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

**DANYLO ADRIANO SOARES VIEIRA**

# GERENCIAMENTO DE CONSUMO EM BARES E RESTAURANTES

Assis
2014

Av. Getúlio Vargas, 1200 – Vila Nova Santana – Assis – SP – 19807-634
Fone/Fax: (0XX18) 3302-1055 homepage: www.fema.edu.br

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

**DANYLO ADRIANO SOARES VIEIRA**

# GERENCIAMENTO DE CONSUMO EM BARES E RESTAURANTES

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso Análise e Desenvolvimento de Sistemas.

**Orientando:** Danylo Adriano Soares Vieira

**Orientador:** Prof. MS. Felipe Alexandre Cardoso Pazinatto

**Área de Concentração:** Informática

Assis

Av. Getúlio Vargas, 1200 – Vila Nova Santana – Assis – SP – 19807-634
Fone/Fax: (0XX18) 3302-1055 homepage: www.fema.edu.br

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

2014

FICHA CATALOGRÁFICA

VIEIRA, Danylo Adriano Soares

Gerenciamento de Consumo em Bares e Restaurantes / Danylo Adriano Soares Vieira. Fundação Educacional do Município de Assis – FEMA – Assis, 2014.

Orientador: Prof. MS. Felipe Alexandre Cardoso Pazinatto

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

1. Software 2. Gerenciamento de Consumo.

CDD: 001.61

Biblioteca da FEMA

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

**DANYLO ADRIANO SOARES VIEIRA**

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso Análise e Desenvolvimento de Sistemas, analisado pela seguinte comissão examinadora:

**Orientador:** Prof. MS. Felipe Alexandre Cardoso Pazinatto

**Analisador:** Prof. Dr. Luiz Carlos Begosso

**Área de Concentração:** Informática

Assis
2014

# DEDICATÓRIA

Dedico este trabalho primeiramente a Deus, por ter me dado destreza, discernimento, força de vontade e muita garra para concluí-lo da forma mais correta possível, diante tantos obstáculos.

# AGRADECIMENTOS

Primeiramente agradeço a Deus por ter me fortalecido nesta jornada e me dado forças para seguir em frente à cada dia.

Agradeço ao meu orientador Felipe Alexandre Cardoso Pazinatto que me auxilia em todas minhas as dúvidas, e que me mostrou os caminhos para chegar até onde estou agora.

Aos meus colegas de sala, que me acompanharam até onde estou agora e que me apoiaram em momentos difíceis e me ajudaram de alguma forma.

A todos os meus professores da faculdade que caminharam comigo até agora, e que me transmitem conhecimento sem medir esforços e apoiam sempre que preciso, colaborando sempre para nosso crescimento.

E principalmente a minha família e a minha namorada, que tem me dado todo apoio e força neste momento, que está sempre ao meu lado para me ajudar no que preciso de todas as formas possíveis.

*“Acredite em si próprio e chegará um dia em que os outros não terão outra escolha senão acreditar com você.”*

*Cynthia Kersey*

# RESUMO

O sistema desenvolvido tem como objetivo principal organizar e facilitar o controle de movimentação dentro de um estabelecimento.
Esse software garante que a cobrança dos gastos obtidos pelo cliente seja feita de forma regular, sem a possibilidade das taxas serem burladas através da consumação de créditos.

O desenvolvimento do sistema foi realizado na linguagem de programação Java com a ferramenta Eclipse. Para armazenar as informações foi utilizado a ferramenta MySQL.

Palavras Chave: Software; Gerenciamento; Consumo;

# ABSTRACT

This system software has the task of control and to organize a restaurant. The software will help with control of charges between customer and the business chief, using the credit account consummation with pre charge on demand.

The system development will be done in the Java programming language with Eclipse tool. To store the information the software will use a MySQL database management system.

Keywords: Software; Management; Consumption;

# LISTA DE TABELAS

**Tabela 1 – Lista de Eventos** ..... 03
**Tabela 2 – Efetuar Login** ..... 30
**Tabela 3 – Cadastrar Cliente** ..... 31
**Tabela 4 – Cadastrar Produto** ..... 32
**Tabela 5 – Cadastrar Mesa** ..... 33
**Tabela 6 – Atualizar Quantidade Produto** ..... 34
**Tabela 7 – Vender Crédito** ..... 35
**Tabela 8 – Baixar Crédito Restante** ..... 36
**Tabela 9 – Retornar Dinheiro**..... 37
**Tabela 10 – Relacionar Cliente Mesa** ..... 38
**Tabela 11 – Vender Produto Cliente** ..... 39
**Tabela 12 – Retirar Crédito** ..... 40
**Tabela 13 – Atualizar Quantidade Produto**..... 41
**Tabela 14 – Vender Produto Mesa** ..... 42
**Tabela 15 – Retirar Crédito Ponderado** ..... 43
**Tabela 16 – Atualizar Quantidade Produto**..... 44
**Tabela 17 – Cronograma das Atividades** ..... 45

# LISTA DE ILUSTRAÇÕES

**Figura 1 – Diagrama do Gerenciamento**..... 28
**Figura 2 – Caso de Uso Geral**..... 29
**Figura 3 – Efetuar Login** ..... 30
**Figura 4 – Cadastrar Cliente** ..... 31
**Figura 5 – Cadastrar Produto**..... 32
**Figura 6 – Cadastrar Mesa** ..... 33
**Figura 7 – Atualizar Quantidade Produto** ..... 34
**Figura 8 – Vender Crédito** ..... 35
**Figura 9 – Baixar Crédito Restante**..... 36
**Figura 10 – Retornar Dinheiro** ..... 37
**Figura 11 – Relacionar Cliente Mesa** ..... 38
**Figura 12 – Vender Produto Cliente** ..... 39
**Figura 13 – Retirar Crédito** ..... 40
**Figura 14 – Atualizar Quantidade Produto** ..... 41
**Figura 15 – Vender Produto Mesa**..... 42
**Figura 16 – Retirar Crédito Ponderado** ..... 43
**Figura 17 – Atualizar Quantidade Produto** ..... 44
**Figura 18 – Diagrama de Atividade**..... 45
**Figura 19 – Diagrama de Classe** ..... 46
**Figura 20 – Diagrama de Entidade Relacionado** ..... 47
**Figura 21 – Sequenciamento das Atividades** ..... 48
**Figura 22 – Tela de Acesso** ..... 53
**Figura 23 – Tela de Venda** ..... 54
**Figura 24 – Tela de Realização de Pedidos** ..... 54
**Figura 25 – Tela de Fluxo de Caixa** ..... 55

# LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| WBS | Work Breakdown Structure |
| UML | Unified Modeling Language |
| SQL | Structure Query Language |
| RAM | Random Acess Memory |
| HD | Hard Disk |
| JSE | Java Standard Edition |
| CRUD | Cread Read Update Delete |
| IDE | Integrated Development Environment |
| EPL | Eclipse Public License |

SUMÁRIO

**1. INTRODUÇÃO ..... 16**
1.1 OBJETIVOS ..... 16
1.2 JUSTIFICATIVA ..... 16
1.3 PÚBLICO ALVO ..... 17
1.4 METODOLOGIA DO TRABALHO ..... 17
**2. MATERIAIS E MÉTODOS ..... 18**
2.1 Classificação de Materiais ..... 18
2.1.1 Requisitos Computacionais ..... 18
2.1.2 Requisitos Humanos ..... 18
2.2 DESCRIÇÃO DA METODOLOGIA DE ANÁLISE ..... 18
2.2.1 Definição de UML ..... 19
2.3 DESCRIÇÃO DO AMBIENTE DE DESENVOLVIMENTO ..... 20
2.4 FERRAMENTAS DE DESENVOLVIMENTO ..... 20
2.4.1 Astah Profissional ..... 20
2.4.2 Microsoft Visio 2013 ..... 20
2.4.3 DB Designer Fork ..... 21
2.4.4 Java ..... 21
2.4.5 Eclipse ..... 21
2.4.6 MySQL ..... 22
**3. ANÁLISE DE REQUISITOS ..... 23**
3.1 LEVANTAMENTO DE REQUISITOS ..... 23
3.2 VALIDAÇÃO DOS REQUISITOS ..... 23
3.3 CONFLITOS PONTENCIAIS ENCONTRADOS ..... 24
3.4 PROPOSTA DE SOLUÇÃO ..... 24
3.5 LISTA DE EVENTOS ..... 26
3.6 RECURSOS NECESSÁRIOS ..... 28
3.6.1 Recursos Computacionais ..... 28
3.6.2 Recursos Físicos ..... 28
3.6.3 Recursos Humanos ..... 28
3.7 ESTIMATIVA DE CUSTOS ..... 29

3.8 WBS – WORK BREAKDOWN STRUCTURE ..... 30
**4. PROJETO DO SISTEMA ..... 32**
4.1 DIAGRAMA DE CASOS DE USO (USE CASE) ..... 32
4.1.1 Caso de Uso Geral ..... 32
4.1.2 Efetuar Login ..... 33
4.1.3 Cadastrar Cliente ..... 34
4.1.4 Cadastrar Produto ..... 35
4.1.5 Cadastrar Mesa ..... 36
4.1.6 Atualizar Quantidade Produto ..... 37
4.1.7 Vender Crédito ..... 38
4.1.8 Baixar Crédito Restante ..... 39
4.1.9 Retornar Dinheiro ..... 40
4.1.10 Relacionar Cliente Mesa ..... 41
4.1.11 Vender Produto Cliente ..... 42
4.1.12 Retirar Crédito ..... 43
4.1.13 Atualizar Quantidade Produto ..... 44
4.1.14 Vender Produto Mesa ..... 45
4.1.15 Retirar Crédito Ponderado ..... 46
4.1.16 Atualizar Quantidade Produto ..... 47
4.2 DIAGRAMA DE ATIVIDADE ..... 48
4.3 DIAGRAMA DE CLASSES ..... 49
4.4 DIAGRAMA DE ENTIDADE RELACIONAMENTO ..... 50
**5. CRONOGRAMA ..... 51**
5.1 SEQUENCIAMENTO DAS ATIVIDADES ..... 51
5.2 CRONOGRAMA DE REALIZAÇÃO DAS ATIVIDADES ..... 52
**6. IMPLEMENTAÇÃO ..... 53**
6.1 BANCO DE DADOS ..... 53
6.1.1 Script da tabela itensvendas ..... 53
6.1.2 Script da Trigger adiciona_estoque ..... 53
6.1.3 Classe de conexão com Banco de dados ..... 53
**7. RESULTADOS ..... 54**

7.1 TELA DE ACESSO ..... 54
7.2 TELA DE VENDA ..... 55
7.3 TELA DE REALIZAÇÃO DE PEDIDOS ..... 55
7.4 TELA DE FLUXO DE CAIXA ..... 56
**8. CONCLUSÃO ..... 57**
**REFERÊNCIAS ..... 58**
**ANEXO ..... 59**
BOTÃO CADASTRO DE CRÉDITO ..... 59
BOTÃO CONFIRMAR VENDA ..... 60
BOTÃO ADICIONAR PRODUTO AO CLIENTE ..... 61

# 1. INTRODUÇÃO

Em nosso dia a dia nos deparamos com a tecnologia por toda parte, em diversos estabelecimentos, assim como em bares e lanchonetes, pois assegura mais conforto e comodidade aos proprietários, estabelecendo mais segurança em suas transações. Porém existe uma grande falta de confiabilidade na dedicação no pagamento de consumistas, pois fica impossível um controle total da movimentação de entrada e saída de clientes. Portanto o sistema sugere que o cliente efetue o pagamento antecipadamente, evitando quaisquer conflitos futuros, tendo em vista que será impossível burlar o pagamento de suas consumações.

## 1.1 OBJETIVOS

O sistema tem como objetivo eliminar a falta de confiabilidade e segurança na estadia dos clientes dentro do estabelecimento, controlando a entrada e saída de caixa, como também identificar o cliente que será atendido, trazendo assim mais comodidade e organização ao estabelecimento e aos usuários.

## 1.2 JUSTIFICATIVA

A necessidade do desenvolvimento deste software se dá pela grande movimentação de transações e a falta de segurança e controle dentro de estabelecimentos comerciais, pois diversas vezes a cobrança, na realidade, não é feita de forma correta e isso acaba sendo prejudicial à empresa, com o auxílio deste sistema a ocorrência destes fatos será extinta, proporcionando mais conforto e dedicação ao cliente e a empresa.

## 1.3 PÚBLICO ALVO

O sistema tem como publico alvo empresas no segmento alimentício, empresas de pequeno porte, como bares e lanchonetes. Empresas destas características podem informatizar-se com este sistema, proporcionando uma significativa mudança para seus negócios, de forma mais ágil e prestativa, auxiliando em um desenvolvimento mais agregado de sua empresa.

## 1.4 METODOLOGIA DO TRABALHO

Com a metodologia utilizada neste trabalho é possível controlar e monitorar as transações realizadas de forma segura e eficiente.
O método requer que o sistema se comunique com um servidor, transmitindo todas as informações a ele, com base no que o cliente requisitar. O sistema foi desenvolvido na plataforma cliente/servidor na linguagem Java.

# 2. MATERIAIS E MÉTODOS

## 2.1 Classificação de Materiais

O desenvolvimento do sistema deverá levar em consideração diversos requisitos funcionais visando aperfeiçoar a comunicação cliente servidor, com verificação dos erros, fazendo o mesmo funcionar a contento quando implantado no ambiente do estabelecimento comercial.

