



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

RAUL GOULART RODRIGUES

GESTÃO DE CONTROLE DE MADEIRAS DE DEMOLIÇÃO

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

Orientador: Prof. Domingos de Carvalho Villela Junior

Área de Concentração: Desenvolvimento de Sistemas

Assis

2012



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

GESTÃO DE CONTROLE DE MADEIRAS DE DEMOLIÇÃO

RAUL GOULART RODRIGUES

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

Orientador: Prof. Domingos de Carvalho Villela Junior

Avaliador: Prof. Dr. Luiz Carlos Begosso

Assis

2012



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

FICHA CATALOGRÁFICA

GOULART RODRIGUES, Raul.

**GESTÃO DE CONTROLE DE MADEIRAS DE DEMOLIÇÃO / Raul Goulart
Rodrigues. Fundação Educacional do Município de Assis – FEMA – Assis, 2012.
52p.**

Orientador: Prof. Domingos de Carvalho Villela Junior

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

1. Madeireira 2. Sistema *Desktop*

**CDD: 001.61
Biblioteca da FEMA**

DEDICATÓRIA

Dedico este trabalho a minha esposa Mariana da Silva Paião, meus filhos Isadora Paião Rodrigues e Caio Paião Rodrigues e a todos os amigos e familiares por apoiarem minha luta e incentivarem e valorizarem meus esforços.

AGRACECIMENTOS

Em primeiro lugar agradeço a Deus que proporcionou tudo que obtive até hoje, minha família, meus estudos, meus bens, minha vontade de vencer e ser um grande profissional.

Ao Sr. Domingos de Carvalho Villela Junior, professor e orientador que me instruiu neste trabalho com muita responsabilidade e eficiência, e a todos os outros professores que me ajudaram seja em aulas ou em dúvidas diversas.

Meus pais, Rubens Bento Rodrigues e Rozana Goulart Rodrigues, que contribuíram de todas as maneiras possíveis no decorrer do curso.

RESUMO

Com a tendência da informatização e automatização de quase tudo que conhecemos as madeiras não poderiam ficar de fora. Mesmo que seja um trabalho manual e de origens conservatórias, é necessário que haja um gerenciamento de um grande volume de dados gerados pelas suas movimentações. Esses dados necessitam ser validados e processados para proporcionar determinadas informações de caráter decisório no rumo e objetivos da empresa.

Neste projeto será desenvolvido um sistema de gestão para controle empresarial da madeira Peroba Rosa – Madeiras de Demolição. Inicialmente foram coletados requisitos primordiais para análise e desenvolvimento da documentação e também foram discutidos os pontos críticos para possível melhoria e os pontos positivos para maximização.

O sistema contará com o conceito de ERP para seu desenvolvimento e estrutura e será desenvolvido na plataforma Microsoft Visual Studio 2010 (.NET) com linguagem C# e banco de dados MySQL.

Palavras-Chave: Gestão de Controle. Madeireira. Madeira de Demolição.

ABSTRACT

With the trend of computerization and automation of almost everything that we know the timber could not be left out. Even if it is a manual origins and conservatories, there needs to be a management of a large volume of data generated by their movements. These data need to be validated and processed to provide certain information bearing on the decision of character and objectives of the company.

This project will develop a management system to control the logging business Peroba Rosa - Demolition Woods. Initially were collected prime requisites for analysis and development of documentation and also discussed the critical points for possible improvement and to maximize the positives.

The system will rely on the concept of ERP for its development and structure and will be developed on the Microsoft Visual Studio 2010 (. NET) with C # language and MySQL database

Keyword: Management Control. Timber. Demolition wood.

LISTA DE FIGURAS

Figura 1 – Diagrama de Casos de Uso.....	25
Figura 2 - UC 1 – Manter Usuário.....	26
Figura 3 - UC 2 – Manter Vendedor	27
Figura 4 - UC 3 – Manter Cliente	28
Figura 5 - UC 4 – Manter Fornecedor	29
Figura 6 - UC 5 – Manter Produto.....	30
Figura 7 - UC 6 – Manter Grupo	31
Figura 8 - UC 7 – Manter Sub-Grupo	32
Figura 9 - UC 8 – Manter Localização de Estoque	33
Figura 10 - UC 9 – Manter Conta.....	34
Figura 11 - UC 10 – Manter Banco.....	35
Figura 12 - UC 11 – Manter Tipo de Movimentação Bancária.....	36
Figura 13 - UC 12 – Manter Grupo de Classificação Financeira	37
Figura 14 - UC 13 – Manter Tipo de Baixa Financeira	38
Figura 15 - UC 14 – Movimentar Venda	39
Figura 16 - UC 15 – Movimentar Compra	40
Figura 17 - UC 16 – Movimentar Estoque	41
Figura 18 - UC 17 – Movimentar Contas a Receber	42
Figura 19 - UC 18 – Movimentar Contas a Pagar	43
Figura 20 - UC 19 – Movimentar Banco e Caixa.....	44
Figura 21 – Emitir Relatório Cliente.....	45
Figura 22 – Emitir Relatório Vendas.....	45
Figura 23 – Emitir Relatório Carregamento.....	46
Figura 24 – Emitir Relatório Fornecedor.....	46
Figura 25 – Emitir Relatório Compras.....	47
Figura 26 – Emitir Relatório Produto	47
Figura 27 – Emitir Relatório Estoque	48
Figura 28 – Emitir Relatório Contas a Receber	48
Figura 29 – Emitir Relatório Contas a Pagar	49
Figura 30 – Emitir Relatório Banco e Caixa	49
Figura 31 – Diagrama de Classes	50
Figura 32 - Cronograma	51

LISTA DE TABELAS

TABELA 1 - MANTER USUÁRIO	26
TABELA 2 - MANTER VENDEDOR	27
TABELA 3 - MANTER CLIENTE	28
TABELA 4 - MANTER FORNECEDOR	29
TABELA 5 - MANTER PRODUTO.....	30
TABELA 6 - MANTER GRUPO	31
TABELA 7 - MANTER SUB-GRUPO	32
TABELA 8 - MANTER BANCO	35
TABELA 9 - MANTER TIPO DE MOVIMENTAÇÃO BANCÁRIA	36
TABELA 10 - MANTER GRUPO DE CLASSIFICAÇÃO FINANCEIRA.....	37
TABELA 11 - MANTER TIPO DE BAIXA FINANCEIRA	38
TABELA 12 - MOVIMENTAR VENDA	39
TABELA 13 - MOVIMENTAR COMPRA.....	40
TABELA 14 - MOVIMENTAR ESTOQUE.....	41
TABELA 15 - MOVIMENTAR CONTAS A RECEBER	42
TABELA 16 - MOVIMENTAR CONTAS A PAGAR.....	43

SUMÁRIO

ORÇAMENTO.....	12
CAPÍTULO 1 – INTRODUÇÃO	13
1.1. INTRODUÇÃO	13
1.2. OBJETIVO	14
1.3. PÚBLICO ALVO.....	14
1.4. JUSTIFICATIVA.....	14
1.5. METODOLOGIA DE DESENVOLVIMENTO.....	15
1.5.1. MICROSOFT VISUAL STUDIO 2010	15
1.5.2. LINGUAGEM C#.....	16
1.5.3. BANCO DE DADOS MYSQL	19
1.5.4. PROGRAMAÇÃO ORIENTADA A OBJETOS.....	20
1.5.5. ASTAH	20
1.6. ESTRUTURA DO TRABALHO.....	20
1.7. METODOLOGIA DE ANÁLISE.....	21
1.7.1. LEVANTAMENTO E ESPECIFICAÇÃO DOS REQUISITOS	21
CAPÍTULO 2 – DOCUMENTAÇÃO.....	23
2.1. LISTA DE EVENTOS	23
2.2. DIAGRAMA DE CASOS DE USO.....	25
2.3. ESPECIFICAÇÕES DE CASOS DE USO.....	26
2.3.1. UC 1 – MANTER USUÁRIO	26
2.3.2. UC 2 – MANTER VENDEDOR.....	27
2.3.3. UC 3 – MANTER CLIENTE	28
2.3.4. UC 4 – MANTER FORNECEDOR.....	29
2.3.5. UC 5 – MANTER PRODUTO	30
2.3.6. UC 6 – MANTER GRUPO.....	31
2.3.7. UC 7 – MANTER SUB-GRUPO	32
2.3.8. UC 8 – MANTER LOCALIZAÇÃO DE ESTOQUE	33
2.3.9. UC 9 – MANTER CONTA	34
2.3.10. UC 10 – MANTER BANCO.....	35
2.3.11. MANTER TIPO DE MOVIMENTAÇÃO BANCÁRIA	36
2.3.12. UC 12 – MANTER GRUPO DE CLASSIFICAÇÃO FINANCEIRA	37
2.3.13. UC 13 – MANTER TIPO DE BAIXA FINANCEIRA.....	38

2.3.14. UC 14 – MOVIMENTAR VENDA.....	39
2.3.15. UC 15 – MOVIMENTAR COMPRA	40
2.3.16. UC 16 – MOVIMENTAR ESTOQUE	41
2.3.17. UC 17 – MOVIMENTAR CONTAS A RECEBER.....	42
2.3.18. UC 18 – MOVIMENTAR CONTAS A PAGAR	43
2.3.19. UC 19 – MOVIMENTAR BANCO E CAIXA.....	44
2.3.20. UC 20 – EMITIR RELATÓRIO CLIENTE	45
2.3.21. UC 21 – EMITIR RELATÓRIO VENDAS.....	45
2.3.22. UC 22 – EMITIR RELATÓRIO CARREGAMENTO.....	46
2.3.23. UC 23 – EMITIR RELATÓRIO FORNECEDOR.....	46
2.3.24. UC 24 – EMITIR RELATÓRIO COMPRAS.....	47
2.3.25. UC 25 – EMITIR RELATÓRIO PRODUTO	47
2.3.26. UC 26 – EMITIR RELATÓRIO ESTOQUE	48
2.3.27. UC 27 – EMITIR RELATÓRIO CONTAS A RECEBER	48
2.3.28. UC 28 – EMITIR RELATÓRIO CONTAS A PAGAR	49
2.3.29. UC 29 – EMITIR RELATÓRIO BANCO E CAIXA.....	49
DIAGRAMA DE CLASSES	50
CRONOGRAMA DE ATIVIDADES.....	51
REFERÊNCIAS BIBLIOGRÁFICAS	51

ORÇAMENTO

Neste tópico será descrito os recursos necessários para o desenvolvimento do projeto bem como seus respectivos custos.

01 Analista/Programador

- Custo Diário = R\$ 40,00
- Total de dias = 122
- **Total do Custo = (Custo Diário * Total de dias) = R\$ 4.880,00**

01 Notebook

- Valor Unitário = R\$ 2.000,00
- Dias de uso = 30 dias (08 horas de trabalho diário) = 240 horas
- Depreciação = R\$ 2.000,00 / 24 meses = R\$ 83,33/mês
- Custo dia = R\$ 83,33 / (30 dias) = R\$ 2,78/dia
- **Custo do notebook = R\$ 2,78 * 122 dias = R\$ 339,16.**

01 Impressora

- Valor Unitário = R\$ 350,00
- Dias de uso = 122 dias
- Depreciação: R\$ 350,00 / 24 = R\$ 14,58/mês
- Custo dia = R\$ 14,58 / 30 = R\$ 0,49/dia
- **Custo da impressora = R\$ 0,49 * 122 dias = R\$ 59,78.**

Software

- Microsoft Visual Studio 2010 – Free version;
- SAP Crystal Reports – Free version;
- Navicat for MySQL – Free version;
- Banco de Dados MySQL 5.5 – Open Source;

CUSTO TOTAL DO PROJETO = R\$ 5.278,94

CAPÍTULO 1 – INTRODUÇÃO

1.1. INTRODUÇÃO

A Peroba Rosa - Madeira de Demolição é uma empresa que está atuando no mercado de venda de madeira para reutilização, sendo possível a construção de móveis e instalação de pisos em madeira maciça. A madeira é originada da demolição de antigas casas, fazendas e barracões. A empresa atua no ramo devido a ser um ramo ecologicamente correto e que se preocupa com o meio ambiente e com a preservação das matas. A empresa também comercializa utensílios encontrados nas propriedades demolidas, que estejam em bom estado, como: luminárias, vasos, candelabros, fechaduras, etc.

A própria empresa faz a compra, demolição e coleta das propriedades armazenando em estoques próprios que podem estar em locais físicos diferentes. Há então a necessidade de um software que faça todo esse controle.

Na atual situação, a empresa Peroba Rosa – Madeira de Demolição têm executado os processos e tarefas de forma manual por seus colaboradores. Não há nenhum controle sobre quanto há disponível em estoque para venda e nem quanto há previsto para demolição. Os principais problemas de tudo ser feito de forma manual são: a falta de informação em tempo real para tomada de decisão, erros de cálculo em carregamentos que saem da empresa, falta de agilidade nos processos, necessidade de mão-de-obra especializada para tarefas básicas, centralização de informação em determinados colaboradores, falta de planejamento do cumprimento das obrigações e falta de acompanhamento das contas a receber.

