



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

DIRCEU BERNEGOSSI DE SOUZA JÚNIOR

SISTEMA DE GERENCIAMENTO DE LOJA

Assis

2012

DIRCEU BERNEGOSSI DE SOUZA JÚNIOR

SISTEMA DE GERENCIAMENTO DE LOJA

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

Orientador: Domingos de Carvalho Villela Júnior

Área de Concentração: Análise e Desenvolvimento de Sistemas

Assis

2012

FICHA CATALOGRÁFICA

BERNEGOSI, Dirceu

Sistema de Gerenciamento de Loja / Dirceu Bernegossi de Souza Júnior. Fundação Educacional do Município de Assis – FEMA -- Assis, 2012.

58p.

Orientador: Domingos de Carvalho Villela Júnior

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

1. Sistema. 2. Conclusão. 3. Trabalho.

CDD: 001.61

Biblioteca da FEMA

SISTEMA DE GERENCIAMENTO DE LOJA

DIRCEU BERNEGOSSI DE SOUZA JÚNIOR

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

Orientador: Domingos de Carvalho Villela Júnior

Analisador: Luiz Carlos Begosso

Assis

2012

DEDICATÓRIA

Dedico este trabalho aos meus professores,
aos meus familiares e a todos os amigos
do curso e também fora dele.

AGRADECIMENTOS

Primeiramente a Deus, por me permitir viver esta vida maravilhosa que apesar de simples eu não poderia desejar coisa melhor.

Ao professor Domingos de Carvalho Villela Júnior, que sempre teve muita paciência e sempre demonstrou muita humildade e simplicidade, e também a todos os outros professores de anos anteriores e do atual que sempre com muita disposição demonstraram que com esforço é possível conseguir coisas grandes.

E por último e não menos importante a todos os amigos que não citarei os nomes porque são tantos e poderei correr o risco de cometer uma injustiça esquecendo-me de citar alguns deles.

RESUMO

O trabalho em questão tem o objetivo de analisar a forma como são executadas as tarefas e os problemas que são gerados no decorrer da execução destas pela PB Digital, visando um software que ao mesmo tempo em que seja de simples manuseio, também forneça todos os recursos necessários para uma experiência completa e satisfatória.

O Sistema de Gerenciamento de Loja terá um importante papel operacional, porque irá controlar todos os fluxos de entradas e saídas de recursos financeiros, assim como gerenciamento de estoque e até mesmo classificara as mercadorias e serviços prestados para que haja uma maior organização e facilite a produção de históricos consistentes evitando assim o retrabalho.

O sistema será desenvolvido com a IDE NetBeans e será usada a linguagem Java, que tem a vantagem de ser multiplataforma e a linguagem mais usada e divulgada atualmente, aliando-se a isso o projeto será voltado para aplicação web resultando em mobilidade e facilidade de uso através de uma interface robusta, mas ao mesmo tempo simples, que permite que mesmo um usuário que tenha pouco conhecimento em informática trabalhe de forma satisfatória sem nenhuma dificuldade devido à interface simplificada.

Para a armazenagem de informações será usado o banco de dados MySQL que permite uma implementação simples da aplicação, mas com desempenho consistente e que tem o seu maior foco em pequenas empresas que exigem um baixo fluxo de informações por se tratar de uma aplicação sem custos.

Palavras-chave: NetBeans; Java; multiplataforma; MySQL.

ABSTRACT

The work in question aims to examine how the tasks are performed and the problems that are generated during the execution of these by PB Digital, a software aimed at the same time it is simple to use, also provide all resources needed for a complete and satisfying experience.

The Store Management System will have an important operational role, because it will control all incoming and outgoing flows of financial resources, as well as inventory management and even had classified the goods and services so that there is greater organization and facilitates the production consistent history of avoiding rework.

The system will be developed with the NetBeans IDE and the language used is Java, which has the advantage of being multi-platform and language most used and disseminated currently allying themselves with this project will be focused on web application resulting in mobility and ease of use through a robust interface but at the same time simple, which allows even a user who has little knowledge in computing work satisfactorily without any difficulty due to the simplified interface.

For the storage of information will be used MySQL database that allows a simple implementation of the application, but with consistent performance and is aimed at small businesses that require a low flow of information because it is an application without costs.

Keywords: NetBeans; Java; multi-platform; MySQL.

LISTA DE ILUSTRAÇÕES

Figura 01. Estrutura Analítica do Projeto (WBS).....	27
Figura 02. UC – Sistema de Gerenciamento de Loja.....	30
Figura 03. UC – Manter Fornecedores.....	31
Figura 04. UC – Manter Grupos de Mercadorias.....	32
Figura 05. UC – Efetuar Compra de Mercadorias.....	33
Figura 06. UC – Efetuar Venda de Mercadorias.....	35
Figura 07. UC – Manter Tipos de Serviços.....	36
Figura 08. UC – Efetuar Prestação de Serviços.....	37
Figura 09. UC – Manter Contas a Receber.....	38
Figura 10. UC – Manter Contas a Pagar.....	40
Figura 11. UC – Manter Cliente.....	41
Figura 12. UC – Emitir Relatório de Fornecedores.....	42
Figura 13. UC – Emitir Relatório de Vendas.....	43
Figura 14. UC – Emitir Relatório de Serviços.....	44
Figura 15. UC – Emitir Relatório de Clientes.....	45
Figura 16. UC – Emitir Relatório de Estoque.....	46
Figura 17. UC – Emitir Relatório Financeiro.....	47
Figura 18 – Diagrama de Classes.....	48
Figura 19 – Diagrama de Sequência – Fornecedor.....	49
Figura 20 – Diagrama de Sequência – Serviços.....	50

Figura 21 – Diagrama de Sequência – Clientes.....	51
Figura 22 – Diagrama de Sequência – Contas a Pagar.....	52
Figura 23 – Diagrama de Sequência – Contas a Receber.....	53
Figura 24 – Diagrama de Sequência – Mercadoria.....	54
Figura 25 – Diagrama de Atividades – Compra.....	55
Figura 26 – Diagrama de Atividades – Venda.....	56
Figura 27 – Cronograma.....	57

LISTA DE TABELAS

Tabela 1. UC – Manter Fornecedores.....	31
Tabela 2. UC – Manter Grupos de Mercadorias.....	32
Tabela 3. UC – Efetuar Compra de Mercadorias.....	33
Tabela 4. UC – Efetuar Vendas de Mercadorias.....	35
Tabela 5. UC – Manter Tipos de Serviços.....	36
Tabela 6. UC – Efetuar Prestação de Serviços.....	37
Tabela 7. UC – Manter Contas a Receber.....	38
Tabela 8. UC – Manter Contas a Pagar.....	40
Tabela 9. UC – Manter Cliente.....	41
Tabela 10. UC – Emitir Relatório de Fornecedores.....	42
Tabela 11. UC – Emitir Relatório de Vendas.....	43
Tabela 12. UC – Emitir Relatório de Serviços.....	44
Tabela 13. UC – Emitir Relatório de Clientes.....	45
Tabela 14. UC – Emitir Relatório de Estoque.....	46
Tabela 15. UC – Emitir Relatório Financeiro.....	47

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 JUSTIFICATIVAS E MOTIVAÇÕES.....	15
1.2 OBJETIVOS.....	16
2 LEVANTAMENTO DE REQUISITOS.....	17
2.1. ANÁLISE DE REQUISITOS.....	17
3 TÉCNOLOGIAS UTILIZADAS.....	19
3.1 LINGUAGEM UML.....	19
3.2 JAVA.....	19
3.3 JSF.....	21
3.4 MYSQL.....	22
3.5 NETBEANS.....	22
3.6 HIBERNATE E JPA.....	23
3.7 WEB SERVER.....	25
3.8 RIA.....	25
4 PLANEJAMENTO DO PROJETO.....	27
4.1 ESTRUTURA ANALÍTICA DO PROJETO (WBS).....	27
4.2 SEQUÊNCIAMENTO DE ATIVIDADES.....	28
5 LISTA DE EVENTOS.....	29

6 CASOS DE USO.....	30
7 DIAGRAMA DE CLASSES.....	48
8 DIAGRAMA DE SEQUÊNCIA.....	49
9 DIAGRAMA DE ATIVIDADES.....	55
10 CRONOGRAMA.....	57
11 REFERÊNCIAS.....	58

1. INTRODUÇÃO

O sistema de Gerenciamento de Loja terá como foco principal o controle de prestação de serviços e venda de mercadorias, assim como a gestão do fluxo de caixa da loja PB Digital. Além dos benefícios citados o software também contribuirá com relatórios detalhados sobre os segmentos mais importantes da loja, mantendo o proprietário a par de todas as movimentações que ocorrerem em tempo real.