### 2.1.1 Requisitos Computacionais

Para maior funcionalidade e desempenho do Sistema no servidor local o ideal é que o mesmo contenha os seguintes requisitos:

Dois computadores com processador Intel Core i7 que possui quatro núcleos de alto desempenho, com 16GB de RAM (Radom Access Memory), 2TB de HD (Hard Disk).

### 2.1.2 Requisitos Humanos

Um profissional com habilidades na área de informática que saiba manusear o sistema de forma prática e ágil.

## 2.2 DESCRIÇÃO DA METODOLOGIA DE ANÁLISE

A metodologia do sistema foi modelada através da Análise Orientada a Objetos, que constitui uma linguagem de análise chamada UML (Unify Modeling Language).

### 2.2.1 Definição de UML

UML é uma linguagem de modelagem, utilizada por métodos para mostrar e expressar projetos, é uma maneira de sugerir como e quais passos devem ser seguidos para elaboração de um projeto.

Segundo Guedes (2010), UML, ou em português, Linguagem de Modelagem Unificada, é uma linguagem visual para modelagem de softwares baseada no paradigma da orientação a objetos, sendo possível aplica-la, juntamente com seu propósito geral, a todos os domínios de aplicação, tornando-se nos últimos anos, a linguagem padrão de modelagem, sendo adotada internacionalmente pela indústria de engenharia de software.

Segundo Larman (2000), UML é um padrão emergente, que está sendo aceito pela indústria, para a modelagem orientada a objetos. Ela começou como um esforço conjunto de Grady Booch e Jim Rumbaugh em 1994, para combinar seus dois métodos populares – os métodos Booch e OMT(Object Modeling Technique), em resposta a uma solicitação para definir uma linguagem e notação de modelagem padronizada.

## 2.3 DESCRIÇÃO DO AMBIENTE DE DESENVOLVIMENTO

O sistema desenvolvido utilizando a ferramenta Eclipse, no ambiente de desenvolvimento *JSE(Java Standard Edition)* onde é realizada todas as aplicações do projeto. Para o armazenamento de dados é utilizado o banco *MySQL*. O sistema foi implementado unindo as ferramentas, o projeto ao seu banco, visando um resultado positivo.

## 2.4 FERRAMENTAS DE DESENVOLVIMENTO

### 2.4.1 Astah Profissional

O *Astah* é uma ferramenta de modelagem em *UML*, que permite a construção de vários diagramas essenciais para a análise de software. Ele possibilita a construção dos diagramas de forma ágil e útil, como os de Casos de Uso, de Atividade, de *CRUD(Create, Read, Update, Delete)*, de Entidade-Relacionamento, de Classe entre outros. O *Astah* é capaz de proporcionar diagramas prontos para a linguagem Java.

### 2.4.2 Microsoft Visio 2013

O *Microsoft Visio 2013* é bem conhecido por conter os diagramas técnicos e profissionais, com imagens vetoriais, que podem ser ampliados e manipulados com facilidade. O Visio pode ser utilizado para gerar diagramas de diversos tipos, como organogramas, fluxogramas, modelagem de dados, casos de uso, entre muitos outros. (FILHO, 2014)

### 2.4.3 DB Designer Fork

O *DB Designer Fork* é uma ferramenta para modelagem de dados, trabalhando com modelos lógicos e desenvolvida pela *fabFORCE*. Possui licença livre, sendo multiplataforma e implementado em *Delphi/Kylix*. O *DB Designer* permite a modelagem, criação e manutenção de banco de dados, bem como fazer a engenharia reversa, ou seja, gerando o modelo de dados a partir de um banco existente, sincronizando-os. (ARAÚJO, 2014)

### 2.4.4 Java

*Java* é a uma base de desenvolvimento para todos os tipos de aplicações em rede, e é o padrão global de implementação e distribuição de aplicações, jogos, dispositivos móveis, conteúdo baseado em *Web*, softwares corporativos. Com milhares de pessoas que desenvolvem no mundo todo, e de forma eficiente, o *Java* possibilita que você desenvolva, implante, e use aplicações e serviços estimulantes. (BALAGTAS, 2013)

### 2.4.5 Eclipse

A plataforma *Eclipse* é um *IDE(Integrated Development Environment)* capaz de suportar várias linguagens através de plug-ins, porém desenvolvido especialmente para a linguagem *Java*, a plataforma segue o modelo *open source* de desenvolvimento de software. O eclipse possui sua liberação em *EPL(Eclipse Public License)*, portanto é totalmente aberto ao público. (GALLARDO, 2012)

### 2.4.6 MySQL

Um banco de dados é uma coleção de dados estruturados. Ele pode fornecer qualquer coisa, uma simples lista de compras, uma galeria de imagens ou até grandes quantidades de informações de uma rede coorporativa.

O MySQL é o mais popular banco de dados SQL livre. Ele é um sistema que faz gerenciamento de bancos de dados. MySQL foi desenvolvido para armazenar grandes informações de dados e de maneira muito mais rápida que outras soluções existentes e está sendo usado em ambientes de produção que exige alta demanda, e por diversos anos de maneira bem sucedida. Um Banco de dados relacional armazena dados em tabelas separadas embora todos os dados estejam armazenados em um só local. Isso proporciona velocidade e flexibilidade. As tabelas são unidas por relações definidas tornando possível combinar dados de diferentes tabelas nas requisições. (PEDROSO, 2007)

# 3. ANÁLISE DE REQUISITOS

## 3.1 LEVANTAMENTO DE REQUISITOS

Para desenvolvimento do sistema foram levantados os seguintes requisitos necessários:

**Administrador**

- Efetuar Login de Acesso

**Atendente**

- Cadastrar Cliente
- Cadastrar Produto
- Cadastrar Mesa
- Atualizar Quantidade Produto
- Vender Crédito
- Baixar Crédito Restante
- Retornar Dinheiro
- Relacionar Cliente Mesa

**Garçom**

- Vender Produto Cliente
- Vender Produto Mesa
- Retirar Crédito
- Retirar Crédito Ponderado
- Atualizar Quantidade Produto

## 3.2 VALIDAÇÃO DOS REQUISITOS

Após a conclusão parcial do sistema, alguns testes reais foram feitos para verificação da qualidade da usabilidade do sistema.

• Requisitos estão completos?

(x) SIM ( )NÃO

• Os requisitos são reais?

(x) SIM ( ) NÃO

• Houve algum tipo de redundância ou falta de clareza dos requisitos?

(x) SIM ( ) NÃO

## 3.3 CONFLITOS PONTENCIAIS ENCONTRADOS

Alguns conflitos possíveis foram encontrados, tais como:

- A falta de habilidades do usuário na manipulação do sistema.
- Ocorrência de falta de energia.
- Manutenção técnica mal sucedida.
- Problema nos aparelhos computacionais.
- Espaço de armazenamento de dados insuficiente.

## 3.4 PROPOSTA DE SOLUÇÃO

- Manter usuários sempre capacitados para a utilização do sistema, através de treinamentos e estudos.

- Utilização de nobreaks nos computadores, para evitar a perca de dados em caso de falta de energia.
- Manutenção mensal para reparar e atualizar o sistema.
- Fazer manutenção aos equipamentos periodicamente.
- Atualização dos aparelhos conforme a necessidade do usuário.

## 3.5 LISTA DE EVENTOS

| Nº | Evento | Caso de Uso |
|---|---|---|
| 1 | Administrador faz login | Efetuar Login |
| 2 | Atendente cadastra Cliente | Cadastro de Cliente |
| 3 | Atendente cadastra Produto | Cadastro de Produto |
| 4 | Atendente cadastra Mesa | Cadastro de Mesa |
| 5 | Atendente atualiza quantidade de produtos | Atualizar Quantidade Produto |
| 6 | Atendente associa créditos ao cliente | Vender Crédito |
| 7 | Atendente da baixa de crédito restante do cliente | Baixar Crédito Restante |
| 8 | Atendente faz a devolução em dinheiro ao cliente | Retornar Dinheiro |
| 9 | Atendente associa cliente à mesa | Relacionar Cliente Mesa |
| 10 | Garçom efetua venda ao cliente | Vender Produto Cliente |
| 11 | Garçom debita crédito do cliente | Retirar Crédito |
| 12 | Garçom atualiza a quantidade de produtos | Atualizar Quantidade Produto |
| 13 | Garçom efetua venda a uma mesa de clientes | Vender Produto Mesa |
| 14 | Garçom debita crédito de mesa de clientes | Retirar Crédito Ponderado |
| 15 | Garçom atualiza a quantidade de produtos | Atualizar Quantidade Produto |

**Tabela 1 – Lista de Eventos**

Esta análise visa proporcionar maior visão e clareza diante dos requisitos necessários para o desenvolvimento do sistema, este conhecimento torna o caminho e os conflitos do projeto mais nítidos, portanto mais específico.

A análise feita auxilia também na compreensão do usuário diante o sistema, deixando claro todos os procedimentos que o sistema irá tomar de acordo com cada função determinada pelo o usuário.

## 3.6 RECURSOS NECESSÁRIOS

Para a elaboração do sistema foram necessários alguns recursos, sendo eles: recursos físicos, como computadores e impressoras, recursos computacionais, como softwares, e recursos humanos, como o profissional que irá trabalhá-lo. Foi um projeto de pequeno porte, portanto foram utilizados poucos recursos para o desenvolvimento do sistema.

### 3.6.1 Recursos Computacionais

- Windows 7 Ultimate;
- Eclipse;
- Microsoft Visio 2013;
- MySQL;
- DB Designer Fork;
- Astah Profissional.

### 3.6.2 Recursos Físicos

- Um notebook Samsung RV411;
- Uma impressora HP Deskjet 3516.

### 3.6.3 Recursos Humanos

- Um Analista de Sistema.

## 3.7 ESTIMATIVA DE CUSTOS

Recursos:

- Computacionais:

- Eclipse Freeware

- Microsoft Visio 2013 R$ 550,00

- MySQL Freeware

- DB Designer Fork Freeware

- Astah Profissional R$ 89,80

Custo Total Software = R$550,00 + 89,00 = R$639,00

- Físicos:

- Notebook Samsung RV411 R$1800,00

Depreciação (2 anos) = R$1.800,00 / 24 = R$ 75,00/mês

Custo por dia = R$75,00 / 26 (dias) = R$2,88 (ao dia)

Custo do total do computador = R$2,88 * 330 = R$950,40.

- Impressora HP Deskjet 3516 R$ 300,00

Depreciação = R$300,00 / 12 = R$25,00/mês

Custo dia = R$25,00 / 26 = R$0,96 (ao dia)

Custo impressora = R$0,96 * 330 = R$316,80

Custo Total Equipamento = R$950,00 + R$316,80 = R$1266,80

- Humanos:

| Analista | Período/Dias | | Custo Diário | Total |
|---|---|---|---|---|
| Danylo A. S. Vieira | Análise | 180 | R$ 25,00 | R$ 4.500,00 |
| | Desenv. | 150 | R$ 25,00 | R$ 3.750,00 |
| **Total** | | | | R$ 8.250,00 |

Custo total do Sistema: **R$10.155,80**

## 3.8 WBS – WORK BREAKDOWN STRUCTURE

Segundo o Guia PMI de Gerenciamento de Projetos (2004, p. 112):

A *WBS*, ou *Estrutura Analítica de Projeto*, é uma decomposição hierárquica orientada à entrega do trabalho a ser executado pela equipe do projeto, para atingir os objetivos do projeto e criar as entregas necessárias. A *EAP* organiza e define o escopo total do projeto. A *EAP* subdivide o trabalho do projeto em partes menores e mais facilmente gerenciáveis, em que cada nível descendente da *EAP* representa uma definição cada vez mais detalhada do trabalho do projeto. É possível agendar, estimar custos, monitorar e controlar o trabalho planejado contido nos componentes de nível mais baixo da *EAP*, denominados pacotes de trabalho.

