



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

JOÃO HENRIQUE DE OLIVEIRA

SAECE- SISTEMA ADMINISTRATIVO E CONTROLE DE ESTOQUE

Assis
2010

JOÃO HENRIQUE DE OLIVEIRA

SAECE- SISTEMA ADMINISTRATIVO E CONTROLE DE ESTOQUE

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

Orientador: Prof. Domingos de Carvalho Villela Júnior
Área de Concentração: Desenvolvimento de Sistema

Assis
2010

FICHA CATALOGRÁFICA

Oliveira, João Henrique
Sistema Administrativo e controle de Estoque / João Henrique de
Oliveira. Fundação Educacional do Município de Assis – FEMA – Assis, 2010.
50 Páginas

Orientador: Prof. Domingos de Carvalho Villela Júnior
Trabalho de Conclusão de Curso – Instituto Municipal de Ensino
Superior de Assis – IMESA

1.Loja. 1.Sistema Gerencial

CDD: 001.61
Biblioteca da FEMA

SAECE- SISTEMA ADMINISTRATIVO E CONTROLE DE ESTOQUE

JOÃO HENRIQUE DE OLIVEIRA

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

Orientador: Prof. Domingos de Carvalho Villela Júnior

Analisador: Prof. Fabio Eder Cardoso

Assis
2010

AGRADECIMENTOS

Agradeço primeiramente a Deus por me dar força e ajuda para obter conhecimentos para cumprir mais uma etapa de minha vida e por dar coragem para enfrentar todos os desafios que foram impostos ao longo desse caminho.

Ao Prof. Domingos de Carvalho Villela Júnior, pela orientação, compreensão e toda a paciência durante o desenvolvimento e ao Prof. Fabio Eder Cardoso pelas críticas e elogios feitos na qualificação.

A todos os familiares, em especial meu Pai Ary José de Oliveira, Mãe Maria Rosa de Castro, minha Esposa Caroline de Oliveira, meu irmão Rodrigo de Castro, sua esposa Lais, minha querida Avó Joana, tia Cida, prima Michelli.

Aos meus amigos sempre presentes em especial meu sócio Fabiano Albuquerque que esta sempre junto comigo desde o inicio e agradecer aos amigos de sala Luiz Ângelo, Luis Fernando, Andre Do Lago, Paulo Vitor e também em especial dois amigos que sempre me ajudaram em aula Anderson Pacheco e Jefferson.

RESUMO

Este trabalho tem como objetivo principal desenvolver um sistema Administrativo e controle de estoque, visando um melhoramento no controle de estoque da empresa com relatórios claros, administrar a entrada e saída de dinheiro diariamente emitindo relatórios concretos dos gastos e lucros. Tendo como finalidade facilitar a administração do caixa da empresa.

O sistema será implantado visando atender todas as necessidades do usuário, dando suporte a todos os tipos de venda e estoque da empresa.

Utilizando a metodologia orientada a objetos programando em camadas para Java será utilizado Eclipse com a linguagem Java, utilizando a base de banco de dados *PostgreSQL* e *HiberNate* como forma de acesso aos mesmos.

Palavras Chaves: Orientada a objetos, Java, Eclipse, *PostgreSQL*, *Hibernate*.

ABSTRACT

This work has as main objective to develop a system administrative and inventory control, seeking an improvement in inventory control of the company with clear reports, manage the entry and exit of money by issuing daily reports of actual expenditures and profits. Taking intended to facilitate the administration the company's cash.

The system will be deployed to meet all user needs, supporting all types of stock and sale of the company.

The system will be deployed to meet the needs of the school, supporting all activities related to the library environment.

Using the method object-oriented programming in Java layers to be used Eclipse with Java, using the basic database and *Postgres HiberNet* as a way to access them.

Keywords: Object Oriented, Java, Eclipse, *Postgres*, *Hibernete*.

Índice de Ilustrações

Figura 1 - Mapa Mental do Sistema Administrativo e Controle de Estoque	18
Figura 2 - Caso de Uso Visão Geral - Efetuar Login	22
Figura 3 - Caso de Uso Visão Geral - Movimentações	23
Figura 4 - Caso de Uso Visão Geral – Consultar Relatórios.....	24
Figura 5 – UC – Manter Usuário	25
Figura 6 – UC – Manter Produto	26
Figura 7 – UC – Manter Dados do Cliente	27
Figura 8 - UC – Manter Cliente	28
Figura 9 - UC – Manter Tipo de Pagamento	29
Figura 10 - UC – Movimentar vendas.....	30
Figura 11 - UC – Movimentar Caixa	31
Figura 12 - UC – Movimentar Contas a Pagar	32
Figura 13 - UC – Movimentar Contas a receber	33
Figura 14 - UC – Verificar Produtos	34
Figura 15 - UC – Verificar Cliente	35
Figura 16 - UC – Verificar Fornecedor	36
Figura 17 - UC – Verificar Vendas	37
Figura 18 – UC – Efetuar devolução	38
Figura 34 - Diagrama de Classes	40
Figura 35 - Diagrama de Atividades - Cadastros	41
Figura 36 - Diagrama de Atividades - Movimentações.....	42
Figura 37 - Diagrama de Atividades - Relatórios	43
Figura 38 - WBS (Work Breakdown Structure)	44
Figura 39 - Sequenciamento de Atividades	45

Índice de Tabelas

Tabela 1 – Lista de Eventos.....	21
Tabela 2 – Especificação do Caso de Uso 01.....	26
Tabela 3 – Especificação do Caso de Uso 02.....	27
Tabela 4 – Especificação do Caso de Uso 03.....	28
Tabela 5 – Especificação do Caso de Uso 04.....	29
Tabela 6 – Especificação do Caso de Uso 05.....	30
Tabela 7 – Especificação do Caso de Uso 06.....	31
Tabela 8 – Especificação do Caso de Uso 07.....	32
Tabela 9 – Especificação do Caso de Uso 08.....	33
Tabela 10 – Especificação do Caso de Uso 09.....	34
Tabela 11 – Especificação do Caso de Uso 10.....	35
Tabela 12 – Especificação do Caso de Uso 11.....	36
Tabela 13 – Especificação do Caso de Uso 12.....	37
Tabela 14 – Especificação do Caso de Uso 13.....	38
Tabela 15 – Especificação do Caso de Uso 14.....	39
Tabela 16 – Especificação do Caso de Uso 15.....	40

SUMÁRIO

1.1	A Loja.....	12
1.2	PÚBLICO ALVO.....	12
1.3	JUSTIFICATIVA.....	13
1.4	METODOLOGIA	13
1.5	Eclipse	13
1.6	java	13
1.7	jude.....	14
1.8	Postgres.....	14
1.9	iREPORT	14
2.	ESTRUTURA DE DESENVOLVIMENTO DO SISTEMA.....	15
2.1.	UML	15
2.2	CLASSES.....	15
2.3	OBJETOS.....	15
2.4	ATRIBUTOS.....	16
2.5	MÉTODOS	16
2.6	CONSTRUTORES	16
2.7	ENCAPSULAMENTO.....	17
2.8	HERANÇA.....	17
2.9	POLIMORFISMO	17
3	ESPECIFICAÇÃO DO SISTEMA	18
3.1	OBJETIVO DO TRABALHO.....	18
3.2	MAPA MENTAL.....	18
3.3	LEVANTAMENTO DE REQUISITOS	18
3.3.1	Entrevista realizada na Info House	18
3.3.2	Funcionalidades do Sistema	19
3.4	ANÁLISE DOS REQUISITOS	19
3.4.2	Lista de eventos	20
3.5	VALIDAÇÃO DOS REQUISITOS	21
4	DIAGRAMAS.....	22
4.1	DIAGRAMA DE CASO DE USO GERAL	22
4.1.1	Efetuar Login	22
4.1.2	Movimentações	23
4.1.3	Consultar Relatórios	24

4.2 DIAGRAMAS DE CASO DE USO ESPECÍFICO	25
4.2.1 UC 01 – Manter Usuário	25
4.2.2 UC 02 – Manter Produto	26
4.2.3 UC 03 – Manter Cliente.....	27
4.2.4 UC 04 – Manter Fornecedor.....	28
4.2.5 UC 05 – Manter Pagamento	29
4.2.6 UC 06 – Movimentar vendas.....	30
4.2.7 UC 07 – Movimentar Caixa.....	31
4.2.8 UC 08 – Movimentar Contas a Pagar	32
4.2.9 UC 09 – Movimentar Contas a Receber	33
4.2.10 UC 10 – Verificar Produtos	34
4.2.11 UC 11 – Verificar Cliente	35
4.2.12 UC 12 – Verificar Fornecedor	36
4.2.13 UC 13 – Verificar vendas.....	37
4.2.14 UC 14 – Verificar Contas a Receber.....	38
4.2.15 UC 15 – Verificar Contas a Pagar	39
4.3 DIAGRAMA DE CLASSE	40
4.4 DIAGRAMA DE ATIVIDADES	41
4.4.1 Cadastros.....	41
4.4.2 Movimentações	42
4.4.3 Relatórios.....	43
5. PROJETO.....	44
5.1 ESTRUTURA ANALÍTICA DO PROJETO (WBS)	44
5.2 SEQUENCIAMENTO DE ATIVIDADES	45
5.3 ESPECIFICAÇÃO DOS CUSTOS.....	46
5.3.1 Recursos necessários para o desenvolvimento	46
5.3.2 Estimativa de custos.....	46
5.3.3 Custos materiais	46
5.3.4 Orçamento do sistema.....	47
5.4 CRONOGRAMA.....	48
6. CONCLUSÃO.....	49
7 REFERÊNCIAS.....	50
REFERÊNCIAS ELETRÔNICAS.....	50

INTRODUÇÃO

Atualmente o uso de um software de gerenciamento vem crescendo muito nas empresas ou qualquer tipo de entidade, isso se deve pelo fato de haver inúmeras informações a serem guardadas e manipuladas e a dificuldade de ser realizado manualmente.

Com isso, a utilização de um software de Gerenciamento resulta na exatidão e agilidade nos resultados obtidos, que são de grande valia para qualquer tipo de organização.

O software a ser desenvolvido trata-se de um Sistema Administrativo e Controle de Estoque de uma Loja de Informática. O sistema será implantado na loja Info House, na cidade de Cândido Mota, estado de São Paulo.

Atualmente, a loja já possui um software, mas de acordo com o proprietário ele possui muitas falhas, e não tem um retorno eficaz e rápido das informações necessárias para uma venda ágil e ate mesma para gerar relatórios dos gastos e lucros.

O sistema ajudará a reduzir muito o tempo gasto em algumas operações, tais como o cadastro de clientes e produtos, e uma enorme facilidade em geral relatórios finais das despesas e lucros do dia, mês e ano.

O software permitira ao Proprietário cadastrar usuário, cliente, fornecedor, produto, controle de caixa com saída e entrada, gerando relatórios.

1.1 A LOJA

Info House, Loja de informática especializada em manutenção de computadores e vendas de peças e acessórios, estabelecidos na cidade de Cândido Mota- SP.

1.2 PÚBLICO ALVO

Lojas de informática que visam informatizar e organizar seu estoque e obter um controle de caixa prático com relatórios exatos.

1.3 JUSTIFICATIVA

A justificativa de implantação de um Sistema Administrativo e Controle de Estoque de loja de informática vêm para facilitar a tomada de decisão por parte dos administradores da empresa.

MÉTODO DE DESENVOLVIMENTO

1.4 METODOLOGIA

Para o desenvolvimento do sistema foi utilizado a plataforma Eclipse com a linguagem Java, usando metodologia orientada a objetos e trabalhando em camadas. Modo SWT (Framework para interface gráfica), como base de banco de dados foi utilizado o PostgreSQL e Hibernate como forma de acesso ao banco.