O seguinte trabalho oferece uma ferramenta capaz de proporcionar o gerenciamento de todas as movimentações físicas referentes a madeiras e utensílios, controle dos serviços prestados a clientes e comercialização de móveis rústicos. Também oferece suporte a controle financeiro com enfoque na condução do cumprimento das obrigações a pagar e obtenção e acompanhamento das contas a receber afinando e proporcionando um simplificado plano de ação financeira.

1.2. OBJETIVO

O objetivo deste trabalho é desenvolver um software que gerencie principalmente os processos de compras e vendas de madeiras e utensílios originados de demolições. Também, não menos importante, controlará outras áreas funcionais da empresa como estoque, contas a pagar, contas a receber, bancos e caixa.

1.3. PÚBLICO ALVO

O projeto é voltado para empresas que trabalham com produtos originados de demolições, no geral, madeireiras que trabalham com a compra e a venda de madeiras, utensílios, etc.

1.4. JUSTIFICATIVA

O software está sendo desenvolvido para atender a uma necessidade solicitada pelo Adilson, proprietário da empresa Peroba Rosa onde a partir de uma conversa informal ele abordou as dificuldades encontradas em gerenciar uma empresa de médio porte sem ter um sistema de informação. A empresa vem crescendo a cada dia e tem-se a necessidade de guardar todas as informações e gerenciá-las de forma que proporcione um crescimento estruturado para a empresa. Outro fator também é a concorrência cada vez mais competitiva então para manter seu produto no mercado é necessário estar sempre procurando melhorar a empresa em todos os sentidos, e o sistema tem o objetivo de minimizar as dificuldades de mercado através de um bom planejamento de compras e logística de vendas.

1.5. METODOLOGIA DE DESENVOLVIMENTO

Para criação dos casos de uso, diagramas e mapa mental serão utilizadas as ferramentas Astah Professional e DBDesigner. Na fase do desenvolvimento do projeto será utilizada a plataforma Microsoft Visual Studio 2010 para programação em linguagem C# e integração com o banco de dados MySQL.

1.5.1. MICROSOFT VISUAL STUDIO 2010

O Microsoft Visual Studio é um pacote de programas da Microsoft para desenvolvimento de software especialmente dedicado ao .NET Framework e às linguagens Visual Basic (VB), C, C++, C# (C Sharp) e J# (J Sharp). Também é um grande produto de desenvolvimento na área web, usando a plataforma do ASP.NET. As linguagens com maior frequência nessa plataforma são: C# (lê-se Cê Sharp).

A 12 de Abril de 2010 a Microsoft lançou o Visual Studio 2010, de nome de código Dev10, e a .NET Framework 4. Foi lançado com o objetivo de ser a IDE mais completa disponível no mercado. Para isso conta com suporte a desenvolvimento de aplicativos Web, aplicativos para Windows Phone, SharePoint além de aprimorar os já conhecidos, Windows Forms, Web Forms e também plataformas como o Microsoft XNA. Oferece IntelliTrace, Gerenciamento de Ciclo de Vida de Aplicativos (ALM), uma nova interface desenvolvida com WPF (Windows Presentation Foundation), para tornar a IDE mais intuitiva, sistema de busca muito mais eficaz, dentre outros. Alguns recursos:

- Construir customizações para SharePoint;
- Desenvolver aplicações para Windows 7, utilizando seus recursos (Multitouch, Pin Bar, etc.);
- Compreender de forma mais simples códigos e arquitetura já existentes;
- Identificar por testes, os impactos de alterações no código.
- Customizar o Visual Studio para atender às suas necessidades.

1.5.2. LINGUAGEM C#

C# é uma linguagem de programação orientada a objetos criada pela Microsoft, faz parte da sua plataforma .Net. A companhia baseou C# na linguagem C++ e Java.

A linguagem C# foi criada junto com a arquitetura .NET. Embora existam várias outras linguagens que suportam essa tecnologia (como VB.NET, C++, J#), C# é considerada a linguagem símbolo do .NET pelas seguintes razões:

- Foi criada praticamente do zero para funcionar na nova plataforma, sem preocupações de compatibilidade com código de legado.
- O compilador C# foi o primeiro a ser desenvolvido.
- A maior parte das classes do .NET Framework foram desenvolvidas em C#.

A criação da linguagem, embora tenha sido feita por vários desenvolvedores, é atribuída principalmente a Anders Hejlsberg, hoje um Distinguished Engineer na Microsoft. Anders era desenvolvedor de compiladores na Borland, e entre suas criações estão o Turbo Pascal e o Delphi.

Características

C# (pronuncia-se "cê chárp" em português ou "cí charp" em inglês) é, de certa forma, a linguagem de programação que mais diretamente reflete a plataforma .NET sobre a qual todos os programas .NET executam. C# está de tal forma ligado a esta plataforma que não existe o conceito de código não-gerenciado (unmanaged code) em C#. Suas estruturas de dados primitivas são objetos que correspondem a tipos em .NET. A desalocação automática de memória por garbage collector além de várias de suas abstrações tais como classes, interfaces, delegados e exceções são nada mais que a exposição explícita recursos do ambiente .NET.

Quando comparada com C e C++, a linguagem é restrita e melhorada de várias formas incluindo:

- Ponteiros e aritmética sem checagem só podem ser utilizados em uma modalidade especial chamada modo inseguro (unsafe mode). Normalmente os acessos a objetos são realizados através de referências seguras, as quais não podem ser invalidadas e normalmente as operações aritméticas são checadas contra sobrecarga (overflow).
- Objetos não são liberados explicitamente, mas através de um processo de coleta de lixo (garbage collector) quando não há referências aos mesmos, prevenindo assim referências inválidas.
- Destrutores não existem. O equivalente mais próximo é a interface Disposable, que juntamente com a construção using block permitem que recursos alocados por um objeto sejam liberados prontamente. Também existem finalizadores, mas como em Java sua execução não é imediata.
- Como no Java, não é permitida herança múltipla, mas uma classe pode implementar várias interfaces abstratas. O objetivo principal é simplificar a implementação do ambiente de execução.
- C# é mais seguro com tipos que C++. As únicas conversões implícitas por default são conversões seguras, tais como ampliação de inteiros e conversões de um tipo derivado para um tipo base. Não existem conversões implícitas entre inteiros e variáveis lógicas ou enumerações. Não existem ponteiros nulos (void pointers) (apesar de referências para Object serem parecidas). E qualquer conversão implícita definida pelo usuário deve ser marcada explicitamente, diferentemente dos construtores de cópia de C++.
- A sintaxe para a declaração de vetores é diferente ("int[] a = new int[5]" ao invés de "int a[5]").
- Membros de enumeração são colocados em seu próprio espaço de nomes (namespace).
- C# não possui modelos (templates), mas C# 2.0 possui genéricos (generics).
- Propriedades estão disponíveis, as quais permitem que métodos sejam chamados com a mesma sintaxe de acesso a membros de dados.
- Recursos de reflexão completos estão disponíveis

Apesar de C# ser freqüentemente tido como similar a Java, existem uma série de diferenças importantes, tais como:

- Java não implementa propriedades nem sobrecarga de operadores.
- Java não implementa um modo inseguro que permita a manipulação de ponteiros e aritmética sem checagem.
- Java possui exceções checadas, enquanto exceções em C# são não checadas como em C++.
- Java não implementa o goto como estrutura de controle, mas C# sim.
- Java utiliza-se de comentários Javadoc para gerar documentação automática a partir de arquivos fonte. C# utiliza comentários baseados em XML para este propósito.
- C# suporta indexadores e delegados.

Bibliotecas de códigos

Ao contrário das outras linguagens de programação, nenhuma implementação de C# atualmente inclui qualquer conjunto de bibliotecas de classes ou funções. Ao invés disso, C# está muito vinculada ao framework .Net, do qual C# obtém suas classes ou funções de execução. O código é organizado em um conjunto de namespaces que agrupam as classes com funções similares. Por exemplo: System.Drawing para gráficos, System.Collections para estrutura de dados e System.Windows.Forms para o sistema Windows Form.

Um nível de organização superior é fornecido pelo conceito de montador (assembly). Um montador pode ser um simples arquivo ou múltiplos arquivos ligados juntos (como em al.exe) que podem conter muitos namespaces ou objetos. Programas que precisam de classes para realizar uma função em particular podem se referenciar a montadores como System.Drawing.dll e System.Windows.Forms.dll assim como a biblioteca core (conhecida como mscorlib.dll na implementação da Microsoft).

1.5.3. BANCO DE DADOS MYSQL

É um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL como interface. É atualmente um dos bancos de dados mais populares, com mais de 10 milhões de instalações pelo mundo. Suporta Unicode, Full Text Indexes, replicação, Hot Backup, GIS, OLAP e muitos outros recursos de banco de dados. O MySQL é um banco de dados completo, robusto e extremamente rápido, com todas as características existentes nos principais bancos de dados disponíveis no mercado. O banco de dados MySQL é um sério competidor para os maiores sistemas de banco de dados existentes para aplicações de pequeno e médio porte. Uma das grandes vantagens do MySQL é suportar mais de uma dúzia de tipos de dados (MILANI, 2007).

Características:

- Portabilidade;
- Compatibilidade (existem drivers ODBC, JDBC e .NET e módulos de interface para diversas linguagens de programação, como Delphi, Java, C/C++, C#, Visual Basic, Python, Perl, PHP, ASP e Ruby);
- Excelente desempenho e estabilidade;
- Pouco exigente quanto a recursos de hardware;
- Facilidade de uso;
- É um Software Livre com base na GPL (entretanto, se o programa que acessar o Mysql não for GPL, uma licença comercial deverá ser adquirida);
- Contempla a utilização de vários Storage Engines como MyISAM, InnoDB, Falcon, BDB, Archive, Federated, CSV, Solid;
- Suporta controle transacional;
- Suporta Triggers;
- Suporta Cursors (Non-Scrollable e Non-Updatable);
- Suporta Stored Procedures e Functions;
- Replicação facilmente configurável;

- Interfaces gráficas (MySQL Toolkit) de fácil utilização cedidos pela MySQL Inc.

1.5.4. PROGRAMAÇÃO ORIENTADA A OBJETOS

A programação Orientada a Objeto é um conceito de desenvolvimento usado para fazer abstrações do mundo real, que possibilita “agrupar” abstrações de mesma CLASSE, onde cada classe determina o comportamento (definido nos métodos) que são ações que a classe executa e os estados possíveis (atributos) de seus objetos (DEITEL, 2003).

1.5.5. ASTAH

A UML (Linguagem de Modelagem Unificada) caracteriza-se como uma linguagem de modelagem de terceira geração. A UML não diz o que você deve fazer, portanto ela não pode ser definida como uma metodologia de desenvolvimento, seu objetivo é auxiliar na visualização, no desenho e na comunicação entre objetos de um software ou sistema. A UML permite aos desenvolvedores visualizem os artefatos, caracterizado neste contexto como digramas, gerados pelo processo de desenvolvimento de software. A linguagem possui uma notação gráfica independente de qualquer tipo de processos que venha a ser utilizado.

Além de incorporar os nove diagramas de compõe a UML, o **astah** professional traz consigo o digrama de fluxo de dados (os velhos e bons pressupostos de Tom De Marco), notações advindas do flowchart, o diagrama ER, ferramenta para criação dos mapas mentais e o diagrama de requisitos, artefato que potencializa o desenvolvimento de software utilizando a idéia de processos ágeis.

1.6. ESTRUTURA DO TRABALHO

Para atender da melhor maneira as necessidades e objetivos do projeto, este trabalho foi dividido cinco em Capítulos. No primeiro Capítulo será feita

uma introdução do projeto a ser desenvolvido com base nas necessidades da empresa Peroba Rosa e uma breve descrição das tecnologias utilizadas na implementação do projeto. No segundo Capítulo, será feita a documentação do software.

No terceiro Capítulo é onde será desenvolvido o Banco de Dados e também onde será implementado o software.

Já no quarto Capítulo, será feita uma bateria de testes no software já pronto e será feita a implantação no cliente. Se for necessário, serão feitas correções ou modificações no software.

Finalmente no quinto Capítulo, o projeto será apresentado e finalizado.

1.7. METODOLOGIA DE ANÁLISE

Para a modelagem do software será utilizada a metodologia Orientada a Objetos e com ela a foi utilizada a linguagem de modelagem UML onde é possível especificar, visualizar, construir e documentar um sistema (UML, 2012).

