seja positiva o sistema retorna ao início do caso de uso.  Em **{Verificar Opção}**, se o usuário clicar no botão “Alterar”:  1. O sistema verifica se todos os campos foram preenchidos.  2. O sistema valida os dados digitados.  3. O sistema altera o cadastro com as informações fornecidas no banco de dados.  4. O sistema exibe uma mensagem de confirmação ao usuário e verifica se o usuário deseja continuar na tela de cadastro.  5. Caso a resposta do usuário seja positiva o sistema retorna ao início do caso de uso.  Em **{Verificar Opção}**, se o usuário clicar no botão “Pesquisar”:  1. O sistema realiza a pesquisa no banco de dados de acordo com o ID inserido.  2. Se o cadastro for localizado, o sistema exibe os dados na tela permite que sejam realizadas alterações. Se não, o sistema informa ao usuário que não foram encontrados registros de acordo com o ID digitado. |

Fonte: Quadro nosso.

### 4.3.6 Caso de Uso Manter Preços de Produtos

Quadro 11: Caso de Uso Manter Preços de Produtos

|  |
| --- |
| **Caso de uso:** Manter Preços de Produtos |
| **ID:** UC006 |
| **Resumo:**  Processo pelo qual um usuário define e altera os preços de produtos. |
| **Atores primários:**  1. Gerente  2. Estoquista |
| **Atores secundários:**  Nenhum |
| **Pré-condições:**  1. O usuário deve estar logado no sistema.  2. O usuário deve possuir permissão para cadastrar preços. |
| **Fluxo de evento principal:**  1. O usuário clica no menu “Cadastro”.  2. O usuário clica no item de menu “Cadastro de Preços”.  3. O usuário seleciona o produto a ter o preço definido.  4. O usuário digita o preço.  5. O usuário clica no botão “Gravar”.  **{Validar Preço}**  6. O sistema grava o preço no banco de dados.  7. O sistema exibe a página principal ao usuário. |
| **Pós-condições:**  1. O usuário está logado no sistema.  2. O sistema está no menu principal aguardando a ação do usuário. |
| **Fluxo de eventos alternativo:**  Em **{Validar Preço}**, se a o campo “Novo preço” estiver vazio ou com o valor igual a zero:  1. O sistema exibe uma caixa de diálogo informando que é necessário preencher o campo.  2. O sistema volta ao início do caso de uso. |

Fonte: Quadro nosso.

### 4.3.7 Caso de Uso Obter Consulta de Estoque

Quadro 12: Caso de Uso Obter Consulta de Estoque

|  |
| --- |
| **Caso de uso:** Obter consulta de Estoque |
| **ID:** UC007 |
| **Resumo:**  Processo pelo qual um usuário tem acesso aos dados dos produtos em estoque: quantidade disponível, valor total, descrição do produto, etc. |
| **Atores primários:**  1. Gerente  2. Estoquista |
| **Atores secundários:**  Nenhum |
| **Pré-condições:**  1. O usuário deve estar logado no sistema.  2. O usuário deve possuir permissão para consultar o estoque. |
| **Fluxo de evento principal:**  1. O usuário clica no menu “Estoque”.  2. O usuário clica no item de menu “Consulta de Estoque”.  3.O sistema exibe, em uma tabela, a lista de todos os produtos em estoque seguidos da quantidade e o valor total. |
| **Pós-condições:**  1. O usuário está logado no sistema.  2. O sistema está no menu principal aguardando a ação do usuário. |

Fonte: Quadro nosso.

### 4.3.8 Caso de Uso Observar Caixas

Quadro 13: Caso de Uso Observar Caixas

|  |
| --- |
| **Caso de uso:** Observar Caixas |
| **ID:** UC008 |
| **Resumo:**  Processo pelo qual um usuário tem acesso às informações referentes aos caixas abertos: número do caixa, operador de caixa, valor em caixa, etc. |
| **Atores primários:**  1. Gerente |
| **Atores secundários:**  Nenhum |
| **Pré-condições:**  1. O usuário deve estar logado no sistema.  2. O usuário deve possuir permissão de acesso às informações fornecidas. |
| **Fluxo de evento principal:**  1. O usuário clica no menu “Caixas”.  2. O usuário clica no item de menu “Observar Caixas”.  3. O sistema pesquisa os caixas que estão abertos.  4. O sistema exibe uma tabela com as informações de todos os caixas que estão abertos. |
| **Pós-condições:**  1. O usuário está logado no sistema.  2. O sistema está no menu principal aguardando a ação do usuário. |

Fonte: Quadro nosso.

### 4.3.9 Caso de Uso Retirar Produtos de Estoque

Quadro 14: Caso de Uso Retirar Produtos de Estoque

|  |
| --- |
| **Caso de uso:** Retirar Produtos de Estoque |
| **ID:** UC009 |
| **Resumo:**  Processo em que o sistema recebe os dados de produtos que foram vendidos ou que foram retirados de estoque por diferentes motivos e altera as informações do estoque no banco de dados. |
| **Atores primários:**  Nenhum |
| **Atores secundários:**  1. Operador de Caixa  2. Estoquista |
| **Pré-condições:**  1. Uma venda ou uma transação de retirada de produto de estoque deve ter ocorrido.  2. Deve haver produtos suficientes para retirar do estoque. |
| **Fluxo de evento principal:**  1. O sistema recebe os dados do produto e da quantidade que foi retirada de estoque.  2. O sistema pesquisa o item referente ao produto que foi retirado de estoque.  3. O sistema altera a quantidade e o valor total de estoque de acordo com a quantidade do produto retirado. |
| **Pós-condições:**  1. O estoque está atualizado. |

Fonte: Quadro nosso.

### 4.3.10 Caso de Uso Registrar Entrada de Produtos

Quadro 15: Caso de Uso Registrar Entrada de Produtos

|  |
| --- |
| **Caso de uso:** Registrar Entrada de Produtos |
| **ID:** UC010 |
| **Resumo:**  Processo pelo qual um usuário insere os dados de produtos que foram adicionados ao estoque. |
| **Atores primários:**  1. Gerente  2. Estoquista |
| **Atores secundários:**  Nenhum |
| **Pré-condições:**  1. O usuário deve estar logado no sistema.  2. O usuário deve possuir permissão para registrar a entrada de produtos.  3. Deve haver produtos cadastrados.  4. Deve haver fornecedores cadastrados. |
| **Fluxo de evento principal:**  1. O usuário clica no menu “Estoque”.  2. O usuário clica no item de menu “Registrar Entrada de Produtos”.  3.O sistema exibe o formulário para que o usuário registre as informações.  4. O usuário seleciona o produto.  5. O usuário seleciona o fornecedor.  6. O usuário insere a quantidade e o valor unitário de compra.  7. O usuário clica no botão “Gravar”.  8. O sistema altera o banco de dados de acordo com a inserção de produtos. |
| **Pós-condições:**  1. O usuário está logado no sistema.  2. O sistema está no menu principal aguardando a ação do usuário. |

Fonte: Quadro nosso.