1.5 ECLIPSE

Eclipse é uma IDE desenvolvida em Java, com código aberto para a construção de programas de computador. O projeto Eclipse foi iniciado na IBM que desenvolveu a primeira versão do produto e doou-o como software livre para a comunidade. Uma plataforma de desenvolvimento ótima para programadores que gostam de trabalhar com códigos limpos, sem arrastar nada, tudo feito manualmente.

(<http://www.eclipse.org/>)

1.6 JAVA

Uma linguagem considerada ainda nova na área de programação, mas que já tomou um grande e considerável espaço. Java é uma linguagem de programação orientada a objeto desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems. Diferentemente das linguagens

convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um "bytecode" que é executado por uma máquina virtual. A linguagem de programação Java é a linguagem convencional da Plataforma Java, mas não sua única linguagem.

(<http://www.java.sun.com/features/1998/05/birthday.html>)

1.7 JUDE

JUDE é uma IDE para Modelagem de Dados (UML) criada com Java e de uso fácil e intuitivo. Com a IDE JUDE é possível realizar uma modelagem de dados complexa, apresenta os dados para o usuário de forma clara e ainda possui a vantagem de seu layout ser bem intuitivo.

(<http://www.br-linux.org/linux/node/3335>)

1.8 POSTGRES

Um grande gerenciador de banco de dados, com mais de 15 anos de mercado ativo, uma arquitetura que comprovadamente ganhou forte reputação de confiabilidade, integridade de dados e conformidade a padrões. Roda em todos os grandes sistemas operacionais.

(<http://www.postgresql.org.br/sobre>)

1.9 IREPORT

O iReport é um programa Open Source, capaz de criar relatórios para aplicações Java no formato da biblioteca JasperReports. Com ele é capaz de criar qualquer tipo de relatório simples de forma rápida.

(Gonçalves, Edson 2008)

2. ESTRUTURA DE DESENVOLVIMENTO DO SISTEMA

A análise será Orientada a Objetos.

2.1. UML

Unified Modeling Language é uma tentativa de padronizar a modelagem Orientada a Objetos, de forma que qualquer sistema possa ser modelado corretamente, a UML é constituída por elementos gráficos, utilizados na modelagem que permitem representar os conceitos do paradigma da Orientação a Objetos, através destes elementos gráficos podemos construir vários diagramas como: Diagrama de Caso de Uso, Classe, Atividade, Seqüência, Colaboração, Componente, Objeto, etc.

(BEZERRA, 2002, p.14).

2.2 CLASSES

Classe é o termo técnico utilizado em linguagens orientadas a objetos que descreve um conjunto de dados estruturados que são caracterizados por propriedades comuns. Também pode ser interpretado como uma estrutura modular completa que descreve as propriedades estáticas e dinâmicas dos elementos manipulados pelo programa.

2.3 OBJETOS

O que caracteriza a programação orientada a objetos são os objetos. De um modo geral podemos encarar os objetos como sendo os objetos físicos do mundo real, tal como: carro, avião, cachorro, casa, telefone, computador, etc., por isso que às vezes é dito que orientação a objetos representa os problemas mais próximo ao mundo real, dando assim mais facilidade a programação como um todo, mais isso não é sempre verdade, porque às vezes temos problemas que são extremamente funcionais¹. Nesses problemas funcionais é difícil representar a estrutura lógica em

torno de objetos. Com isso, não são todos os problemas que giram em torno dos objetos facilmente visíveis.

De maneira simples, um objeto é uma entidade lógica que contém dados e código para manipular esses dados. Os dados são denominados como sendo atributos do objeto, ou seja, a estrutura que o objeto tem, e o código que o manipula denominamos método. Um método é uma função que manipula a estrutura de dados do objeto.

2.4 ATRIBUTOS

Um atributo é um dado para o qual cada objeto tem seu próprio valor.

Atributos são, basicamente, a estrutura de dados que vai representar a classe.

2.5 METODOS

Métodos são declarados dentro de uma classe para representar as operações que os objetos pertencentes a esta classe podem executar.

Um método é a implementação de uma rotina, ou seja, o código propriamente dito. Pode ser comparado a um procedimento ou função das linguagens imperativas.

2.6 CONSTRUTORES

Construtores são métodos especiais chamados pelo sistema no momento da criação de um objeto. Eles não possuem valor de retorno, porque você não pode chamar um construtor para um objeto, você só usa o construtor no momento da inicialização do objeto. Construtores representam uma oportunidade de inicializar seus dados de forma organizada.

Um construtor tem sempre o mesmo nome da classe a qual pertence. Para a classe String, pré-definida na linguagem o construtor tem a forma String ("Constante do tipo String"); com o argumento entre aspas que é especificado pelo programador. Ele seria chamado automaticamente no momento da criação, declaração de uma String, sem necessidade de uso do nome do construtor como método.

2.7 ENCAPSULAMENTO

O conceito de encapsulamento é decorrente do fato de se combinar os dados (atributos) e o código que manipula estes dados (métodos) em um único Objeto. Ele garante que a única forma de acesso aos dados é através dos métodos disponíveis ao usuário (chamados públicos). Os demais métodos e os atributos da classe ficam sendo privados, ou seja, apenas funções-membro da classe têm acesso direto aos mesmos.

2.8 HERANÇA

Herança é a propriedade dos objetos que permite a criação de uma hierarquia entre eles, onde os descendentes herdam o acesso ao código e estruturas de dados dos seus ancestrais.

2.9 POLIMORFISMO

Polimorfismo é a propriedade de uma ou mais classes responderem a mesma mensagem, cada uma de uma forma diferente. Numa linguagem orientada a objeto, uma referência polimórfica é tal que, no decorrer do desenvolvimento do software, refere-se a mais de uma classe. Desta forma é possível explorar similaridades entre diferentes classes de objetos. Este conceito é útil para distinguir mensagens de um método. Um objeto emissor envia uma mensagem, se o objeto receptor implementa um método com a mesma assinatura, ele poderá respondê-la. Diferentes respostas serão possíveis, dependendo de como os métodos dos receptores estão implementados.

3 ESPECIFICAÇÃO DO SISTEMA

3.1 OBJETIVO DO TRABALHO

O sistema desenvolvido tem como objetivo preencher todas as necessidades da empresa com uma grande facilidade de uso, ajudando o funcionário a desenvolver suas atividades com maior perfeição e agilidade, com isso o proprietário terá um ganho de tempo e dinheiro.

3.2 MAPA MENTAL

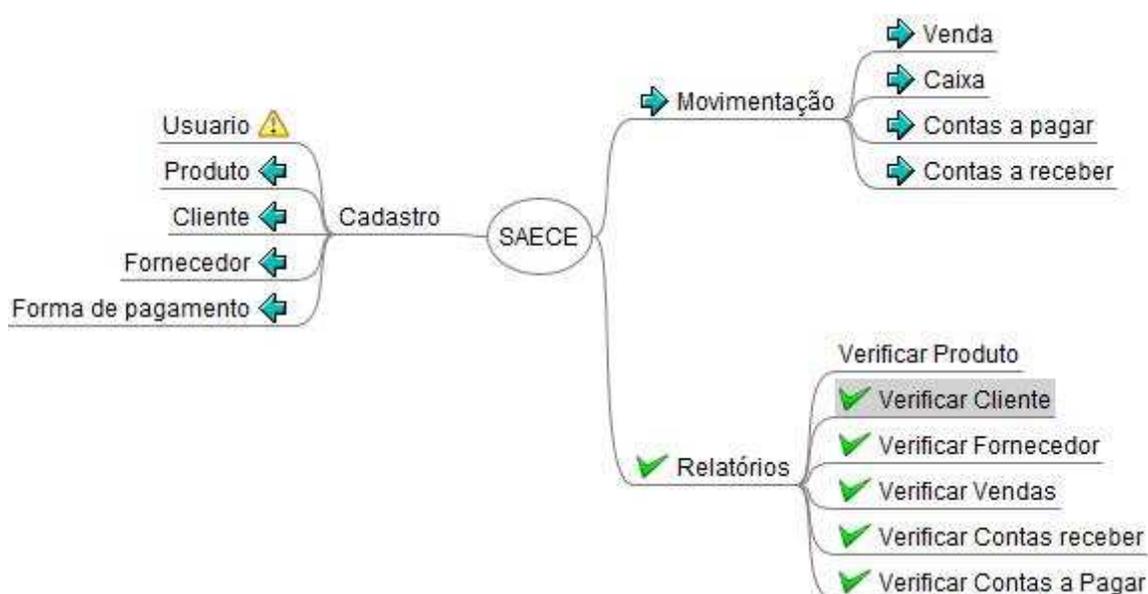


Figura 1 - Mapa Mental do Sistema Administrativo e Controle de Estoque

3.3 LEVANTAMENTO DE REQUISITOS

3.3.1 Entrevista realizada na Info House

a) Qual o ramo da Instituição?

R: A instituição atua no setor de informática, oferecendo venda de peças, acessórios e manutenção em computadores.

b) Quais as pessoas que fazem parte da empresa?

R: A empresa é formada pelo proprietário, vendedores, secretaria e técnicos de informática.

c) Descreva o problema do software já existente:

R: Os softwares utilizados pela empresa contem muitas falhas, não dando suporte a relatórios e um sistema muito lento.

d) Qual o resultado esperado com a implantação de um novo software?

R: Com o novo software espera-se ter um controle eficaz do caixa e estoque com relatórios concretos semanais e mensais.

e) No que influencia essas informações obtidas pelo software?

R: Essas informações são de muita importância, pois através dela é possível saber se a empresa esta tendo lucros e se os funcionários estão trabalhando corretamente devido ao controle do caixa.

3.3.2 Funcionalidades do Sistema

- Login de Acesso ao Sistema;
- Cadastros (Usuário, Produto, Fornecedor, Cliente, Forma de Pagamento)
- Controle administrativo do sistema;
- Controle de Caixa;
- Gerar Relatórios.

3.4 ANÁLISE DOS REQUISITOS

3.4.2 Lista de eventos

Nº	Descrição	Use case
01	Administrador gerencia contas de logins para novos usuários	Manter Usuário
02	Usuário cadastra Produto	Manter Produto
03	Usuário cadastra Cliente	Manter Cliente
04	Usuário cadastra Fornecedor	Manter Fornecedor
05	Usuário cadastra Forma de Pagamento	Manter Pagamento
06	Usuário	Movimentar vendas
07	Usuário	Movimentar Caixa
08	Usuário	Movimentar Contas a pagar
09	Usuário	Movimentar Contas a receber
10	Administrador solicita relatório	Verificar Produtos
11	Administrador solicita relatório	Verificar Cliente
12	Administrador solicita relatório	Verificar Fornecedor
13	Administrador solicita relatório	Verificar Vendas
14	Administrador solicita relatório	Verificar Contas a receber
15	Administrador solicita relatório	Verificar Contas a pagar

Tabela 1- Lista de Eventos

3.5 VALIDAÇÃO DOS REQUISITOS

a) Requisitos levantados através de entrevistas estão completos?

SIM (X) NÃO ()

b) As informações levantadas atende as necessidades do cliente?

SIM (X) NÃO ()

c) Os requisitos são reais?

SIM (X) NÃO ()

d) Houve algum tipo de redundância e falta de clareza dos requisitos?

SIM () NÃO (X)

e) A análise está de acordo com as necessidades solicitadas pelo cliente?