Além das funções citadas acima, o software terá um diferencial sobre os demais sistemas do setor presentes no mercado, ele será implementado inteiramente com ferramentas livres, e usará tecnologias atuais e que futuramente através de atualizações poderá obter novas funcionalidades acrescentadas através das necessidades que o cliente demonstrar sobre o uso do sistema.

Entre as tecnologias usadas estão a linguagem UML, NetBeans, MySQL, JSF, Web Server, Hibernate, JPA e RIA que serão mais detalhadas no decorrer do trabalho.

1.1 JUSTIFICATIVAS E MOTIVAÇÕES

O projeto foi criado através da necessidade de se obter maior confiabilidade nos controles de mercadorias, serviços, caixa entre outros setores de importância fundamental para a loja (PB Digital).

A PB Digital possui muitas dificuldades com relação à organização de estoque e também na realização de serviços, isso devido à falta de um software gerenciador de entradas e saídas, agendamento e produção de relatórios que detalhem com maior precisão os setores que requerem mais atenção.

E juntando-se a isso também existe a necessidade de maior mobilidade para que a proprietária realize essas operações em qualquer hora ou lugar sem se ater a obter tais dados somente enquanto cumpre o horário de trabalho comercial, portanto o que justifica plenamente o sistema no formato Web.

1.2 OBJETIVOS

Este trabalho de conclusão de curso tem por objetivo o desenvolvimento do projeto e implementação do sistema PB Digital. O referido sistema executa o controle de mercadorias, fluxo de caixa, prestação de serviços e uso de materiais, assim como visa manter um ambiente de trabalho organizado e ágil.

2. LEVANTAMENTO DE REQUISITOS

Os requisitos são parte fundamental em um desenvolvimento de software, se feita de maneira inadequada pode causar sérios problemas como atrasos, falta de funcionalidades necessárias, funcionalidades que o cliente não necessita, insatisfação do cliente contratante entre outros tantos motivos que podem levar um projeto ao insucesso.

Para que o projeto tivesse a abordagem correta do que o cliente precisaria, foram vistas uma a uma as dificuldades em se trabalhar sem um sistema de gerenciamento e a seguir serão descritos os pontos chaves resultantes dessa análise feita junto ao cliente.

2.1 ANÁLISE DE REQUISITOS

Requisito: Controlar as Contas a Pagar e Receber

Descrição: As informações sobre as contas tanto de despesas quanto a receber são feitas através de papéis, o que pode causar transtorno caso haja rasura ou perda desses papéis e conseqüentemente falta de controle sobre o fluxo de caixa.

Proposta de Solução: Desenvolver um sistema que gerencie este fluxo e mantenha o proprietário sempre informado da situação e data de vencimento das contas através de relatórios detalhados.

Requisito: Melhorar o Controle de Estoque

Descrição: Por não contar com um sistema de controle de estoque, ocorreram muitas vezes situações em que não se tinha controle se uma mercadoria tinha sido repostada no estoque causando perda de tempo, entendendo-se que o estoque teria que ser recontado peça a peça, e também sobre qual valor os lotes do mesmo produto, mas que foram comprados em diferentes épocas possuíam.

Proposta de Solução: Desenvolver um sistema que cadastre um novo produto assim que seja feito um orçamento com os fornecedores e que seja feita toda a coleta de

dados deste produto assim que ele é entregue no estoque, e também um alerta quando a peça atinge um estoque mínimo determinado pela loja.

Requisito: Melhorar o controle do material gasto com prestação de serviços

Descrição: O maior problema abordado pelo cliente é controlar o uso de material gasto com prestações de serviços, exemplo: Papel para Xerox, Tinta para recargas de Cartucho além do material gasto com o próprio consumo da loja, o que de certa forma é difícil de gerenciar porque são mercadorias que além dos serviços prestados devem ter seu valor agregado no preço final e que devido ao controle mal feito acaba gerando prejuízos para a loja por não ter o valor exato de material gasto com o serviço, impedindo assim de se calcular um valor final com uma margem de lucro segura em relação ao que foi gasto para tal.

Proposta de Solução: Desenvolver um sistema que cadastre os diferentes serviços prestados e o material usado em cada um, e com a geração de relatórios obter o valor e tempo gasto com cada prestação de serviço e suas particularidades

3. TECNOLOGIAS UTILIZADAS

3.1 LINGUAGEM UML

A UML - *Unified Modeling Language* – é um modelo de linguagem para modelagem de dados orientados a objetos. Através desta técnica é possível representar todas as etapas e relacionamentos de um projeto de desenvolvimento de software tornando a interação entre o usuário e o sistema mais dinâmica e consistente.

Esta linguagem surgiu da união entre outras técnicas já existentes, aproveitando as qualidades que cada uma possuía criando assim um padrão que hoje é referência no mercado de criação de software.

A UML foi inserida no manual PMBOK que regulamenta as boas práticas de desenvolvimento de projetos, vale ressaltar que o PMBOK não se prende apenas a programação, e pode ser aplicada a qualquer tipo de projeto e praticamente em todos os ramos de atividade.

É importante destacar que o manual PMBOK foi reconhecido no ano de 1999 como um padrão de gerenciamento de projetos pelo ANSI – *American National Standards Institute*, e foi criado pelo *Standards Committee* (Comitê de Padronização), (informação pessoal)¹.

3.2 JAVA

Java é uma linguagem orientada a objetos que foi criada pela Sun, que mais tarde foi comprada pela Oracle que a mantém até hoje, é uma das linguagens mais usadas no mundo, estando presente em diversos dispositivos.

A linguagem Java está presente desde Celulares, Tablets, PCs e até mesmo em outros sistemas que se utilizam do código Java com algumas modificações em suas bibliotecas como é o caso do hoje líder de mercado em dispositivos portáteis Android da Google.

¹ Informação obtida através de consulta realizada diretamente no endereço: <<http://www.curso-pmi.com.br/artigos/o-que-e-pmbok/>> acesso em 24 de junho de 2012.

Portanto a linguagem Java pode ser aplicada a qualquer dispositivo que tenha livre acesso para tal, o segredo dessa independência de plataformas é que diferentemente de outras linguagens que utilizam um *Framework* (que nada mais são que um conjunto de instruções) que é responsável por organizar o código produzido e executa-lo sob um Kernel (Núcleo do Sistema Operacional), como é o caso do C#, linguagem proprietária da Microsoft que roda exclusivamente no Windows, o Java não possui *Framework*, mas sim JVM (*Java Virtual Machine*).

Então contanto que sua plataforma tenha uma JVM rodando, o código feito em Java executará sem problemas porque o próprio JVM se encarregará de interpretar e gerar um Bytecode (Código Binário), de acordo com que o sistema operacional necessita para executar as instruções, simplificando, o programador não precisa se preocupar com o sistema operacional porque a própria linguagem Java se encarrega de fazer essa “ponte”.

É preciso salientar que durante o início do Java a sua execução era considerada lenta comparada com outras linguagens como o C++, e por esse motivo quando a aplicação necessitava de velocidade de execução como jogos não era conveniente ser feito em linguagem Java, pela lógica fica muito claro o porquê do Java ter tido no início uma lentidão em relação a outras linguagens.

Enquanto as linguagens nativas do sistema operacional usado executavam diretamente no Kernel do sistema, o código Java necessitava ser interpretado e depois gerado no formato do sistema operacional usado, o que demandava mais tempo em troca de maior compatibilidade, mas com o tempo o Java foi sendo aperfeiçoado e após a criação de novos recursos o Java ficou em igualdade em relação à velocidade com outras linguagens.

Um dos mais importantes recursos citados consiste em desconsiderar código sem função na execução do sistema, evitando assim que a JVM levasse mais tempo para interpretar e gerar o Bytecode com códigos que não eram vitais para a execução do software.

É importante citar que o Java hoje roda sobre três tipos principais de plataformas:

Plataformas Móveis Java ME (*Micro Edition*): Voltada para dispositivos com pouca memória e restrição de bateria como Celulares, Tablets, Rádios Automotivos.

Plataforma Web Java EE (*Enterprise Edition*): Voltada para sistemas corporativos que usam rede e são executadas através do Browser.

Plataforma Desktop Java SE (*Standard Edition*): Sistemas que são criados basicamente para PCs.

3.3 JSF (*Java Server Faces*)

JSF ou Java Server Faces é uma tecnologia utilizada para desenvolvimento web, que utiliza um modelo de interface gráfica baseada em eventos. Esta tecnologia foi definida pelo JCP (*Java Community Process*), o que a torna um padrão de desenvolvimento de interfaces visuais.

O JSF é baseado no padrão MVC (*Model-View-Controller*), o que facilita muito o desenvolvimento porque a divisão entre a regra de negócios e a interface é muito clara, tornando a programação muito mais dinâmica e produtiva.