A Figura, 1 ilustra a WBS do sistema proposto:

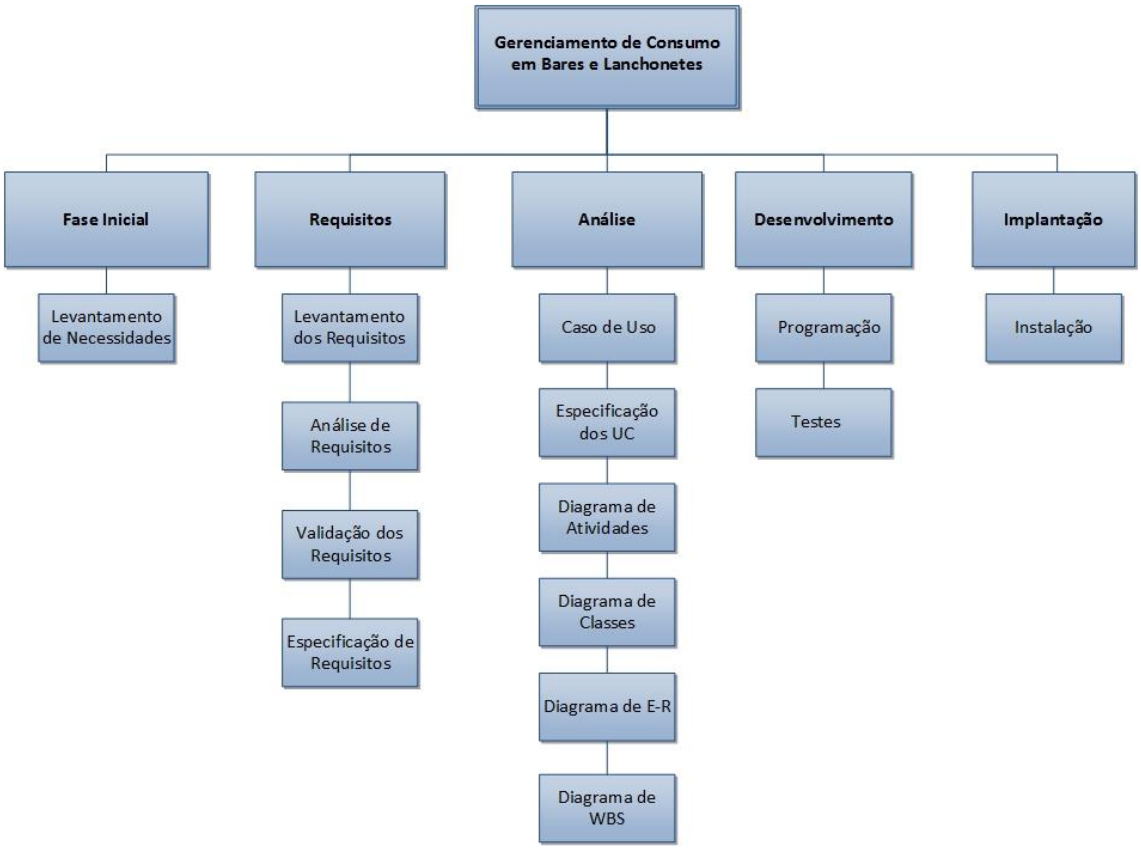


**Figura 1 – Diagrama do Gerenciamento de Consumo em Bares e Restaurantes**

# 4. PROJETO DO SISTEMA

Essa análise tornou o desenvolvimento do projeto ainda mais aprofundado em seus detalhes, mostrando de forma nítida e transparente todos os problemas e dificuldades que ocorreriam durante o desenvolvimento, desta forma, surge à necessidade de trabalhar com mais flexibilidade e visão durante todo o trajeto da aplicação.

## 4.1 DIAGRAMA DE CASOS DE USO (USE CASE)

Segundo Larman (2000): Um caso de uso é um documento narrativo que descreve a sequência de eventos de um ator que usa um sistema para completar um processo. Eles são história ou casos de utilização de um sistema. Casos de uso não são exatamente especificação de requisitos ou especificação funcional, mais ilustram e implicam requisitos na história que eles contam.

### 4.1.1 Caso de Uso Geral

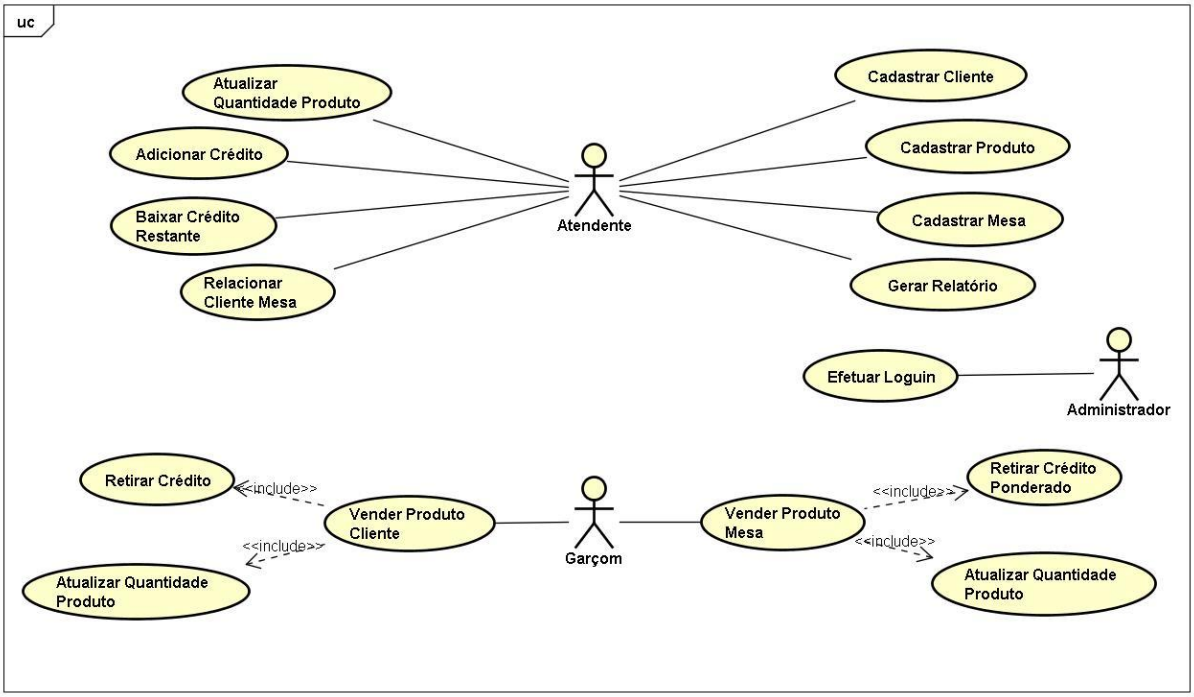


**Figura 2 – Caso de Uso Geral**

### 4.1.2 Efetuar Login

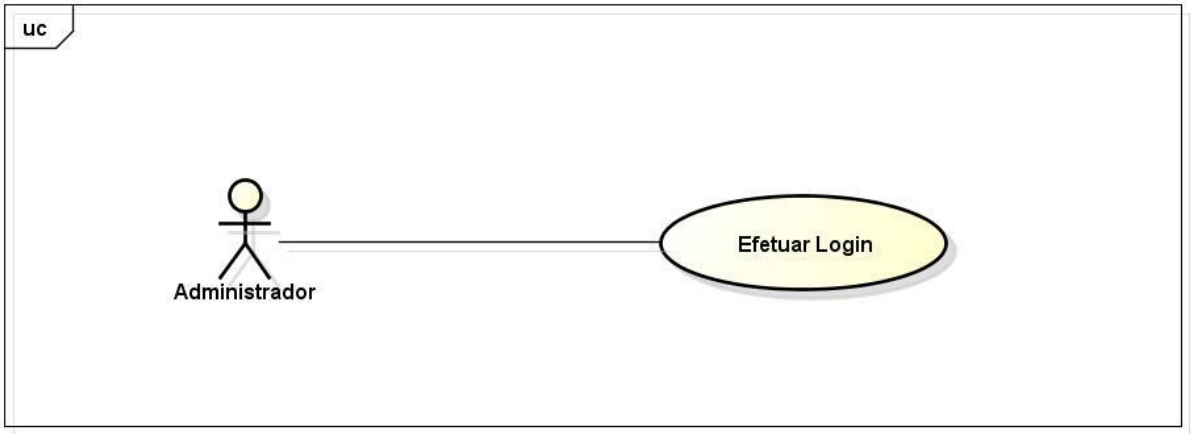


**Figura 3 – Efetuar Login**

| | |
|---|---|
| **Nome do Caso de Uso** | Fazer Login |
| **Ator Principal** | Administrador |
| **Atores Secundários** | |
| **Resumo** | Permite o usuário acessar o sistema |
| **Pré-Condições** | O usuário deve teve conter um registro no sistema |
| **Pós-Condições** | O usuário terá acesso a todos os dados e movimentações no sistema. |
| **Fluxo Principal** | 1. Usuário informa nome de login e senha.<br>2. O Sistema verifica os registros informados, e libera o acesso. |
| **Fluxos Alternativos** | 1. Usuário cancela a entrada ao sistema.<br>2. O Sistema encerra. |
| **Fluxos de Exceções** | 1. Usuário digita login ou senha incorreta.<br>2. O Sistema emite mensagem de “Incorreto”, e cancela a entrada. |

**Tabela 2 – Efetuar Login**

### 4.1.3 Cadastrar Cliente

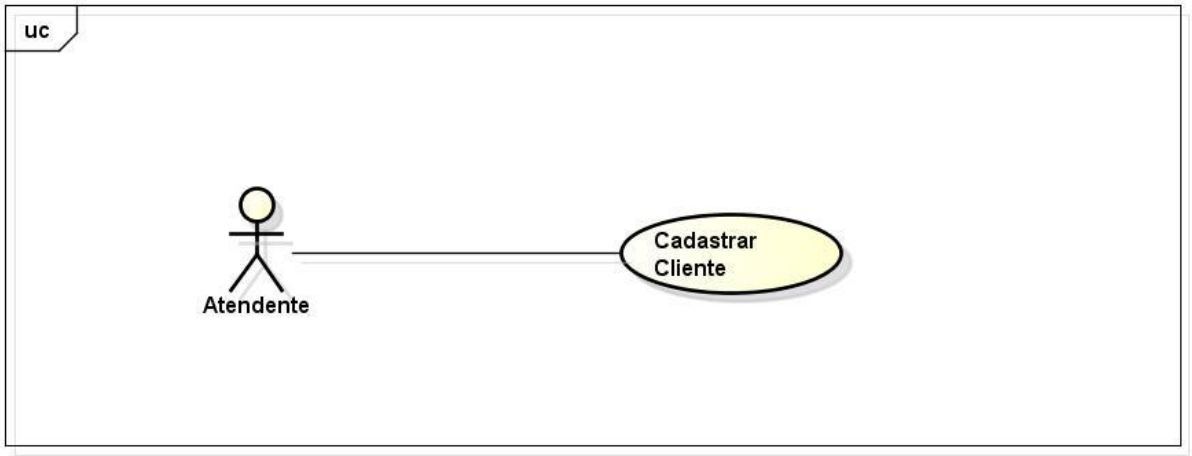


**Figura 4 – Cadastrar Cliente**

| | |
|---|---|
| **Nome do Caso de Uso** | Cadastrar Cliente |
| **Ator Principal** | Atendente |
| **Atores Secundários** | |
| **Resumo** | Cria o cadastro de um novo cliente. |
| **Pré-Condições** | O usuário deve estar conectado ao sistema. |
| **Pós-Condições** | O usuário terá acesso a todos os dados e movimentações no sistema. |
| **Fluxo Principal** | 1. Usuário deve clicar em Cadastrar Cliente.<br>2. Sistema irá abrir uma página para o preenchimento dos campos necessários.<br>3. Usuário deve preencher os campos, e clicar em salvar.<br>4. Sistema emite uma mensagem de confirmação.<br>5. Usuário deve confirmar a mensagem.<br>6. Sistema conclui exibindo uma mensagem de “Cadastrado com sucesso”.<br>7. Usuário clica “Ok”. |
| **Fluxos Alternativos** | 1. Usuário cancela o cadastro de cliente.<br>2. Sistema retorna a página anterior. |
| **Fluxos de Exceções** | 1. Usuário não preenche corretamente todos os campos.<br>2. Sistema emite uma mensagem de falha. |

**Tabela 3 – Cadastrar Cliente**

### 4.1.4 Cadastrar Produto

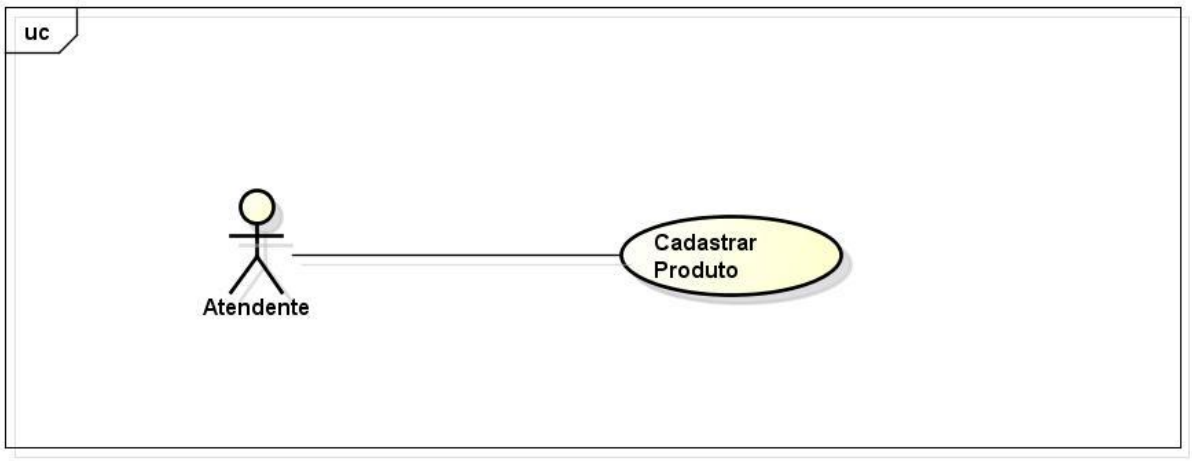


**Figura 5 – Cadastrar Produto**

| | |
|---|---|
| **Nome do Caso de Uso** | Cadastro de Produto |
| **Ator Principal** | Atendente |
| **Atores Secundários** | |
| **Resumo** | Cria o Cadastro de um novo produto. |
| **Pré-Condições** | O usuário deve estar conectado ao sistema. |
| **Pós-Condições** | O usuário terá acesso a todos os dados e movimentações no sistema. |
| **Fluxo Principal** | 1. Usuário deve clicar em Cadastrar Produto.<br>2. O sistema irá abrir uma nova página para preenchimento dos campos necessário.<br>3. Usuário deve digitar corretamente todos os dados do novo produto.<br>4. Sistema irá verificar todos os campos e liberar a confirmação de cadastro.<br>5. Usuário terá que confirmar o cadastro.<br>6. Sistema emite uma mensagem de confirmação.<br>7. Usuário deve confirmar a mensagem.<br>8. Sistema conclui emitindo uma mensagem de “Cadastro efetuado” e limpa os campos para um novo cadastro. |
| **Fluxos Alternativos** | 1. Usuário cancela o cadastro de produto.<br>2. Sistema retorna a página anterior. |
| **Fluxos de Exceções** | 1. Usuário cadastra um produto já cadastrado.<br>2. Sistema emite uma mensagem de erro, e reinicia o cadastro. |