SIM (X) NÃO ()

4 DIAGRAMAS

4.1 DIAGRAMA DE CASO DE USO GERAL

4.1.1 Efetuar Login

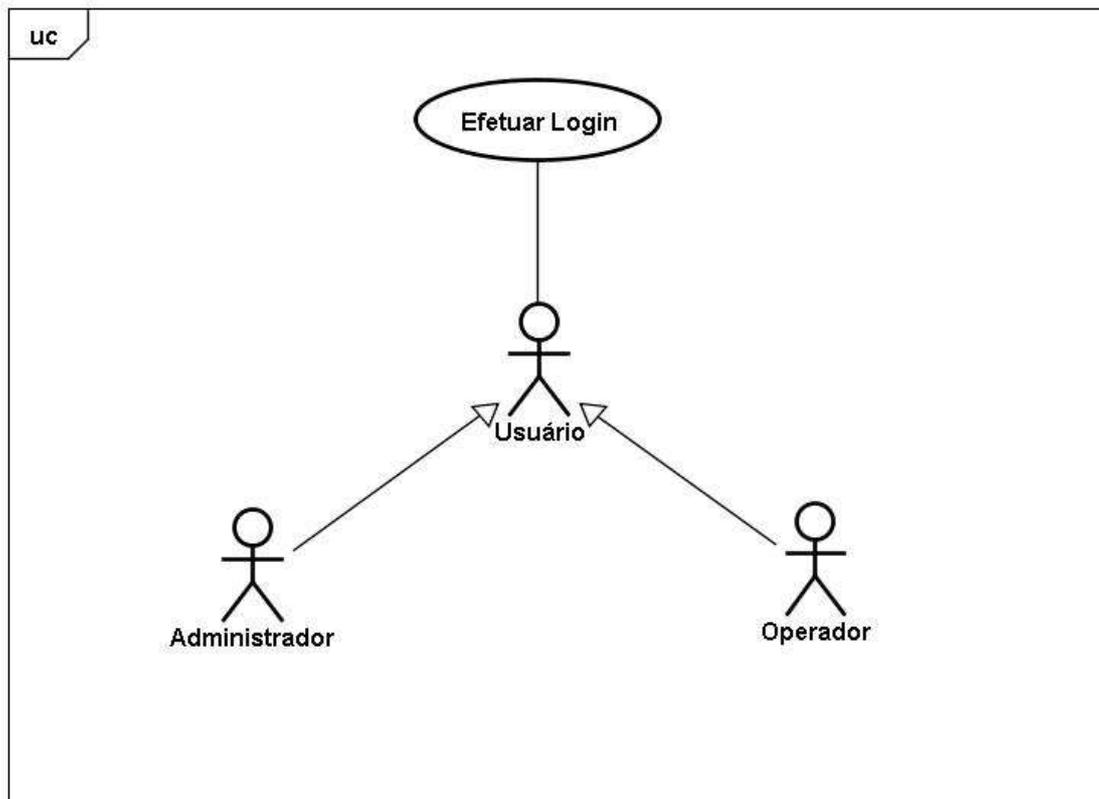


Figura 2 - Caso de Uso Visão Geral - Efetuar Login

4.1.2 Movimentações

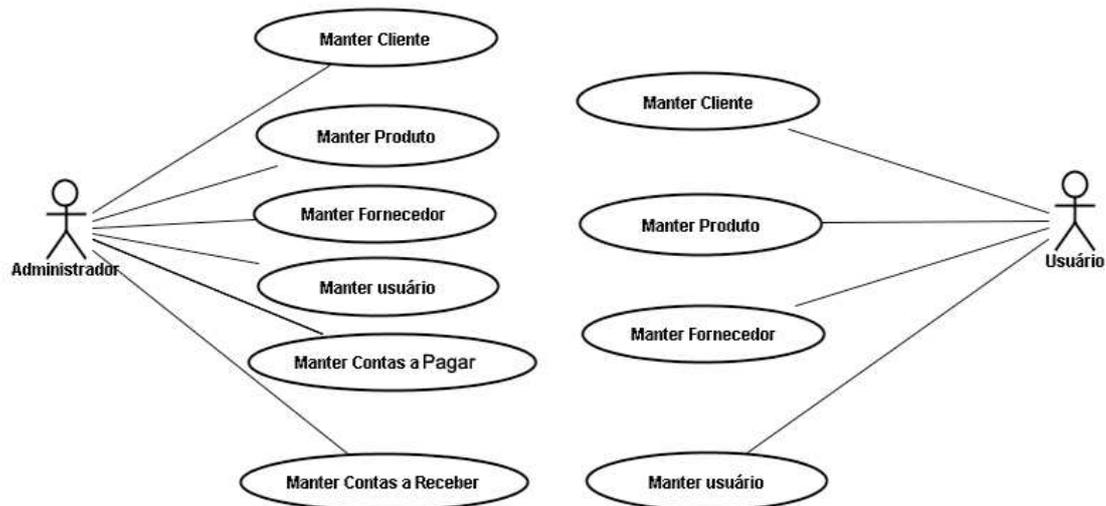


Figura 3 - Caso de Uso Visão Geral - Movimentações

4.1.3 Consultar Relatórios

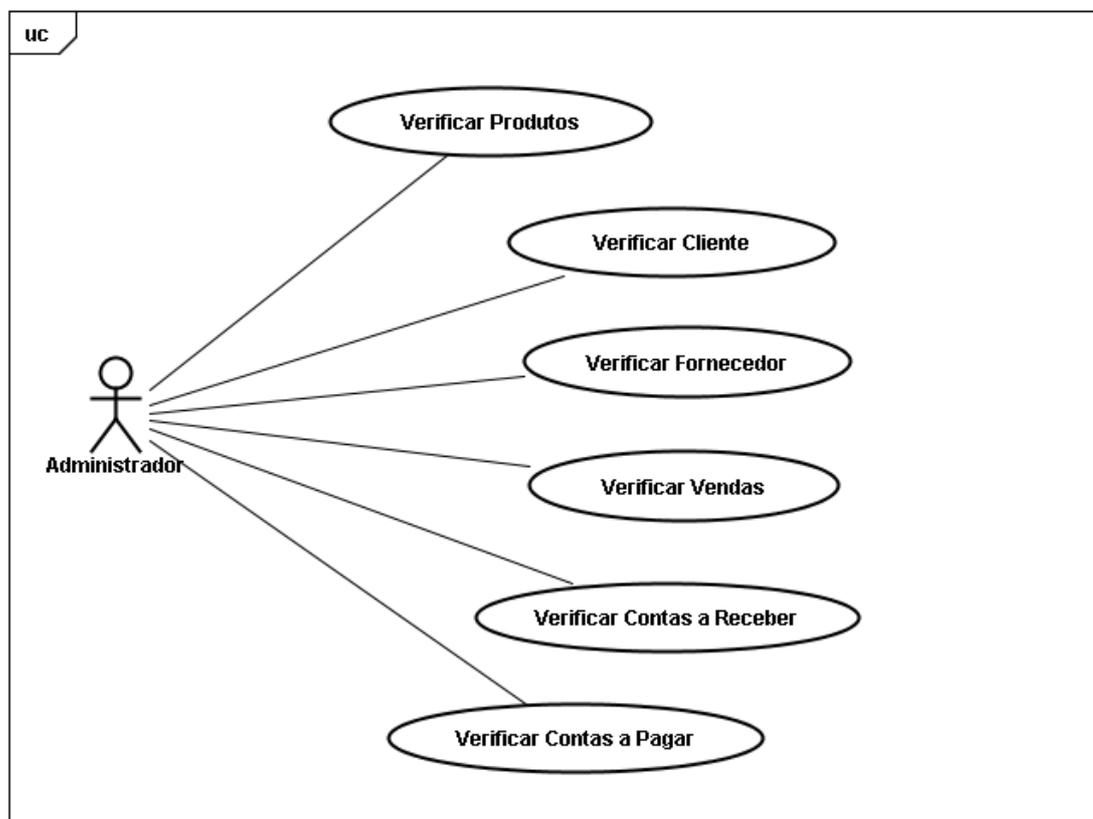


Figura 4 - Caso de Uso Visão Geral – Consultar Relatórios

4.2 DIAGRAMAS DE CASO DE USO ESPECÍFICO

4.2.1 UC 01 – Manter Usuário

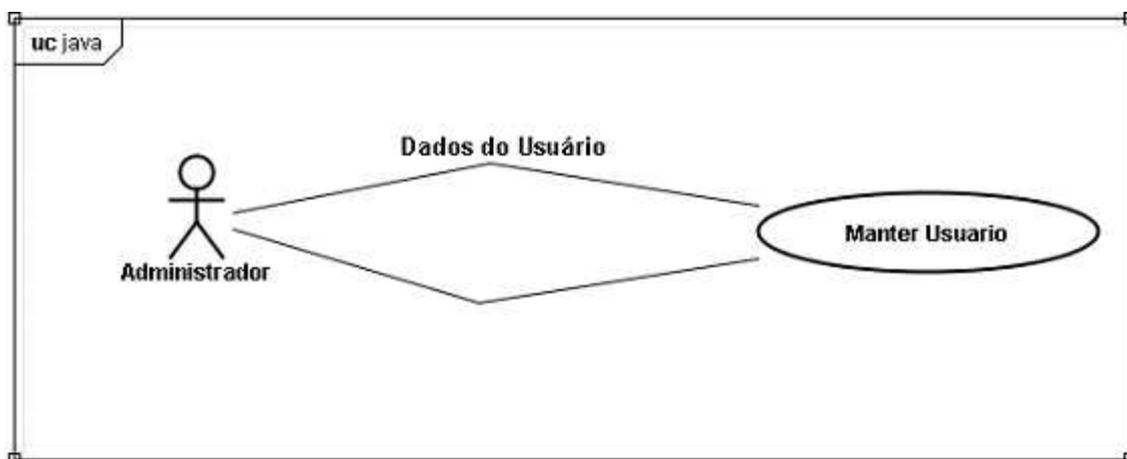


Figura 5 – UC – Manter Usuário

Finalidade/Objetivo:	Permitir a inclusão de novos usuários. O sistema terá um usuário previamente cadastrado durante sua implantação.
Atores:	Administrador do sistema;
Pré-condições:	O Administrador deverá estar autenticado (“logado”) no sistema;
Evento Inicial:	O Administrador do Sistema escolhe a opção “Cadastro de Usuários” na tela principal da aplicação (<i>ele foi até o menu e escolheu a opção</i>);
Fluxo Principal:	1 – O sistema solicita os dados necessários para o cadastro do usuário; 2 – O sistema efetiva a inclusão dos dados;
Fluxo Alternativo	1 – A qualquer momento, o Administrador do sistema poderá sair da tela de cadastro de usuários; 2 – A qualquer momento, o Administrador do sistema poderá sair do aplicativo;

Tabela 2 – Especificação do Caso de Uso 01

4.2.2 UC 02 – Manter Produto

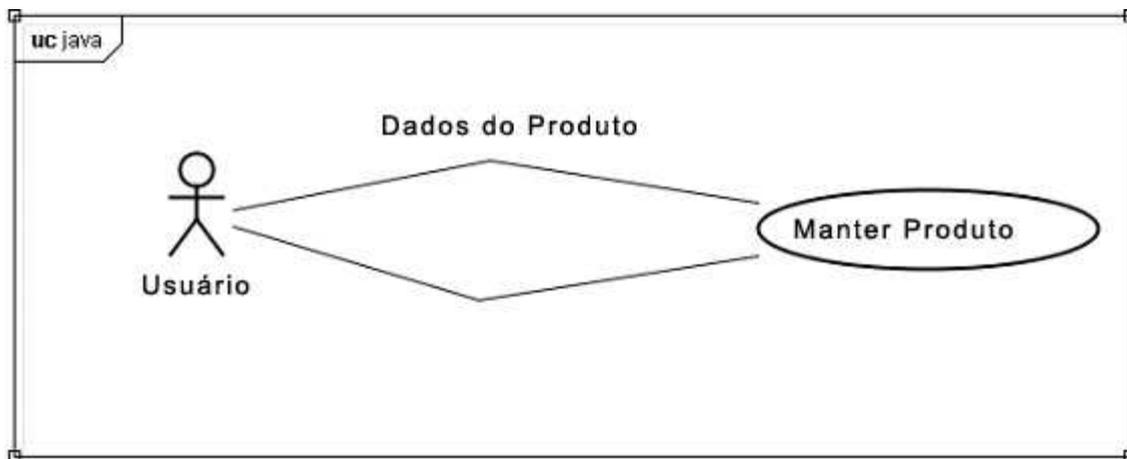


Figura 6 – UC – Manter Produto

Finalidade/Objetivo:	Permitir a inclusão de dados sobre o produto, cadastrar, editar, excluir.
Atores:	Usuário;
Pré-condições:	O Usuário deverá estar autenticado (“logado”) no sistema;
Evento Inicial:	O Usuário do Sistema escolhe a opção “Cadastro de Produto” (<i>ele foi até o menu e escolheu a opção</i>);
Fluxo Principal:	1 – O sistema solicita os dados necessários para o cadastro de Produto; 2 – O sistema efetiva a inclusão dos dados;
Fluxo Alternativo	1 – A qualquer momento, o Usuário do sistema poderá sair da tela de cadastro de Produto; 2 – A qualquer momento, o Usuário do sistema poderá sair do aplicativo;