### 4.3.11 Caso de Uso Registrar Retirada de Produtos

Quadro 16: Caso de Uso Registrar Retirada de Produtos

|  |
| --- |
| **Caso de uso:** Registrar Retirada de Produtos |
| **ID:** UC011 |
| **Resumo:**  Processo em que o usuário registra a retirada de produtos de estoque por motivos diversos, sem que haja venda. |
| **Atores primários:**  1. Gerente  2. Estoquista |
| **Atores secundários:**  Nenhum |
| **Pré-condições:**  1. O usuário deve estar logado no sistema.  2. O usuário deve possuir permissão retirar produtos de estoque.  3. Deve haver quantidade suficiente em estoque do produto a ser retirado. |
| **Fluxo de evento principal:**  1. O usuário clica no menu “Estoque”.  2. O usuário clica no item de menu “Registrar Saída de Produtos”.  3. O sistema exibe um formulário para que o usuário insira as informações.  4. O usuário seleciona o produto a ser retirado.  5. O usuário informa a quantidade e o motivo da retirada do produto.  6. O usuário clica no botão “Gravar”.  **{Verificar Estoque}**  **{Executar Caso de Uso Retirar Produtos de Estoque}** |
| **Pós-condições:**  1. O usuário está logado no sistema.  2. O sistema está no menu principal aguardando a ação do usuário. |
| **Fluxo de eventos alternativo:**  Em **{Verificar Estoque}**, se a quantidade em estoque for insuficiente:  1. O sistema exibe uma mensagem de erro informando que a quantidade é insuficiente.  2. O sistema volta ao início do caso de uso. |

Fonte: Quadro nosso.

### 4.3.12 Caso de Uso Iniciar Caixa

Quadro 17: Caso de Uso Iniciar Caixa

|  |
| --- |
| **Caso de uso:** Iniciar Caixa |
| **ID:** UC012 |
| **Resumo:**  Processo em que o usuário realiza a abertura de caixa. |
| **Atores primários:**  1. Operador de Caixa |
| **Atores secundários:**  1. Gerente |
| **Pré-condições:**  1. O usuário deve estar logado no sistema.  2. O usuário deve possuir permissão para realizar abertura de caixa. |
| **Fluxo de evento principal:**  1. O usuário clica no menu “Venda”.  2. O usuário clica no item de menu “Abrir Caixa”.  3. O sistema exibe um formulário com a lista dos caixas fechados.  4. O usuário seleciona o caixa a ser aberto.  5. O usuário digita o valor inicial em caixa.  6. O usuário clica no botão “Abrir caixa”.  **{Verificar Valor Inserido}**  7. O sistema insere os dados referentes a abertura do caixa no banco de dados.  8. O sistema informa que o caixa foi aberto e aguarda a ação do usuário. |
| **Pós-condições:**  1. O usuário está logado no sistema.  2. O sistema está no menu principal aguardando a ação do usuário. |
| **Fluxo de eventos alternativo:**  Em **{Verificar Valor Inserido}**, se o campo “Valor inicial” estiver vazio:  1. O sistema exibe uma caixa de diálogo para verificar se o usuário deseja iniciar o caixa com o valor de R$ 0,00.  2. Caso o usuário responda positivamente o sistema inicia o caixa normalmente. Caso não, o sistema retorna ao início do caso de uso. |

Fonte: Quadro nosso.

### 4.3.13 Caso de Uso Adicionar Item à Venda

Quadro 18: Caso de Uso Adicionar Item à Venda

|  |
| --- |
| **Caso de uso:** Adicionar Item à Venda |
| **ID:** UC013 |
| **Resumo:**  Processo em que um item é adicionado à venda. |
| **Atores primários:**  1. Operador de Caixa |
| **Atores secundários:**  1. Gerente |
| **Pré-condições:**  1. O usuário deve estar logado no sistema.  2. O usuário deve possuir permissão para realizar venda. |
| **Fluxo de evento principal:**  1. No formulário de venda, o usuário digita o código de barras do produto e tecla “Enter”.  **{Pesquisar Produto}**  **{Pesquisar Preço do Produto}**  2. O sistema preenche os campos do formulário com os dados do produto: descrição, valor por unidade ou quilo, etc.  3. O usuário digita a quantidade.  **{Verificar Quantidade}**  4. O sistema calcula o subtotal do item.  5. O sistema adiciona o item à lista de itens da venda.  6. O sistema atualiza o valor total da venda. |
| **Pós-condições:**  1. O usuário está logado no sistema.  2. O sistema exibe a tela de venda e aguarda a ação do usuário. |
| **Fluxo de eventos alternativo:**  Em **{Pesquisar Produto}**, se o produto não for localizado no banco de dados:  1. O sistema exibe uma caixa de diálogo e informa ao usuário que o produto não foi localizado.  2. O sistema retorna ao início do caso de uso.  Em **{Pesquisar Preço do Produto}**, se o preço do produto não for localizado no banco de dados:  1. O sistema exibe uma caixa de diálogo e informa que não é possível vender o produto pois o preço ainda não foi cadastrado.  2. O sistema retorna ao início do caso de uso.  Em **{Verificar Quantidade}**, se a quantidade disponível em estoque for menor do que a quantidade digitada pelo usuário:  1. O sistema exibe uma caixa de diálogo informando que não há quantidade suficiente do produto em estoque.  2. O sistema retorna ao início do caso de uso. |

Fonte: Quadro nosso.

### 4.3.14 Caso de Uso Remover Item da Venda

Quadro 19: Caso de Uso Remover Item da Venda

|  |
| --- |
| **Caso de uso:** Remover Item da Venda |
| **ID:** UC014 |
| **Resumo:**  Processo em que um item é removido da venda. |
| **Atores primários:**  1. Operador de Caixa |
| **Atores secundários:**  1. Gerente |
| **Pré-condições:**  1. O usuário deve estar logado no sistema.  2. O usuário deve possuir permissão para realizar venda. |
| **Fluxo de evento principal:**  1. No formulário de venda, o usuário seleciona o item a ser removido na tabela de itens da venda.  **{Verificar Item Removido}**  2. O sistema define o item como cancelado.  3. O sistema altera a coluna “Cancelado” da tabela de itens e atribui o valor “Sim”.  4. O sistema atualiza o valor total da venda. |
| **Pós-condições:**  1. O usuário está logado no sistema.  2. O sistema exibe a tela de venda e aguarda a ação do usuário. |
| **Fluxo de eventos alternativo:**  Em **{Verificar Item Removido}**, se o item já estiver sido removido previamente:  1. O sistema retorna ao início do caso de uso. |

Fonte: Quadro nosso.