1.7.1. LEVANTAMENTO E ESPECIFICAÇÃO DOS REQUISITOS

Os requisitos foram levantados através de uma reunião feita com Adilson Bento Rodrigues, o proprietário da empresa Peroba Rosa. Ele relatou não ter controle de nada e que seria necessário começar do “zero”. Conversamos bastante e chegamos aos seguintes requisitos:

- CADASTROS
 - USUÁRIO;
 - VENDEDOR;
 - CLIENTE;
 - FORNECEDOR;
 - PRODUTO;
 - GRUPO;
 - SUB-GRUPO;
 - LOCALIZAÇÃO DE ESTOQUE;

- CONTA;
- BANCO;
- TIPO DE MOVIMENTAÇÃO BANCÁRIA;
- GRUPO DE CLASSIFICAÇÃO FINANCEIRA;
- TIPO DE BAIXA FINANCEIRA;

- MOVIMENTAÇÃO
 - VENDAS;
 - COMPRAS;
 - ESTOQUE;
 - CONTAS A RECEBER;
 - CONTAS A PAGAR;
 - BANCO E CAIXA;

- RELATÓRIOS
 - CLIENTE;
 - VENDAS;
 - CARREGAMENTO;
 - FORNECEDOR;
 - COMPRAS;
 - PRODUTO;
 - ESTOQUE;
 - CONTAS A RECEBER;
 - CONTAS A PAGAR;
 - BANCO E CAIXA;

CAPÍTULO 2 – DOCUMENTAÇÃO

A documentação proporciona uma base segura e confiável para a implementação do sistema onde tudo já foi pensado e planejado. Isso proporciona mais agilidade e rapidez para implementar ou dar manutenção no sistema. Segue abaixo os componentes deste capítulo:

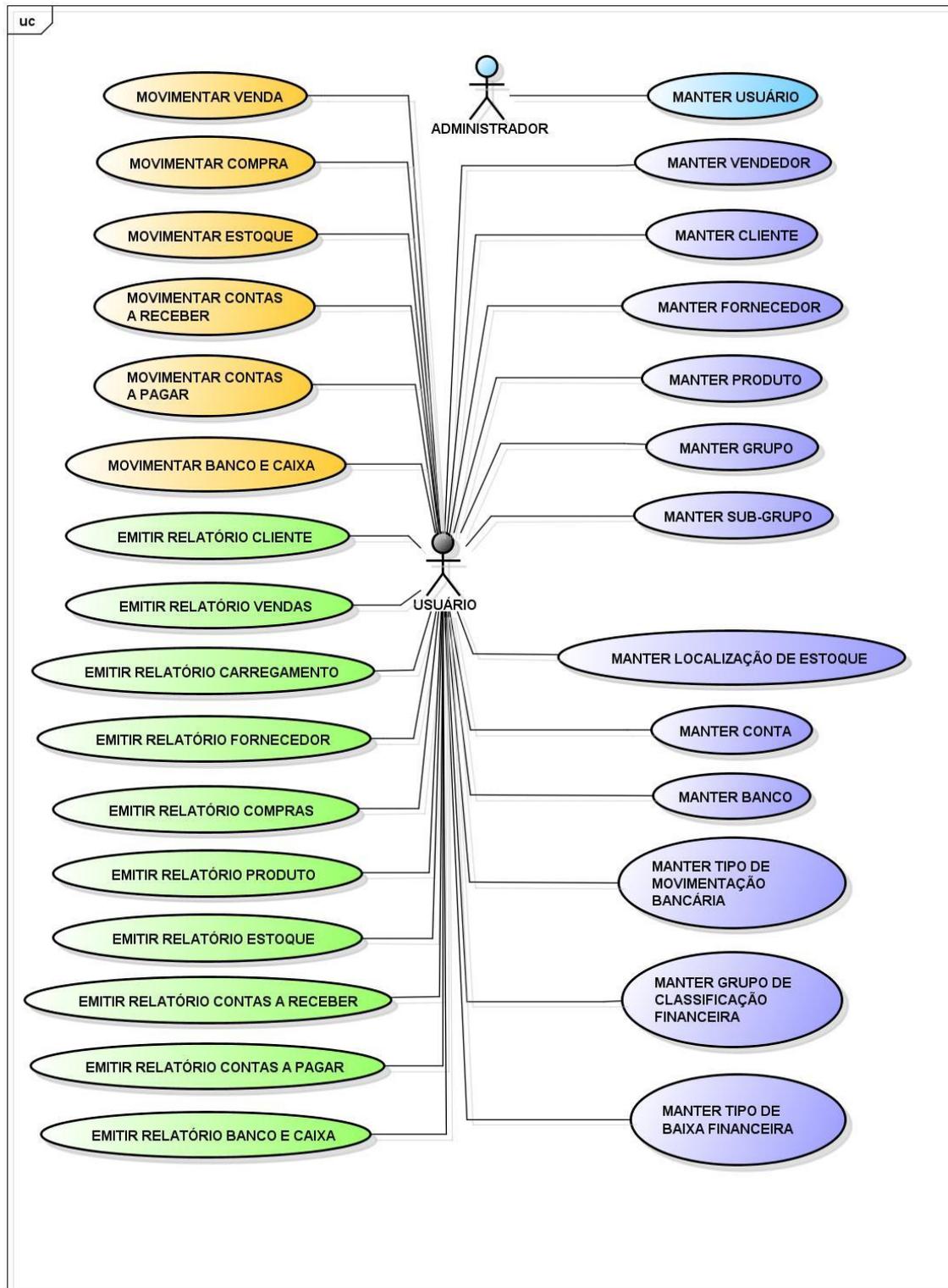
2.1. LISTA DE EVENTOS

Para a modelagem do sistema foi feita uma relação dos eventos relacionados às necessidades encontradas. São eles:

- Manter Usuário;
- Manter Vendedor;
- Manter Cliente;
- Manter Fornecedor;
- Manter Produto;
- Manter Grupo;
- Manter Sub-Grupo;
- Manter Localização de Estoque;
- Manter Conta;
- Manter Banco;
- Manter Tipo de Movimentação Bancária;
- Manter Grupo de Classificação Financeira;
- Manter Tipo de Baixa Financeira;
- Movimentar Venda;
- Movimentar Compra;
- Movimentar Estoque;
- Movimentar Contas a Receber;
- Movimentar Contas a Pagar;
- Movimentar Banco e Caixa;
- Emitir Relatório Cliente;
- Emitir Relatório Vendas;
- Emitir Relatório Carregamento;

- Emitir Relatório Fornecedor;
- Emitir Relatório Compras;
- Emitir Relatório Produto;
- Emitir Relatório Estoque;
- Emitir Relatório Contas a Receber;
- Emitir Relatório Contas a Pagar;
- Emitir Relatório Banco e Caixa;

2.2. DIAGRAMA DE CASOS DE USO



powered by astah

Figura 1 – Diagrama de Casos de Uso

2.3. ESPECIFICAÇÕES DE CASOS DE USO

2.3.1. UC 1 – MANTER USUÁRIO

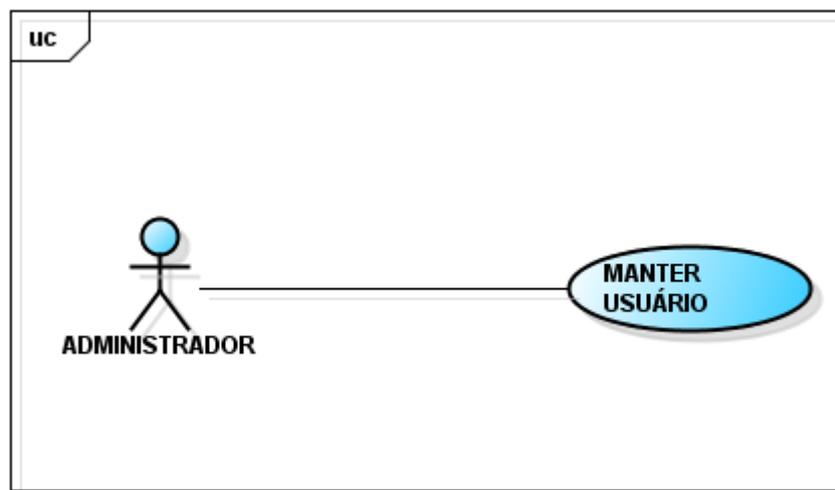


Figura 2 - UC 1 – Manter Usuário

Funcionalidade/Objetivo	Inserir, alterar, excluir e pesquisar usuário.
Ator (es)	Administrador
Pré-Condição	O administrador deverá estar autenticado no sistema.
Cenário Principal	<p>1- O sistema solicita os dados necessários para o cadastro do usuário.</p> <p>2- O administrador informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O sistema solicita os dados para o cadastro da função.[A1]</p> <p>4- O administrador informa os dados necessários.[A2]</p> <p>5- O administrador seleciona a opção Cadastrar.</p> <p>6- O sistema emite a mensagem “Usuário cadastrado com sucesso”.</p> <p>7- O sistema cadastra o usuário.</p>
Cenário Alternativo	<p>A1 - Se o administrador não informar os dados para o cadastro da função, o sistema informa que o usuário não foi cadastrado.</p> <p>A2 - O administrador poderá cancelar o processo durante o cadastro.</p>
Casos de Teste	<p>4.1- O sistema verifica se os campos foram preenchidos corretamente.</p> <p>4.2- O sistema não confirma o cadastro e emite uma mensagem de erro.</p> <p>4.3- O sistema cancela a operação.</p>

TABELA 1 - MANTER USUÁRIO

2.3.2. UC 2 – MANTER VENDEDOR

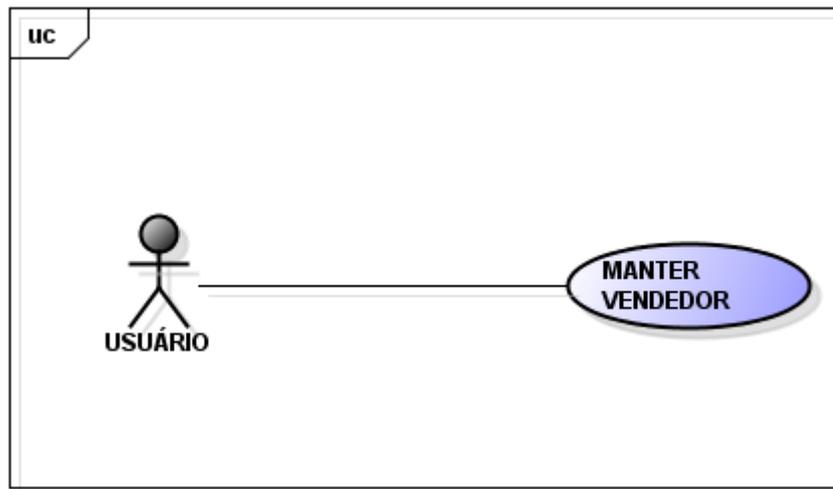


Figura 3 - UC 2 – Manter Vendedor

Funcionalidade/Objetivo	Inserir, alterar, excluir e pesquisar vendedor.
Ator (es)	Usuário
Pré-Condição	O usuário deverá estar autenticado no sistema.
Cenário Principal	<p>1- O sistema solicita os dados necessários para o cadastro do vendedor.</p> <p>2- O usuário informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O sistema solicita os dados para o cadastro da função.[A1]</p> <p>4- O usuário informa os dados necessários.[A2]</p> <p>5- O usuário seleciona a opção Cadastrar.</p> <p>6- O sistema emite a mensagem “Vendedor cadastrado com sucesso”.</p> <p>7- O sistema cadastra o vendedor.</p>
Cenário Alternativo	<p>A1 - Se o usuário não informar os dados para o cadastro da função, o sistema informa que o vendedor não foi cadastrado.</p> <p>A2 - O usuário poderá cancelar o processo durante o cadastro.</p>
Casos de Teste	<p>4.1- O sistema verifica se os campos foram preenchidos corretamente.</p> <p>4.2- O sistema não confirma o cadastro e emite uma mensagem de erro.</p> <p>4.3- O sistema cancela a operação.</p>

TABELA 2 - MANTER VENDEDOR

2.3.3. UC 3 – MANTER CLIENTE

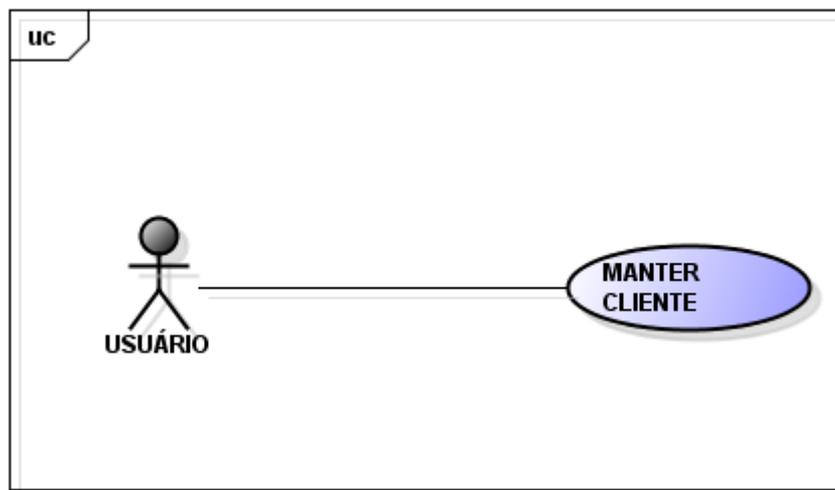


Figura 4 - UC 3 – Manter Cliente

Funcionalidade/Objetivo	Inserir, alterar, excluir e pesquisar cliente.
Ator (es)	Usuário
Pré-Condição	O usuário deverá estar autenticado no sistema.
Cenário Principal	<p>1- O sistema solicita os dados necessários para o cadastro do cliente.</p> <p>2- O usuário informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O sistema solicita os dados para o cadastro da função.[A1]</p> <p>4- O usuário informa os dados necessários.[A2]</p> <p>5- O usuário seleciona a opção Cadastrar.</p> <p>6- O sistema emite a mensagem “Cliente cadastrado com sucesso”.</p> <p>7- O sistema cadastra o cliente.</p>
Cenário Alternativo	<p>A1 - Se o usuário não informar os dados para o cadastro da função, o sistema informa que o cliente não foi cadastrado.</p> <p>A2 - O usuário poderá cancelar o processo durante o cadastro.</p>
Casos de Teste	<p>4.1- O sistema verifica se os campos foram preenchidos corretamente.</p> <p>4.2- O sistema não confirma o cadastro e emite uma mensagem de erro.</p> <p>4.3- O sistema cancela a operação.</p>