Tabela 3 – Especificação do Caso de Uso 02

4.2.3 UC 03 – Manter Cliente

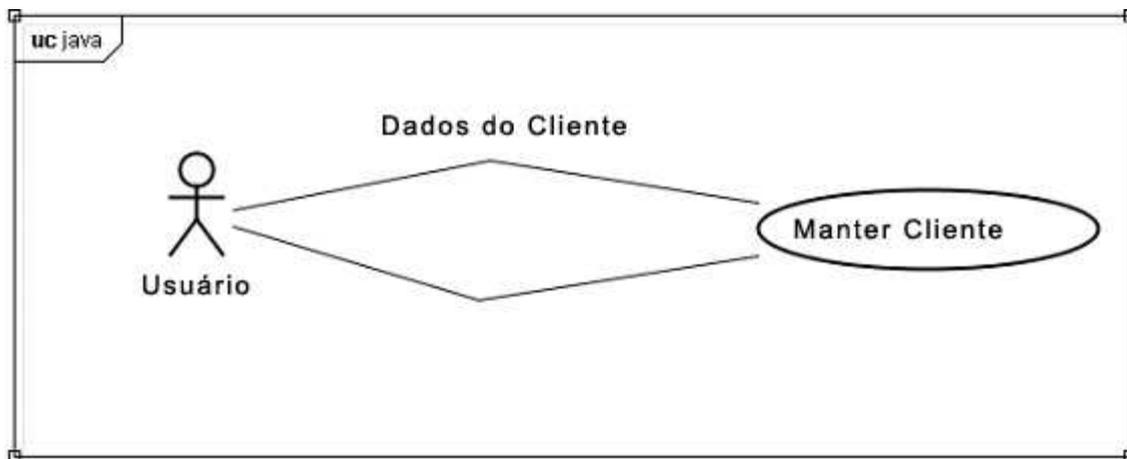


Figura 7 – UC – Manter Dados do Cliente

Finalidade/Objetivo:	Permitir a inclusão de dados sobre o Cliente, cadastrar, editar, excluir.
Atores:	Usuário;
Pré-condições:	O Usuário deverá estar autenticado ("logado") no sistema;
Evento Inicial:	O Usuário do Sistema escolhe a opção "Cadastro de Cliente" (<i>ele foi até o menu e escolheu a opção</i>);
Fluxo Principal:	1 – O sistema solicita os dados necessários para o cadastro de Cliente; 2 – O sistema efetiva a inclusão dos dados;
Fluxo Alternativo	1 – A qualquer momento, o Administrador do sistema poderá sair da tela de cadastro de Cliente; 2 – A qualquer momento, o Administrador do sistema poderá sair do aplicativo;

Tabela 4 – Especificação do Caso de Uso 03

4.2.4 UC 04 – Manter Fornecedor

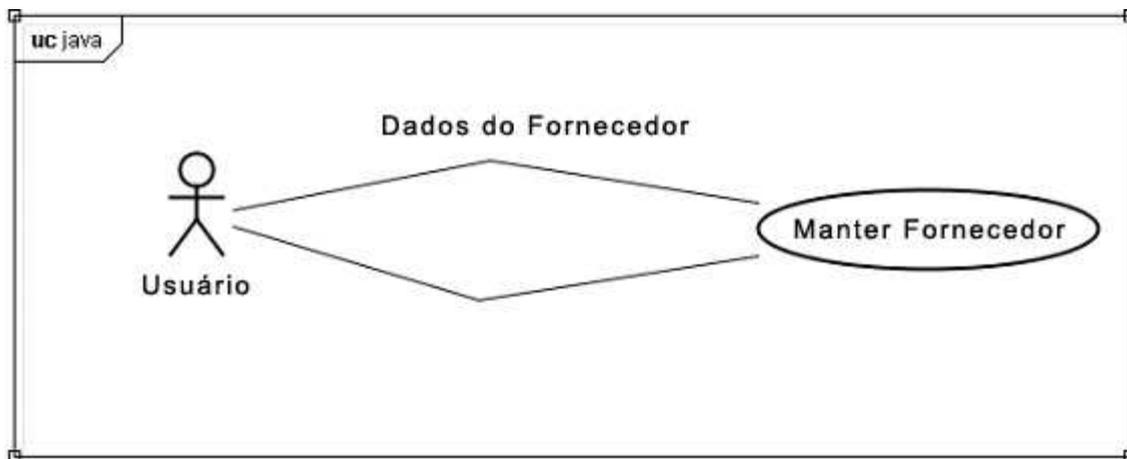


Figura 8 - UC – Manter Cliente

Finalidade/Objetivo:	Permitir a inclusão de dados sobre o fornecedor, cadastrar, editar, excluir.
Atores:	Usuário;
Pré-condições:	O Usuário deverá estar autenticado ("logado") no sistema;
Evento Inicial:	O Usuário do Sistema escolhe a opção "Cadastro de Fornecedor" <i>(ele foi até o menu e escolheu a opção)</i> ;
Fluxo Principal:	1 – O sistema solicita os dados necessários para o cadastro do fornecedor; 2 – O sistema efetiva a inclusão dos dados;
Fluxo Alternativo	1 – A qualquer momento, o Usuário do sistema poderá sair da tela de cadastro de usuário; 2 – A qualquer momento, o Usuário do sistema poderá sair do aplicativo;

Tabela 5 – Especificação do Caso de Uso 04

4.2.5 UC 05 – Manter Pagamento

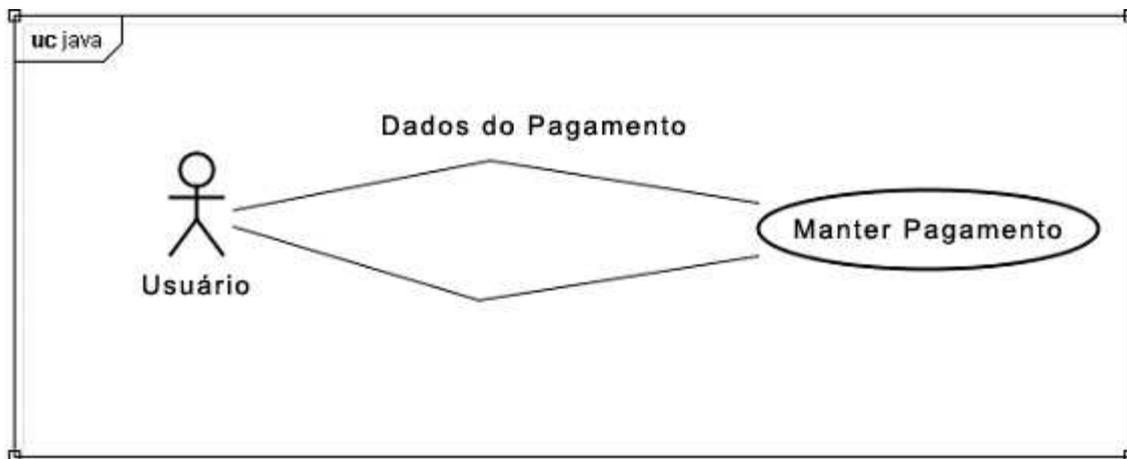


Figura 9 - UC – Manter Tipo de Pagamento

Finalidade/Objetivo:	Permitir que o Usuário entre no sistema;
Atores:	Usuário;
Pré-condições:	O Usuário deverá estar autenticado ("logado") no sistema;
Evento Inicial:	O usuário do Sistema escolhe a opção "Vendas/Pagamentos" (<i>ele foi até o menu e escolheu a opção</i>);
Fluxo Principal:	1 – O sistema solicita tipo de venda; 2 – O sistema efetiva a inclusão dos dados; 3- O sistema gera cadastro da vendas;
Fluxo Alternativo	1 – A qualquer momento, o Usuário do sistema poderá sair da tela de relatórios; 2 – A qualquer momento, o Usuário do sistema poderá sair do aplicativo;

Tabela 6 – Especificação do Caso de Uso 05

4.2.6 UC 06 – Movimentar vendas

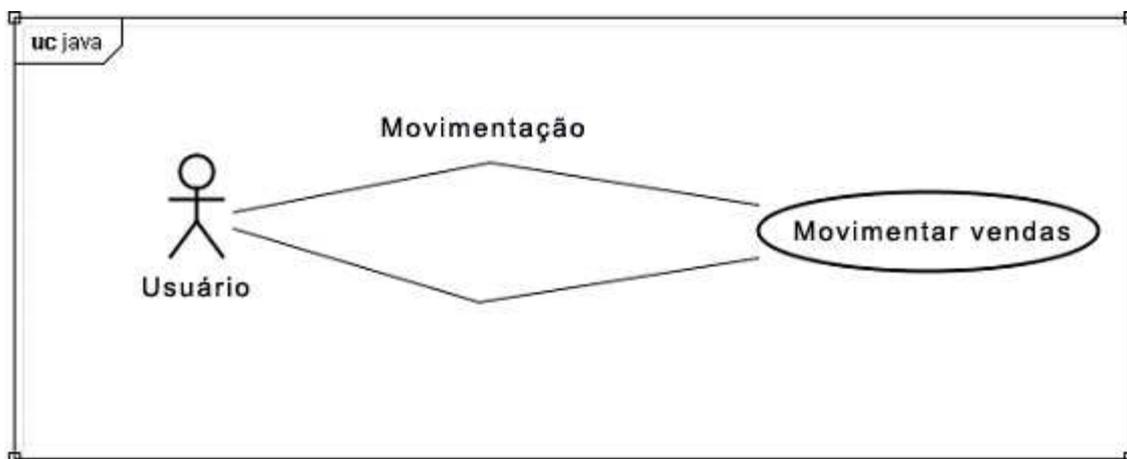


Figura 10 - UC – Movimentar vendas

Finalidade/Objetivo:	Permitir que o Usuário entre no sistema;
Atores:	Usuário;
Pré-condições:	O Usuário deverá estar autenticado ("logado") no sistema;
Evento Inicial:	O usuário do Sistema escolhe a opção "Vendas" (<i>ele foi até o menu e escolheu a opção</i>);
Fluxo Principal:	1 – O sistema solicita tipo de venda; 2 – O sistema efetiva a inclusão dos dados; 3- O sistema gera cadastro da vendas;
Fluxo Alternativo	1 – A qualquer momento, o Usuário do sistema poderá sair da tela de relatórios; 2 – A qualquer momento, o Usuário do sistema poderá sair do aplicativo;