Basicamente o MVC é a divisão do sistema em três grandes camadas que resumidamente tem as seguintes funções:

Model ou Modelo – é onde fica o modelo de dados com todos os objetos e seus serviços.

View ou Interface – é a forma gráfica que a aplicação é vista pelo usuário, através disso é manipulada pelo usuário conforme a necessidade deste.

Controller ou Controlador – como o próprio nome diz tem a função de fazer a “ponte” entre o modelo de dados e a interface, no momento que o usuário requisita um serviço através da interface, o controlador recebe a informação e se comunica com o modelo de dados para que o serviço requisitado seja concluído ou em alguns casos relatem ao usuário mensagens específicas.

É importante citar que o MVC pode ser usado na maioria das linguagens orientadas a objetos, mas esse recurso se adequa muito bem ao JSF devido ao tratamento que pode ser feito dentro da própria página (Interface), evitando assim aumentar o acesso a outras camadas de forma a simplificar a programação.

3.4 MySQL

O banco de dados MySQL pertence a Oracle e apesar de também trabalhar em conjunto com aplicações Desktop é muito usado para aplicações Web, é um banco de dados *Free* (sem custos), tem muita estabilidade, segurança e principalmente possui uma interface simples e objetiva.

Por se tratar de um banco de dados sem custos é bastante recomendado para trabalhos acadêmicos, fornecendo uma excelente ferramenta sem nenhum ônus ao estudante.

O MySQL tem uma integração muito boa com a linguagem Java por se tratar de produtos da mesma empresa (Oracle), e para manipula-lo é mais simples que alguns outros bancos, por se tratar de linguagem SQL (*Structured Query Language*) padrão Oracle o que significa dizer que é uma linguagem muito difundida e padronizada.

3.5 NETBEANS

O NetBeans é uma IDE (*Integrated Development Environment*) ou Ambiente de Desenvolvimento Integrado, é desenvolvido pela Sun Microsystems que pertence a Oracle e é construído usando a linguagem Java.

É uma IDE de código aberto e que suporta várias linguagens sendo a mais usada à linguagem Java, curiosamente o NetBeans não foi criado pela criadora do Java (Sun Microsystems), a IDE foi criada por estudantes tchecos na universidade de Charles em 1996, com o objetivo de rivalizar com o Delphi que era a IDE mais popular naquele momento, só que a intenção dos estudantes era criar uma IDE própria para Java que não era popular como é hoje.

O NetBeans foi criado com o nome Xelfi, em alusão ao Delphi e em 1999 o projeto evoluiu para uma IDE proprietária e passou a se chamar NetBeans Developer X2, que mais tarde viria a ser comprada pela Sun após o fracasso de sua IDE própria chamada Java Workshop, (informação pessoal)².

² Informação obtida através de consulta realizada diretamente no endereço: <http://www.oficinadanet.com.br/artigo/1061/o_que_e_o_netbeans> acesso em 24 de junho 2012.

Como vem acontecendo com muita frequência no mundo da tecnologia, o NetBeans só deslanchou em junho de 2000, após a Sun Microsystems transformá-lo em uma IDE *OpenSource*, em tradução livre software de código aberto, que permitiu a todos os desenvolvedores colaborarem em seu amadurecimento e desenvolvimento no mercado.

O NetBeans pode ser instalado em todos os sistemas operacionais mais populares por se tratar de um software programado em Java, tem suporte a todas as plataformas Java (EE, ME, SE) e tem funcionalidades muito intuitivas e automatizadas para a produção de código padronizado e de alta produtividade, o NetBeans está atualmente na versão 7.2 e tem como principal rival como IDE Java o Eclipse que também é bem difundido no mercado de programação Java.

3.6 HIBERNATE E JPA

Hibernate é um *framework* de mapeamento objeto/relacional, antigamente diversos programadores usavam os *entity beans* do EJB 2.1 como tecnologia para implementar a camada de persistência. Este modelo foi adotado largamente pela indústria de software.

Especialistas da comunidade acharam que o modelo de desenvolvimento com EJB (principalmente os *entity beans*) deveria ser melhorado, então a Sun Microsystems (criadora do Java) iniciou uma nova especificação para simplificar o EJB, que foi nomeada de *Enterprise JavaBeans 3.0*, conhecida também por JSR-220.

Os desenvolvedores do Hibernate se tornaram parte da equipe de especialistas desta especificação e ajudaram nesta evolução, além de várias outras empresas e pessoas. Como o Hibernate já era um framework de persistência popular em todo o mundo por sua estabilidade e facilidade, vários conceitos foram aproveitados na especificação do EJB 3.0.

As especificações do EJB 3.0 possuem várias partes: uma que define o modelo de programação EJB para *session beans*, *message-driven beans* e etc., outra parte, chamada de *Java Persistence API (JPA)*, trata apenas de persistência, como

entidades, mapeamento objeto/relacional e linguagem para consulta de dados (query).

Existem produtos que implementam um container completo EJB 3.0 e outros que implementam somente a parte de persistência.

As implementações de JPA devem ser plugáveis, o que significa que você poderá trocar um produto por outro sem afetar em nada seu sistema. Os produtos JPA devem também funcionar fora de um ambiente EJB 3.0 completo, ou seja, pode ser usado em aplicativos simples desenvolvidos em Java.

Conseqüentemente existem várias opções de produtos para empresas e desenvolvedores. A maioria dos produtos oferece funcionalidades que não fazem parte da especificação, como uma melhoria de desempenho ou uma API para consultas mais simples que a especificada. Você pode usar essas funcionalidades, porém deve-se lembrar que isso dificultará em uma futura migração de produto.

O Hibernate implementa a JPA, e aproveitando a flexibilidade que a especificação proporciona, você pode combinar diversos módulos do Hibernate:

- *Hibernate Core*: é a base para o funcionamento da persistência, com APIs nativas e metadados de mapeamentos gravados em arquivos XML. Possui uma linguagem de consultas chamada HQL (parecido com SQL), um conjunto de interfaces para consultas usando critérios (Criteria API) e queries (consultas) baseadas em objetos exemplos.

- *Hibernate Annotations*: uma nova forma de definir metadados se tornou disponível graças ao Java 5.0. O uso de anotações dentro do código Java substituiu arquivos XML e tornou o desenvolvimento ainda mais fácil. O uso de *Hibernate Annotations* com *Hibernate Core* possibilita que os metadados sejam descritos dentro do código-fonte das classes de domínio em forma de anotações. A especificação JPA possui um conjunto de anotações para que o mapeamento objeto/relacional seja feito, e o *Hibernate Annotations* estende essas anotações para adicionar funcionalidades extras.

- *Hibernate EntityManager*: este módulo encapsula o *Hibernate Core* para fornecer compatibilidade com a JPA.

3.7 WEB SERVER

Web Server ou Servidor Web é um programa de computador responsável por aceitar pedidos HTTP de clientes, geralmente os navegadores, e servi-los com respostas HTTP, incluindo opcionalmente dados, que geralmente são páginas web, tais como documentos HTML com objetos embutidos (imagens, etc.).

Os pedidos HTTP que se referem habitualmente a páginas HTML são normalmente feitos através de browsers. O processo se inicia com a conexão entre o computador onde está instalado o servidor web e o computador do cliente; como na web não é possível prever a que hora se dará essa conexão, os servidores web precisam estar disponíveis dia e noite. A partir daí é processado o pedido do cliente, e conforme as restrições de segurança e a existência da informação solicitada, o servidor devolve os dados.

Atualmente há cada vez mais programas que fazem pedidos HTTP (leitores de RSS e outros) e quase desde o início da web que os ficheiros servidos pelo *Web Server* vão para além dos ficheiros HTML, incluindo imagens, ficheiros de som, pdfs, etc.

Genericamente tudo o que se enquadre no conceito de ficheiro pode ser enviado como resultado de um pedido HTTP, finalmente os servidores web também podem executar programas e scripts, interagindo mais com o usuário, (informação pessoal)³.

3.8 RIA

Aplicações web ricas, também chamadas de Aplicações de Internet Ricas (*Rich Internet Application* - RIA) são uma linha de aplicações web mais sofisticadas que se comportam de maneira similar a programas de software para Desktop.

Comparadas com as páginas "normais", elas são ricas em interação, ricas em conteúdo e ricas em funcionalidade, ou seja, possuem uma linhagem própria. Uma das principais vantagens das aplicações web ricas é uma experiência do usuário melhorada. Essas aplicações frequentemente usam um modelo de página

³ Informação obtida através de consulta realizada diretamente no endereço: <http://www.oficinadanet.com.br/artigo/servidores/o_que_e_um_servidor_web>

única no qual as informações atualizadas são rapidamente trazidas à página.