TABELA 3 - MANTER CLIENTE

2.3.4. UC 4 – MANTER FORNECEDOR

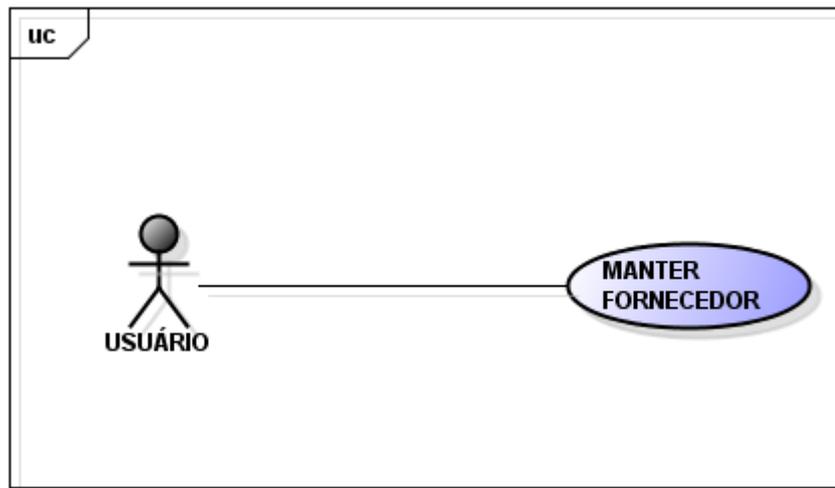


Figura 5 - UC 4 – Manter Fornecedor

Funcionalidade/Objetivo	Inserir, alterar, excluir e pesquisar fornecedor.
Ator (es)	Usuário
Pré-Condição	O usuário deverá estar autenticado no sistema.
Cenário Principal	<p>1- O sistema solicita os dados necessários para o cadastro do fornecedor.</p> <p>2- O usuário informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O sistema solicita os dados para o cadastro da função.[A1]</p> <p>4- O usuário informa os dados necessários.[A2]</p> <p>5- O usuário seleciona a opção Cadastrar.</p> <p>6- O sistema emite a mensagem “Fornecedor cadastrado com sucesso”.</p> <p>7- O sistema cadastra o fornecedor.</p>
Cenário Alternativo	<p>A1 - Se o usuário não informar os dados para o cadastro da função, o sistema informa que o fornecedor não foi cadastrado.</p> <p>A2 - O usuário poderá cancelar o processo durante o cadastro.</p>
Casos de Teste	<p>4.1- O sistema verifica se os campos foram preenchidos corretamente.</p> <p>4.2- O sistema não confirma o cadastro e emite uma mensagem de erro.</p> <p>4.3- O sistema cancela a operação.</p>

TABELA 4 - MANTER FORNECEDOR

2.3.5. UC 5 – MANTER PRODUTO

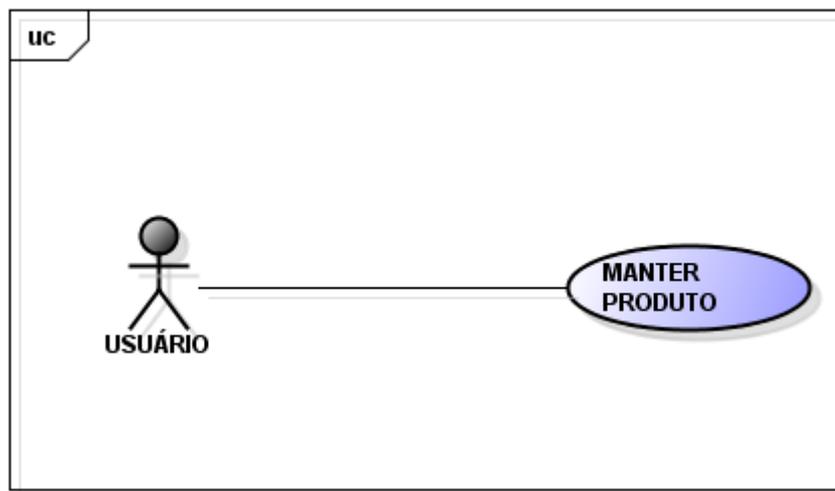


Figura 6 - UC 5 – Manter Produto

Funcionalidade/Objetivo	Inserir, alterar, excluir e pesquisar produto.
Ator (es)	Usuário
Pré-Condição	1 - O usuário deverá estar autenticado no sistema. 2 - O fornecedor deverá estar cadastrado.
Cenário Principal	1- O sistema solicita os dados necessários para o cadastro do produto. 2- O usuário informa os dados de acordo com os campos a serem preenchidos. 3- O sistema solicita os dados para o cadastro da função.[A1] 4- O usuário informa os dados necessários.[A2] 5- O usuário seleciona a opção Cadastrar. 6- O sistema emite a mensagem “Produto cadastrado com sucesso”. 7- O sistema cadastra o produto.
Cenário Alternativo	A1 - Se o usuário não informar os dados para o cadastro da função, o sistema informa que o produto não foi cadastrado. A2 - O usuário poderá cancelar o processo durante o cadastro.
Casos de Teste	4.1- O sistema verifica se os campos foram preenchidos corretamente. 4.2- O sistema não confirma o cadastro e emite uma mensagem de erro. 4.3- O sistema cancela a operação.

TABELA 5 - MANTER PRODUTO

2.3.6. UC 6 – MANTER GRUPO

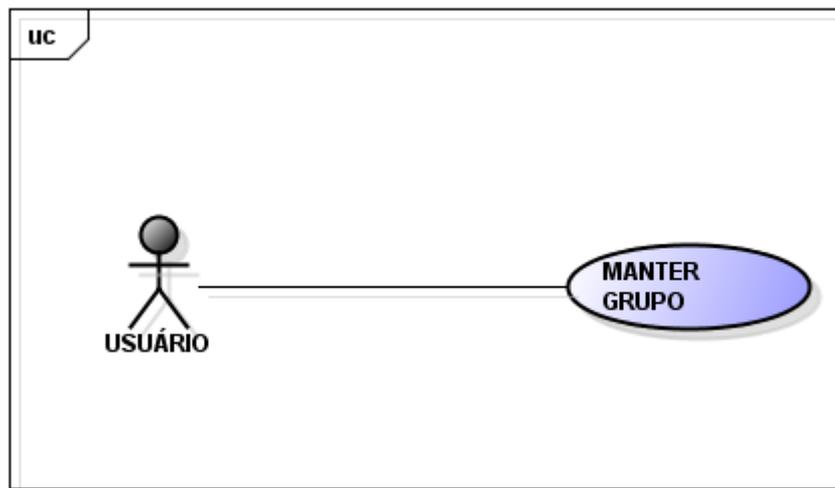


Figura 7 - UC 6 – Manter Grupo

Funcionalidade/Objetivo	Inserir, alterar, excluir e pesquisar grupo.
Ator (es)	Usuário
Pré-Condição	O usuário deverá estar autenticado no sistema.
Cenário Principal	<p>1- O sistema solicita os dados necessários para o cadastro do grupo.</p> <p>2- O usuário informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O sistema solicita os dados para o cadastro da função.[A1]</p> <p>4- O usuário informa os dados necessários.[A2]</p> <p>5- O usuário seleciona a opção Cadastrar.</p> <p>6- O sistema emite a mensagem “Grupo cadastrado com sucesso”.</p> <p>7- O sistema cadastra o grupo.</p>
Cenário Alternativo	<p>A1 - Se o usuário não informar os dados para o cadastro da função, o sistema informa que o grupo não foi cadastrado.</p> <p>A2 - O usuário poderá cancelar o processo durante o cadastro.</p>
Casos de Teste	<p>4.1- O sistema verifica se os campos foram preenchidos corretamente.</p> <p>4.2- O sistema não confirma o cadastro e emite uma mensagem de erro.</p> <p>4.3- O sistema cancela a operação.</p>

TABELA 6 - MANTER GRUPO

2.3.7. UC 7 – MANTER SUB-GRUPO

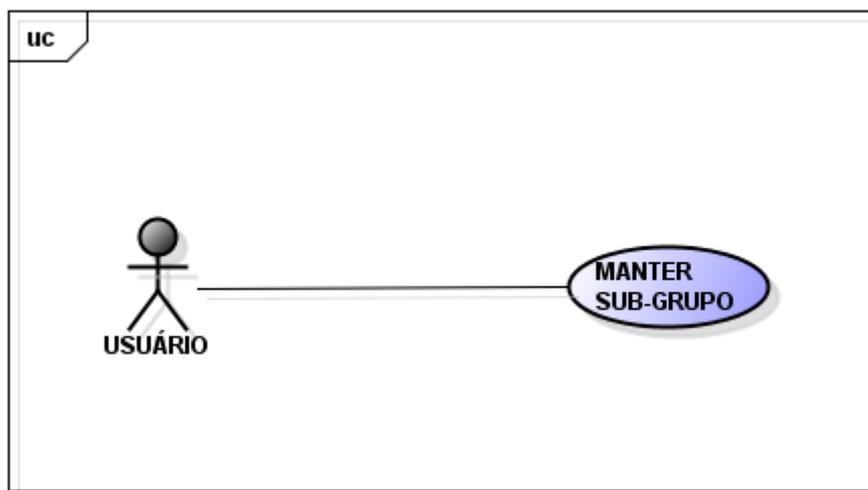


Figura 8 - UC 7 – Manter Sub-Grupo

Funcionalidade/Objetivo	Inserir, alterar, excluir e pesquisar sub-grupo.
Ator (es)	Usuário
Pré-Condição	1 - O usuário deverá estar autenticado no sistema. 2 - O grupo deverá estar cadastrado.
Cenário Principal	1- O sistema solicita os dados necessários para o cadastro do sub-grupo e a qual grupo ele pertence. 2- O usuário informa os dados de acordo com os campos a serem preenchidos. 3- O sistema solicita os dados para o cadastro da função.[A1] 4- O usuário informa os dados necessários.[A2] 5- O usuário seleciona a opção Cadastrar. 6- O sistema emite a mensagem “Sub-Grupo cadastrado com sucesso”. 7- O sistema cadastra o sub-grupo.
Cenário Alternativo	A1 - Se o usuário não informar os dados para o cadastro da função, o sistema informa que o sub-grupo não foi cadastrado. A2 - O usuário poderá cancelar o processo durante o cadastro.
Casos de Teste	4.1- O sistema verifica se os campos foram preenchidos corretamente. 4.2- O sistema não confirma o cadastro e emite uma mensagem de erro. 4.3- O sistema cancela a operação.