Tabela 7 – Especificação do Caso de Uso 06

4.2.7 UC 07 – Movimentar Caixa

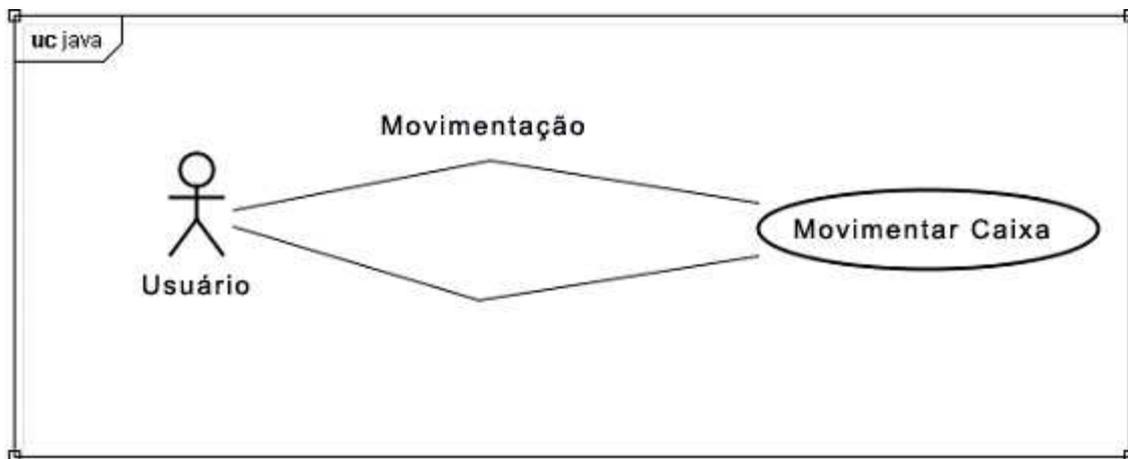


Figura 11 - UC – Movimentar Caixa

Finalidade/Objetivo:	Permitir que o usuário entre no caixa.
Atores:	Usuário;
Pré-condições:	O Usuário deverá estar autenticado ("logado") no sistema;
Evento Inicial:	O Administrado do Sistema escolhe a opção "Caixa" (<i>ele foi até o menu e escolheu a opção</i>);
Fluxo Principal:	1 – O sistema solicita a opção ; 2 – O sistema abre a opcao;
Fluxo Alternativo	1 – A qualquer momento, o Usuário do sistema poderá sair da tela de caixa; 2 – A qualquer momento, o Usuário do sistema poderá sair do aplicativo;

Tabela 8 – Especificação do Caso de Uso 07

4.2.8 UC 08 – Movimentar Contas a Pagar

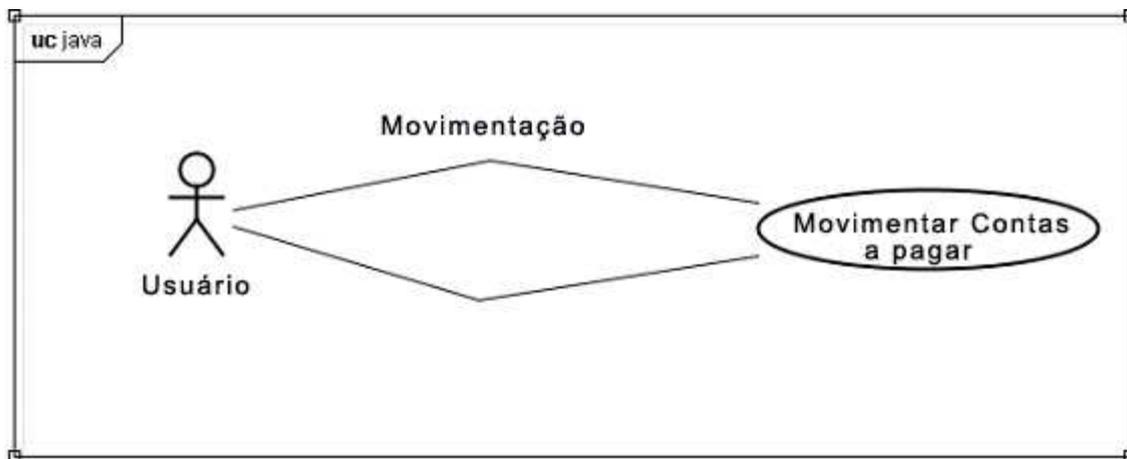


Figura 12 - UC – Movimentar Contas a Pagar

Finalidade/Objetivo:	Permitir que o Usuário entre no caixa;
Atores:	Usuário;
Pré-condições:	O Usuário deverá estar autenticado ("logado") no sistema;
Evento Inicial:	O Usuário do Sistema escolhe a opção "Caixa/Contas a pagar" (<i>ele foi até o menu e escolheu a opção</i>);
Fluxo Principal:	1 – O sistema solicita a data; 2 – O sistema efetiva a inclusão dos dados; 3- O sistema gera Boletos;
Fluxo Alternativo	1 – A qualquer momento, o Usuário do sistema poderá sair da tela de relatórios; 2 – A qualquer momento, o Usuário do sistema poderá sair do aplicativo;

Tabela 9 – Especificação do Caso de Uso 08

4.2.9 UC 09 – Movimentar Contas a Receber

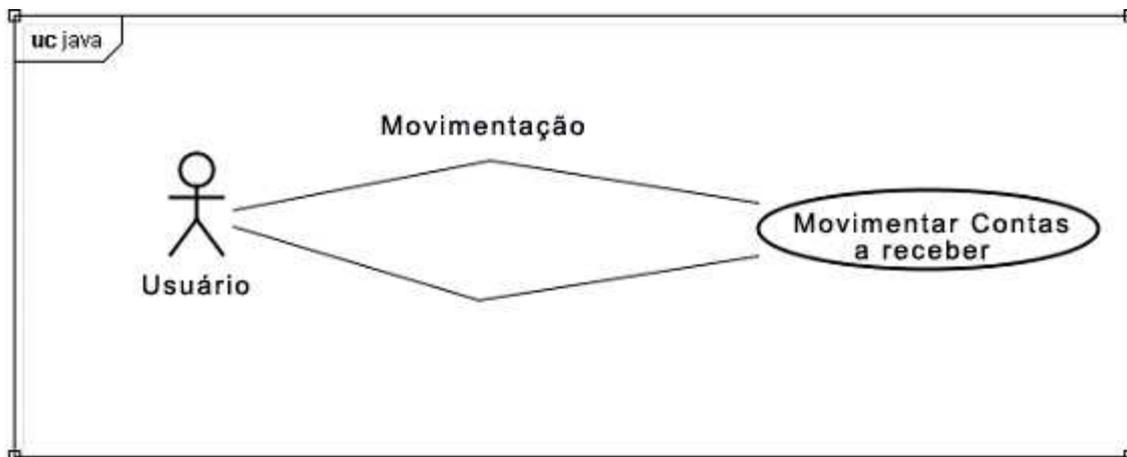


Figura 13 - UC – Movimentar Contas a receber

Finalidade/Objetivo:	Permitir que o Usuário entre no caixa;
Atores:	Usuário;
Pré-condições:	O Usuário deverá estar autenticado ("logado") no sistema;
Evento Inicial:	O usuário do Sistema escolhe a opção "Caixa/Contas a Receber" <i>(ele foi até o menu e escolheu a opção);</i>
Fluxo Principal:	1 – O sistema solicita a data; 2 – O sistema efetiva a inclusão dos dados; 3- O sistema gera cadastro da conta;
Fluxo Alternativo	1 – A qualquer momento, o Usuário do sistema poderá sair da tela de relatórios; 2 – A qualquer momento, o Usuário do sistema poderá sair do aplicativo;

Tabela 10 – Especificação do Caso de Uso 09

4.2.10 UC 10 – Verificar Produtos

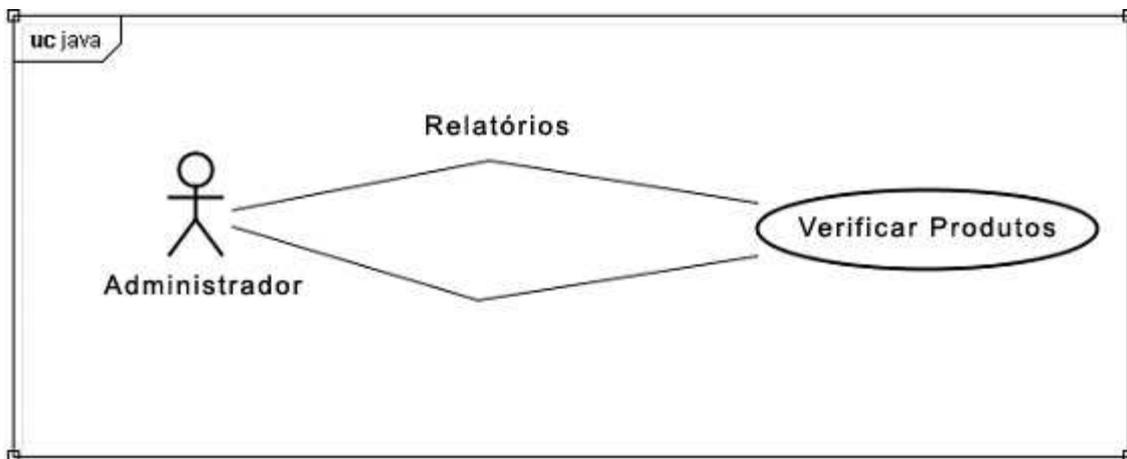


Figura 14 - UC – Verificar Produtos

Finalidade/Objetivo:	Permitir que o Administrador verifica relatórios ;
Atores:	Administrador;
Pré-condições:	O Usuário deverá estar autenticado (“logado”) no sistema;
Evento Inicial:	O Administrado do Sistema escolhe a opção “Relatórios/Produto” <i>(ele foi até o menu e escolheu a opção)</i> ;
Fluxo Principal:	1 – O sistema solicita a data para geral o relatório; 2 – O sistema efetiva a inclusão dos dados; 3- O sistema gera o relatório
Fluxo Alternativo	1 – A qualquer momento, o Administrador do sistema poderá sair da tela de relatórios; 2 – A qualquer momento, o Administrador do sistema poderá sair do aplicativo;

Tabela 11 – Especificação do Caso de Uso 10

4.2.11 UC 11 – Verificar Cliente

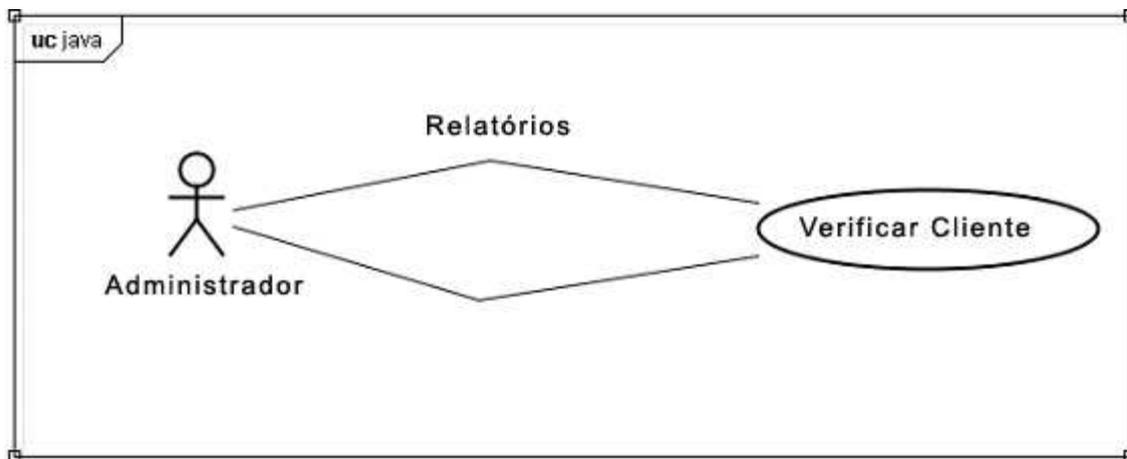


Figura 15 - UC – Verificar Cliente

inalidade/Objetivo:	Permitir que o Administrador verifica relatórios;
Atores:	Administrador;
Pré-condições:	O Usuário deverá estar autenticado ("logado") no sistema;
Evento Inicial:	O Administrado do Sistema escolhe a opção "Relatórios/Cadastro/Cliente " (<i>ele foi até o menu e escolheu a opção</i>);
Fluxo Principal:	<ol style="list-style-type: none"> 1 – O sistema solicita a data para geral o relatório; 2 – O sistema efetiva a inclusão dos dados; 3- O sistema gera o relatório
Fluxo Alternativo	<ol style="list-style-type: none"> 1 – A qualquer momento, o Administrador do sistema poderá sair da tela de relatórios; 2 – A qualquer momento, o Administrador do sistema poderá sair do aplicativo;