**Tabela 4 – Cadastrar Produto**

### 4.1.5 Cadastrar Mesa

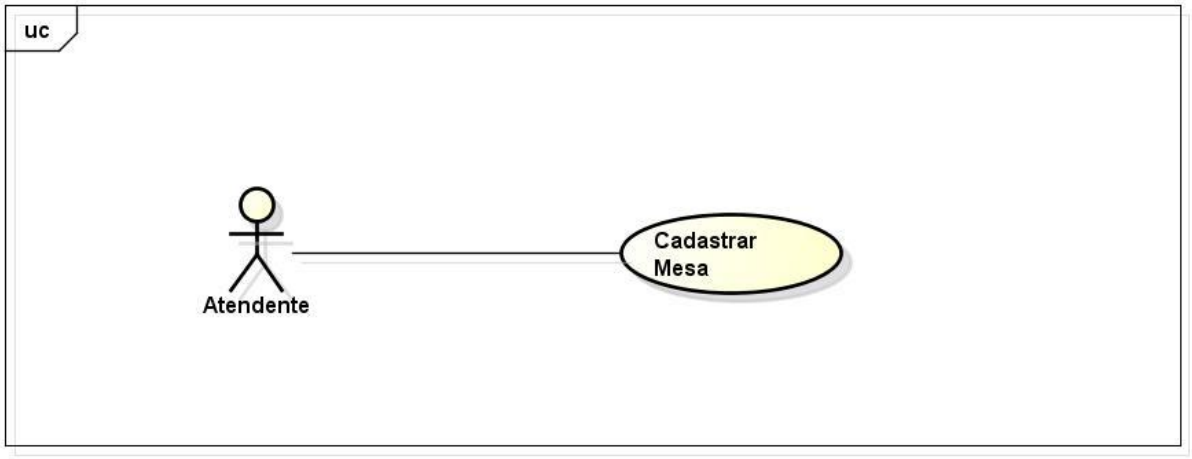


**Figura 6 – Cadastrar Mesa**

| | |
|---|---|
| **Nome do Caso de Uso** | Cadastro de Mesa |
| **Ator Principal** | Atendente |
| **Atores Secundários** | |
| **Resumo** | Permite cadastrar uma nova Mesa. |
| **Pré-Condições** | O usuário deve estar conectado ao sistema. |
| **Pós-Condições** | O usuário terá uma nova mesa registrada no sistema. |
| **Fluxo Principal** | 1. Usuário deve clicar em Cadastrar Mesa.<br>2. Sistema irá abrir uma nova página requisitando a descrição da mesa.<br>3. Usuário terá que preencher e clicar em “Ok”.<br>4. Sistema irá emitir uma mensagem de confirmação do cadastro.<br>5. Usuário deve clicar em “Concluir”.<br>6. Sistema emitirá uma mensagem de “Mesa cadastrada” e finaliza. |
| **Fluxos Alternativos** | 1. Usuário cancela o cadastro de mesa.<br>2. Sistema retorna a página anterior. |
| **Fluxos de Exceções** | 1. Usuário tenta cadastrar uma mesa já registrada.<br>2. Sistema emite mensagem de “Erro”. |

**Tabela 5 – Cadastrar Mesa**

### 4.1.6 Atualizar Quantidade Produto

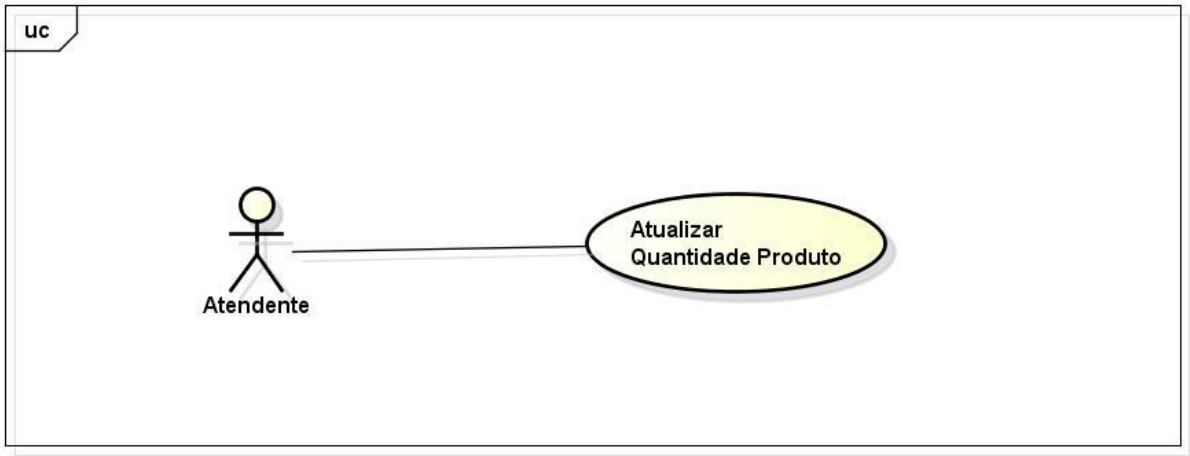


**Figura 7 – Atualizar Quantidade Produto**

| | |
|---|---|
| **Nome do Caso de Uso** | Atualizar Quantidade Produto |
| **Ator Principal** | Atendente |
| **Atores Secundários** | |
| **Resumo** | Permite a atualização da quantidade de produtos em estoque. |
| **Pré-Condições** | O usuário deve estar conectado ao sistema. |
| **Pós-Condições** | O banco de dados do sistema atualizará com novos registros. |
| **Fluxo Principal** | 1. Usuário abre pagina de atualização de quantidade de produto.<br>2. Sistema libera uma lista com todos produtos registrados.<br>3. Usuário clica em Alterar Quantidade.<br>4. Sistema libera os campos para edição de quantidade.<br>5. Usuário digita a nova quantidade, e confirma.<br>6. Sistema atualiza página, e informa os dados atualizados. |
| **Fluxos Alternativos** | 1. Usuário cancela a alteração de quantidade.<br>2. Sistema retorna a página anterior. |
| **Fluxos de Exceções** | 1. Usuário digita um valor negativo em quantidade.<br>2. Sistema emite mensagem de “Quantidade incorreta”. |

**Tabela 6 – Atualizar Quantidade Produto**

### 4.1.7 Vender Crédito

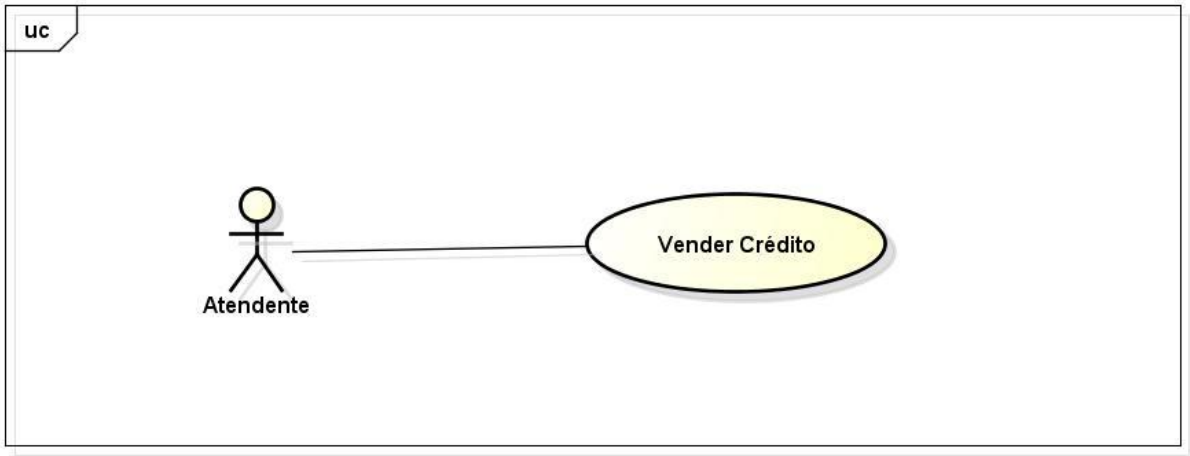


**Figura 8 – Vender Crédito**

| | |
|---|---|
| **Nome do Caso de Uso** | Vender Crédito |
| **Ator Principal** | Atendente |
| **Atores Secundários** | |
| **Resumo** | Associa Créditos ao Cliente. |
| **Pré-Condições** | O usuário deve estar conectado ao sistema. |
| **Pós-Condições** | O cliente terá créditos associados a seu cadastro. |
| **Fluxo Principal** | 1. Usuário abre pagina de Vender Crédito.<br>2. Sistema abre página com uma nova tabela de crédito.<br>3. Usuário associa o nome do cliente já registrado com um novo crédito.<br>4. Sistema cria um novo crédito associado a um cliente.<br>5. Usuário digita o valor à ser creditado no crédito do cliente, e confirma.<br>6. Sistema emite mensagem de “Cliente Creditado”. |
| **Fluxos Alternativos** | 1. Usuário cancela a criação do novo crédito.<br>2. Sistema retorna a página anterior. |
| **Fluxos de Exceções** | 1. Usuário informa um cliente não cadastrado.<br>2. Sistema emite mensagem de “Cliente sem cadastro”. |

**Tabela 7 – Vender Crédito**

### 4.1.8 Baixar Crédito Restante

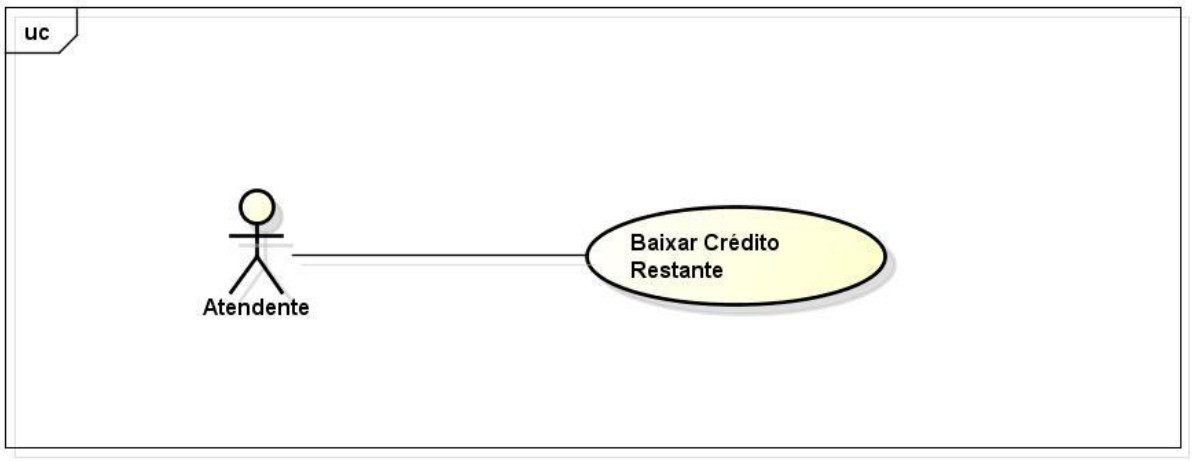


**Figura 9 – Baixar Crédito Restante**

| | |
|---|---|
| **Nome do Caso de Uso** | Baixar Crédito Restante |
| **Ator Principal** | Atendente |
| **Atores Secundários** | |
| **Resumo** | Retorna o valor restante creditado, ao cliente, após termino de seu consumo. |
| **Pré-Condições** | O usuário deve estar conectado ao sistema. |
| **Pós-Condições** | A conta do cliente terá seu crédito estornado, o registro de crédito será finalizado do registro do cliente. |
| **Fluxo Principal** | 1. Usuário Busca o nome do Cliente.<br>2. Sistema abre uma página identificando a condição atual do cliente solicitado.<br>3. Usuário clica em Dar baixar no crédito.<br>4. Sistema retorna o saldo restante que o cliente possui creditado.<br>5. Usuário finaliza o consumo do cliente, e confirma.<br>6. Sistema realiza o fechamento de registro de crédito do cliente. |
| **Fluxos Alternativos** | 1. Usuário cancela a solicitação de baixar de crédito.<br>2. Sistema retorna uma mensagem de confirmação.<br>3.Usuário confirma.<br>4. Sistema retorna a pagina anterior |
| **Fluxos de Exceções** | 1. Usuário informa um cliente não cadastrado.<br>2. Sistema emite mensagem de “Cliente sem cadastro”. |