TABELA 7 - MANTER SUB-GRUPO

2.3.8. UC 8 – MANTER LOCALIZAÇÃO DE ESTOQUE

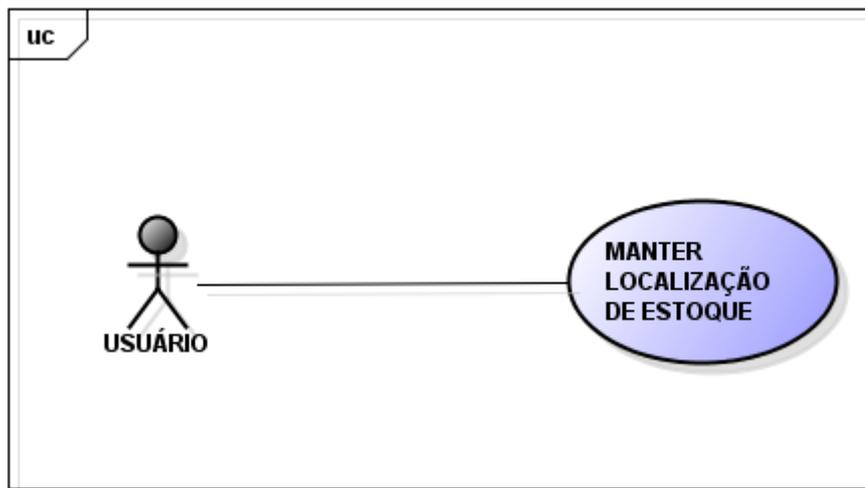


Figura 9 - UC 8 – Manter Localização de Estoque

Funcionalidade/Objetivo	Inserir, alterar, excluir e pesquisar localização de estoque.
Ator (es)	Usuário
Pré-Condição	O usuário deverá estar autenticado no sistema.
Cenário Principal	<p>1- O sistema solicita os dados necessários para o cadastro da localização de estoque.</p> <p>2- O usuário informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O sistema solicita os dados para o cadastro da função.[A1]</p> <p>4- O usuário informa os dados necessários.[A2]</p> <p>5- O usuário seleciona a opção Cadastrar.</p> <p>6- O sistema emite a mensagem “Localização de estoque cadastrada com sucesso”.</p> <p>7- O sistema cadastra a localização de estoque.</p>
Cenário Alternativo	<p>A1 - Se o usuário não informar os dados para o cadastro da função, o sistema informa que a localização de estoque não foi cadastrada.</p> <p>A2 - O usuário poderá cancelar o processo durante o cadastro.</p>
Casos de Teste	<p>4.1- O sistema verifica se os campos foram preenchidos corretamente.</p> <p>4.2- O sistema não confirma o cadastro e emite uma mensagem de erro.</p> <p>4.3- O sistema cancela a operação.</p>

TABELA 8 - MANTER LOCALIZAÇÃO DE ESTOQUE

2.3.9. UC 9 – MANTER CONTA

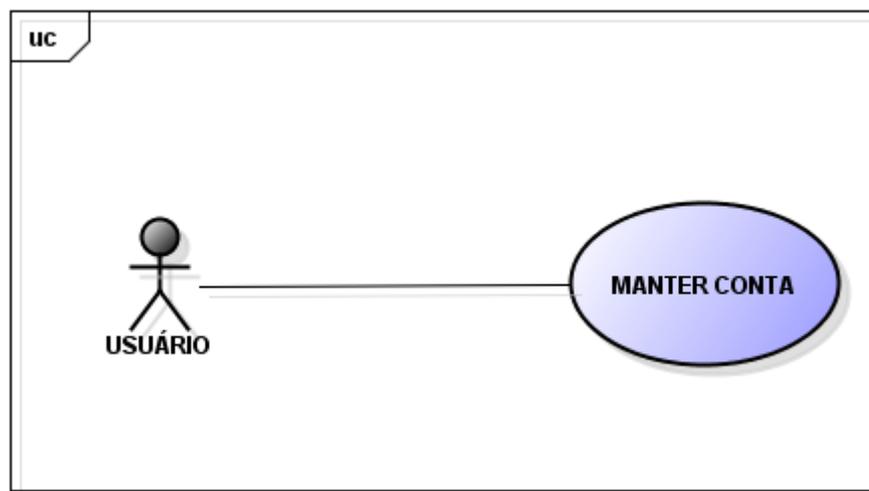


Figura 10 - UC 9 – Manter Conta

Funcionalidade/Objetivo	Inserir, alterar, excluir e pesquisar conta.
Ator (es)	Usuário
Pré-Condição	1 - O usuário deverá estar autenticado no sistema. 2 - O banco deverá estar cadastrado.
Cenário Principal	1- O sistema solicita os dados necessários para o cadastro da conta. 2- O usuário informa os dados de acordo com os campos a serem preenchidos. 3- O sistema solicita os dados para o cadastro da função.[A1] 4- O usuário informa os dados necessários.[A2] 5- O usuário seleciona a opção Cadastrar. 6- O sistema emite a mensagem “Conta cadastrada com sucesso”. 7- O sistema cadastra a conta.
Cenário Alternativo	A1 - Se o usuário não informar os dados para o cadastro da função, o sistema informa que a conta não foi cadastrada. A2 - O usuário poderá cancelar o processo durante o cadastro.
Casos de Teste	4.1- O sistema verifica se os campos foram preenchidos corretamente. 4.2- O sistema não confirma o cadastro e emite uma mensagem de erro. 4.3- O sistema cancela a operação.

TABELA 9 - MANTER CONTA

2.3.10. UC 10 – MANTER BANCO

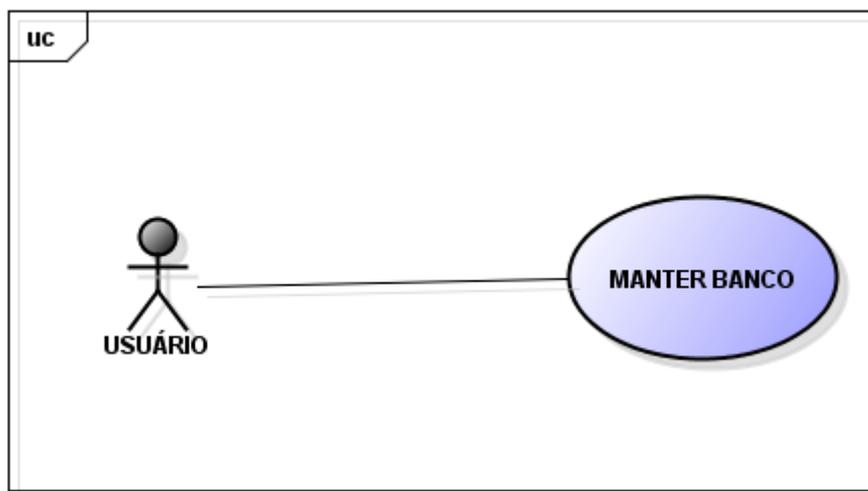


Figura 11 - UC 10 – Manter Banco

Funcionalidade/Objetivo	Inserir, alterar, excluir e pesquisar banco.
Ator (es)	Usuário
Pré-Condição	O usuário deverá estar autenticado no sistema.
Cenário Principal	<p>1- O sistema solicita os dados necessários para o cadastro do banco.</p> <p>2- O usuário informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O sistema solicita os dados para o cadastro da função.[A1]</p> <p>4- O usuário informa os dados necessários.[A2]</p> <p>5- O usuário seleciona a opção Cadastrar.</p> <p>6- O sistema emite a mensagem “Banco cadastrado com sucesso”.</p> <p>7- O sistema cadastra o banco.</p>
Cenário Alternativo	<p>A1 - Se o usuário não informar os dados para o cadastro da função, o sistema informa que o banco não foi cadastrado.</p> <p>A2 - O usuário poderá cancelar o processo durante o cadastro.</p>
Casos de Teste	<p>4.1- O sistema verifica se os campos foram preenchidos corretamente.</p> <p>4.2- O sistema não confirma o cadastro e emite uma mensagem de erro.</p> <p>4.3- O sistema cancela a operação.</p>

TABELA 8 - MANTER BANCO

2.3.11. MANTER TIPO DE MOVIMENTAÇÃO BANCÁRIA

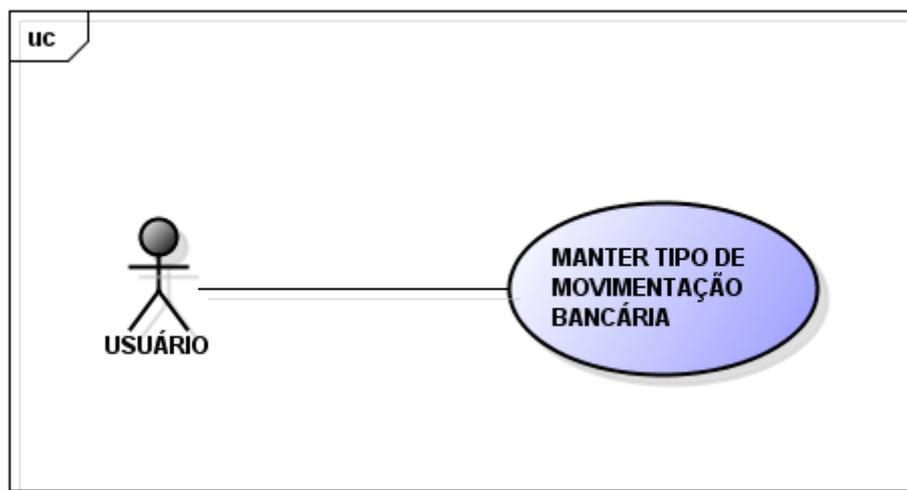


Figura 12 - UC 11 – Manter Tipo de Movimentação Bancária

Funcionalidade/Objetivo	Inserir, alterar, excluir e pesquisar tipo de movimentação bancária.
Ator (es)	Usuário
Pré-Condição	O usuário deverá estar autenticado no sistema.
Cenário Principal	<p>1- O sistema solicita os dados necessários para o cadastro do tipo de movimentação bancária.</p> <p>2- O usuário informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O sistema solicita os dados para o cadastro da função.[A1]</p> <p>4- O usuário informa os dados necessários.[A2]</p> <p>5- O usuário seleciona a opção Cadastrar.</p> <p>6- O sistema emite a mensagem “Tipo de movimentação bancária cadastrado com sucesso”.</p> <p>7- O sistema cadastra o tipo de movimentação bancária.</p>
Cenário Alternativo	<p>A1 - Se o usuário não informar os dados para o cadastro da função, o sistema informa que o tipo de movimentação bancária não foi cadastrado.</p> <p>A2 - O usuário poderá cancelar o processo durante o cadastro.</p>
Casos de Teste	<p>4.1- O sistema verifica se os campos foram preenchidos corretamente.</p> <p>4.2- O sistema não confirma o cadastro e emite uma mensagem de erro.</p> <p>4.3- O sistema cancela a operação.</p>

TABELA 9 - MANTER TIPO DE MOVIMENTAÇÃO BANCÁRIA

2.3.12. UC 12 – MANTER GRUPO DE CLASSIFICAÇÃO FINANCEIRA

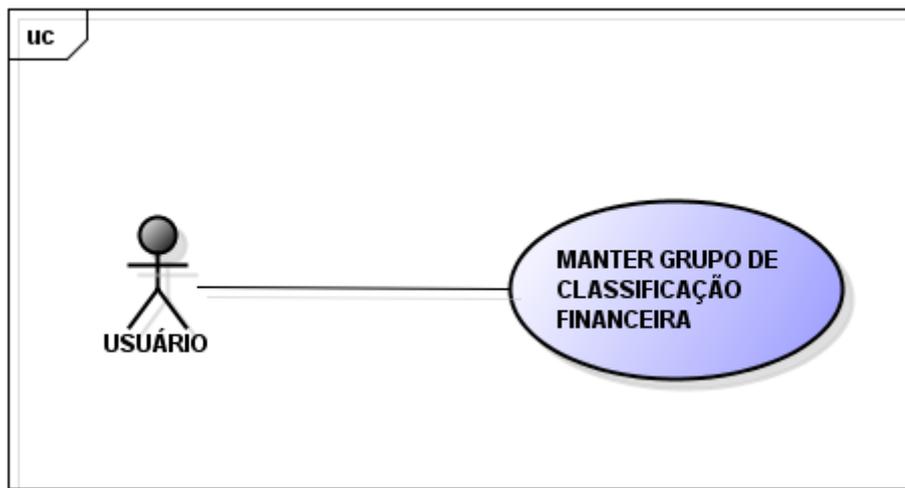


Figura 13 - UC 12 – Manter Grupo de Classificação Financeira

Funcionalidade/Objetivo	Inserir, alterar, excluir e pesquisar grupo de classificação financeira.
Ator (es)	Usuário
Pré-Condição	O usuário deverá estar autenticado no sistema.
Cenário Principal	<p>1- O sistema solicita os dados necessários para o cadastro do grupo de classificação financeira.</p> <p>2- O usuário informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O sistema solicita os dados para o cadastro da função.[A1]</p> <p>4- O usuário informa os dados necessários.[A2]</p> <p>5- O usuário seleciona a opção Cadastrar.</p> <p>6- O sistema emite a mensagem “Grupo de classificação financeira cadastrado com sucesso”.</p> <p>7- O sistema cadastra o grupo de classificação financeira.</p>
Cenário Alternativo	<p>A1 - Se o usuário não informar os dados para o cadastro da função, o sistema informa que o grupo de classificação financeira não foi cadastrado.</p> <p>A2 - O usuário poderá cancelar o processo durante o cadastro.</p>
Casos de Teste	<p>4.1- O sistema verifica se os campos foram preenchidos corretamente.</p> <p>4.2- O sistema não confirma o cadastro e emite uma mensagem de erro.</p> <p>4.3- O sistema cancela a operação.</p>