Tabela 12 – Especificação do Caso de Uso 11

4.2.12 UC 12 – Verificar Fornecedor

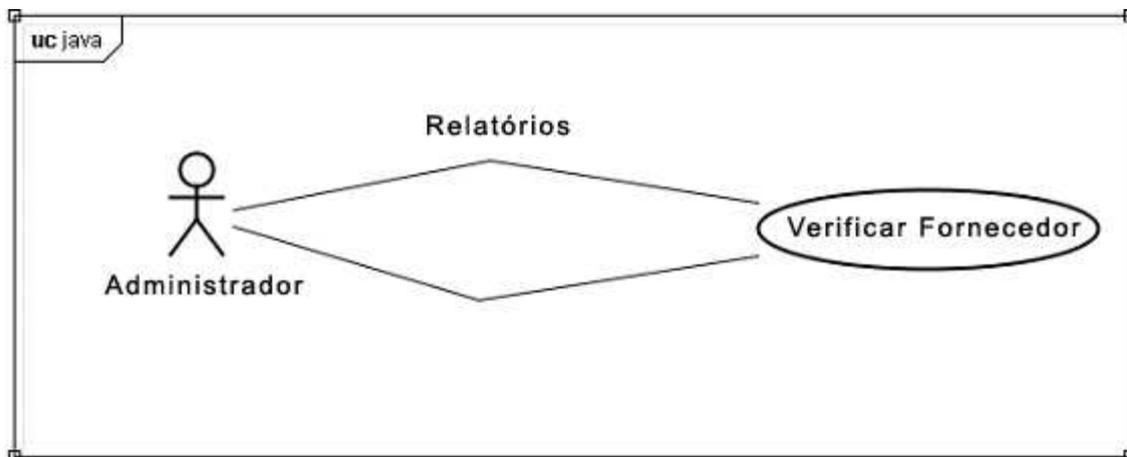


Figura 16 - UC – Verificar Fornecedor

Finalidade/Objetivo:	Permitir que o Administrador verifica relatórios sobre Fornecedor;
Atores:	Administrador;
Pré-condições:	O Administrador deverá estar autenticado (“logado”) no sistema;
Evento Inicial:	O Administrado do Sistema escolhe a opção “Relatórios/Fornecedor” <i>(ele foi até o menu e escolheu a opção)</i> ;
Fluxo Principal:	1 – O sistema solicita a data para geral o relatório; 2 – O sistema efetiva a inclusão dos dados; 3- O sistema gera o relatório
Fluxo Alternativo	1 – A qualquer momento, o Administrador do sistema poderá sair da tela de relatórios; 2 – A qualquer momento, o Administrador do sistema poderá sair do aplicativo;

Tabela 13 – Especificação do Caso de Uso 12

4.2.13 UC 13 – Verificar vendas

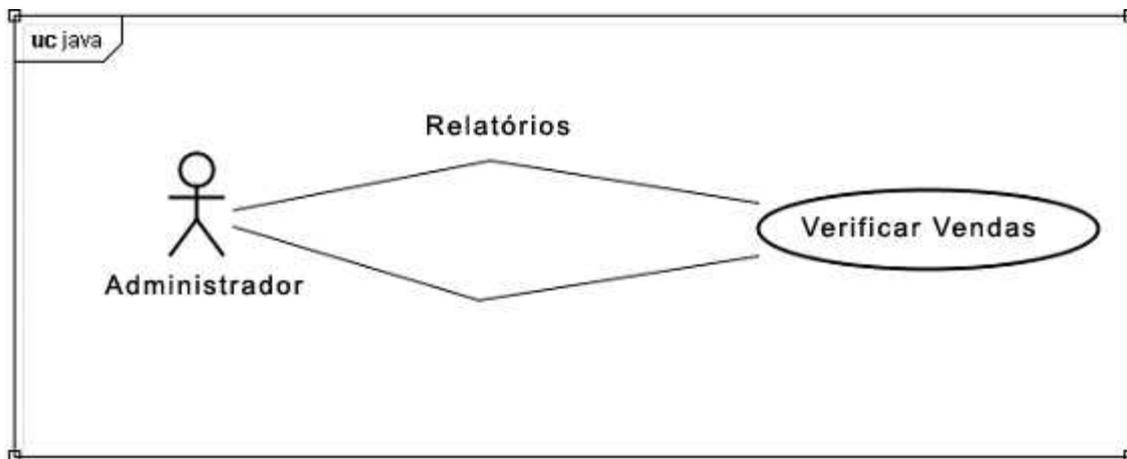


Figura 17 - UC – Verificar Vendas

Finalidade/Objetivo:	Permitir que o Administrador verifica relatório;
Atores:	Administrador;
Pré-condições:	O Usuário deverá estar autenticado ("logado") no sistema;
Evento Inicial:	O Administrado do Sistema escolhe a opção "Relatórios/Verificar Vendas" (<i>ele foi até o menu e escolheu a opção</i>);
Fluxo Principal:	1 – O sistema solicita a data para geral o relatório; 2 – O sistema efetiva a inclusão dos dados; 3- O sistema gera o relatório
Fluxo Alternativo	1 – A qualquer momento, o Administrador do sistema poderá sair da tela de relatórios; 2 – A qualquer momento, o Administrador do sistema poderá sair do aplicativo;

Tabela 14 – Especificação do Caso de Uso 13

4.2.14 UC 14 – Verificar Contas a Receber

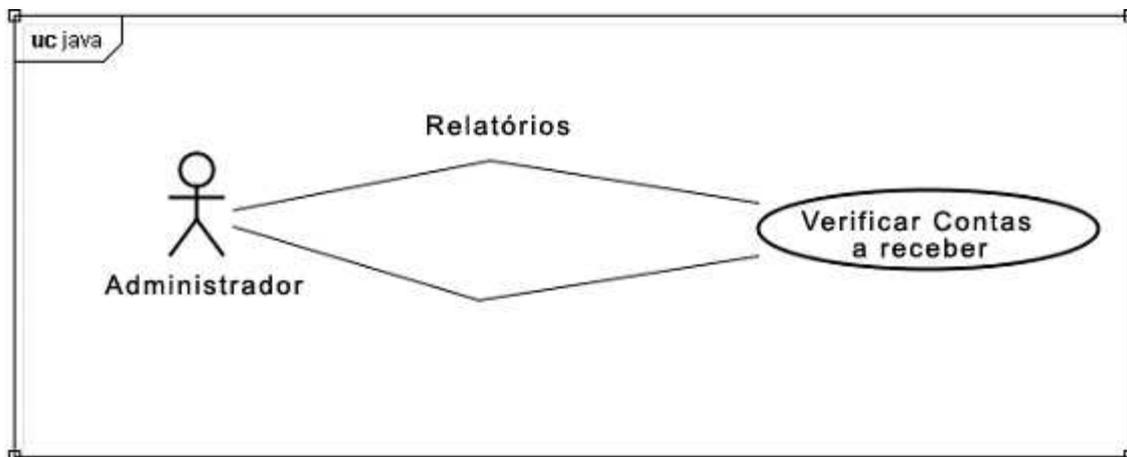


Figura 18 – UC – Efetuar devolução

Finalidade/Objetivo:	Permitir que o Administrador verifica relatórios ;
Atores:	Administrador;
Pré-condições:	O Administrador deverá estar autenticado ("logado") no sistema;
Evento Inicial:	O Administrado do Sistema escolhe a opção "Relatórios/Caixa/contas a Receber" (<i>ele foi até o menu e escolheu a opção</i>);
Fluxo Principal:	1 – O sistema solicita a data para geral o relatório; 2 – O sistema efetiva a inclusão dos dados; 3- O sistema gera o relatório
Fluxo Alternativo	1 – A qualquer momento, o Administrador do sistema poderá sair da tela de relatórios; 2 – A qualquer momento, o Administrador do sistema poderá sair do aplicativo;

Tabela 15 – Especificação do Caso de Uso 14

4.2.15 UC 15 – Verificar Contas a Pagar

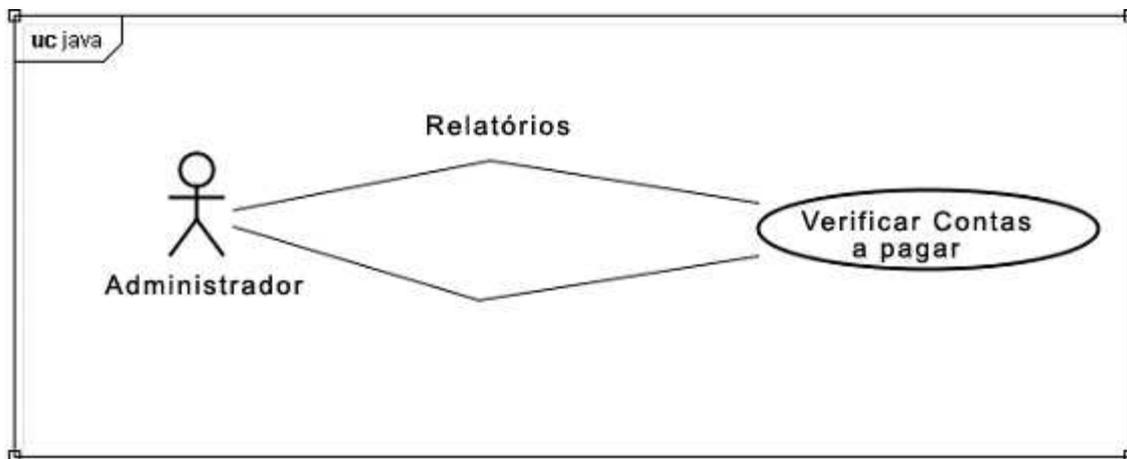


Figura 19 – UC – Verificar Contas a Pagar

Finalidade/Objetivo:	Permitir que o Administrador verifica relatórios;
Atores:	Administrador;
Pré-condições:	O Usuário deverá estar autenticado (“logado”) no sistema;
Evento Inicial:	O Administrado do Sistema escolhe a opção “Relatórios/Caixa/contas a Pagar” (<i>ele foi até o menu e escolheu a opção</i>);
Fluxo Principal:	1 – O sistema solicita a data para geral o relatório; 2 – O sistema efetiva a inclusão dos dados; 3- O sistema gera o relatório
Fluxo Alternativo	1 – A qualquer momento, o Administrador do sistema poderá sair da tela de relatórios; 2 – A qualquer momento, o Administrador do sistema poderá sair do aplicativo;