Utilizando este método, a reorientação em relação às opções de navegação é mínima ou nem mesmo está presente. De um modo geral, podemos dizer que as aplicações web ricas são mais responsivas e permitem maneiras mais rápidas de encontrarmos as informações, enquanto diminuem potencialmente o efeito de deixar o usuário desorientado.

4. PLANEJAMENTO DO PROJETO

4.1 ESTRUTURA ANALÍTICA DO PROJETO (WBS)

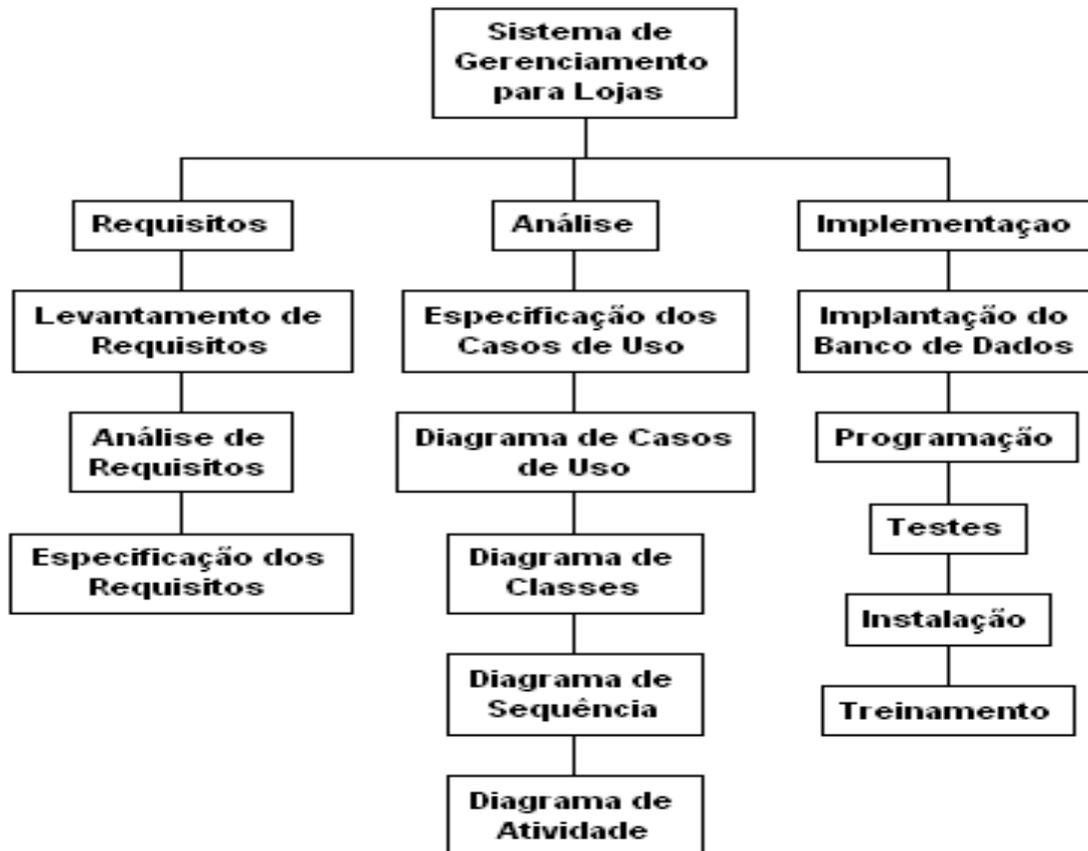
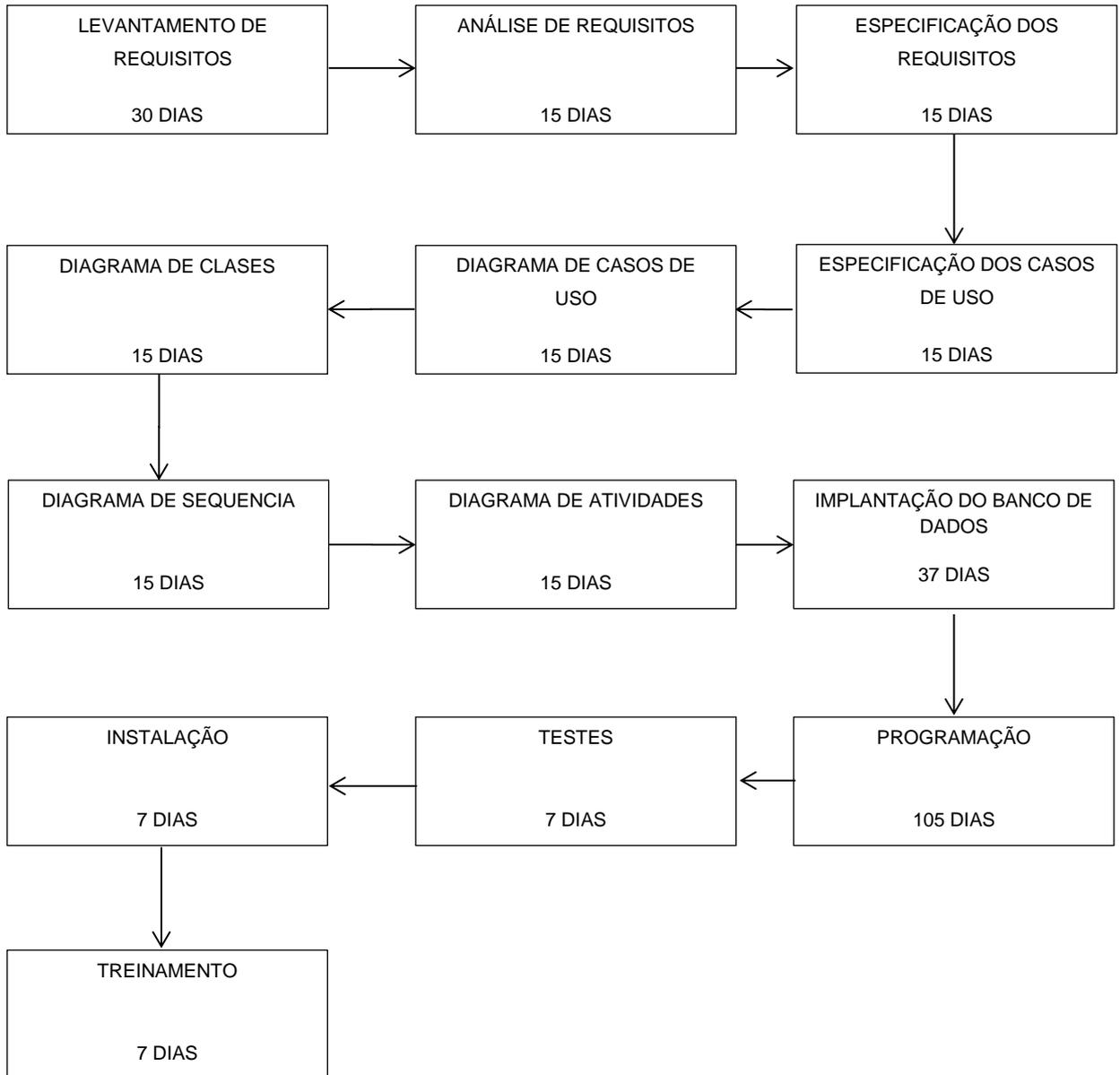


Figura 01. Estrutura Analítica do Projeto (WBS)

4.2 SEQUENCIAMENTO DE ATIVIDADES



5. LISTA DE EVENTOS

1. Manter Fornecedores
2. Manter Grupos de Mercadorias
3. Efetuar Compra de Mercadorias
4. Manter Tipos de Serviços
5. Manter Cliente
6. Efetuar Prestação de Serviços
7. Efetuar Venda de Mercadorias
8. Manter Fluxo de Caixa de Entrada
9. Manter Fluxo de Caixa de Saída
10. Emitir Relatório de Vendas
11. Emitir Relatório de Serviços
12. Emitir Relatório Financeiro
13. Emitir Relatório de Cliente
14. Emitir Relatório de Fornecedores
15. Emitir Relatório de Estoque

6. Casos de Uso

Para que se tenha uma visão do sistema e suas funcionalidades, será mostrada a figura que representa o caso de uso geral do sistema e logo a seguir serão descritos os casos de uso individualmente seguidos de suas especificações.