### 4.3.15 Caso de Uso Cancelar Venda

Quadro 20: Caso de Uso Cancelar Venda

|  |
| --- |
| **Caso de uso:** Cancelar Venda |
| **ID:** UC015 |
| **Resumo:**  Processo em que uma venda é cancelada. |
| **Atores primários:**  1. Operador de Caixa |
| **Atores secundários:**  1. Gerente |
| **Pré-condições:**  1. O usuário deve estar logado no sistema.  2. O usuário deve possuir permissão para realizar venda. |
| **Fluxo de evento principal:**  1. O usuário tecla a tecla de atalho referente a “Cancelar Venda”.  2. O sistema exibe uma caixa de diálogo para confirmar o cancelamento.  **{Confirmar Cancelamento}**  3. O sistema exclui as informações da venda no banco de dados. |
| **Pós-condições:**  1. O usuário está logado no sistema.  2. O sistema exibe a tela de venda e aguarda a ação do usuário. |
| **Fluxo de eventos alternativo:**  Em **{Confirmar Cancelamento}**, se o usuário responder negativamente ao diálogo:  1. O sistema interrompe a execução do caso de uso. |

Fonte: Quadro nosso.

### 4.3.16 Caso de Uso Realizar Venda

Quadro 21: Caso de Uso Realizar Venda

|  |
| --- |
| **Caso de uso:** Realizar Venda |
| **ID:** UC016 |
| **Resumo:**  Processo em que uma venda é finalizada. |
| **Atores primários:**  1. Operador de Caixa |
| **Atores secundários:**  1. Gerente |
| **Pré-condições:**  1. O usuário deve estar logado no sistema.  2. O usuário deve possuir permissão para realizar venda. |
| **Fluxo de evento principal:**  1. O usuário tecla a tecla de atalho referente a “Pagamento”.  2. O sistema exibe um formulário para que o usuário registre o pagamento.  3. O usuário insere o valor pago no campo referente à cada forma de pagamento.  4. O sistema atualiza o valor pago e o valor de troco.  5. O usuário pressiona a tecla de atalho referente a “Confirmar pagamento”.  **{Verificar Valor a Pagar}**  **{Verificar Valor de Troco}**  6. Para cada item da venda: **{Executar Caso de Uso Retirar Produtos de Estoque}**  7. O sistema define a venda como finalizada no banco de dados.  8. O sistema registra o pagamento da venda no banco de dados. |
| **Pós-condições:**  1. O usuário está logado no sistema.  2. O sistema exibe a tela de venda e aguarda a ação do usuário. |
| **Fluxo de eventos alternativo:**  Em **{Verificar Valor a Pagar}**, se o valor pago for menor do que o valor total da venda:  1. O sistema exibe uma caixa de diálogo informando que o valor pago é inferior ao valor total da venda.  2. O sistema retorna ao início do caso de uso.  Em **{Verificar Valor de Troco}**, se o valor de troco for maior do que o valor pago em dinheiro:  1. O sistema exibe uma caixa de diálogo informando que o valor de troco não pode ser maior do que o valor pago em dinheiro.  2. O sistema retorna ao início do caso de uso. |

Fonte: Quadro nosso.

### 4.3.17 Caso de Uso Realizar Fechamento de Caixa

Quadro 22: Caso de Uso Realizar Fechamento de Caixa

|  |
| --- |
| **Caso de uso:** Realizar Fechamento de Caixa |
| **ID:** UC017 |
| **Resumo:**  Processo em que o caixa é finalizado. |
| **Atores primários:**  1. Operador de Caixa |
| **Atores secundários:**  1. Gerente |
| **Pré-condições:**  1. O usuário deve estar logado no sistema.  2. O usuário deve possuir permissão para realizar venda.  3. O usuário deve ter aberto um caixa previamente. |
| **Fluxo de evento principal:**  1. O usuário clica no menu “Venda”.  2. O usuário clica no item de menu “Finalizar Caixa”.  3. O sistema calcula o valor em caixa para dinheiro, débito, crédito, cheque, voucher e outros de acordo com as vendas realizadas e o valor de início de caixa.  4. O sistema exibe o formulário de fechamento de caixa com as informações referentes aos valores em dinheiro, débito, crédito, cheque, voucher e outros que estiverem em caixa.  5. O usuário pressiona a tecla de atalho referente a “Finalizar Caixa”.  6. O sistema atualiza o banco de dados e finaliza a sessão. |
| **Pós-condições:**  1. A sessão do usuário deve estar finalizada. |

Fonte: Quadro nosso.

### 4.3.18 Caso de Uso Consultar Vendas

Quadro 23: Caso de Uso Consultar Vendas

|  |
| --- |
| **Caso de uso:** Consultar Vendas |
| **ID:** UC018 |
| **Resumo:**  Processo em que um usuário obtém uma lista com as informações das vendas realizadas na data atual. |
| **Atores primários:**  1. Gerente |
| **Atores secundários:**  Nenhum |
| **Pré-condições:**  1. O usuário deve estar logado no sistema.  2. O usuário deve possuir permissão para consultar vendas. |
| **Fluxo de evento principal:**  1. O usuário clica no menu “Consulta”.  2. O usuário clica no item de menu “Consulta de Vendas”.  3. O sistema pesquisa no banco de dados e obtém as informações de todas as vendas realizadas na data atual.  4. O sistema exibe uma lista com as informações das vendas. |
| **Pós-condições:**  1. O usuário permanece logado. |

Fonte: Quadro nosso.

### 4.3.19 Caso de Uso Obter Relatórios

Quadro 24: Caso de Uso Obter Relatórios

|  |
| --- |
| **Caso de uso:** Obter Relatórios |
| **ID:** UC019 |
| **Resumo:**  Processo em que um usuário de nível gerencial solicita relatórios para obter informações gerais. |
| **Atores primários:**  1. Gerente |
| **Atores secundários:**  Nenhum |
| **Pré-condições:**  1. O usuário deve estar logado no sistema.  2. O usuário deve possuir permissão para obter relatórios. |
| **Fluxo de evento principal:**  1. O usuário clica no menu “Relatórios”.  2. O usuário escolhe uma opção entre os relatórios listados.  3. O sistema, a partir do relatório solicitado pelo usuário, gera o relatório e exibe. |
| **Pós-condições:**  1. O usuário deve permanecer logado. |

Fonte: Quadro nosso.