Tabela 16 – Especificação do Caso de Uso 15

4.3 DIAGRAMA DE CLASSE

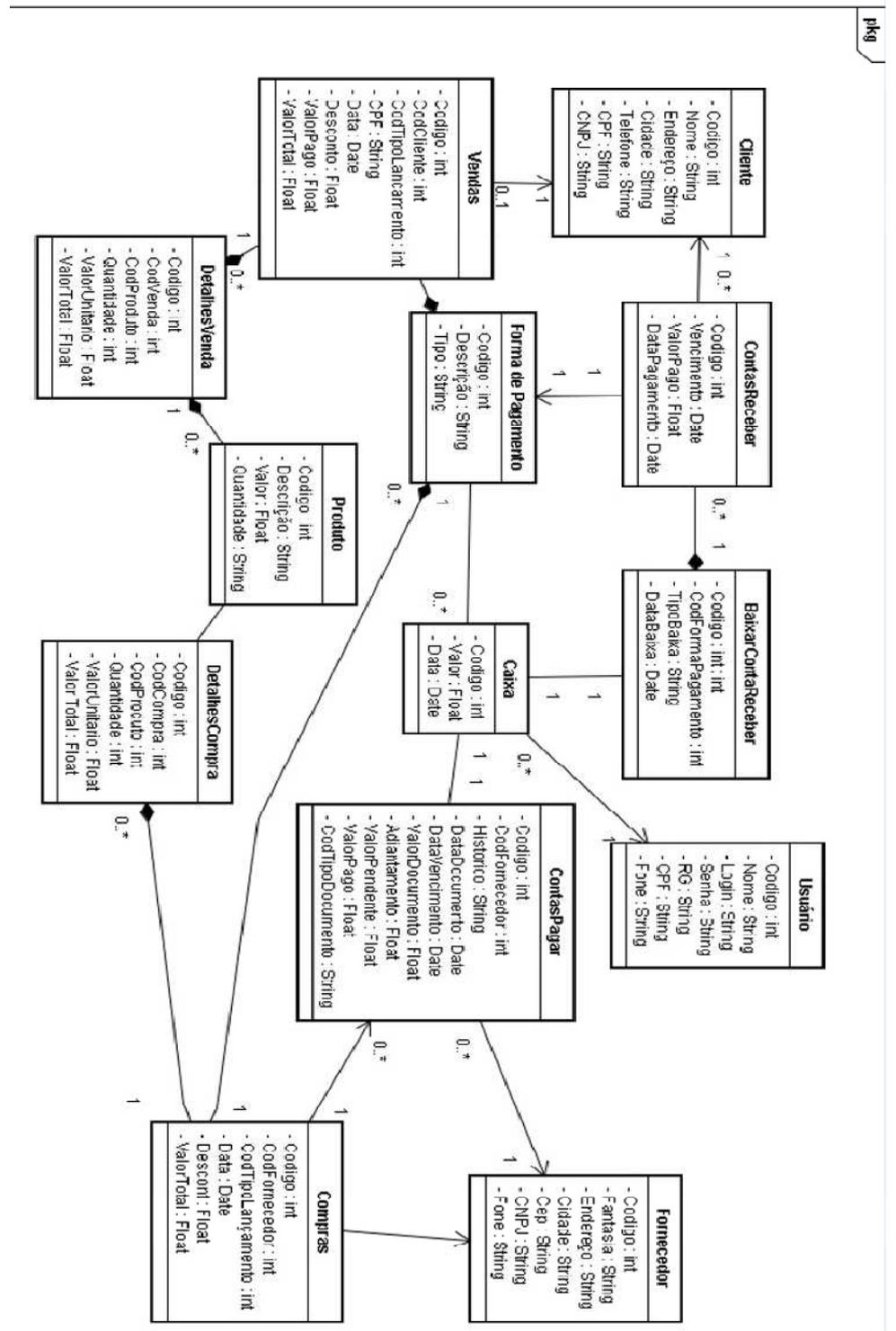


Figura 19 - Diagrama de Classes

4.4 DIAGRAMA DE ATIVIDADES

4.4.1 Cadastros

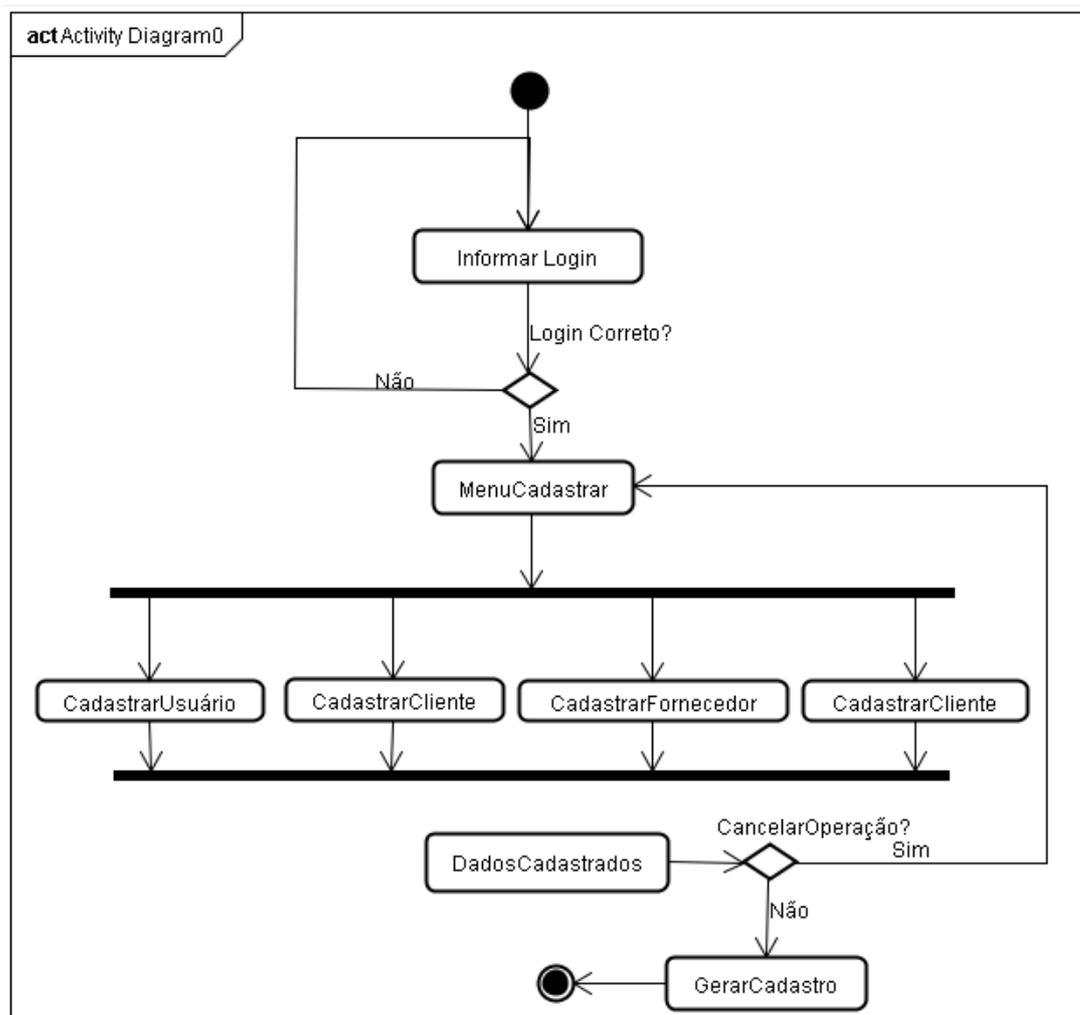


Figura 20 - Diagrama de Atividades - Cadastros

4.4.2 Movimentações

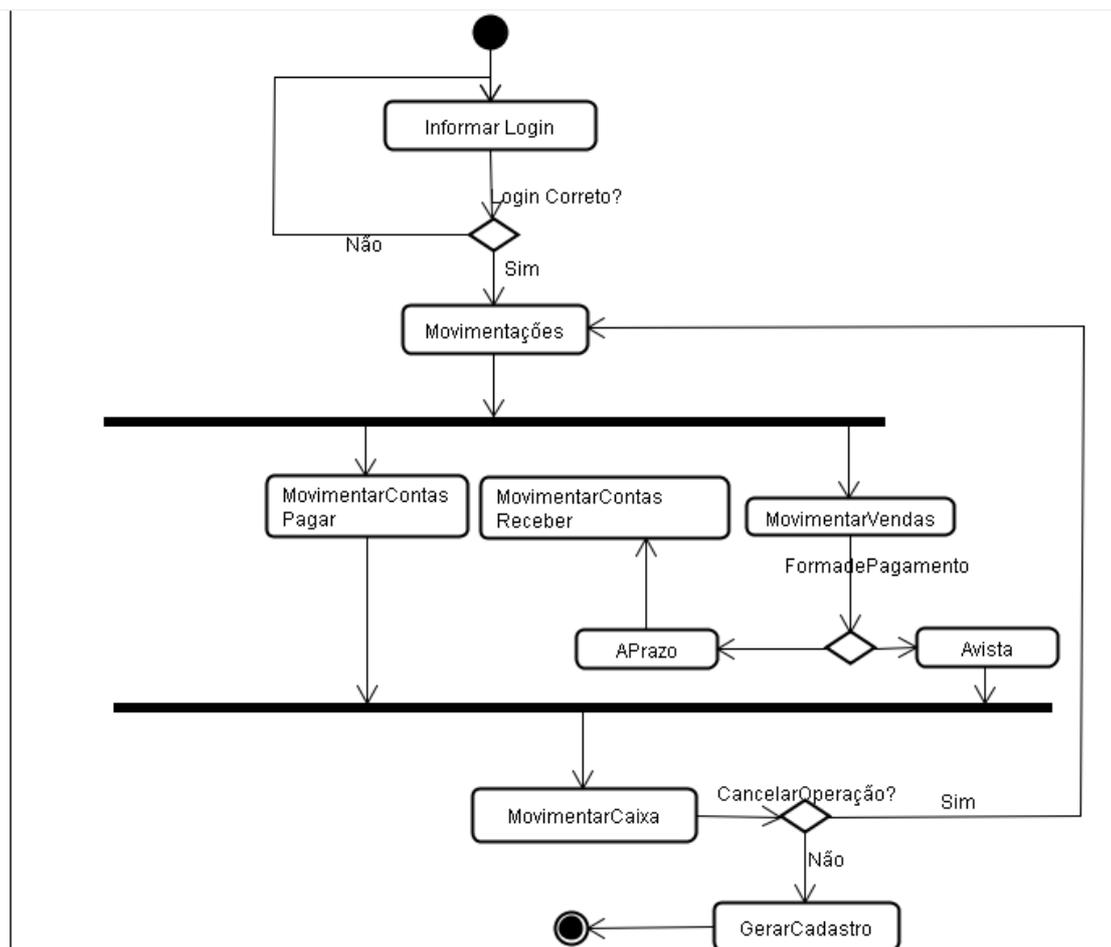


Figura 21 - Diagrama de Atividades - Movimentações

4.4.3 Relatórios

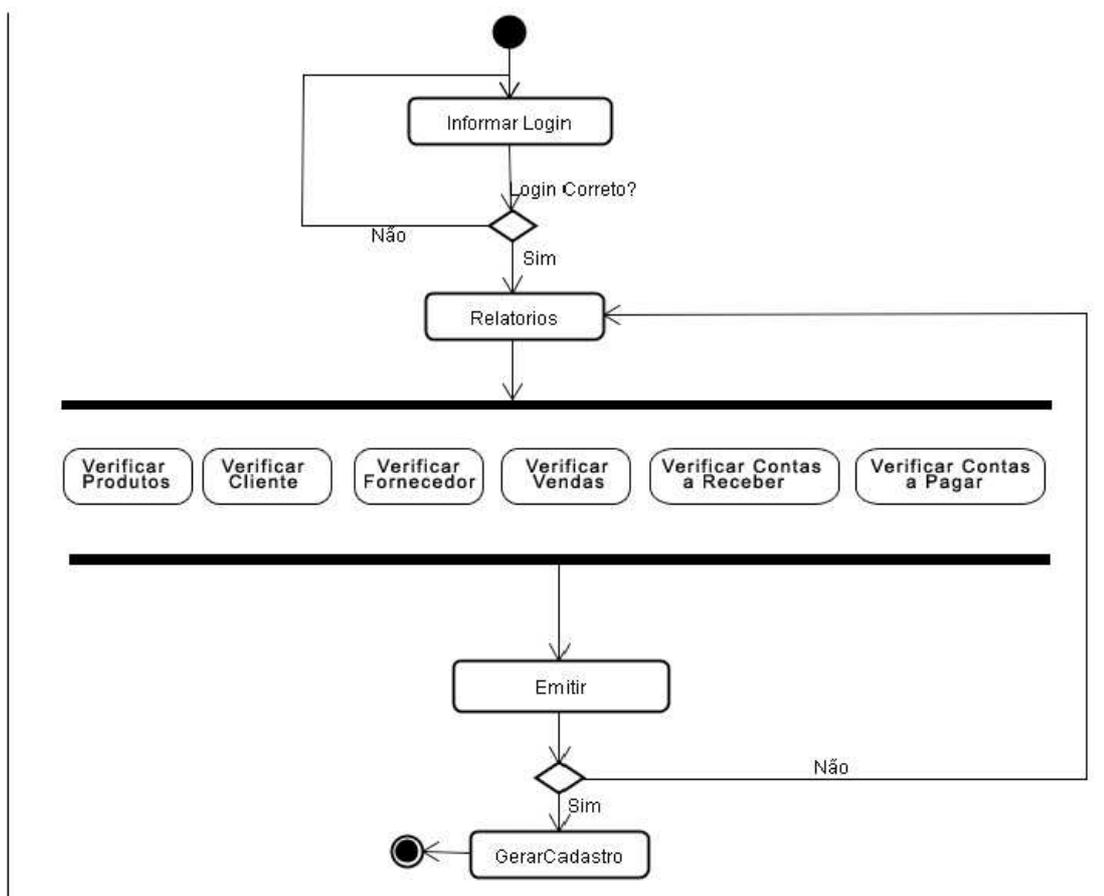


Figura 22 - Diagrama de Atividades - Relatórios

5. PROJETO

5.1 ESTRUTURA ANALÍTICA DO PROJETO (WBS)

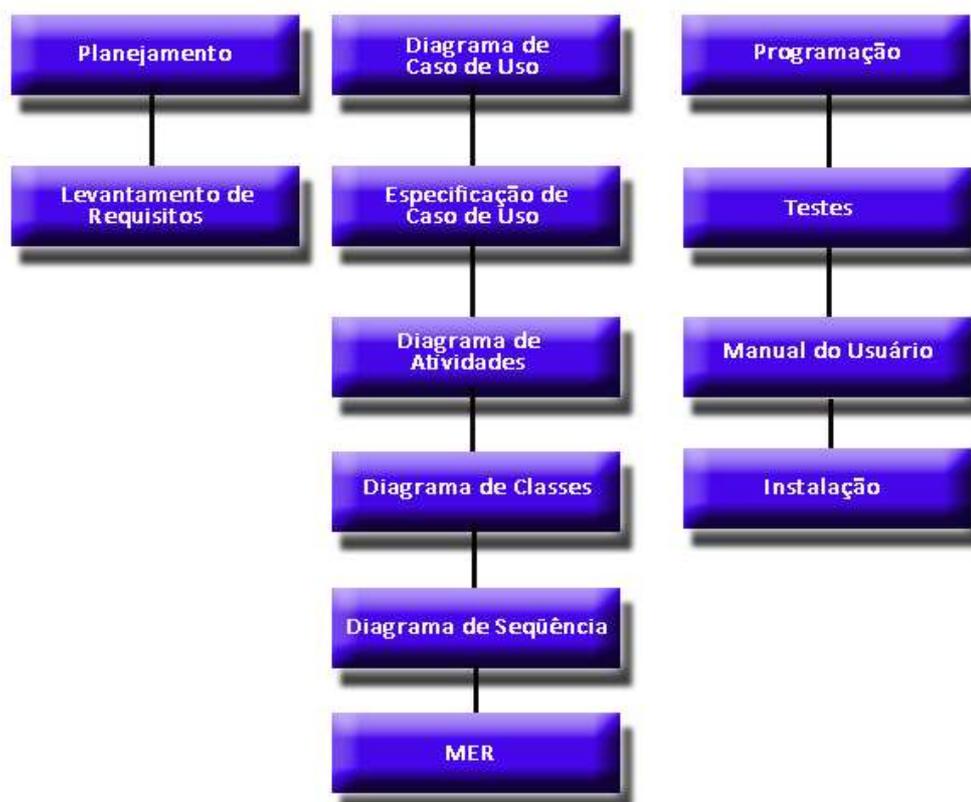


Figura 23 - WBS (Work Breakdown Structure)

5.2 SEQUENCIAMENTO DE ATIVIDADES

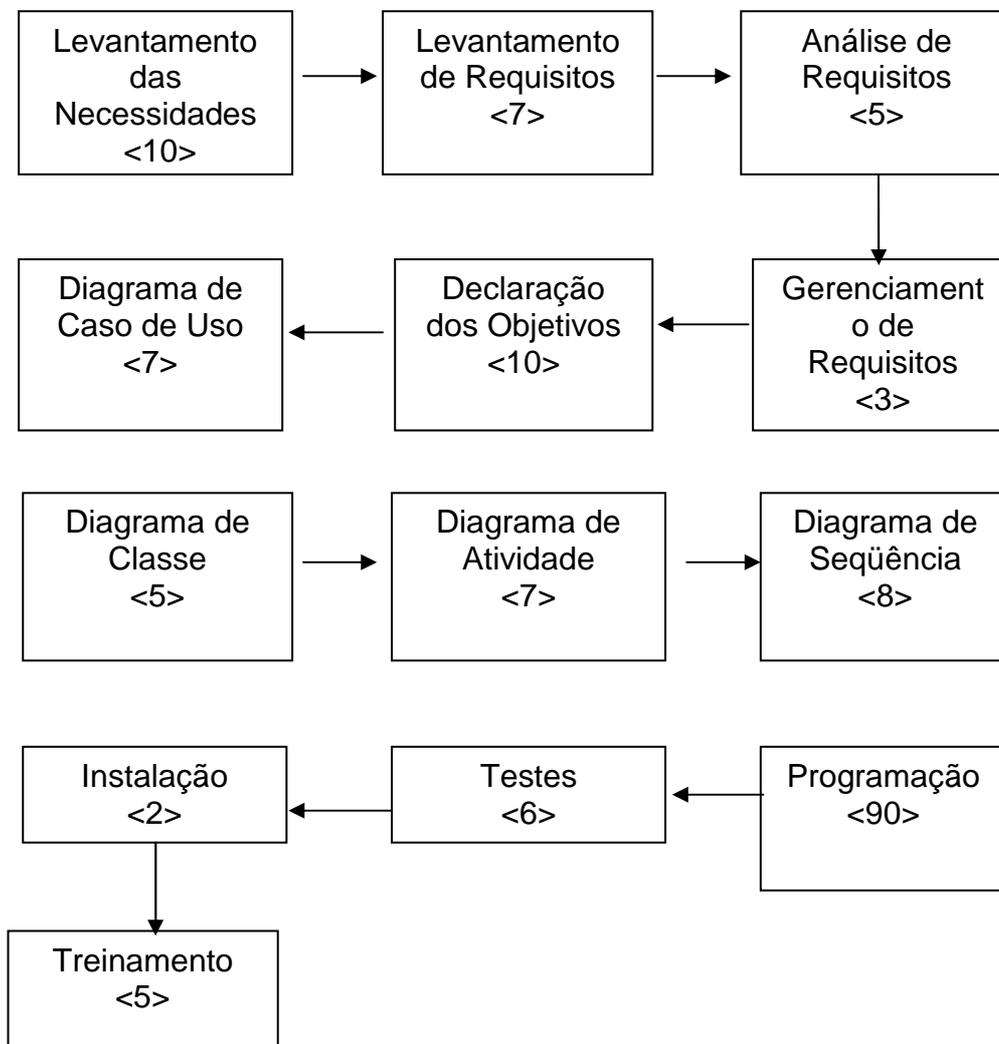


Figura 24 - Seqüenciamento de Atividades

5.3 ESPECIFICAÇÃO DOS CUSTOS

5.3.1 Recursos necessários para o desenvolvimento

- 01 Analista - Programador
- 01 Notebook Intel® Core™2 Duo CPU T6400 @2.00GHz, 4,00 GB RAM;
- 01 Impressora Multifuncional Epson
- Sistema Operacional Windows Seven;
- Eclipse;
- Postgres;
- IReports;

5.3.2 Estimativa de custos

Custo Analista – Programador

Custo Diário: R\$40,00 (Quarenta Reais)

Total de Dias: 167

Custo Total: (Total de Dias * Custo Diário): R\$6.680,00 (Seis mil seiscentos e oitenta reais).

5.3.3 Custos materiais

Notebook: R\$ 2.400,00 (Dois mil e quatrocentos reais)

Depreciação 3 anos: $R\$2.400,00 / 36$ (meses) = R\$66,66 (Sessenta e seis reais e sessenta e seis centavos) por mês;

Custo de um dia: $R\$66,66 / 30$ (dias) = R\$ 2,22 (Dois reais e vinte e dois centavos) por dia;

Custo de 167 dias: $R\$2,22 * 167 = R\$371,11$ (Trezentos e setenta e um reais e onze centavos).

Impressora: R\$ 320,00 (Trezentos e vinte reais)

Depreciação 3 anos: $R\$320,00 / 36$ (meses) = R\$8,88 (Oito reais e oitenta e oito centavos) por mês;

Custo de um dia: $R\$8,88 / 30$ (dias) = R\$ 0,29 (Vinte e nove centavos) por dia;

Custo de 167 dias: $R\$0,29*167 = R\$49,48$ (Quarenta e nove reais e quarenta e oito centavos).

Sistema Operacional Windows Seven Basic: R\$329,00 (Trezentos e vinte e nove reais);

Depreciação 2 anos: $R\$329,00 / 24$ (meses) = R\$13,78 (Treze reais e setenta e oito centavos) por mês;

Custo de um dia: $R\$13,78 / 30$ (dias) = R\$0,45 (Quarenta e cinco centavos) por dia;