TABELA 10 - MANTER GRUPO DE CLASSIFICAÇÃO FINANCEIRA

2.3.13. UC 13 – MANTER TIPO DE BAIXA FINANCEIRA

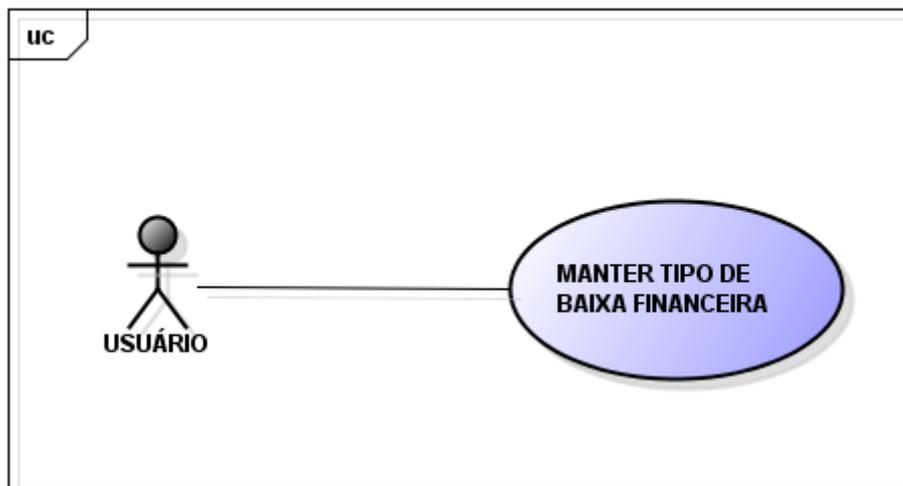


Figura 14 - UC 13 – Manter Tipo de Baixa Financeira

Funcionalidade/Objetivo	Inserir, alterar, excluir e pesquisar tipo de baixa financeira.
Ator (es)	Usuário
Pré-Condição	O usuário deverá estar autenticado no sistema.
Cenário Principal	<p>1- O sistema solicita os dados necessários para o cadastro do tipo de baixa financeira.</p> <p>2- O usuário informa os dados de acordo com os campos a serem preenchidos.</p> <p>3- O sistema solicita os dados para o cadastro da função.[A1]</p> <p>4- O usuário informa os dados necessários.[A2]</p> <p>5- O usuário seleciona a opção Cadastrar.</p> <p>6- O sistema emite a mensagem “Tipo de Baixa financeira cadastrado com sucesso”.</p> <p>7- O sistema cadastra o tipo de baixa financeira.</p>
Cenário Alternativo	<p>A1 - Se o usuário não informar os dados para o cadastro da função, o sistema informa que o tipo de baixa financeira não foi cadastrado.</p> <p>A2 - O usuário poderá cancelar o processo durante o cadastro.</p>
Casos de Teste	<p>4.1- O sistema verifica se os campos foram preenchidos corretamente.</p> <p>4.2- O sistema não confirma o cadastro e emite uma mensagem de erro.</p> <p>4.3- O sistema cancela a operação.</p>

TABELA 11 - MANTER TIPO DE BAIXA FINANCEIRA

2.3.14. UC 14 – MOVIMENTAR VENDA

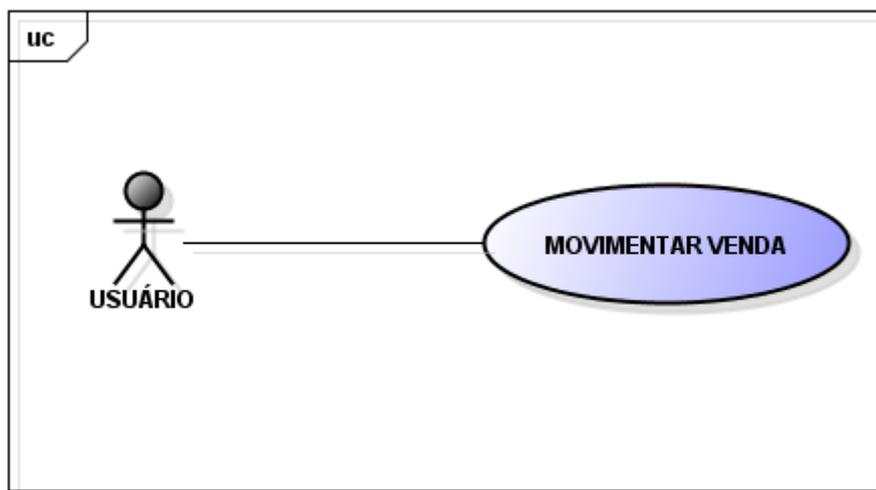


Figura 15 - UC 14 – Movimentar Venda

Funcionalidade/Objetivo	Permite ao usuário fornecer informações para a movimentação de vendas.
Ator (es)	Usuário
Pré-Condição	1 - O usuário deverá estar autenticado no sistema. 2 - O(s) produto(s) vendido deverá(ão) estar cadastrado(s) no sistema e com saldo em estoque.
Cenário Principal	1- O sistema solicita os dados necessários para movimentar venda. 2- O usuário informa os dados de acordo com os campos a serem preenchidos. 3- O sistema solicita os dados para o cadastro da função. [A1] 4- O usuário informa os dados necessários. 5- O usuário seleciona a opção "Concluir" 6- O sistema emite a mensagem "Operação realizada com sucesso".
Cenário Alternativo	A1 - O usuário poderá cancelar o processo durante a movimentação.
Casos de Teste	1- O sistema verifica se os campos foram preenchidos corretamente. 2- O sistema não confirma o cadastro e emite uma mensagem de erro. 3- O sistema cancela a operação.

TABELA 12 - MOVIMENTAR VENDA

2.3.15. UC 15 – MOVIMENTAR COMPRA

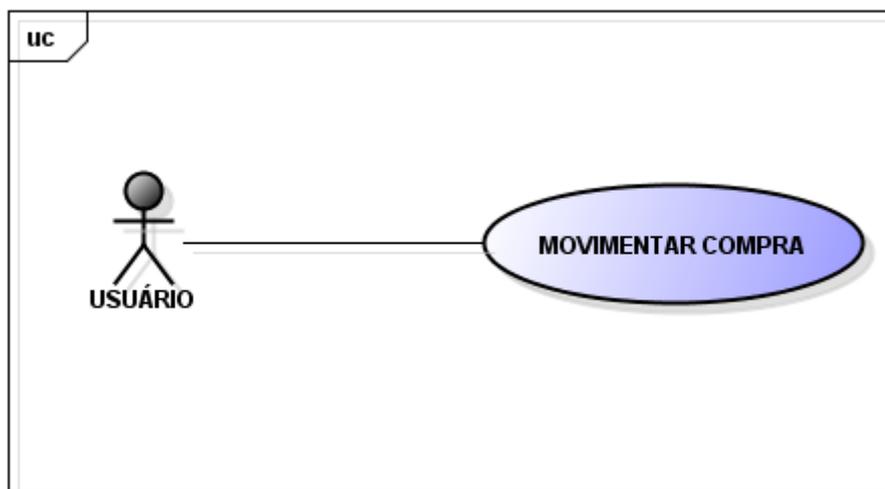


Figura 16 - UC 15 – Movimentar Compra

Funcionalidade/Objetivo	Permite ao usuário fornecer informações para a movimentação de compras.
Ator (es)	Usuário
Pré-Condição	1 - O usuário deverá estar autenticado no sistema. 2 - O(s) produto(s) comprado(s) deverá(ão) estar cadastrado(s).
Cenário Principal	1- O sistema solicita os dados necessários para movimentar compra. 2- O usuário informa os dados de acordo com os campos a serem preenchidos. 3- O sistema solicita os dados para o cadastro da função. [A1] 4- O usuário informa os dados necessários. 5- O usuário seleciona a opção "Concluir" 6- O sistema emite a mensagem "Operação realizada com sucesso".
Cenário Alternativo	A1 - O usuário poderá cancelar o processo durante a movimentação.
Casos de Teste	1- O sistema verifica se os campos foram preenchidos corretamente. 2- O sistema não confirma o cadastro e emite uma mensagem de erro. 3- O sistema cancela a operação.

TABELA 13 - MOVIMENTAR COMPRA

2.3.16. UC 16 – MOVIMENTAR ESTOQUE

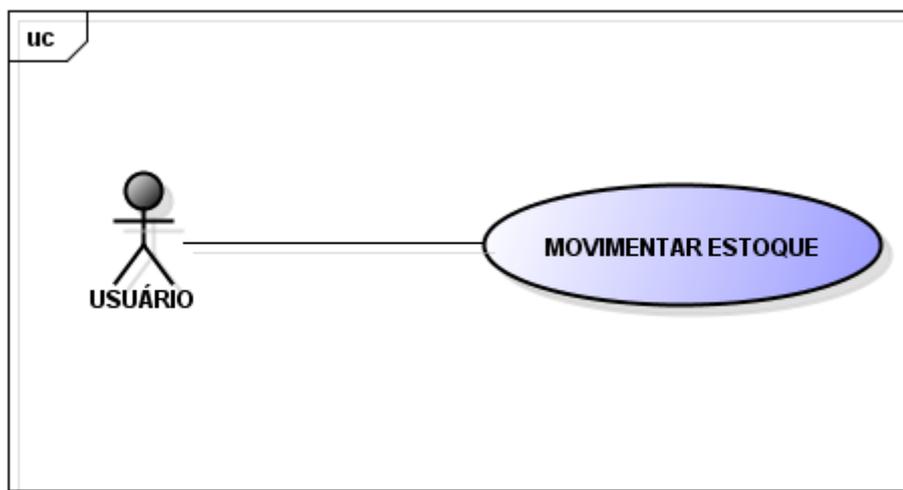


Figura 17 - UC 16 – Movimentar Estoque

Funcionalidade/Objetivo	Permite ao usuário fornecer informações para a movimentação de estoque.
Ator (es)	Usuário
Pré-Condição	1 - O usuário deverá estar autenticado no sistema. 2 - O(s) produto(s) movimentado(s) deverá(ão) estar cadastrado(s).
Cenário Principal	1- O sistema solicita os dados necessários para movimentar o estoque. 2- O usuário informa os dados de acordo com os campos a serem preenchidos. 3- O sistema solicita os dados para o cadastro da função. [A1] 4- O usuário informa os dados necessários. 5- O usuário seleciona a opção "Concluir" 6- O sistema emite a mensagem "Operação realizada com sucesso".
Cenário Alternativo	A1 - O usuário poderá cancelar o processo durante a movimentação.
Casos de Teste	1- O sistema verifica se os campos foram preenchidos corretamente. 2- O sistema não confirma o cadastro e emite uma mensagem de erro. 3- O sistema cancela a operação.

TABELA 14 - MOVIMENTAR ESTOQUE

2.3.17. UC 17 – MOVIMENTAR CONTAS A RECEBER

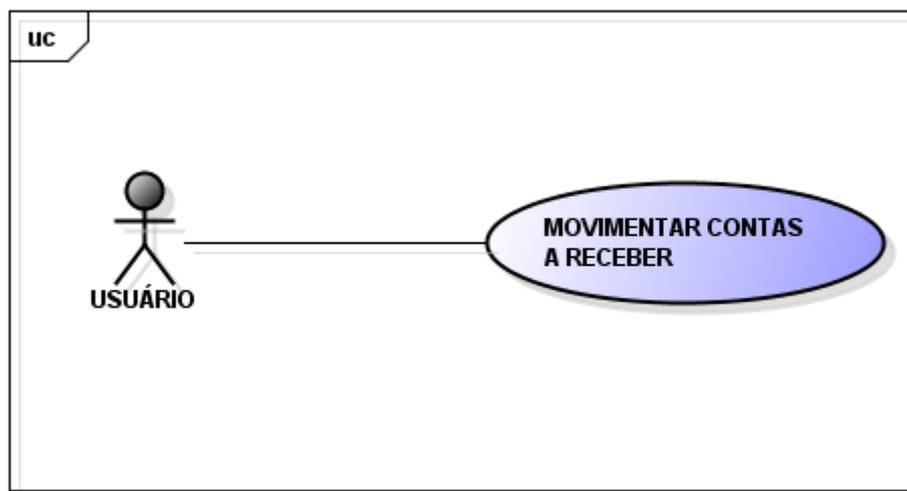


Figura 18 - UC 17 – Movimentar Contas a Receber

Funcionalidade/Objetivo	Permite ao usuário fornecer informações para a movimentação de contas a receber.
Ator (es)	Usuário
Pré-Condição	<ol style="list-style-type: none"> 1 - O usuário deverá estar autenticado no sistema. 2 - O cliente deverá estar cadastrado no sistema. 3 - O grupo de classificação financeira deverá estar cadastrado no sistema. 4 - O tipo de baixa financeira deverá estar cadastrado no sistema.
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema solicita os dados necessários para movimentar contas a receber. 2- O usuário informa os dados de acordo com os campos a serem preenchidos. 3- O sistema solicita os dados para o cadastro da função. [A1] 4- O usuário informa os dados necessários. 5- O usuário seleciona a opção "Concluir" 6- O sistema emite a mensagem "Operação realizada com sucesso".
Cenário Alternativo	A1 - O usuário poderá cancelar o processo durante a movimentação.
Casos de Teste	<ol style="list-style-type: none"> 1- O sistema verifica se os campos foram preenchidos corretamente. 2- O sistema não confirma o cadastro e emite uma mensagem de erro. 3- O sistema cancela a operação.