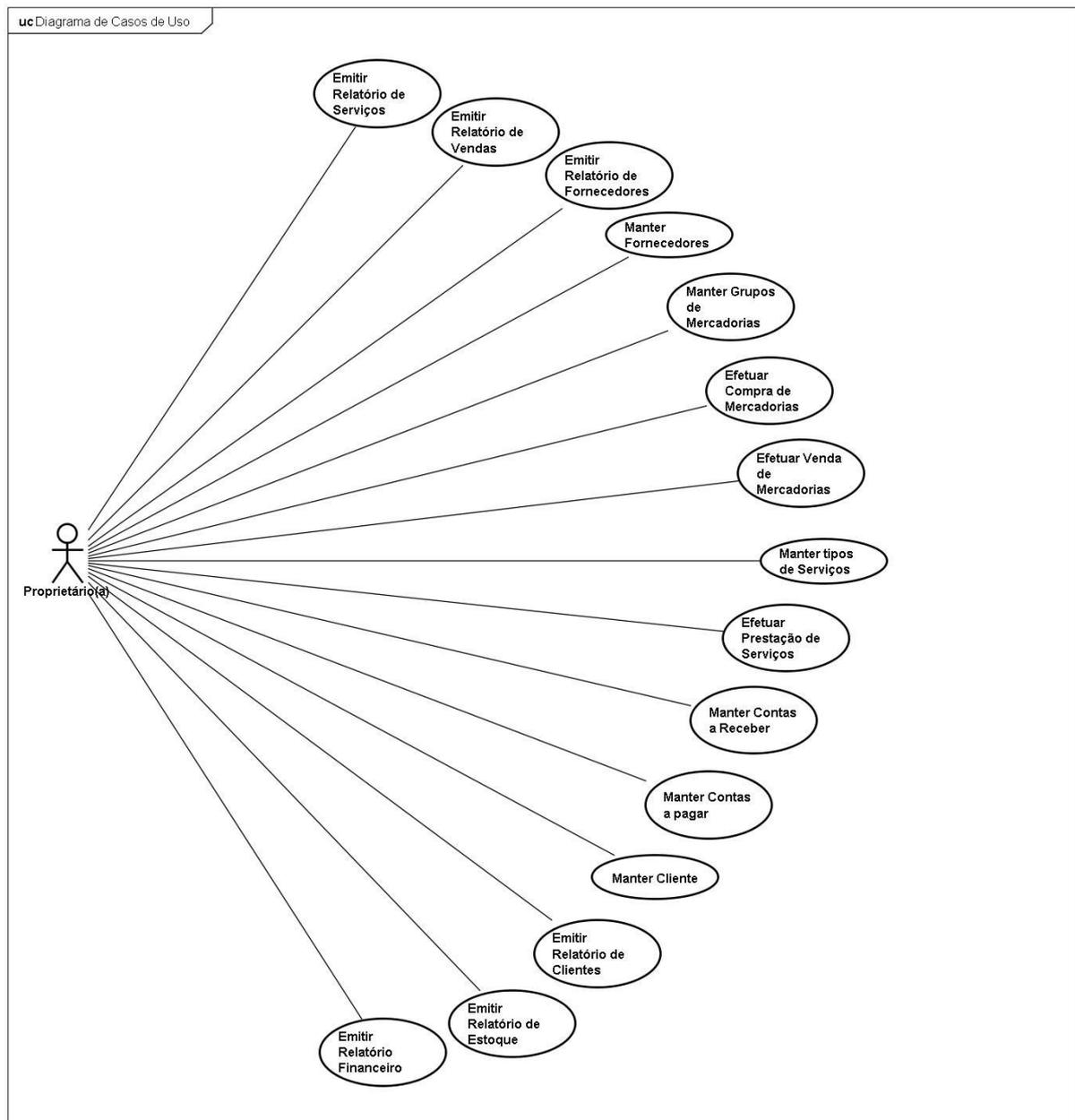


Figura 02. UC Sistema de Gerenciamento de Lojas

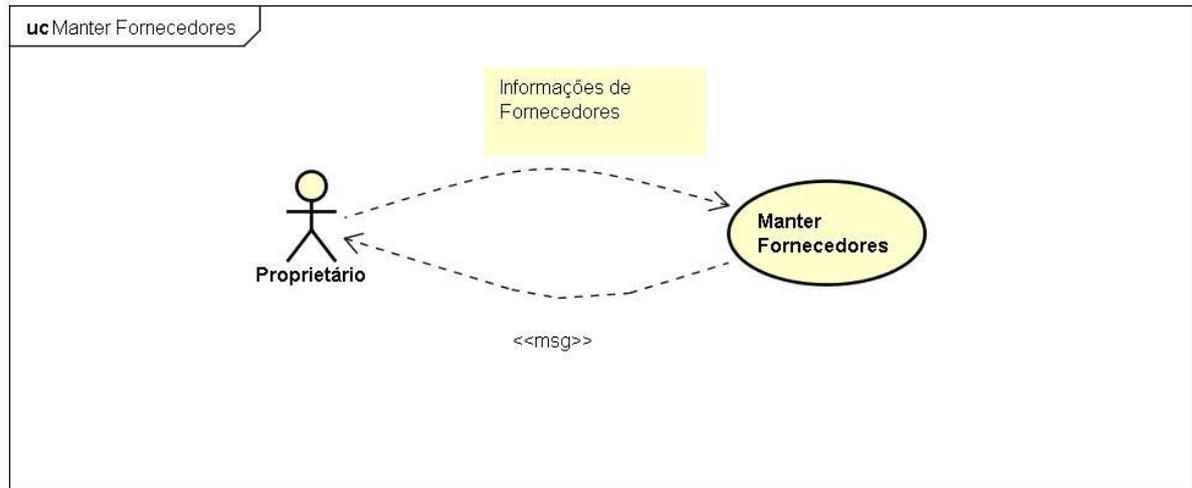


Figura 03. UC - Manter Fornecedores

Nome do Caso de Uso	Manter Fornecedores
Pré-Condição	O proprietário deverá estar logado no sistema.
Ator(es)	Proprietário
Cenário Principal	<p>1 - O sistema solicita os dados referentes ao Fornecedor para efetuar o cadastro.</p> <p>2 - O proprietário preenche todos os campos que constam na tela.</p> <p>3 - O proprietário clica em cadastrar.</p> <p>4 - O sistema envia uma mensagem “Fornecedor inserido com sucesso”.</p> <p>5 - O sistema insere o novo cadastro na lista de Fornecedores.</p>
Cenário Alternativo	O proprietário pode cancelar o cadastro e encerrar a operação.
Casos de Testes	<p>O sistema verifica se todos os dados foram inseridos corretamente.</p> <p>O sistema verifica se existem fornecedores com os mesmos dados cadastrados.</p>

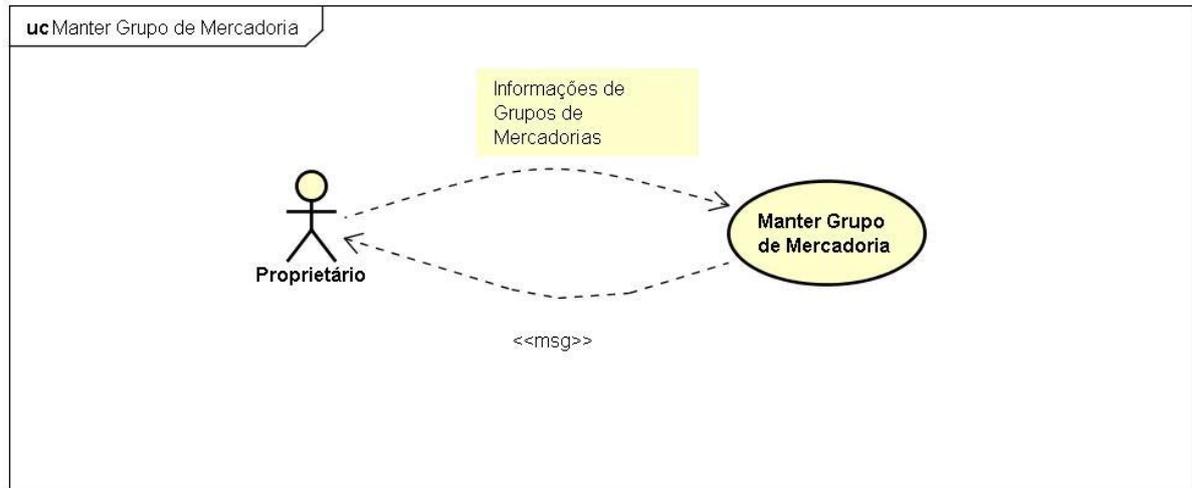


Figura 04. UC - Manter Grupo de Mercadorias

Nome do Caso de Uso	Manter Grupo de Mercadorias.
Pré-Condicion	O proprietário deve estar logado no sistema.
Ator(es)	Proprietário.
Cenário Principal	<p>1 - O sistema exibe a tela para preenchimento de cadastro de novo grupo de mercadorias.</p> <p>2 - O proprietário preenche todos os dados e clica em finalizar cadastro.</p> <p>3 - O sistema exibe a mensagem "Grupo de mercadorias cadastrado com sucesso".</p>
Cenário Alternativo	O proprietário resolve cancelar a operação e sair da tela de cadastro.
Casos de Testes	<p>O sistema verifica se o novo grupo cadastrado já existe.</p> <p>O sistema exibe a mensagem "Dados duplicados, não foi possível realizar a operação".</p>