**Tabela 8 – Baixar Crédito Restante**

### 4.1.9 Retornar Dinheiro

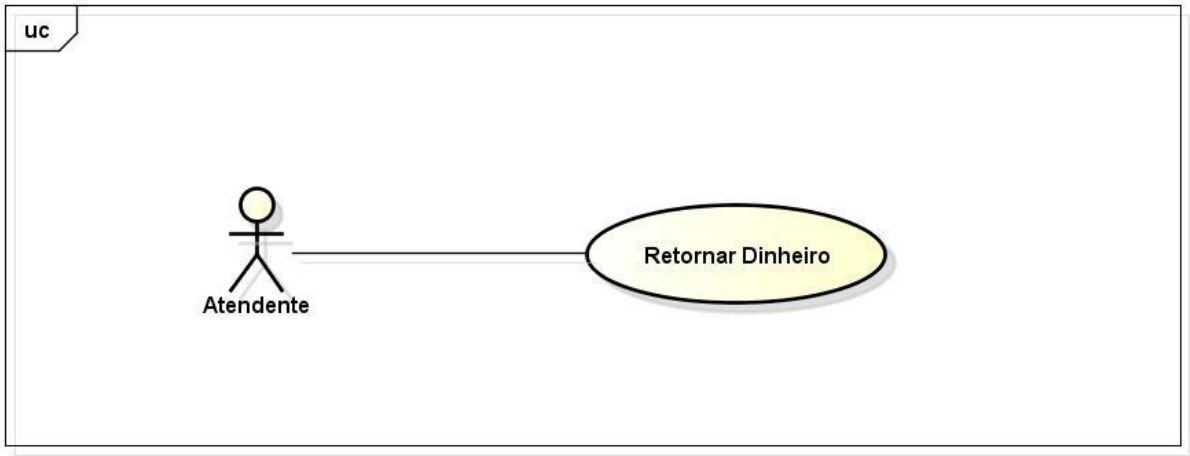


**Figura 10 – Retornar Dinheiro**

| | |
|---|---|
| **Nome do Caso de Uso** | Retornar Dinheiro |
| **Ator Principal** | Atendente |
| **Atores Secundários** | |
| **Resumo** | Realiza a devolução em dinheiro dos créditos restantes do cliente. |
| **Pré-Condições** | O usuário deve estar conectado ao sistema. |
| **Pós-Condições** | O sistema terá finalizado o crédito associado ao cliente. |
| **Fluxo Principal** | 1. Usuário verifica o Crédito restante do cliente.<br>2. Sistema pesquisa o crédito do cliente.<br>3. Usuário finaliza associação do cliente ao crédito restante.<br>4. Sistema emite o saldo em dinheiro restante do crédito do cliente.<br>5. Usuário seleciona Baixar saldo, e confirma.<br>6. Sistema emite comprovante de saldo.<br>7. Usuário finaliza o status do cliente.<br>8. Sistema emite mensagem de “Cliente Finalizado”, e retorna a pagina anterior. |
| **Fluxos Alternativos** | 1. Usuário cancela a alteração do crédito.<br>2. Sistema retorna a pagina anterior |
| **Fluxos de Exceções** | 1. Usuário verifica um cliente sem saldos.<br>2. Sistema emite mensagem de “Saldo Insuficiente”. |

**Tabela 9 – Retornar Dinheiro**

### 4.1.10 Relacionar Cliente Mesa

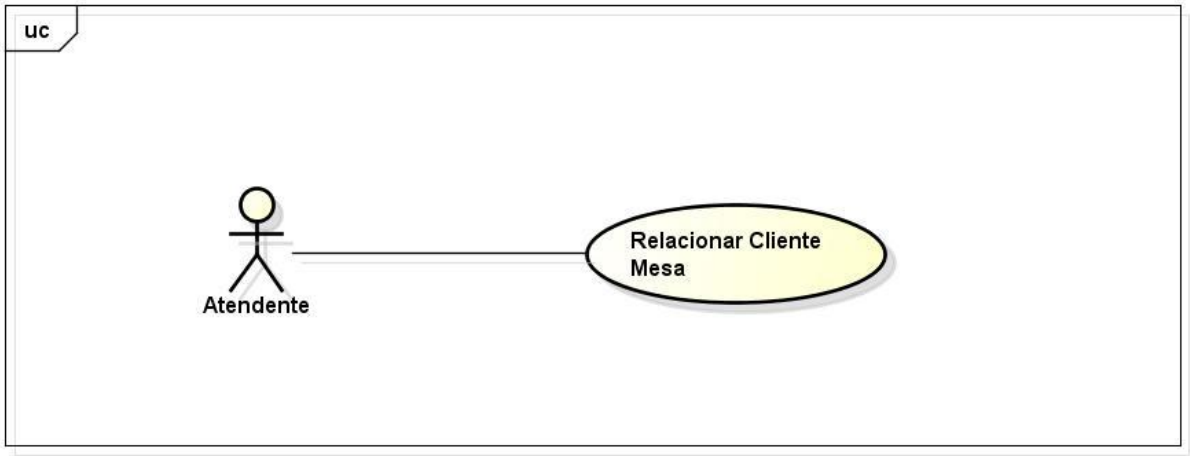


**Figura 11 – Relacionar Cliente Mesa**

| | |
|---|---|
| **Nome do Caso de Uso** | Relacionar Cliente Mesa |
| **Ator Principal** | Atendente |
| **Atores Secundários** | |
| **Resumo** | Associa um cliente já cadastrado, a uma mesa. |
| **Pré-Condições** | O usuário deve estar conectado ao sistema. |
| **Pós-Condições** | O cliente terá uma mesa associada a seu cadastro. |
| **Fluxo Principal** | 1. Usuário abre a pagina de Relacionar Cliente Mesa.<br>2. Sistema abre uma página solicitando o nome do cliente.<br>3. Usuário digita o nome do cliente, e confirma.<br>4. Sistema abre uma seleção de mesas, e permite a escolha de qual mesa se associar.<br>5. Usuário seleciona a mesa desejada, e confirma.<br>6. Sistema emite mensagem de “Mesa associada ao Cliente”. |
| **Fluxos Alternativos** | 1. Usuário cancela a associação de mesa ao cliente.<br>2. Sistema emite mensagem de “Deseja cancelar?”;<br>3. Usuário confirma a mensagem.<br>4. Sistema retorna à página anterior. |
| **Fluxos de Exceções** | 1. Usuário digita o nome do cliente, e confirma.<br>2. Sistema emite mensagem de “Sem mesas disponíveis”. |

**Tabela 10 – Relacionar Cliente Mesa**

### 4.1.11 Vender Produto Cliente

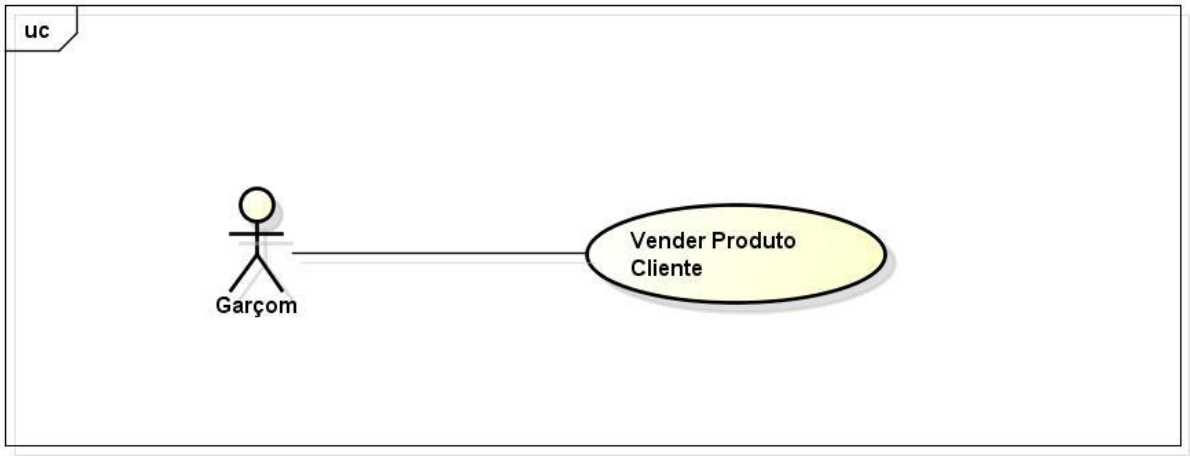


**Figura 12 – Vender Produto Cliente**

| | |
|---|---|
| **Nome do Caso de Uso** | Vender Produto Cliente |
| **Ator Principal** | Garçom |
| **Atores Secundários** | |
| **Resumo** | O usuário realiza a efetuação de venda ao cliente. |
| **Pré-Condições** | O usuário deve estar conectado ao sistema. |
| **Pós-Condições** | O servidor recebe o pedido realizado pelo cliente. |
| **Fluxo Principal** | 1. Usuário verifica o nome do cliente no sistema.<br>2. Sistema informa o nome do cliente e sua situação.<br>3. Usuário digita o pedido do cliente e confirma.<br>4. Sistema emite ao servidor o pedido, e retorna a mensagem “Pedido Realizado”. |
| **Fluxos Alternativos** | 1. Usuário cancela o pedido do cliente.<br>2. Sistema retorna à pagina anterior. |
| **Fluxos de Exceções** | 1. Usuário digita o pedido do cliente e confirma.<br>2. Sistema emite mensagem de “Saldo Insuficiente”. |

**Tabela 11 – Vender Produto Cliente**

### 4.1.12 Retirar Crédito

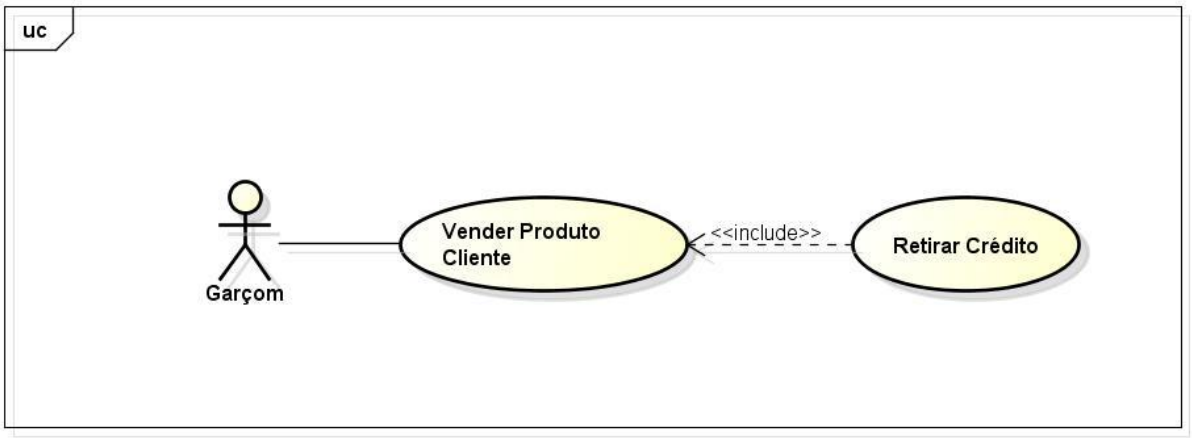


**Figura 13 – Retirar Crédito**

| | |
|---|---|
| **Nome do Caso de Uso** | Retirar Crédito |
| **Ator Principal** | Garçom |
| **Atores Secundários** | |
| **Resumo** | Será debitado do crédito do cliente o valor do produto consumido. |
| **Pré-Condições** | O usuário deve estar conectado ao sistema. |
| **Pós-Condições** | O saldo será atualizado conforme o valor retirado de crédito. |
| **Fluxo Principal** | 1. Usuário emite os produtos adquiridos pelo cliente ao sistema.<br>2. Sistema retorna o valor a ser debitado pelo produto.<br>3. Usuário confirma a retirada de crédito do cliente.<br>4. Sistema emite ao servidor o saldo debitado e atualiza, retornando a mensagem de “Débito efetuado”. |
| **Fluxos Alternativos** | 1. Usuário cancela a transição de debito do cliente.<br>2. Sistema emite a mensagem “Deseja continuar?”, e aguarda confirmação.<br>3. Usuário confirma a mensagem.<br>4. Sistema retorna a pagina anterior. |
| **Fluxos de Exceções** | 1. Usuário digita o pedido do cliente e confirma.<br>2. Sistema emite mensagem de “Saldo Insuficiente”. |

**Tabela 12 – Retirar Crédito**

### 4.1.13 Atualizar Quantidade Produto

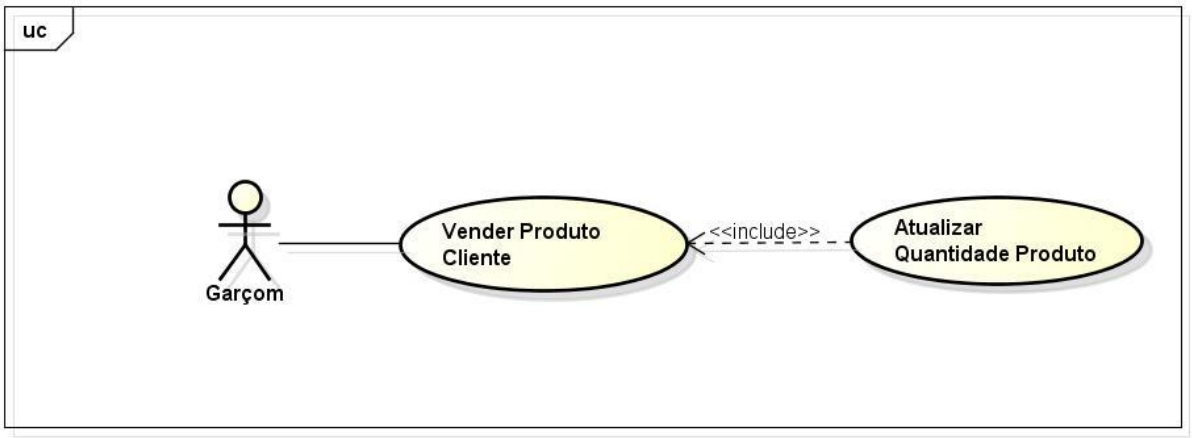


**Figura 14 – Atualizar Quantidade Produto**

| | |
|---|---|
| **Nome do Caso de Uso** | Atualizar Quantidade Produto |
| **Ator Principal** | Garçom |
| **Atores Secundários** | |
| **Resumo** | Realização da atualização da quantidade de produtos disponíveis em estoque. |
| **Pré-Condições** | O usuário deve estar conectado ao sistema. |
| **Pós-Condições** | O sistema retorna todas as atualizações realizadas. |
| **Fluxo Principal** | 1. Usuário clica em Verificar atualizações de estoque.<br>2. Sistema abre uma nova pagina com opções.<br>3. Usuário seleciona Atualização de Produtos, e confirma.<br>4. Sistema atualiza seu banco de dados, e retorna novas atualizações realizadas. |
| **Fluxos Alternativos** | 1. Usuário cancela a verificação de atualizações.<br>2. Sistema retorna a pagina anterior. |