Custo de 167 dias: $R\$0,45*167 = R\$76,31$ (Setenta e seis reais e trinta e um centavos);

Eclipse: R\$0,00;

Postgres: R\$ 0,00;

IReports: R\$ 0,00;

5.3.4 Orçamento do sistema

Recursos Utilizados	Valor
Analista – Programador	R\$ 6.680,00
Notebook	R\$ 371,11
Impressora	R\$ 49,48
Sistema Operacional Windows Seven Basic	R\$ 76,31
Eclipse	R\$ 0,00
Postgres	R\$ 0,00
Ireports	R\$ 0,00
Total	R\$ 7.176,90

Tabela 31 - Orçamento do Projeto

6. CONCLUSÃO

Através deste software, pretendo oferecer um melhor funcionamento para a empresa, tornando o atendimento aos clientes mais ágil, podendo melhorar o controle de atendimento. O caixa agora será controlado junto com o estoque facilitando os relatórios.

Como a informática hoje é essencial em qualquer ramo de atividade, o software irá suprir toda necessidade de organização, fornecendo relatórios gerenciais que são de extrema importância para uma eventual tomada de decisão, assim garantindo a qualidade de serviço prestado.

Desenvolvendo este sistema em linguagem JAVA orientada a objetos que no caso vem a ser uma forma especial de programar, mais próximo de como expressaríamos as coisas na vida real do que outros tipos de programação têm que aprender a pensar as coisas de uma maneira distinta, para escrever nossos programas em termos de objetos, propriedades, métodos e outras. Usando também o método de camada MVC, fica muito simples até mesmo na hora de dar suporte ao sistema quando surgir algum tipo de problema.

7 REFERÊNCIAS

- * BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 1. ed. Rio de Janeiro: Editora Campus, 2002.
- * RUMBAUGH, James. BRAHA, Michael. Modelagem e Projetos Baseados em Objetos com UML 2. James Rumbaugh, Michael Braha, Rio de Janeiro: Elsevier, 2006.
- * Shalloway, Alan. Explicação Padrões de Projeto: Uma Nova Perspectiva em Projeto Orientado a Objeto / Alan Shalloway e James R. Trott; trad. Ana M. De Alencar Price. – Porto Alegre: Bookman, 2004..
- * Sierra, Kathy. Use a Cabeça Java. 2ª Edição, 2005.
- * GONÇALVES, EDSON, Dominando Eclipse.

REFERÊNCIAS ELETRÔNICAS

- * Programação orientada a objetos. Disponível em:
< www.jack.eti.br/www/arquivos/apostilas/java/poo.pdf >
- * Apostila UML. Disponível em:
< <http://www.scribd.com/doc/429906/Apostila-UML> >
- * Eclipse
< <http://www.eclipse.org/> >