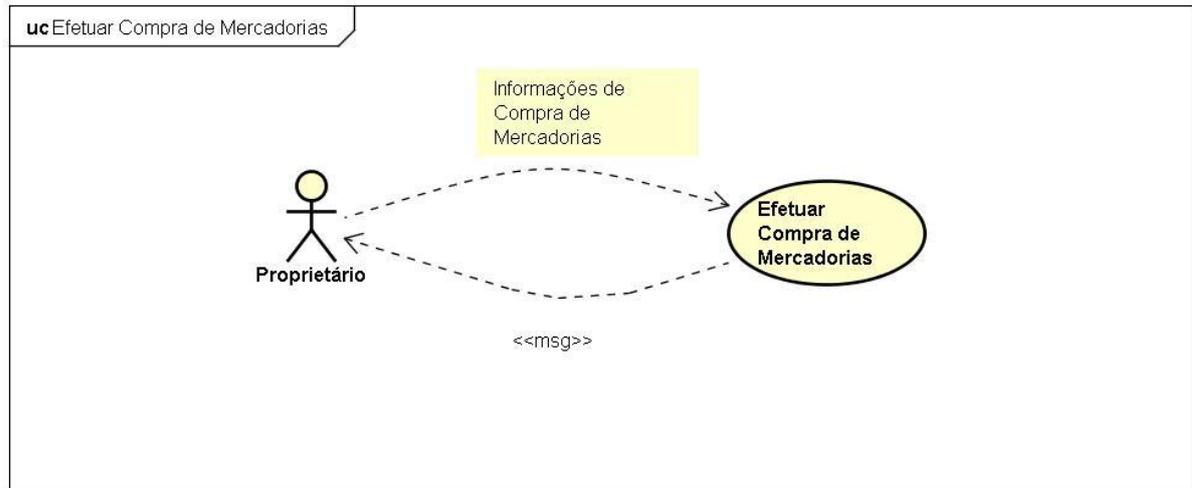


Figura 05. UC – Efetuar Compra de Mercadorias

Nome do Caso de Uso	Efetuar Compra de Mercadorias.
Pré-Condição	O proprietário deve estar logado no sistema.
Ator(es)	Proprietário.
Cenário Principal	<p>1 - O sistema exibe a tela em que o proprietário deve escolher o grupo de mercadorias que ele deseja adquirir.</p> <p>2 - O proprietário escolhe o grupo desejado e clica em prosseguir.</p> <p>3 - O sistema exibe uma tela com os fornecedores cadastrados que vendem este grupo de mercadorias.</p> <p>4 - O proprietário escolhe o Fornecedor e clica em prosseguir.</p> <p>5 - O sistema exibe na tela uma lista de produtos ofertados pelo fornecedor com o valor das últimas compras realizadas.</p> <p>6 - O proprietário clica no produto desejado e escolhe a quantidade a ser adquirida.</p>

	<p>7 - O sistema exibe na tela o valor total da compra.</p> <p>8 - O proprietário clica em finalizar compra.</p> <p>9 - O sistema exibe a mensagem “Compra de produtos realizada com sucesso”.</p>
Cenário Alternativo	O usuário desiste da compra e cancela a operação.
Casos de Testes	O sistema verifica qual é o fornecedor com o menor valor dos produtos adquiridos na última compra.

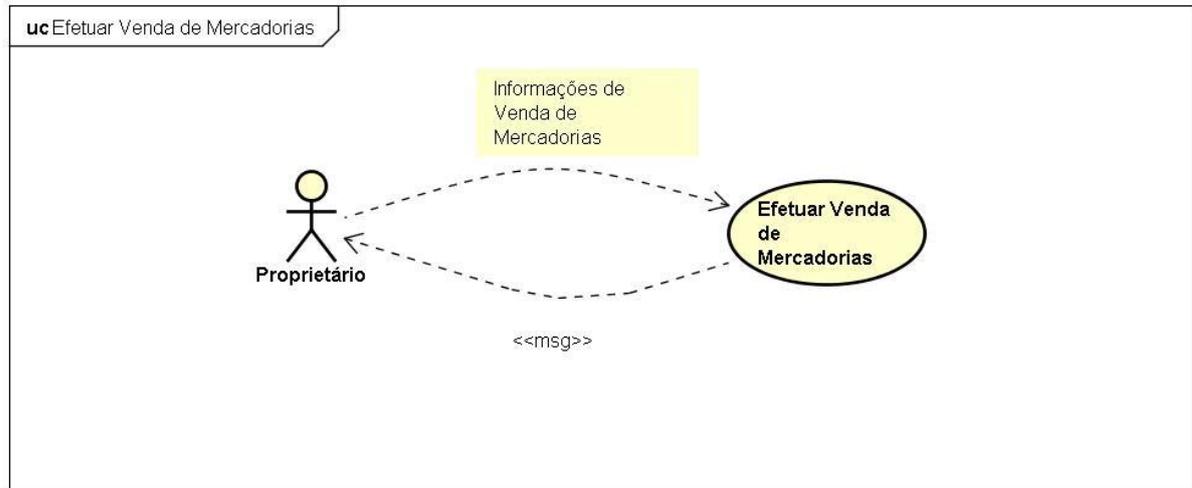


Figura 06. UC – Efetuar Venda de Mercadorias

Nome do Caso de Uso	Efetuar Venda de Mercadorias.
Pré-Condição	Proprietário estar logado no sistema. O sistema exibir como disponível no estoque.
Ator(es)	Proprietário
Cenário Principal	1 - O sistema exibe uma tela para inserir qual grupo de mercadorias o produto pretendido pertence. 2 - O proprietário clica no grupo desejado. 3 - O sistema exibe a quantidade de peças existentes de cada produto do grupo escolhido. 4 - O proprietário clica no produto desejado e digita a quantidade. 5 - O sistema exibe a tela "Venda concluída com sucesso".
Cenário Alternativo	O valor não é o esperado pelo cliente, então o proprietário cancela a venda.
Casos de Testes	O sistema verifica qual é o produto com o valor mais em conta dentro do grupo.

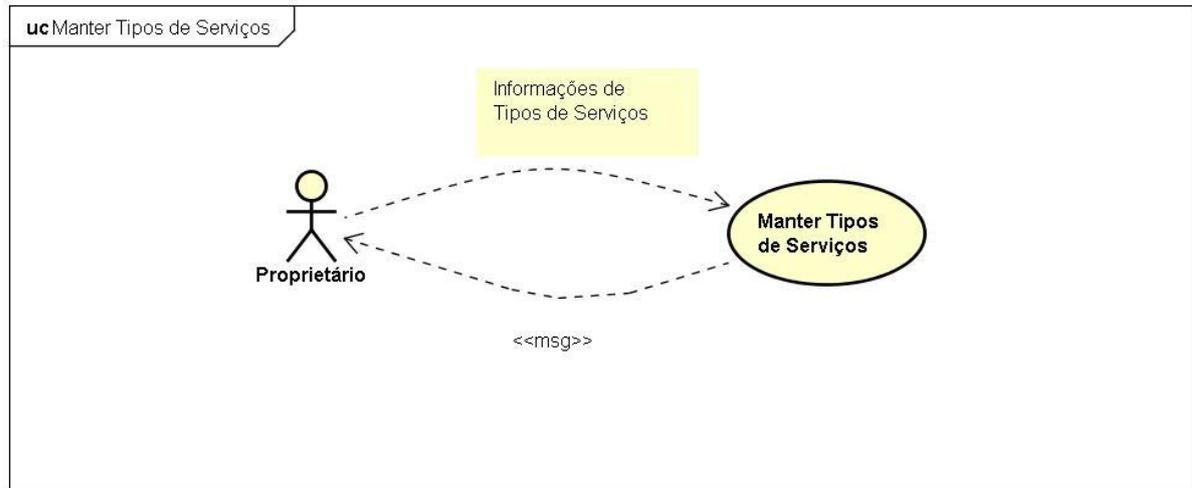


Figura 07. UC – Manter Tipos de Serviços

Nome do Caso de Uso	Manter Tipos de Serviços.
Pré-Condição	Proprietário deve estar logado no sistema.
Ator(es)	Proprietário.
Cenário Principal	<p>1 - O sistema exibe a tela para que o tipo de serviço seja cadastrado.</p> <p>2 - O proprietário preenche todos os dados e clica em próximo.</p> <p>3 - O sistema exibe na tela quais produtos são necessários para se realizar este tipo de serviço.</p> <p>4 - O proprietário clica nos produtos necessários e finaliza a operação.</p> <p>5 - O sistema exibe a mensagem "Tipo de serviço inserido com sucesso".</p>
Cenário Alternativo	O proprietário resolve cancelar a operação durante o cadastro.
Casos de Testes	O sistema verifica se o produto escolhido é compatível com o tipo de serviço.