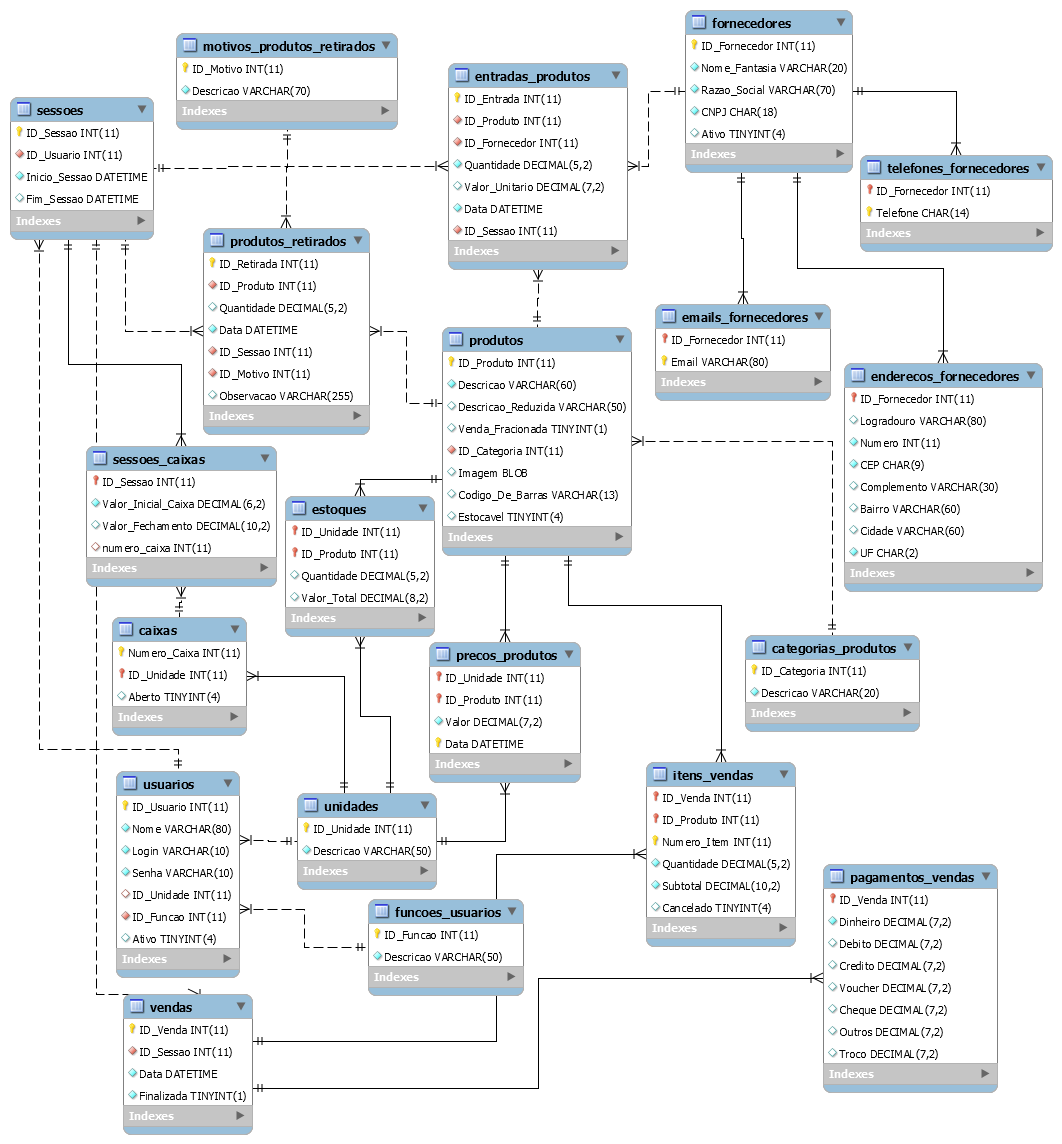
# 5 MODELOS DE ANÁLISE E PROJETO DO SISTEMA

Nesta sessão, utiliza-se o requisitos e casos de uso já apresentados para elaborar a análise orientada a objetos. Os diagramas mostrados a seguir têm a finalidade de apresentar diferentes visões do projeto.

## 5.1 Diagrama Entidade-Relacionamento

O Diagrama Entidade-Relacionamento (DER) ilustra a visão do banco de dados que o sistema utiliza. Neste diagrama é possível verificar a maneira em que as informações são armazenadas no banco de dados.

Figura 3: Diagrama Entidade-Relacionamento



Fonte: Figura nossa.

### 5.1.1 Dicionário de Dados

Quadro 25: Tabela Unidades

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabela: **unidades** | | | | | |
| **Campo** | **Classe** | **Tipo** | **Tamanho** | **Restrições** | **Observações** |
| ID\_Unidade | Determinante (Chave Primária) | INT | 11 | Não nulo e maior que zero | Campo auto-incremento |
| Descricao | Simples | VARCHAR | 255 | Não nulo |  |

Fonte: Quadro nosso.

Quadro 26: Tabela Fornecedores

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabela: **fornecedores** | | | | | |
| **Campo** | **Classe** | **Tipo** | **Tamanho** | **Restrições** | **Observações** |
| ID\_Fornecedor | Determinante (Chave Primária) | INT | 11 | Não nulo e maior que zero | Campo auto-incremento |
| Nome\_Fantasia | Simples | VARCHAR | 80 | Não nulo |  |
| Razao\_Social | Simples | VARCHAR | 120 | Não nulo |  |
| CNPJ | Simples | CHAR | 18 | Não nulo |  |
| Ativo | Simples | TINYINT | 4 | Não nulo | Default 1 |

Fonte: Quadro nosso.

Quadro 27: Tabela Endereços Fornecedores

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabela: **enderecos\_fornecedores** | | | | | |
| **Campo** | **Classe** | **Tipo** | **Tamanho** | **Restrições** | **Observações** |
| ID\_Fornecedor | Determinante (Chave Primária e Chave Estrangeira) | INT | 11 | Não nulo e maior que zero | Campo auto-incremento |
| Logradouro | Simples | VARCHAR | 80 | Não nulo |  |
| Numero | Simples | INT | 11 | Não nulo |  |
| CEP | Simples | CHAR | 9 | Não nulo |  |
| Complemento | Simples | VARCHAR | 30 |  |  |
| Bairro | Simples | VARCHAR | 60 | Não nulo |  |
| Cidade | Simples | VARCHAR | 60 | Não nulo |  |
| UF | Simples | VARCHAR | 255 | Não nulo |  |

Fonte: Quadro nosso.

Quadro 28: Tabela E-mails Fornecedores

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabela: **emails\_fornecedores** | | | | | |
| **Campo** | **Classe** | **Tipo** | **Tamanho** | **Restrições** | **Observações** |
| ID\_Fornecedor | Determinante (Chave Primária e Chave Estrangeira) | INT | 11 | Não nulo e maior que zero | Campo auto-incremento |
| Email | Simples | VARCHAR | 80 | Não nulo |  |

Fonte: Quadro nosso.

Quadro 29: Tabela Telefones Fornecedores

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabela: **telefones\_fornecedores** | | | | | |
| **Campo** | **Classe** | **Tipo** | **Tamanho** | **Restrições** | **Observações** |
| ID\_Fornecedor | Determinante (Chave Primária e Chave Estrangeira) | INT | 11 | Não nulo e maior que zero | Campo auto-incremento |
| Telefone | Simples | CHAR | 14 | Não nulo |  |

Fonte: Quadro nosso.