TABELA 15 - MOVIMENTAR CONTAS A RECEBER

2.3.18. UC 18 – MOVIMENTAR CONTAS A PAGAR

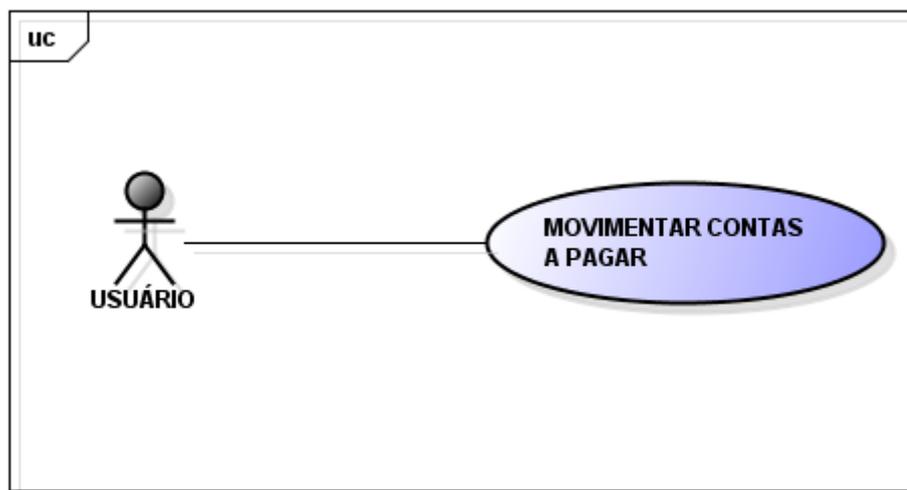


Figura 19 - UC 18 – Movimentar Contas a Pagar

Funcionalidade/Objetivo	Permite ao usuário fornecer informações para a movimentação de contas a pagar.
Ator (es)	Usuário
Pré-Condição	<ol style="list-style-type: none"> 1 - O usuário deverá estar autenticado no sistema. 2 - O fornecedor deverá estar cadastrado no sistema. 3 - O grupo de classificação financeira deverá estar cadastrado no sistema. 4 - O tipo de baixa financeira deverá estar cadastrado no sistema.
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema solicita os dados necessários para movimentar contas a pagar. 2- O usuário informa os dados de acordo com os campos a serem preenchidos. 3- O sistema solicita os dados para o cadastro da função. [A1] 4- O usuário informa os dados necessários. 5- O usuário seleciona a opção "Concluir" 6- O sistema emite a mensagem "Operação realizada com sucesso".
Cenário Alternativo	A1 - O usuário poderá cancelar o processo durante a movimentação.
Casos de Teste	<ol style="list-style-type: none"> 1- O sistema verifica se os campos foram preenchidos corretamente. 2- O sistema não confirma o cadastro e emite uma mensagem de erro. 3- O sistema cancela a operação.

TABELA 16 - MOVIMENTAR CONTAS A PAGAR

2.3.19. UC 19 – MOVIMENTAR BANCO E CAIXA

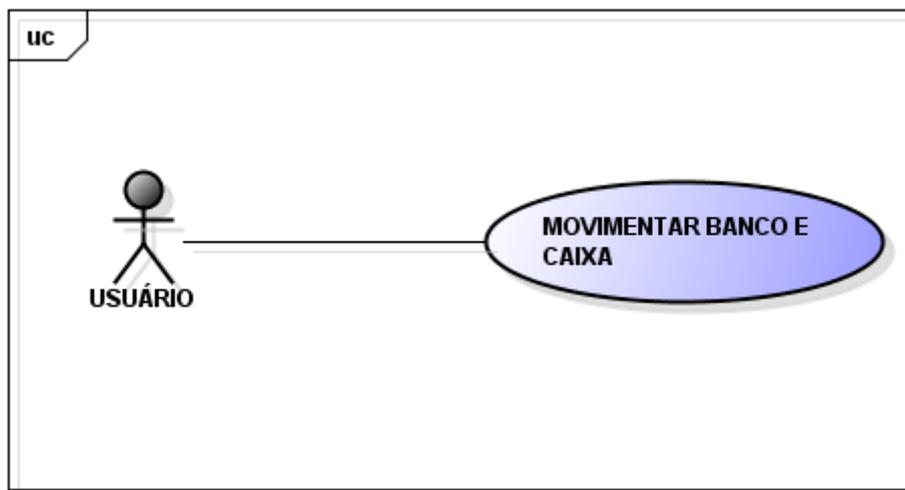


Figura 20 - UC 19 – Movimentar Banco e Caixa

Funcionalidade/Objetivo	Permite ao usuário fornecer informações para a movimentação de lançamentos em contas bancárias, cofre ou caixa.
Ator (es)	Usuário
Pré-Condição	O usuário deverá estar autenticado no sistema.
Cenário Principal	<ol style="list-style-type: none"> 1- O sistema solicita os dados necessários para movimentar lançamentos em contas. 2- O usuário informa os dados de acordo com os campos a serem preenchidos. 3- O sistema solicita os dados para o cadastro da função. [A1] 4- O usuário informa os dados necessários. 5- O usuário seleciona a opção "Concluir" 6- O sistema emite a mensagem "Operação realizada com sucesso".
Cenário Alternativo	A1 - O usuário poderá cancelar o processo durante a movimentação.
Casos de Teste	<ol style="list-style-type: none"> 1- O sistema verifica se os campos foram preenchidos corretamente. 2- O sistema não confirma o cadastro e emite uma mensagem de erro. 3- O sistema cancela a operação.