**Tabela 13 – Atualizar Quantidade Produto**

### 4.1.14 Vender Produto Mesa

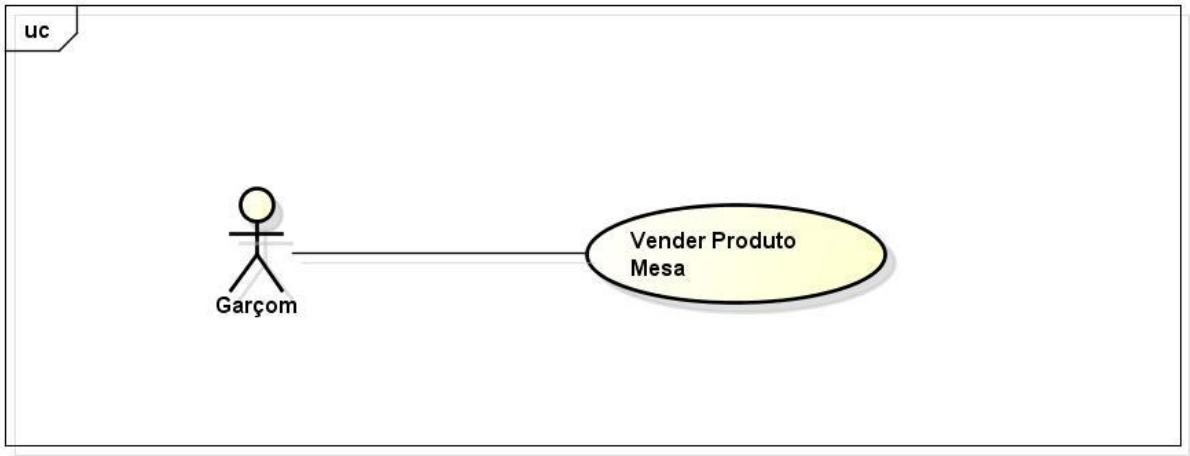


**Figura 15 – Vender Produto Mesa**

| | |
|---|---|
| **Nome do Caso de Uso** | Vender Produto Mesa |
| **Ator Principal** | Garçom |
| **Atores Secundários** | |
| **Resumo** | O usuário realiza a efetuação de venda à uma mesa associada a um ou mais clientes. |
| **Pré-Condições** | O usuário deve estar conectado ao sistema. |
| **Pós-Condições** | O servidor recebe o pedido realizado pela mesa cliente. |
| **Fluxo Principal** | 1. Usuário verifica a descrição da mesa do cliente no sistema.<br>2. Sistema informa o nome dos clientes associados à mesa e informa sua situação.<br>3. Usuário digita o nome e o pedido de cada cliente associado à mesa.<br>4. Sistema emite mensagem de confirmação.<br>5. Usuário confirma a mensagem.<br>6. Sistema emite o pedido ao servidor. |
| **Fluxos Alternativos** | 1. Usuário cancela o pedido.<br>2. Sistema retorna a pagina anterior. |
| **Fluxos de Exceções** | 1. Usuário digita o pedido dos associados a mesa e confirma.<br>2. Sistema emite mensagem de “ Saldo Insuficiente”. |

**Tabela 14 – Vender Produto Mesa**

### 4.1.15 Retirar Crédito Ponderado

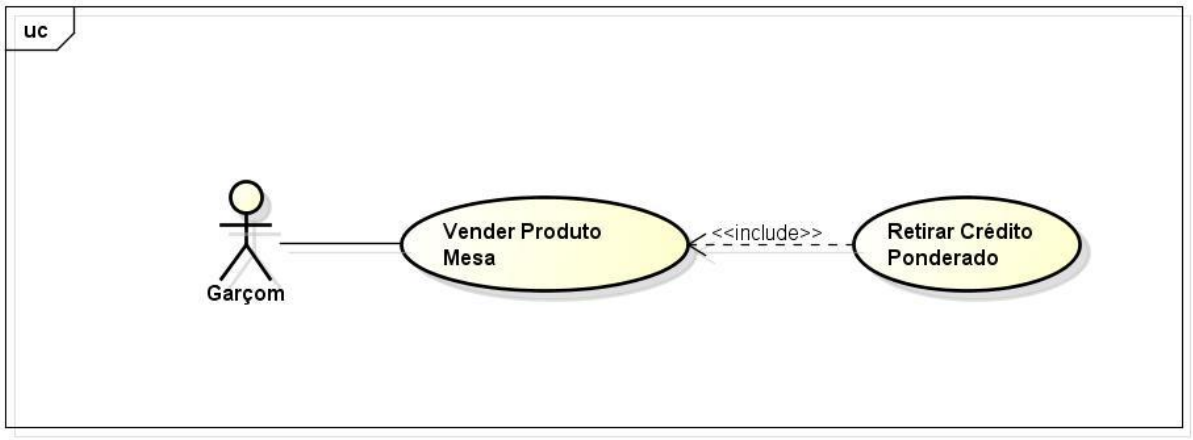


**Figura 16 – Retirar Crédito Ponderado**

| | |
|---|---|
| **Nome do Caso de Uso** | Retirar Crédito Ponderado |
| **Ator Principal** | Garçom |
| **Atores Secundários** | |
| **Resumo** | Será debitado do crédito dos clientes de uma mesa o valor do produto consumido. |
| **Pré-Condições** | O usuário deve estar conectado ao sistema. |
| **Pós-Condições** | O saldo será atualizado conforme o valor retirado de crédito. |
| **Fluxo Principal** | 1. Usuário emite os produtos adquiridos pelo pela mesa ao sistema.<br>2. Sistema retorna o valor a ser debitado pelo produto.<br>3. Usuário confirma a retirada de crédito da mesa.<br>4. Sistema emite ao servidor o saldo debitado e atualiza, retornando a mensagem de “Débito efetuado”. |
| **Fluxos Alternativos** | 1. Usuário cancela a transição de debito da mesa.<br>2. Sistema emite a mensagem “Deseja continuar?”, e aguarda confirmação.<br>3. Usuário confirma a mensagem.<br>4. Sistema retorna à pagina anterior. |
| **Fluxos de Exceções** | 1. Usuário digita o pedido da mesa dos clientes e confirma.<br>2. Sistema emite mensagem de “Saldo Insuficiente”. |

**Tabela 15 – Retirar Crédito Ponderado**

### 4.1.16 Atualizar Quantidade Produto

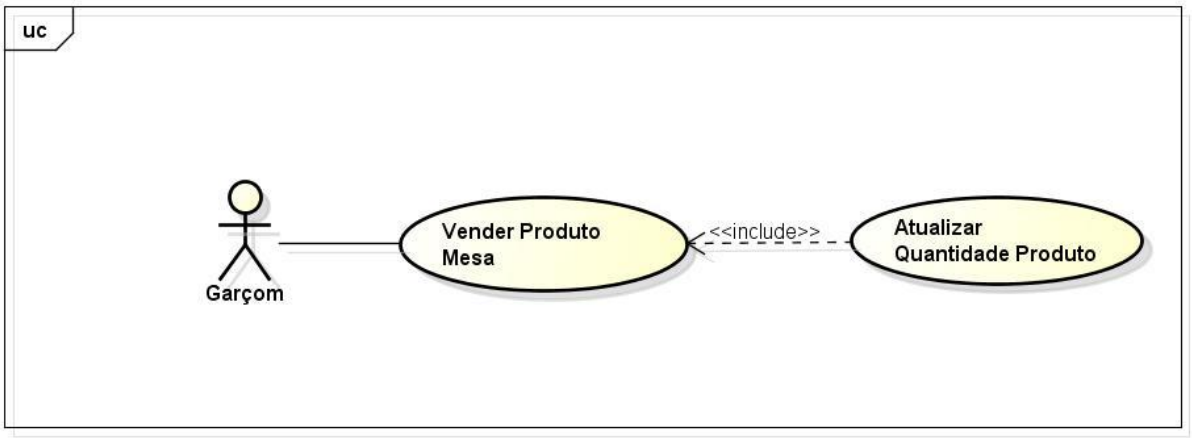


**Figura 17 – Atualizar Quantidade Produto**

| | |
|---|---|
| **Nome do Caso de Uso** | Atualizar Quantidade Produto |
| **Ator Principal** | Garçom |
| **Atores Secundários** | |
| **Resumo** | Realização da atualização da quantidade de produtos disponíveis em estoque. |
| **Pré-Condições** | O usuário deve estar conectado ao sistema. |
| **Pós-Condições** | O sistema retorna todas as atualizações realizadas. |
| **Fluxo Principal** | 1. Usuário clica em Verificar atualizações de estoque.<br>2. Sistema abre uma nova pagina com opções.<br>3. Usuário seleciona Atualização de Produtos, e confirma.<br>4. Sistema atualiza seu banco de dados, e retorna novas atualizações realizadas. |
| **Fluxos Alternativos** | 1. Usuário cancela a verificação de atualizações.<br>2. Sistema retorna a pagina anterior. |