Quadro 30: Tabela Funções Usuários

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabela: **funcoes\_usuarios** | | | | | |
| **Campo** | **Classe** | **Tipo** | **Tamanho** | **Restrições** | **Observações** |
| ID\_Funcao | Determinante (Chave Primária) | INT | 11 | Não nulo e maior que zero | Campo auto-incremento |
| Descricao | Simples | VARCHAR | 50 | Não nulo |  |

Fonte: Quadro nosso.

Quadro 31: Tabela Categorias Produtos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabela: **categorias\_produtos** | | | | | |
| **Campo** | **Classe** | **Tipo** | **Tamanho** | **Restrições** | **Observações** |
| ID\_Categoria | Determinante (Chave Primária) | INT | 11 | Não nulo e maior que zero | Campo auto-incremento |
| Descricao | Simples | VARCHAR | 50 | Não nulo |  |

Fonte: Quadro nosso.

Quadro 32: Tabela Motivos Produtos Retirados

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabela: **motivos\_produtos\_retirados** | | | | | |
| **Campo** | **Classe** | **Tipo** | **Tamanho** | **Restrições** | **Observações** |
| ID\_Motivo | Determinante (Chave Primária) | INT | 11 | Não nulo e maior que zero | Campo auto-incremento |
| Descricao | Simples | VARCHAR | 70 | Não nulo |  |

Fonte: Quadro nosso.

Quadro 33: Tabela Usuários

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabela: **usuarios** | | | | | |
| **Campo** | **Classe** | **Tipo** | **Tamanho** | **Restrições** | **Observações** |
| ID\_Usuario | Determinante (Chave Primária) | INT | 11 | Não nulo e maior que zero | Campo auto-incremento |
| Nome | Simples | VARCHAR | 80 | Não nulo |  |
| Login | Simples | VARCHAR | 10 | Não nulo |  |
| Senha | Simples | VARCHAR | 10 | Não nulo |  |
| ID\_Unidade | Determinante (Chave Estrangeira) | INT | 11 | Não nulo |  |
| ID\_Funcao | Determinante (Chave Estrangeira) | INT | 11 | Não nulo |  |
| Ativo | Simples | TINYINT | 4 | Não nulo | Default 1 |

Fonte: Quadro nosso.

Quadro 34: Tabela Caixas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabela: **caixas** | | | | | |
| **Campo** | **Classe** | **Tipo** | **Tamanho** | **Restrições** | **Observações** |
| Numero\_Caixa | Determinante (Chave Primária) | INT | 11 | Não nulo e maior que zero |  |
| ID\_Unidade | Determinante (Chave Primária e Chave Estrangeira) | INT | 11 | Não nulo e maior que zero |  |
| Aberto | Simples | TINYINT | 4 | Não nulo | Default 0 |

Fonte: Quadro nosso.

Quadro 35: Tabela Sessões

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabela: **sessoes** | | | | | |
| **Campo** | **Classe** | **Tipo** | **Tamanho** | **Restrições** | **Observações** |
| ID\_Sessao | Determinante (Chave Primária) | INT | 11 | Não nulo e maior que zero | Campo auto-incremento |
| ID\_Usuario | Determinante (Chave Estrangeira) | INT | 11 | Não nulo e maior que zero |  |
| Inicio\_Sessao | Simples | DATETIME |  |  |  |
| Fim\_Sessao | Simples | DATETIME |  |  |  |

Fonte: Quadro nosso.

Quadro 36: Tabela Sessões Caixas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabela: **sessões\_caixas** | | | | | |
| **Campo** | **Classe** | **Tipo** | **Tamanho** | **Restrições** | **Observações** |
| ID\_Sessao | Determinante (Chave Primária e Chave Estrangeira) | INT | 11 | Não nulo e maior que zero |  |
| Valor\_Inicial\_Caixa | Simples | DECIMAL | 6,2 | Não nulo e maior ou igual a zero |  |
| Valor\_Fechamento | Simples | DECIMAL | 10,2 |  |  |
| Numero\_Caixa | Determinante (Chave Estrangeira) | INT | 11 | Não nulo |  |

Fonte: Quadro nosso.

Quadro 37: Tabela Produtos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabela: **produtos** | | | | | |
| **Campo** | **Classe** | **Tipo** | **Tamanho** | **Restrições** | **Observações** |
| ID\_Produto | Determinante (Chave Primária) | INT | 11 | Não nulo e maior que zero | Campo auto-incremento |
| Descricao | Simples | VARCHAR | 150 | Não nulo |  |
| Descricao\_Reduzida | Simples | VARCHAR | 50 | Não nulo |  |
| Venda\_Fracionada | Simples | TINYINT | 4 | Não nulo | Default 0 |
| ID\_Categoria | Determinante (Chave Estrangeira) | INT | 11 | Não nulo |  |
| Imagem | Simples | BLOB |  |  |  |
| Codigo\_De\_Barras | Simples | VARCHAR | 13 | Não nulo |  |
| Estocável | Simples | TINYINT | 4 | Não nulo | Default 1 |

Fonte: Quadro nosso.

Quadro 38: Tabela Preços de Produtos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabela: **preços\_produtos** | | | | | |
| **Campo** | **Classe** | **Tipo** | **Tamanho** | **Restrições** | **Observações** |
| ID\_Unidade | Determinante (Chave Primária e Chave Estrangeira) | INT | 11 | Não nulo e maior que zero |  |
| ID\_Produto | Determinante (Chave Primária e Chave Estrangeira) | INT | 11 | Não nulo e maior que zero |  |
| Valor | Simples | DECIMAL | 7,2 | Não nulo e maior que zero |  |
| Data | Simples | DATETIME |  | Não nulo |  |

Fonte: Quadro nosso.

Quadro 39: Tabela Estoques

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabela: **estoques** | | | | | |
| **Campo** | **Classe** | **Tipo** | **Tamanho** | **Restrições** | **Observações** |
| ID\_Unidade | Determinante (Chave Primária e Chave Estrangeira) | INT | 11 | Não nulo e maior que zero |  |
| ID\_Produto | Determinante (Chave Primária e Chave Estrangeira) | INT | 11 | Não nulo e maior que zero |  |
| Quantidade | Simples | DECIMAL | 5,2 | Não nulo e maior ou igual zero |  |
| Valor\_Total | Simples | DECIMAL | 8,2 | Não nulo e maior ou igual zero |  |

Fonte: Quadro nosso.

Quadro 40: Tabela Vendas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabela: **vendas** | | | | | |
| **Campo** | **Classe** | **Tipo** | **Tamanho** | **Restrições** | **Observações** |
| ID\_Venda | Determinante (Chave Primária) | INT | 11 | Não nulo e maior que zero | Campo auto-incremento |
| ID\_Sessao | Determinante (Chave Primária e Chave Estrangeira) | INT | 11 | Não nulo e maior que zero |  |
| Data | Simples | DATETIME |  | Não nulo |  |
| Finalizada | Simples | TINYINT | 4 | Não nulo | Default 0 |

Fonte: Quadro nosso.