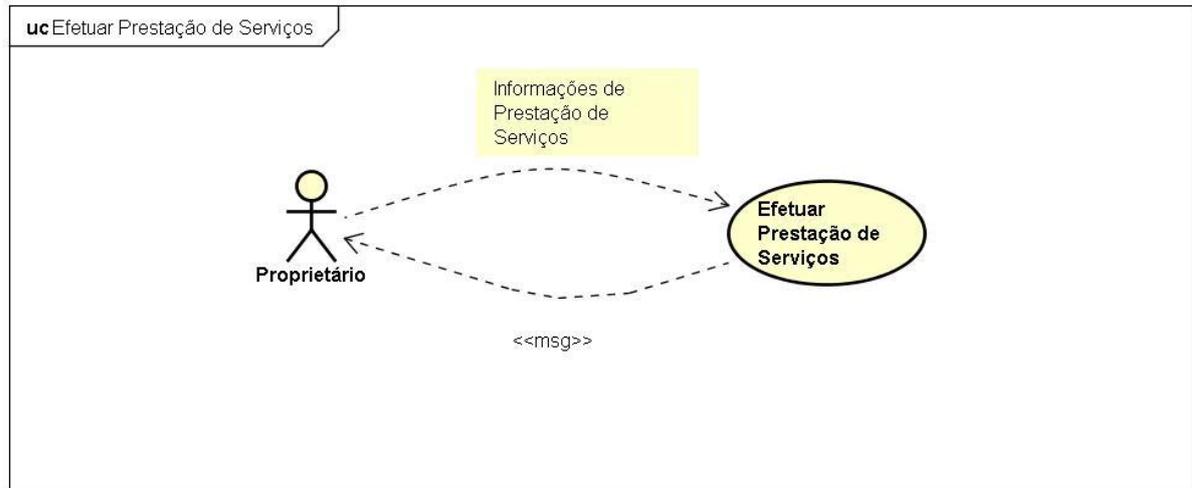


Figura 08. UC – Efetuar Prestação de Serviços

Nome do Caso de Uso	Efetuar Prestação de Serviços
Pré-Condição	Proprietário deve estar logado no sistema. O material necessário para realizar o serviço deve estar disponível em estoque.
Ator(es)	Proprietário.
Cenário Principal	1 - O sistema exibe na tela a escolha de qual tipo de serviço será prestado. 2 - O proprietário escolhe o serviço e clica em próximo. 3 - O sistema exibe na tela a escolha de quanto material será necessário para realizar o serviço. 4 - O proprietário escolhe a quantidade a ser usada e clica em finalizar serviço. 5 - O sistema exibe a mensagem na tela “Serviço realizado com sucesso”.
Cenário Alternativo	O proprietário decide cancelar a operação.
Casos de Testes	O sistema verifica se existe produto em estoque para realizar o serviço.

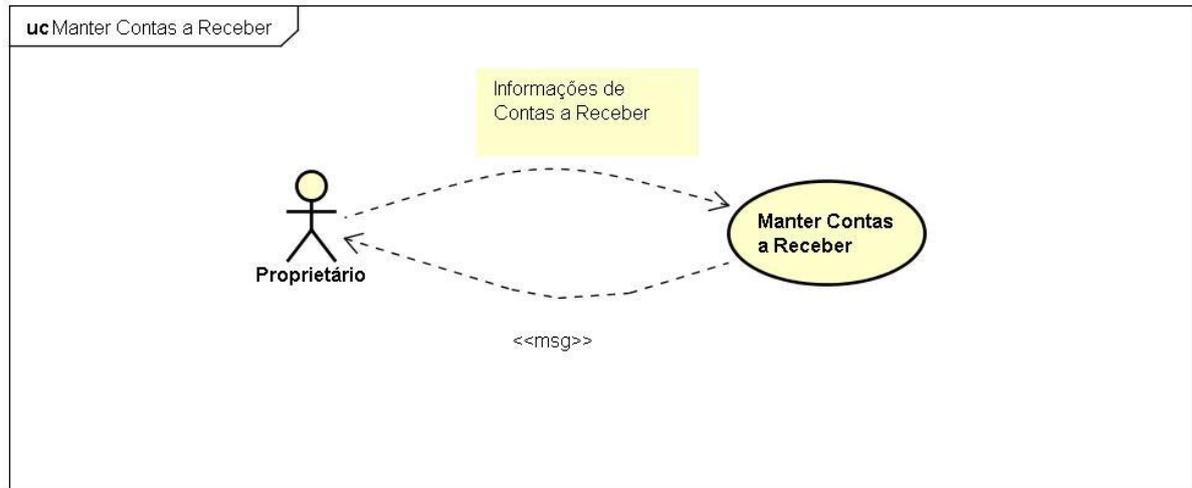


Figura 09. UC – Manter Contas a receber

Nome do Caso de Uso	Manter Contas a Receber.
Pré-Condição	O proprietário deve estar logado no sistema. Alguns produtos devem ter sido vendidos ou serviços prestados.
Ator(es)	Proprietário.
Cenário Principal	<p>1- O sistema exibe na tela a escolha de qual tipo de conta o proprietário está recebendo, serviços prestados ou mercadorias vendidas.</p> <p>2 - O proprietário escolhe a opção desejada e clica em próximo.</p> <p>3 - O sistema exibe uma tela para que o proprietário insira o valor recebido.</p> <p>4 - O proprietário insere o valor recebido e clica em finalizar recebimento.</p> <p>5 - O sistema exibe na tela “Recebimento finalizado com sucesso”.</p>
Cenário Alternativo	O proprietário insere um valor referente a uma parte da conta a receber e clica em finalizar

	<p>recebimento.</p> <p>O sistema exibe na tela a data da próxima fatura e o valor restante com a mensagem “Recebimento finalizado com sucesso”.</p>
Casos de Testes	<p>O sistema calcula se o valor recebido é suficiente para saldar a dívida, se não for o sistema subtrai da conta o valor recebido e mostra o valor restante e a data de vencimento.</p>

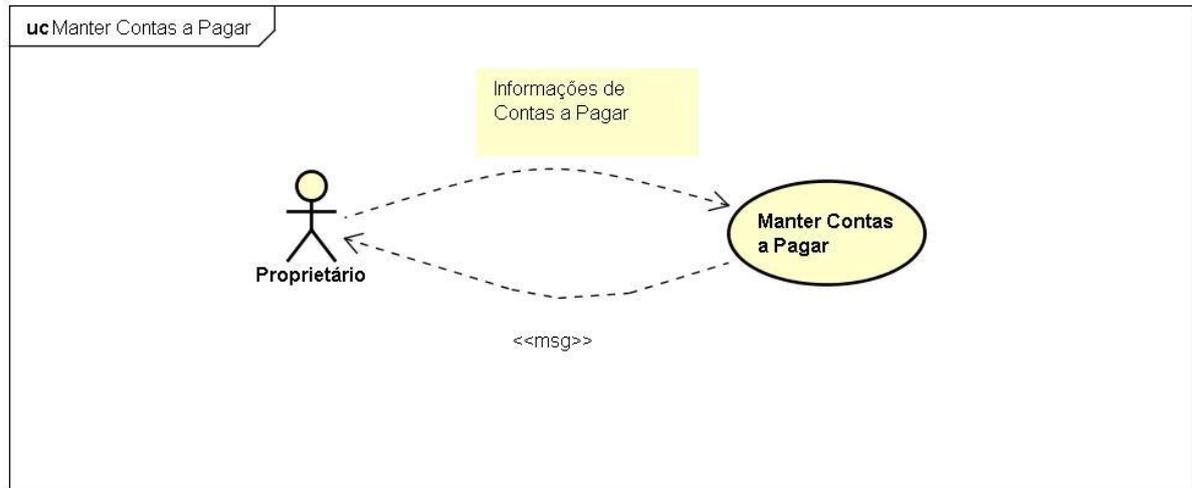


Figura 10. UC – Manter Contas a pagar

Nome do Caso de Uso	Manter Contas a Pagar.
Pré-Condição	O proprietário deve estar logado no sistema.
Ator(es)	Proprietário.
Cenário Principal	<p>1 - O sistema exibe na tela um alerta com o valor, a data de vencimento junto com a descrição da conta a pagar.</p> <p>2 - O proprietário Clica em aviso recebido.</p> <p>3 - O sistema desativa o alerta e volta a tela principal.</p>
Cenário Alternativo	O usuário não visualiza na hora o alerta, então o sistema ativa o alerta entre certos períodos de tempo até que o usuário clique em aviso recebido.
Casos de Testes	O sistema calcula o vencimento das contas com uma margem segura de tempo para que não haja pagamentos com atrasos o que acarretaria em juros e prejuízos.