**Tabela 16 – Atualizar Quantidade Produto**

## 4.2 DIAGRAMA DE ATIVIDADE

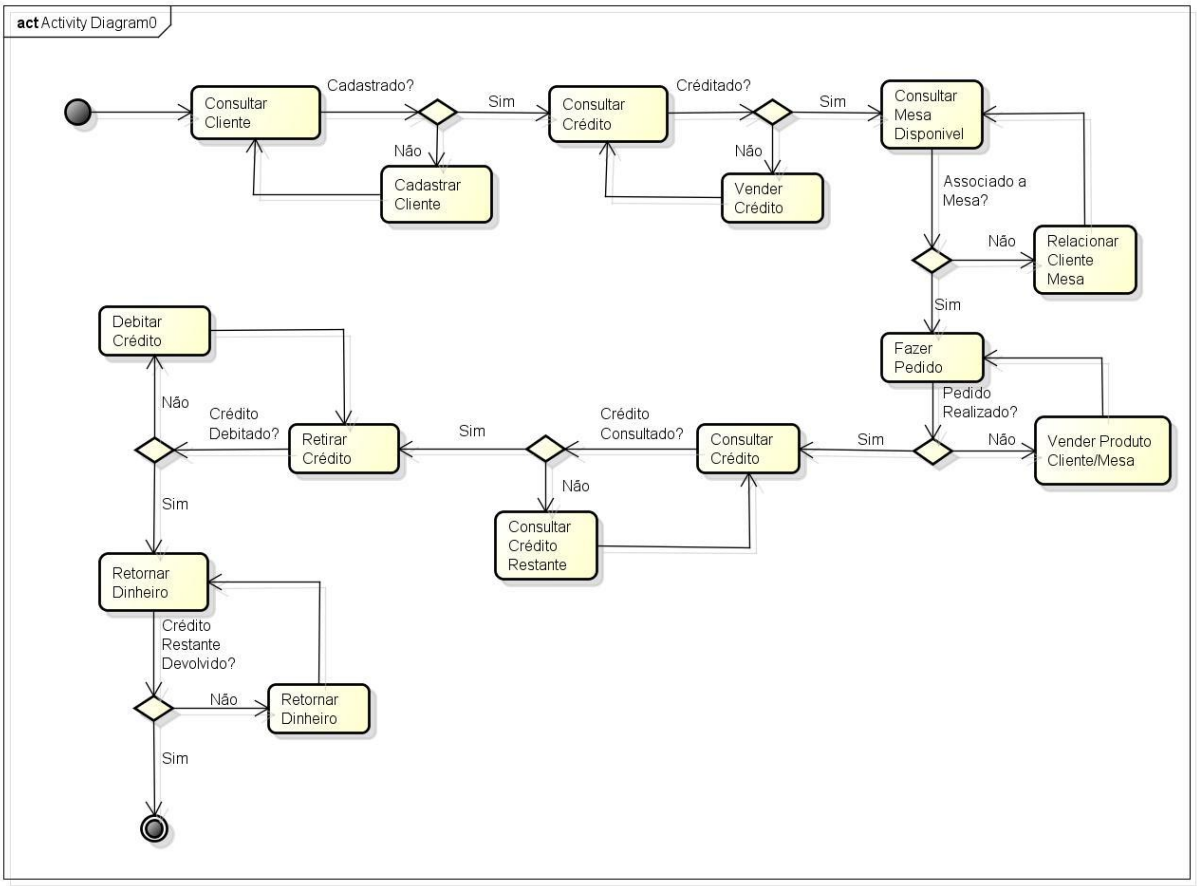


**Figura 18 – Diagrama de Atividade**

## 4.3 DIAGRAMA DE CLASSES

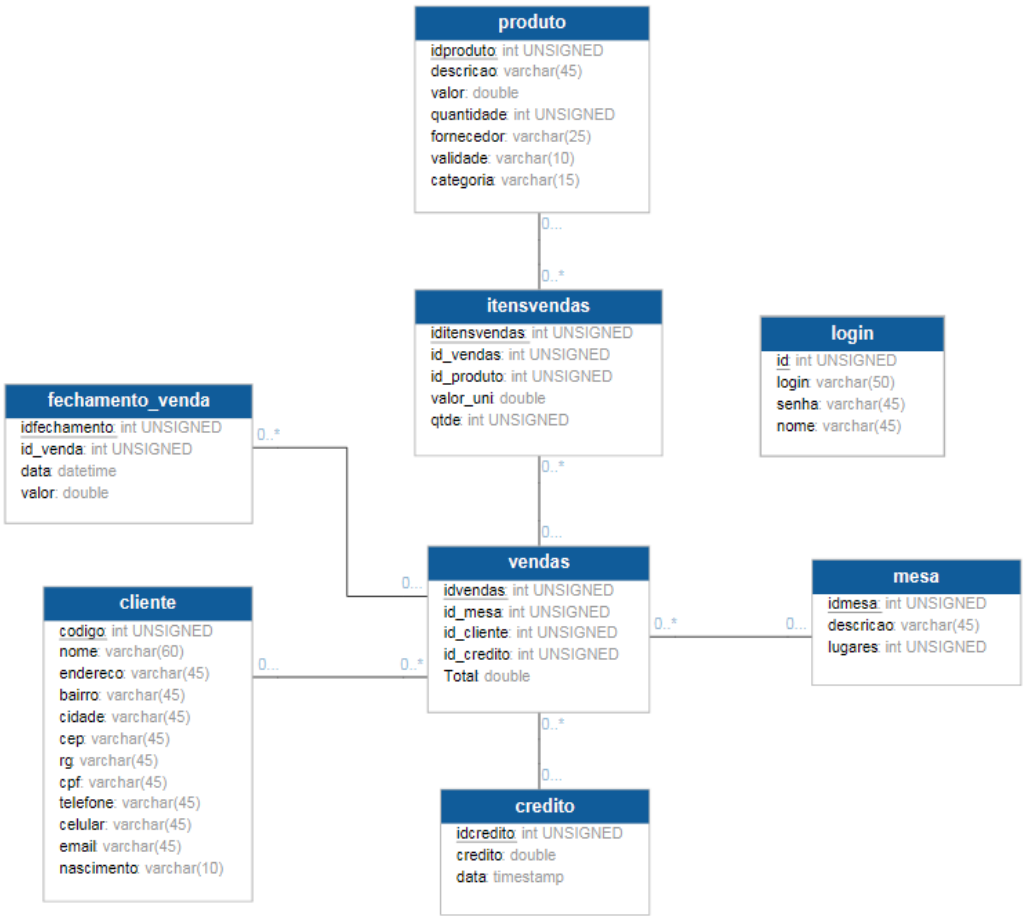


**Figura 19 – Diagrama de Classe**

## 4.4 DIAGRAMA DE ENTIDADE RELACIONAMENTO

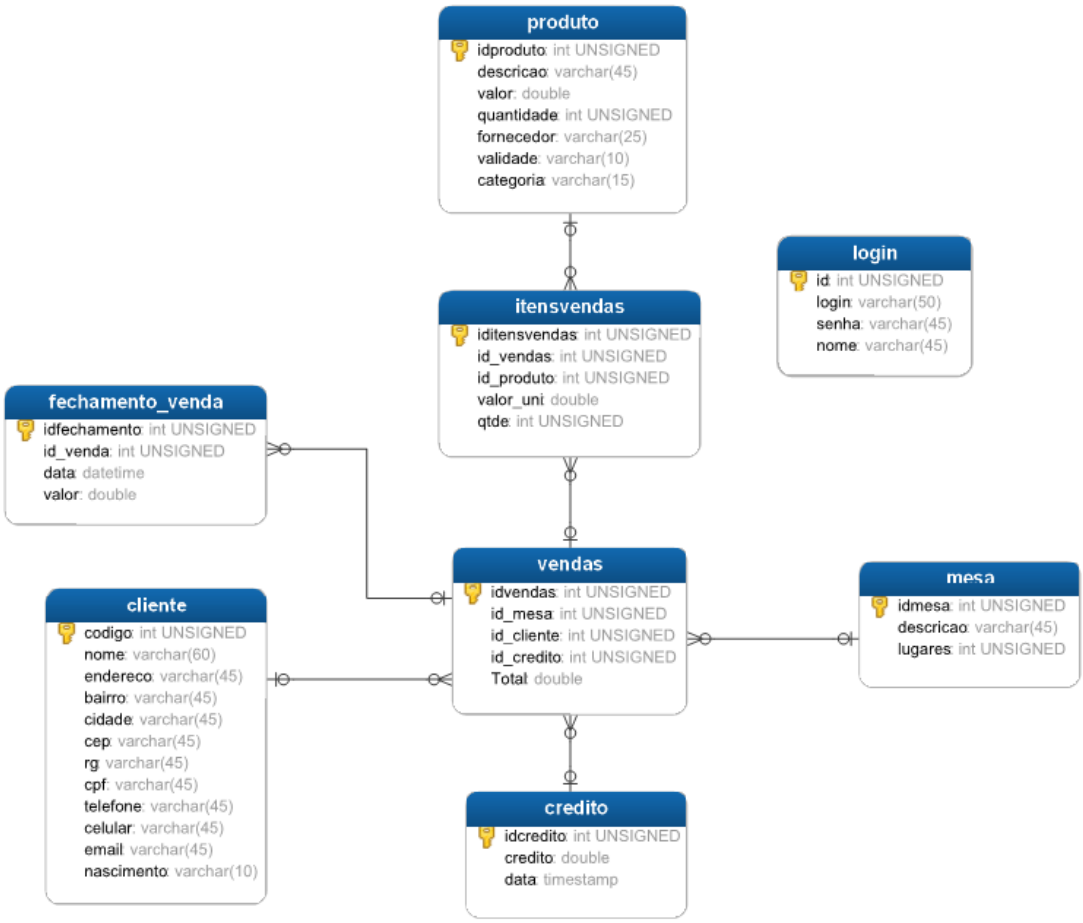


**Figura 20 – Diagrama de Entidade Relacionamento**

# 5. CRONOGRAMA

## 5.1 SEQUENCIAMENTO DAS ATIVIDADES

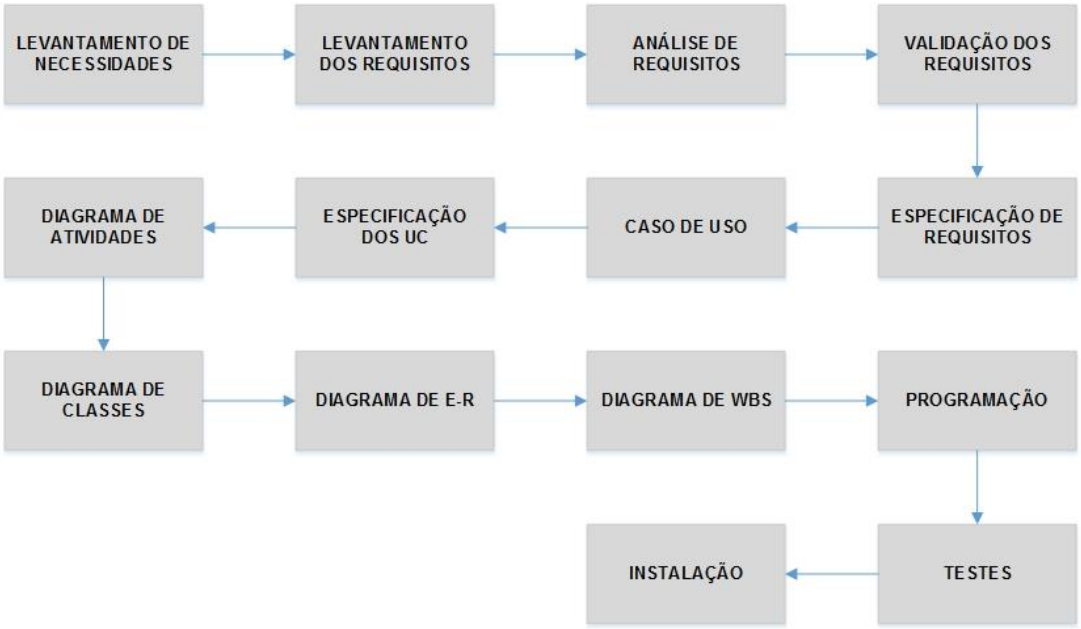


**Figura 21 – Sequenciamento de Atividades**

## 5.2 CRONOGRAMA DE REALIZAÇÃO DAS ATIVIDADES

| Cronograma de Realização das Atividades | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Tarefa** | **Out** | **Nov** | **Dez** | **Jan** | **Fev** | **Mar** | **Abr** | **Mai** | **Jun** | **Jul** | **Ago** |
| Pré-projeto | | | | | | | | | | | |
| Levantamento de Necessidades | | | | | | | | | | | |
| Levantamento de Requisitos | | | | | | | | | | | |
| Análise de Requisitos | | | | | | | | | | | |
| Validação de Requisitos | | | | | | | | | | | |
| Especificação de Requisitos | | | | | | | | | | | |
| Casos de Uso | | | | | | | | | | | |
| Especificação dos UC | | | | | | | | | | | |
| Diagrama de Atividades | | | | | | | | | | | |
| Diagrama de Classe | | | | | | | | | | | |
| Diagrama de E-R | | | | | | | | | | | |
| Diagrama de WBS | | | | | | | | | | | |
| Programação | | | | | | | | | | | |
| Testes | | | | | | | | | | | |
| Instalação | | | | | | | | | | | |

**Tabela 17 – Cronograma das Atividades**

# 6. IMPLEMENTAÇÃO

## 6.1 BANCO DE DADOS

### 6.1.1 Script da tabela itensvendas

```
CREATE TABLE `itensvendas` (
 `iditensvendas` int(10) unsigned NOT NULL AUTO_INCREMENT,  `id_vendas` int(10) unsigned DEFAULT NULL,  `id_produto` int(10) unsigned DEFAULT NULL,  `valor_uni` double NOT NULL,  `qtde` int(10) unsigned NOT NULL, PRIMARY KEY (`iditensvendas`), KEY `produto_estrangeira` (`id_produto`), KEY `vendas_estrangeira` (`id_vendas`),  CONSTRAINT `produto_estrangeira` FOREIGN KEY (`id_produto`) REFERENCES `produto` (`idproduto`), CONSTRAINT `vendas_estrangeira` FOREIGN KEY (`id_vendas`) REFERENCES `vendas` (`idvendas`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

### 6.1.2 Script da Trigger adiciona_estoque

```
DELIMITER $$
CREATE DEFINER = `root`@`localhost` TRIGGER `adiciona_estoque` BEFORE DELETE ON `itensvendas` FOR EACH ROW BEGIN
UPDATE produto SET quantidade =  quantidade + OLD.qtde
WHERE idproduto = OLD.id_produto;
END $$
DELIMITER ;
```

### 6.1.3 Classe de conexão com Banco de dados