Quadro 41: Tabela Itens de Vendas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabela: **itens\_vendas** | | | | | |
| **Campo** | **Classe** | **Tipo** | **Tamanho** | **Restrições** | **Observações** |
| ID\_Venda | Determinante (Chave Primária e Chave Estrangeira) | INT | 11 | Não nulo e maior que zero |  |
| ID\_Produto | Determinante (Chave Primária e Chave Estrangeira) | INT | 11 | Não nulo e maior que zero |  |
| Numero\_Item | Simples | INT | 11 | Não nulo e maior que zero |  |
| Quantidade | Simples | DECIMAL | 5,2 | Não nulo e maior que zero |  |
| Subtotal | Simples | DECIMAL | 10,2 | Não nulo e maior que zero |  |
| Cancelado | Simples | TINYINT | 4 | Não nulo | Default 0 |

Fonte: Quadro nosso.

Quadro 42: Tabela Pagamentos de Vendas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabela: **pagamentos\_vendas** | | | | | |
| **Campo** | **Classe** | **Tipo** | **Tamanho** | **Restrições** | **Observações** |
| ID\_Venda | Determinante (Chave Primária e Chave Estrangeira) | INT | 11 | Não nulo e maior que zero |  |
| Dinheiro | Simples | DECIMAL | 7,2 |  | Default 0.00 |
| Debito | Simples | DECIMAL | 7,2 |  | Default 0.00 |
| Credito | Simples | DECIMAL | 7,2 |  | Default 0.00 |
| Voucher | Simples | DECIMAL | 7,2 |  | Default 0.00 |
| Cheque | Simples | DECIMAL | 7,2 |  | Default 0.00 |
| Outros | Simples | DECIMAL | 7,2 |  | Default 0.00 |
| Troco | Simples | DECIMAL | 7,2 |  | Default 0.00 |

Fonte: Quadro nosso.

Quadro 43: Tabela Produtos Retirados

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabela: **produtos\_retirados** | | | | | |
| **Campo** | **Classe** | **Tipo** | **Tamanho** | **Restrições** | **Observações** |
| ID\_Retirada | Determinante (Chave Primária) | INT | 11 | Não nulo e maior que zero | Campo auto-incremento |
| ID\_Produto | Determinante (Chave Primária e Chave Estrangeira) | INT | 11 | Não nulo e maior que zero |  |
| Quantidade | Simples | DECIMAL | 5,2 | Não nulo e maior que zero |  |
| Data | Simples | DATETIME |  | Não nulo |  |
| ID\_Sessao | Determinante (Chave Primária e Chave Estrangeira) | INT | 11 | Não nulo e maior que zero |  |
| ID\_Motivo | Determinante (Chave Primária e Chave Estrangeira) | INT | 11 | Não nulo e maior que zero |  |
| Observação | Simples | VARCHAR | 255 |  |  |

Fonte: Quadro nosso.

Quadro 44: Tabela Entradas de Produtos

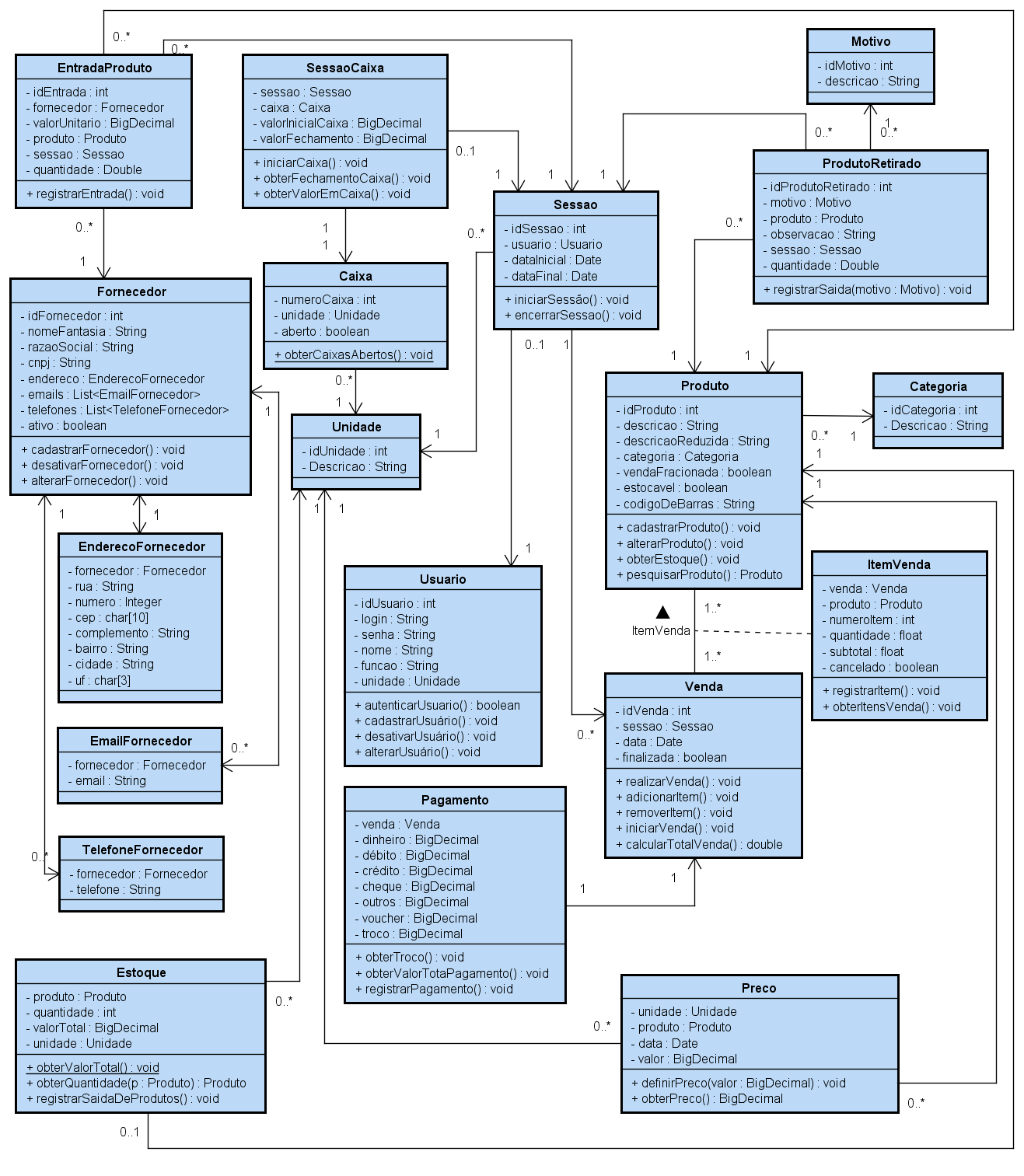
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tabela: **entradas\_produtos** | | | | | |
| **Campo** | **Classe** | **Tipo** | **Tamanho** | **Restrições** | **Observações** |
| ID\_Entrada | Determinante (Chave Primária) | INT | 11 | Não nulo e maior que zero | Campo auto-incremento |
| ID\_Produto | Determinante (Chave Primária e Chave Estrangeira) | INT | 11 | Não nulo e maior que zero |  |
| ID\_Fornecedor | Determinante (Chave Primária e Chave Estrangeira) | INT | 11 | Não nulo e maior que zero |  |
| Quantidade | Simples | DECIMAL | 5,2 | Não nulo e maior que zero |  |
| Valor\_Unitario | Simples | DECIMAL | 7,2 | Não nulo e maior que zero |  |
| Data | Simples | DATETIME |  | Não nulo |  |
| ID\_Sessao | Determinante (Chave Primária e Chave Estrangeira) | INT | 11 | Não nulo e maior que zero |  |
| ID\_Motivo | Determinante (Chave Primária e Chave Estrangeira) | INT | 11 | Não nulo e maior que zero |  |
| Observação | Simples | VARCHAR | 255 |  |  |