TABELA 19 - MOVIMENTAR BANCO E CAIXA

2.3.20. UC 20 – EMITIR RELATÓRIO CLIENTE

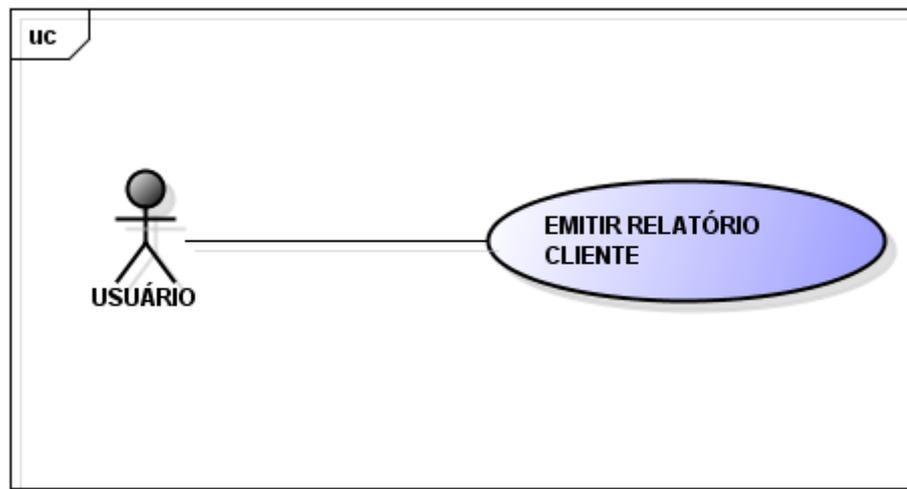


Figura 21 – Emitir Relatório Cliente

2.3.21. UC 21 – EMITIR RELATÓRIO VENDAS

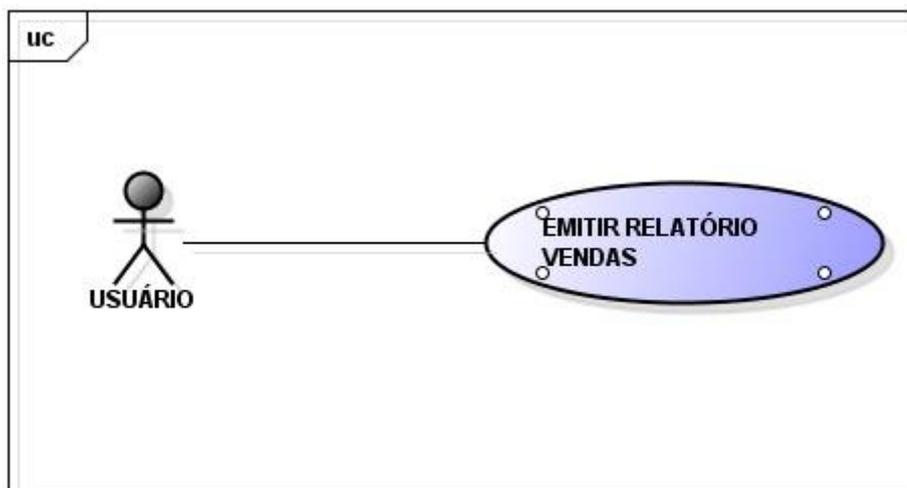


Figura 22 – Emitir Relatório Vendas

2.3.22. UC 22 – EMITIR RELATÓRIO CARREGAMENTO

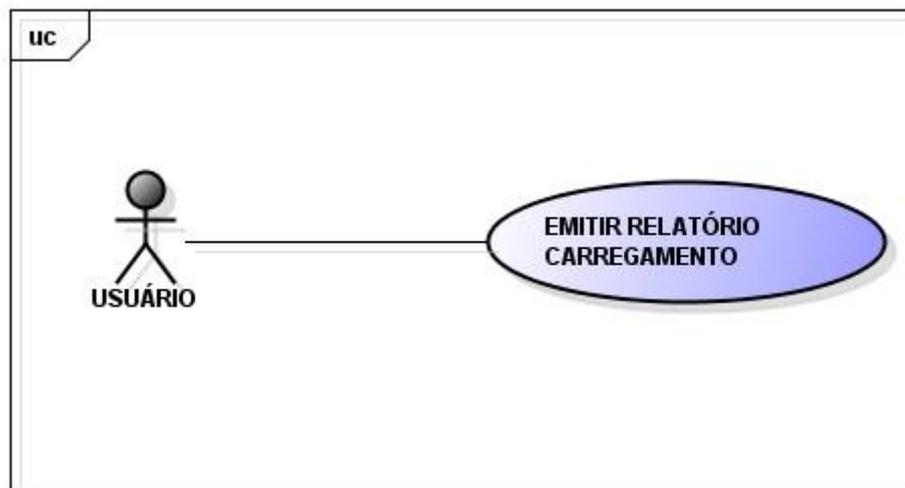


Figura 23 – Emitir Relatório Carregamento

2.3.23. UC 23 – EMITIR RELATÓRIO FORNECEDOR

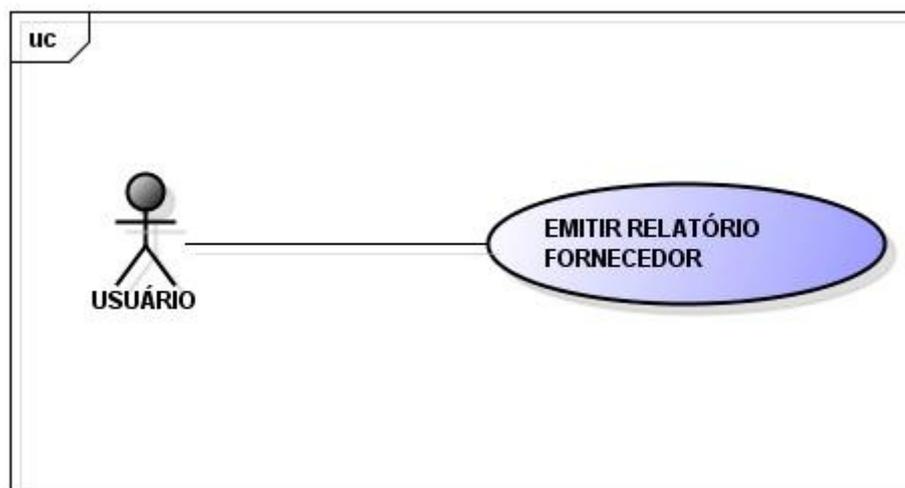


Figura 24 – Emitir Relatório Fornecedor

2.3.24. UC 24 – EMITIR RELATÓRIO COMPRAS

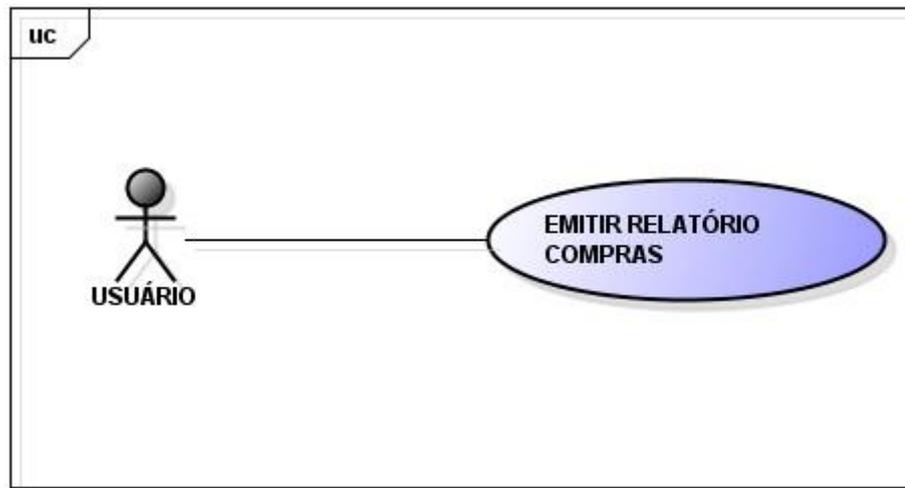


Figura 25 – Emitir Relatório Compras

2.3.25. UC 25 – EMITIR RELATÓRIO PRODUTO

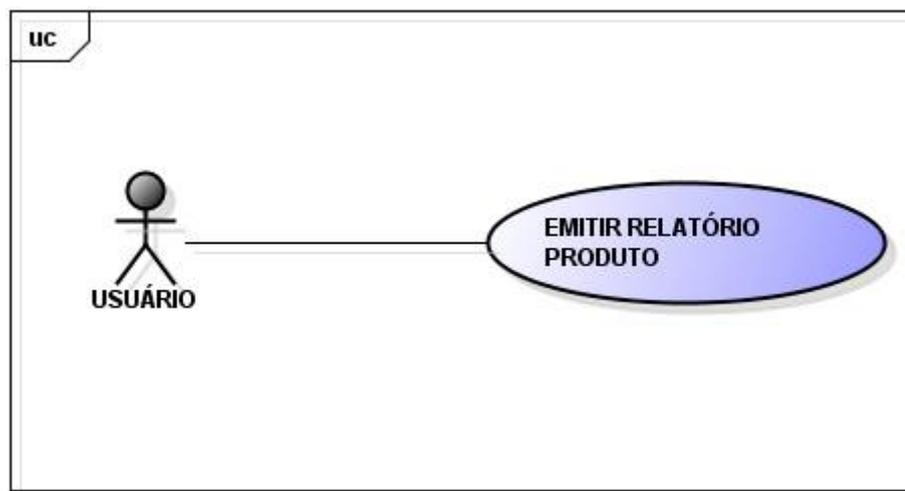


Figura 26 – Emitir Relatório Produto

2.3.26. UC 26 – EMITIR RELATÓRIO ESTOQUE

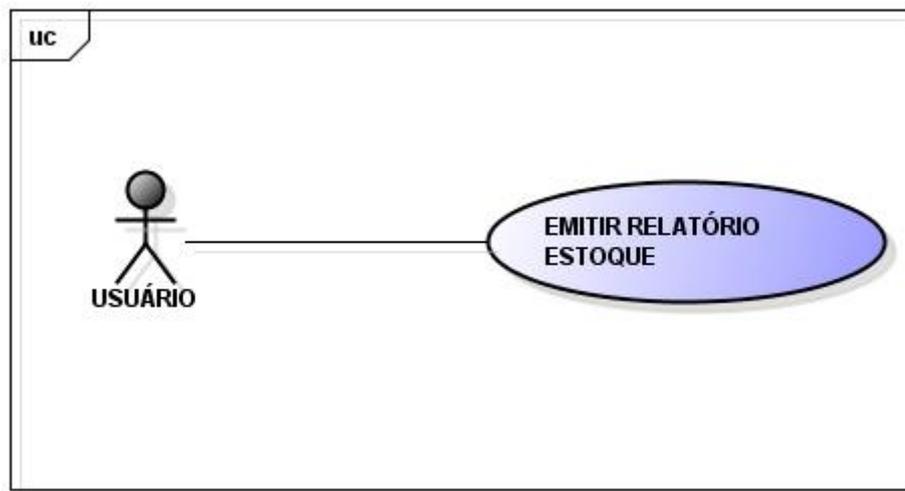


Figura 27 – Emitir Relatório Estoque

2.3.27. UC 27 – EMITIR RELATÓRIO CONTAS A RECEBER

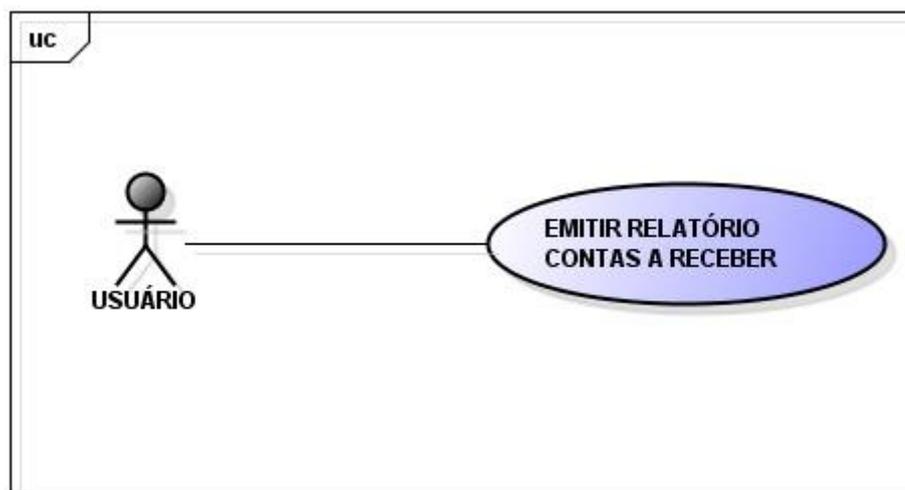


Figura 28 – Emitir Relatório Contas a Receber

2.3.28. UC 28 – EMITIR RELATÓRIO CONTAS A PAGAR

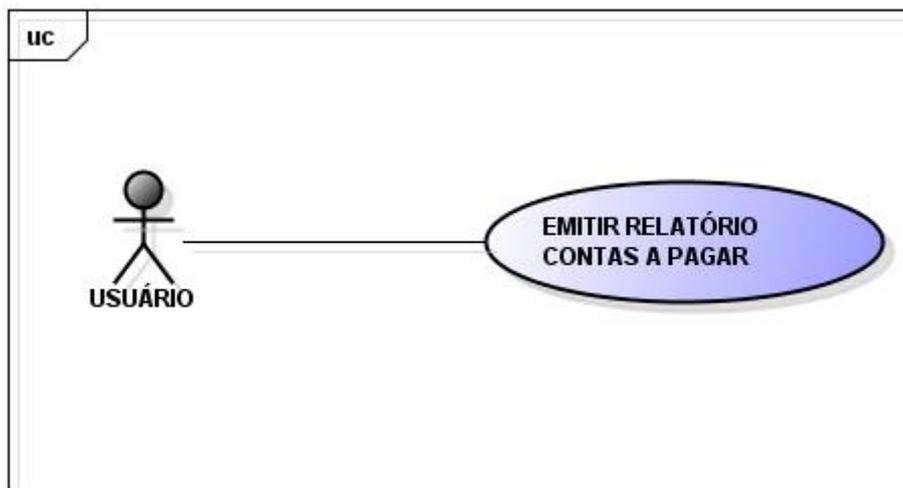


Figura 29 – Emitir Relatório Contas a Pagar

2.3.29. UC 29 – EMITIR RELATÓRIO BANCO E CAIXA

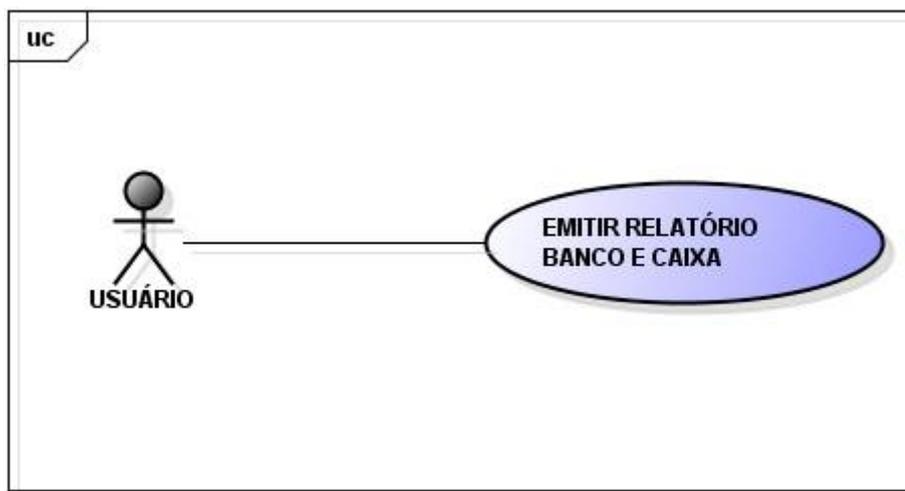
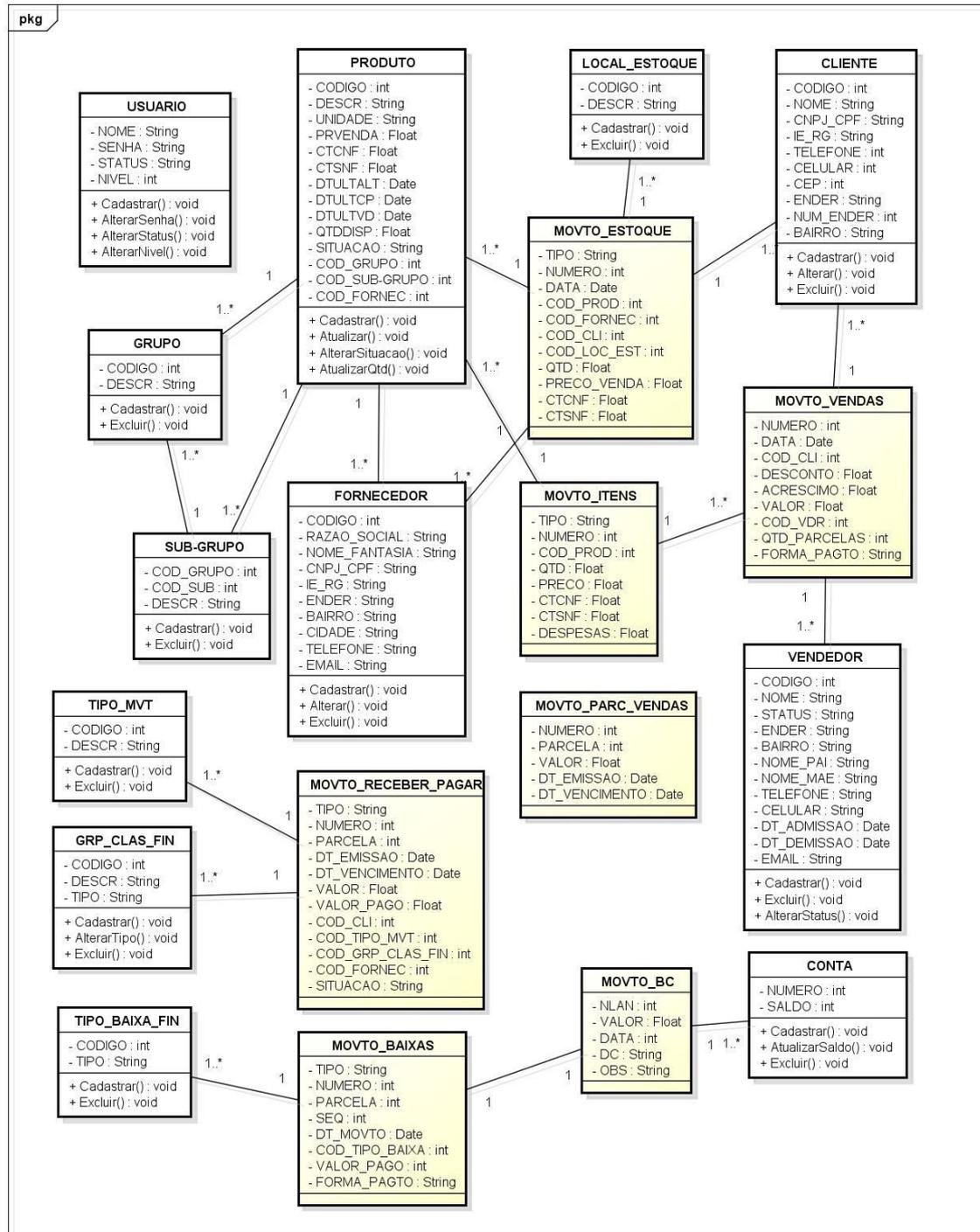


Figura 30 – Emitir Relatório Banco e Caixa

DIAGRAMA DE CLASSES

O Diagrama de Classes apresenta a estrutura de classes e suas cardinalidades (relacionamentos).



powered by astah

Figura 31 – Diagrama de Classes

CRONOGRAMA DE ATIVIDADES

ATIVIDADES	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV
PRÉ-PROJETO										
ANÁLISE DE REQUISITOS										
MODELAGEM DE REQUISITOS										
DOCUMENTAÇÃO										
QUALIFICAÇÃO										
DESENVOLVIMENTO										
TESTE										
CONCLUSÃO DO PROJETO										

Figura 32 - Cronograma

REFERÊNCIAS BIBLIOGRÁFICAS

Santos, L. C., Microsoft Visual C# 2008 Express Edition, 1ª Edição, 4ª Reimpressão. São Paulo, Editora Érica, 2010.

Pelland, P.; Paré, P.; Haines, K., Moving to Microsoft Visual Studio 2010, Microsoft Press, 2011; 336 Páginas.

Ferguson, J.; Patterson, B.; Beres, J.; Boutquin P.; Gupta, M., C# Bible. Indianapolis, Wiley Publishing, Inc., 2002.

ENDEL,

<http://www.sap.com/brazil/solutions/sapbusinessobjects/sme/reporting/crystalreports/index.epx> acesso em setembro de 2012.

MILANI, A. MySQL, Guia do Programador. São Paulo: NOVATEC: 2007.

Engenharia de Software. UML. Disponível em:

<<http://engenhariasoftware.wordpress.com/2011/09/05/as-potencialidades-do-astah-professional-uml-%E2%80%93-parte-1/>>

ASTAH. Astah Community e Astah UML. Software de modelagem UML, documentação e geração de código fonte. Disponível em:

<<http://astah.net>>

Deitel, H. M.; Deitel, P.J. C# Como Programar: 1ª Edição – São Paulo. Makron Books, 2003. 1200 Páginas.