```
public class ConnectionFactory {
public static Connection getConnection() {
final String driver = "com.mysql.jdbc.Driver";
final String local = "jdbc:mysql://127.0.0.1/sistema";
final String login = "root";
final String senha = "";
Connection conexao = null;
try{
    Class.forName(driver);
    conexao = DriverManager.getConnection(local, login, senha);
} catch(ClassNotFoundException e){
    System.out.println("Driver não encontrado!");
    return conexao; }   }
```

# 7. RESULTADOS

## 7.1 TELA DE ACESSO

**Figura 22 – Tela de Acesso**

## 7.2 TELA DE VENDA

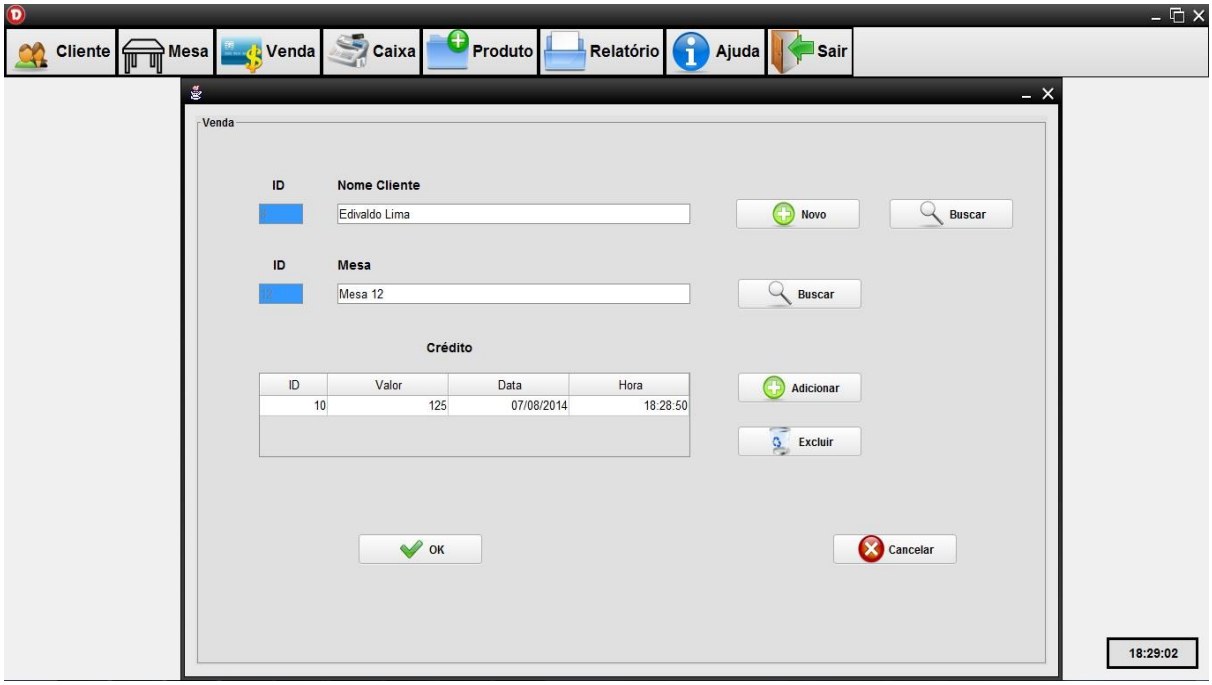


**Figura 23 – Tela de Venda**

## 7.3 TELA DE REALIZAÇÃO DE PEDIDOS

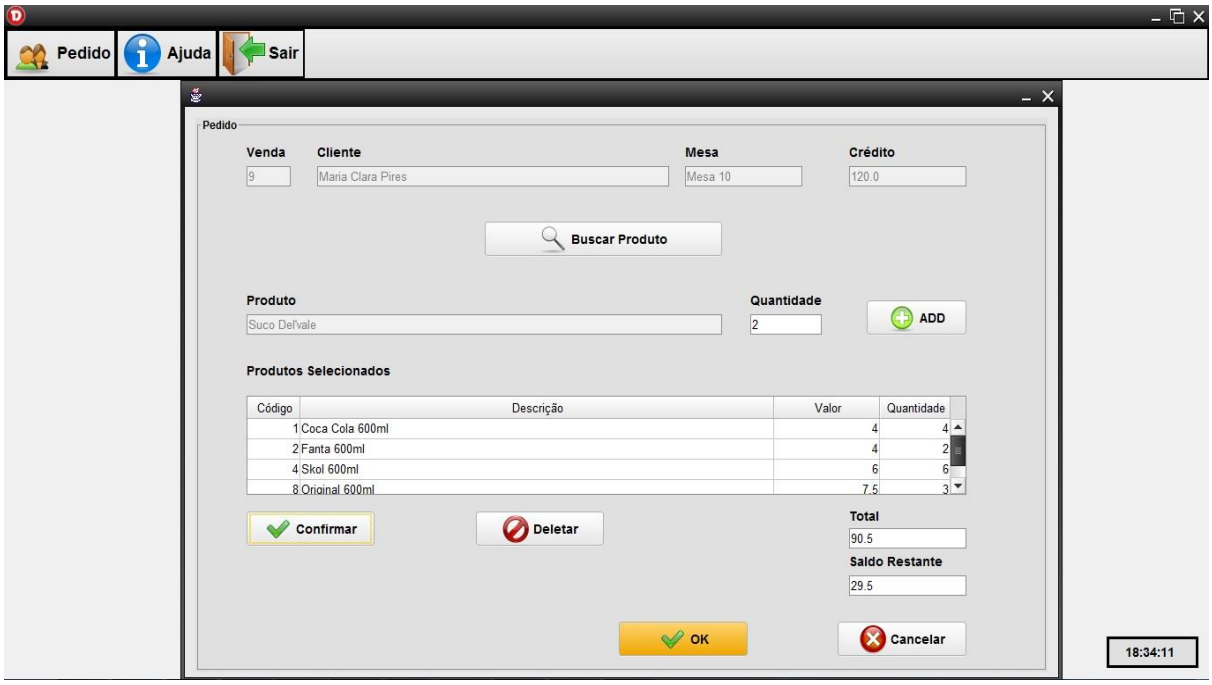


**Figura 24 – Tela de Realização de Pedidos**

## 7.4 TELA DE FLUXO DE CAIXA

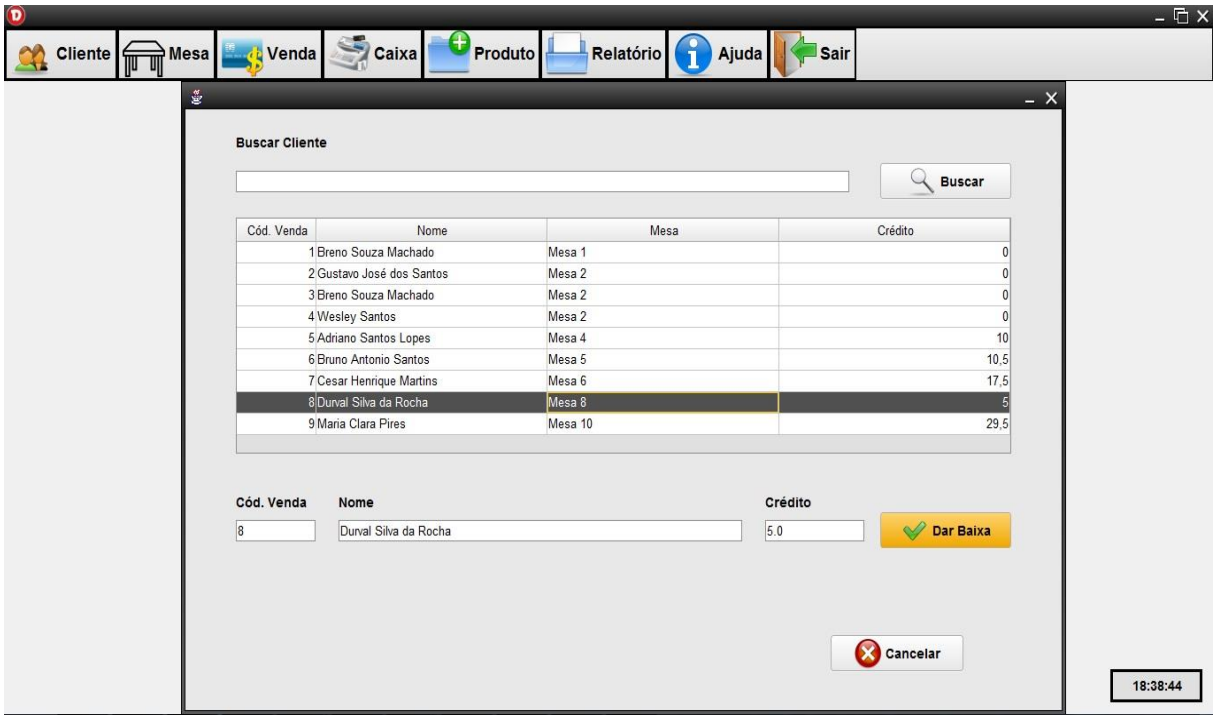


**Figura 25 – Tela de Fluxo de Caixa**

# 8. CONCLUSÃO

Com a conclusão do desenvolvimento sistema pode-se afirmar que o sistema é de grande utilidade para pequenas e médias empresas do seguimento alimentício, pois após diversos testes foi possível notar que o software traz grande segurança e agilidade nas tomadas de decisões e no controle das transações efetuadas diariamente dentro dos devidos estabelecimentos, como na realização de pedidos, pagamentos e relatórios.

Dessa forma, é possível afirmar que os objetivos do sistema foram alcançados, tendo em vista que todas suas definições e funcionalidades foram concluídas com êxito, entretanto, existem algumas regras de negócios suplementares a serem feitas posteriormente para maior complexidade das ações do sistema.

# REFERÊNCIAS

ARAÚJO, Marcos Antônio. Artigo SQL Magazine 35: DB Designer, uma ferramenta gratuita para modelagem de dados. Disponível em: <http://www.devmedia.com.br/dbdesigner-uma-ferramenta-gratuita-para-modelagem-de-dados-artigo-sql-magazine-35/6840>. Acesso em 12 de Fevereiro de 2014.

BALAGTAS, Florence Tiu. Artigo Java Education & Development Initiative. Disponível em: <http://www.oficinadanet.com.br/apostilas/detalhe/760/apostila_completa_de_java_-_programacao_orientada_a_objetos>. Acesso em 23 de Janeiro de 2014.

FILHO, Rafael Dicena. Dominando o MS Visio em 20 passos. Disponível em: <http://www.devmedia.com.br/dominando-ms-visio--em-20-passos-melhores-praticas-em-gestao-de-projetos/20276> Acesso em 03 de Fevereiro de 2014.

GALLARDO, David. Introdução à Plataforma Eclipse. Disponível em: <https://www.ibm.com/developerworks/br/library/os-eclipse-platform/> Acesso em 05 de Março de 2014.

GUEDES, Gilleanes T. A. UML2: Uma abordagem prática. São Paulo: Novatec Editora, 2009.

LARMAN, Craig. Utilizando UML e padrões: uma introdução à analise e ao projeto orientado a objetos. Porto Alegre: Bookman Editora, 2000.

PEDROSO, Robertha Pereira. Guia MySQL. Niterói, Rio de janeiro, 2007.

ORACLE. Obtenha Informações sobre a Tecnologia Java. Disponível em: <http://www.java.com/pt_BR/about>. Acesso em 10 de Fevereiro de 2014.

# ANEXO

BOTÃO CADASTRO DE CRÉDITO

```
private void btnAddCreditoActionPerformed(java.awt.event.ActionEvent evt) {
        Connection con = ConnectionFactory.getConnection();
        String credito = txtCredito.getText();
        if (txtCredito.getText().isEmpty()) {
            JOptionPane.showMessageDialog(null, "Digite um valor", "Campo
Obrigátorio", JOptionPane.ERROR_MESSAGE);
        } else {
            Credito creditomodel = new Credito();
            creditomodel.setCredito(Double.parseDouble(credito));
            creditomodel.setData(new Date());
            try {
                creditoDao creditodao = new creditoDao(con);
                creditodao.CadCredito(creditomodel);

                JOptionPane.showMessageDialog(this, "Creditado com sucesso!",
"Crédito", JOptionPane.INFORMATION_MESSAGE);

                listarCredito();
                dispose();

            } catch (SQLException se) {
                JOptionPane.showMessageDialog(null, "O seguinte Erro ocorreu:" +
se.getMessage(), "", JOptionPane.ERROR_MESSAGE);
            }
        }
    }
```

A ação de o botão Cadastrar Crédito adiciona um valor de crédito definido pelo cliente, trazendo a data e a hora da realização da ação, inserido diretamente ao banco de dados.

BOTÃO CONFIRMAR VENDA

```
private void btnConfirmarVendaActionPerformed(java.awt.event.ActionEvent evt) {
Connection con = ConnectionFactory.getConnection();
        vendasDao vendasdao = new vendasDao(con);

        int idcliente = Integer.parseInt(txtIdcliente.getText());
        int idmesa = Integer.parseInt(txtIdmesa.getText());
        System.out.println(tbCredito.getValueAt(0, 0));
        int idcredito = Integer.parseInt(tbCredito.getValueAt(0, 0) + "");

        Vendas v = new Vendas();

        v.setIdcliente(idcliente);
        v.setIdmesa(idmesa);
        v.setIdcredito(idcredito);

        try {

            int opcao;
            opcao = JOptionPane.showConfirmDialog(null, "Deseja efetuar a venda? ",
"Atenção", JOptionPane.YES_NO_OPTION);
            if (opcao == JOptionPane.YES_OPTION) {
                vendasdao.iniciarVendas(v);
                btnConfirmarVenda.setEnabled(false);
                JOptionPane.showMessageDialog(null, "Venda realizada com sucesso!",
"Venda", JOptionPane.INFORMATION_MESSAGE);
            }
        } catch (SQLException ex) {
            Logger.getLogger(frmVenda.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
```

O botão confirmar venda fica responsável por fazer todas as associações necessárias para efetivar uma venda, vincula o cliente, a mesa e o crédito à tabela vendas, sendo assim, o cliente terá saldo para adquirir produtos.

BOTÃO ADICIONAR PRODUTO AO CLIENTE

```
 private void btnAddProdutoActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel m = (DefaultTableModel) tbProdutosSelecionados.getModel();
m.addRow(new Object[]{produtoselecionado.getIdproduto(),
produtoselecionado.getDescricao(), produtoselecionado.getValor(),
txtQuantidadeProduto.getText()});

        Itens_venda itens = new Itens_venda();

        itens.setId_venda(getVenda().getIdvendas());
        itens.setId_produto(produtoselecionado.getIdproduto());
        itens.setValor_unit(produtoselecionado.getValor());
        itens.setQtde(Integer.parseInt(txtQuantidadeProduto.getText()));

        try {
            ItensVendaDao dao = new ItensVendaDao(con);
            dao.Inserir_produto(itens);

        } catch (SQLException ex) {

            Logger.getLogger(FrmPedido.class.getName()).log(Level.SEVERE, null, ex);
        }

        double soma = 0;
        int qtd = 0;

   for (int i = 0; i < m.getRowCount(); i++) {
   qtd = Integer.parseInt(tbProdutosSelecionados.getValueAt(i, 3) + "");
   soma += Double.parseDouble(tbProdutosSelecionados.getValueAt(i, 2) + "") * qtd;
        }
        txtTotal.setText(Double.toString(soma));
    }
```

A ação de o Botão Adicionar Produto realiza o *insert* do produto selecionado pelo cliente ao banco de dados, verifica se o produto já foi adicionado anteriormente, caso tenha sido, ele realiza um *update* no banco, e então lista o produto em uma tabela do sistema, e por fim realiza a somatória dos produtos adicionados.