Fonte: Quadro nosso.

## 5.2 Diagrama de Classes de Análise

O diagrama de classes de análise abaixo ilustra as classes de entidade do sistema, a partir dele é possível observar os atributos e métodos de cada classe e os relacionamentos entre elas.

Figura 4: Diagrama de Classes



Fonte: Figura nossa.

## 5.2.1 Descrição das Classes

### 5.2.1.1 Classe “Unidade”

A classe “Unidade” representa cada unidade física da empresa ou cliente que utiliza o sistema. Cada empresa ou cliente do sistema pode possuir mais de uma filial e, portanto, mais de um estoque. Cada unidade é tratada individualmente em relação ao estoque e aos preços de venda dos produtos e não há compartilhamento de dados entre as unidades.

Esta classe tem como atributos o número de identificação (inteiro) e a descrição (texto).

### 5.2.1.2 Classe “Fornecedor”

A classe “Fornecedor” representa os fornecedores das mercadorias vendidas nas unidades. Sempre que um registro de entrada de produtos em estoque for realizado, o funcionário responsável deverá informar qual é o fornecedor da mercadoria.

Cada fornecedor deve estar cadastrado previamente para que seja possível associá-lo a um registro de entrada de produto. Para isso, é necessário que um funcionário responsável realize o cadastro de cada fornecedor de mercadorias. Ao cadastrar um novo fornecedor, é necessário entrar com os seguintes dados do fornecedor: CNPJ, nome fantasia, razão social e dados para contato.

Esta classe possui os seguintes atributos: número identificador, nome fantasia, razão social, CNPJ, endereço (instância da classe “Endereço”), e-mails (lista de objetos do tipo “Email”) e telefone (lista de objetos do tipo “Telefone”).

A classe “Fornecedor” possui métodos que permitem manter o cadastro de fornecedores.

### 5.2.1.3 Classe “Endereco”

A classe “Endereco” é a entidade que representa o endereço dos fornecedores de mercadoria. Cada fornecedor é associado a um endereço ao ser cadastrado.

Os atributos desta classe são: fornecedor (instância da classe “Fornecedor”), rua, número, CEP, complemento, bairro, cidade e UF.

### 5.2.1.4 Classe “Telefone”

A classe “Telefone” é a entidade que representa os números telefônicos para contato com os fornecedores de mercadorias. Cada fornecedor pode possuir muitos telefones para contato. Para isso, cada instância da classe “Fornecedor” possui uma referência para uma lista de objetos da classe “Telefone”.

Os atributos da classe são: fornecedor (instância da classe “Fornecedor) e telefone.

### 5.2.1.5 Classe “Email”

A classe “Email” é a entidade que representa os e-mails para contato com os fornecedores de mercadorias. Cada fornecedor pode possuir muitos telefones para contato. Para isso, cada instância da classe Fornecedor possui uma referência para uma lista de objetos da classe Telefone.

Os atributos da classe são: fornecedor (instância da classe “Fornecedor) e e-mail.

### 5.2.1.6 Classe “Usuario”

A classe “Usuario” é a entidade que representa os usuários que operam o sistema. Cada novo usuário deverá ser cadastrado por um outro usuário (que possua permissão para cadastrar novos usuários).

Os atributos desta classe são: número de identificação, nome, login, senha, função e unidade (instância da classe “Unidade”).

Os métodos da classe “Usuario” permitem manter o cadastro de usuários e realizar autenticação para o acesso ao sistema.

### 5.2.1.7 Classe “Caixa”

A classe “Caixa” representa cada um dos caixas de cada unidade. Cada caixa tem um número e pode estar aberto ou fechado. Um usuário que possui permissão pode, por meio do sistema, iniciar o caixa informando o valor de troco já contido em caixa.

Os atributos desta classe são: número do caixa, unidade (objeto do tipo “Unidade”) e aberto (atributo do tipo *boolean* que contém valor verdadeiro caso o caixa esteja aberto).

Esta classe possui o método “obter caixas abertos” que retorna uma lista mostrando os caixas que estão abertos.

### 5.2.1.8 Classe “Sessao”

A classe “Sessao” é a entidade que representa as sessões iniciadas sempre que um usuário faz login no sistema. Sempre que um usuário acessa o sistema uma nova sessão é instanciada e registrada no banco de dados.

Os atributos da classe são: número identificador da sessão, usuário (instância da classe “Usuario”, que contém as informações do usuário que efetuou o login), data inicial e data final. As datas inicial e final contém a data e hora de início de fim da sessão.

Os métodos da classe sessão são: iniciar sessão, que insere as informações da nova sessão no banco de dados, e encerrar sessão, que altera as informações no banco de dados, definindo a hora de encerramento da sessão.

### 5.2.1.9 Classe “SessaoCaixa”

A classe “SessaoCaixa” é a entidade que representa a sessão iniciada quando um caixa é aberto. Para que uma sessão de caixa seja iniciada é necessário que o usuário esteja iniciado uma sessão, ou seja, é necessário que haja uma instância da classe “Sessao” iniciada.

Quando um usuário abre um caixa, informando o número do caixa e o valor inicial em caixa, é instanciado um objeto da classe “SessaoCaixa”, que contém como atributos a sessão (objeto do tipo “Sessão”), o caixa (objeto do tipo “Caixa”), o valor inicial e o valor de fechamento de caixa, este último será definido apenas quando o caixa for finalizado.

Os métodos desta classe são: iniciar caixa (que insere no banco de dados as informações de abertura do caixa), obter fechamento de caixa (que informa o valor em caixa e altera as informações no banco de dados de acordo com o valor em caixa), e obter valor em caixa (que retorna o valor em caixa).

### 5.2.1.10 Classe “Categoria”

A classe “Categoria” é a entidade que representa as categorias dos produtos (ex.: Limpeza, Laticínios, Higiene pessoal, etc.).

Os atributos desta classe são: número identificador da categoria e descrição.

### 5.2.1.11 Classe “Produto”

A classe “Produto” é a entidade que representa o cadastro dos produtos a serem vendidos.

Os atributos da classe são: número identificador, descrição, descrição reduzida, categoria, venda fracionada (valor booleano que, se verdadeiro, indica que o produto é vendido por quilogramas e, se falso, por unidades), estocável (valor booleano) e código de barras.

As operações da classe “Produto” são: cadastrar produto, alterar cadastro, pesquisar produto e obter consulta de estoque.

### 5.2.1.12 Classe “Preco”

A classe “Preco” é a entidade que representa os preços dos produtos. Os preços dos produtos podem variar de tempos em tempos e as diferentes unidades/filiais dos estabelecimentos clientes podem vender produtos a preços diferentes umas das outras.

É necessário manter os preços que os produtos obtiverem ao passar do tempo no banco de dados para possíveis necessidades de relatórios futuros.

Os atributos da classe são: unidade (instancia da classe “Unidade), produto, data e valor.

As operações da classe “Produto” são: definir preço e obter preço.

### 5.2.1.13 Classe “Estoque”

A classe “Estoque”

### 5.2.1.14 Classe “Venda”

A classe “Venda”

### 5.2.1.15 Classe “ItemVenda”

A classe “ItemVenda”

### 5.2.1.16 Classe “Pagamento”

A classe “Pagamento”

### 5.2.1.17 Classe “ProdutoRetirado”

A classe “ProdutoRetirado”

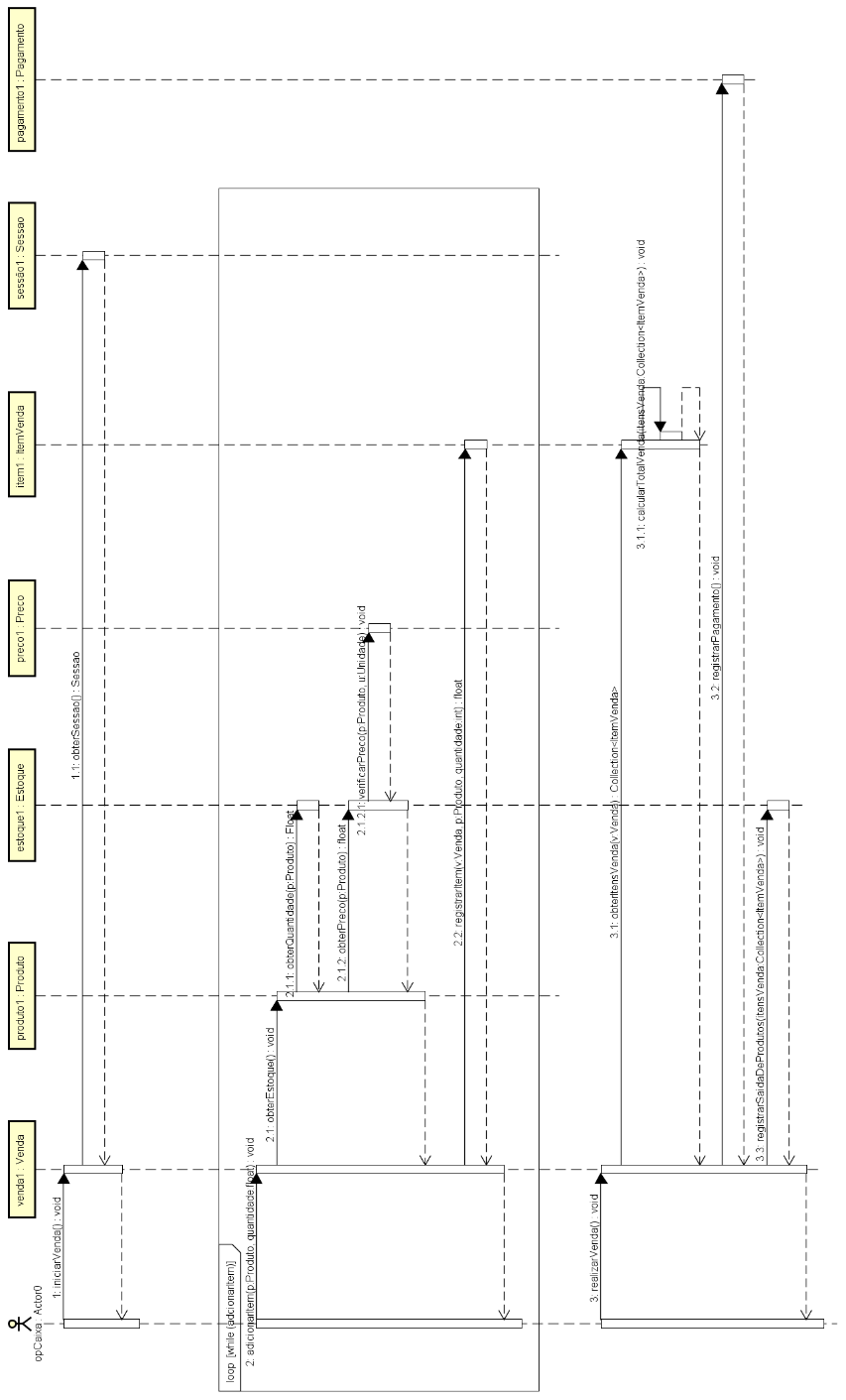
### 5.2.1.18 Classe “Motivo”

A classe “Estoque”

### 5.2.1.19 Classe “EntradaProduto”

A classe “EntradaProduto”

## 5.3 Diagrama de Sequência: Realizar Venda



## 5.4 Diagrama de Pacotes

## 5.5 Diagrama de Componentes

## 5.6 Diagrama de Implantação

# CONSIDERAÇÕES FINAIS

# REFERÊNCIAS

BEZERRA, E. **Princípios de Análise e Projeto de Sistemas com UML**. 3ª ed. Rio de Janeiro: Elsevier, 2015.

DEITEL, H. M. **Java, como programar**. Tradução de Carlos Arthur Lang Lisbôa. 4ª ed. Porto Alegre: Bookman, 2003.

TENÓRIO, F. G. **Tecnologia da Informação Transformando as Organizações e o Trabalho**. 1ª ed. Rio de Janeiro: Editora FGV, 2007.

AQUILES, A; FERREIRA, R. **Controlando versões com Git e GitHub**. 1ª ed. São Paulo: Casa do Código, 2014.

CARVALHO, V. **MySQL:** Comece com o principal banco de dados open source do mercado. 1ª ed. São Paulo: Casa do Código, 2015.

PRESSMAN, R. S. **Engenharia de Software:** Uma abordagem profissional. Tradução de Ariovaldo Griesi. 7ª ed. Porto Alegre: Bookman, 2011.