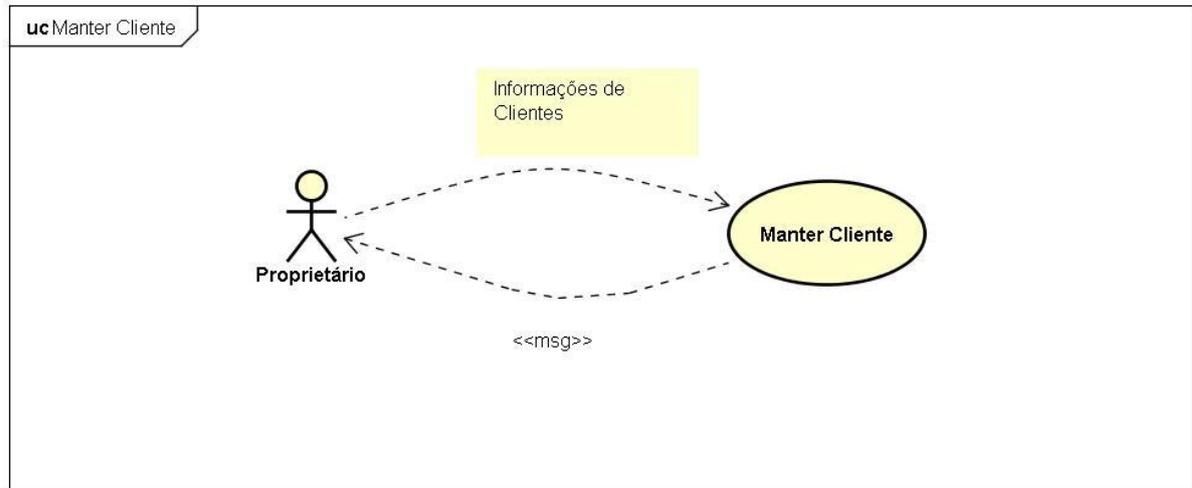


Figura 11. UC – Manter Cliente

Nome do Caso de Uso	Manter Cliente
Pré-Condicion	O proprietário deve estar logado no sistema.
Ator(es)	Proprietário.
Cenário Principal	<p>1 – O sistema exibe a tela de cadastro de clientes.</p> <p>2 – O proprietário preenche todos os campos e clica em finalizar cadastro.</p> <p>3 – O sistema exibe a mensagem “Cadastro concluído com sucesso”.</p>
Cenário Alternativo	O proprietário decide cancelar a operação.
Casos de Testes	O sistema verifica se há duplicidade de dados.

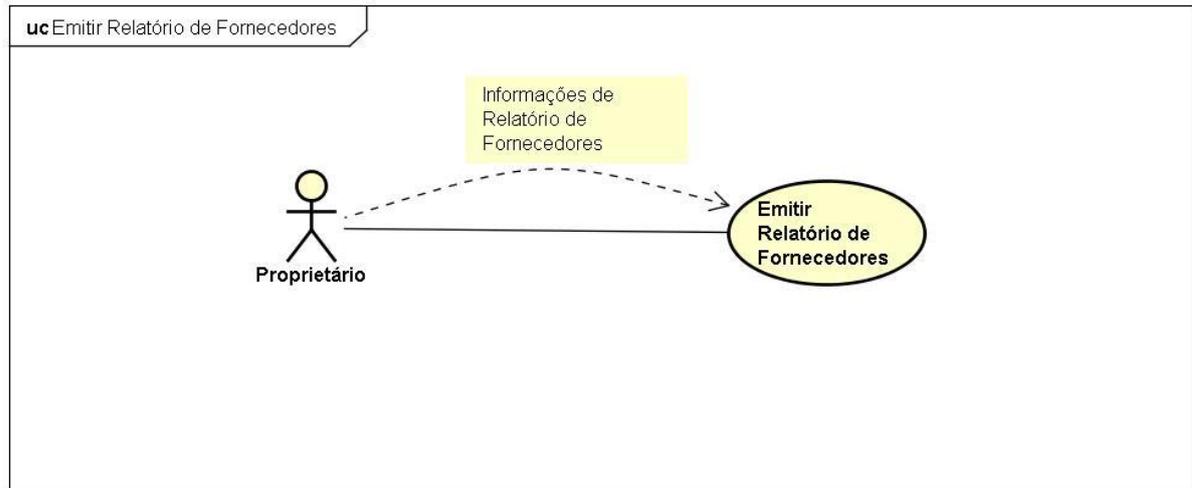


Figura 12. UC – Emitir Relatório de Fornecedores

Nome do Caso de Uso	Emitir relatório de Fornecedores
Pré-Condição	Proprietário deve estar logado no sistema.
Ator(es)	Proprietário.
Cenário Principal	<p>1 - O sistema exibe na tela a opção de gerar relatórios.</p> <p>2 – O proprietário clica na opção.</p> <p>3 – O sistema pede para o proprietário escolher pelo menos um tipo de relatório.</p> <p>4 – O proprietário escolhe emitir relatório de fornecedores.</p> <p>5 – O sistema exibe uma relação detalhada de todos os fornecedores cadastrados no sistema.</p>
Cenário Alternativo	O usuário cancela a operação antes da emissão do relatório.
Casos de Testes	O sistema verifica todas as movimentações feitas por fornecedores e especifica datas, produtos e valores negociados.

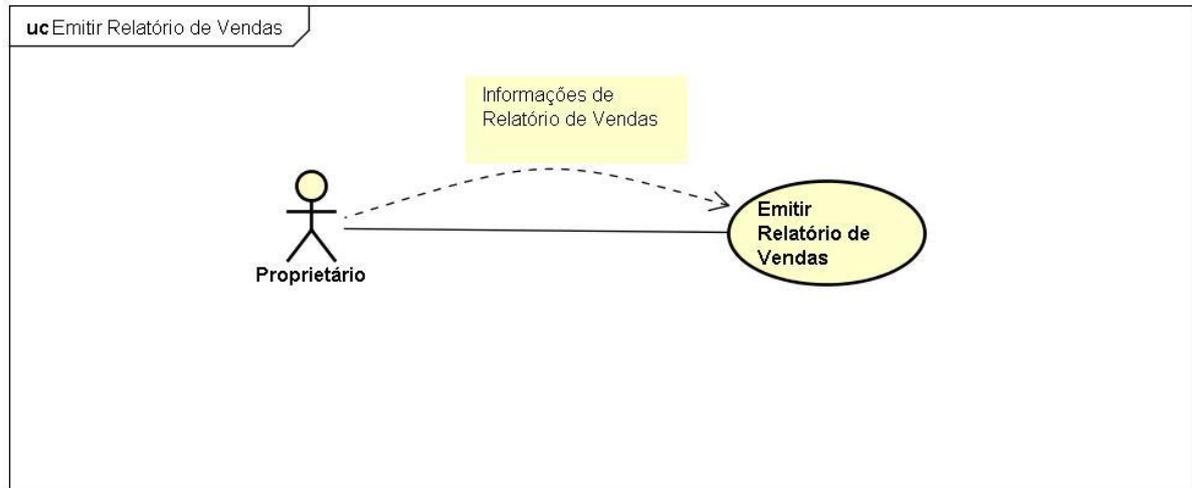


Figura 13. UC – Emitir Relatório de Vendas

Nome do Caso de Uso	Emitir Relatório de Vendas
Pré-Condição	O proprietário deve estar logado no sistema.
Ator(es)	Proprietário.
Cenário Principal	<p>1 - O sistema exibe na tela a opção de gerar relatórios.</p> <p>2 – O proprietário clica na opção.</p> <p>3 – O sistema pede para o proprietário escolher pelo menos um tipo de relatório.</p> <p>4 – O proprietário escolhe emitir relatório de vendas.</p> <p>5 – O sistema exibe os grupos de produtos para que o proprietário escolha uma opção.</p> <p>6 – O proprietário clica no grupo desejado.</p> <p>7 – O sistema exibe uma relação com todas as vendas separadas por períodos, valores, grupos, tendências entre outras informações.</p>
Cenário Alternativo	O usuário cancela a operação antes da emissão do relatório.
Casos de Testes	O sistema faz um levantamento de todos os dados referentes a vendas realizadas.

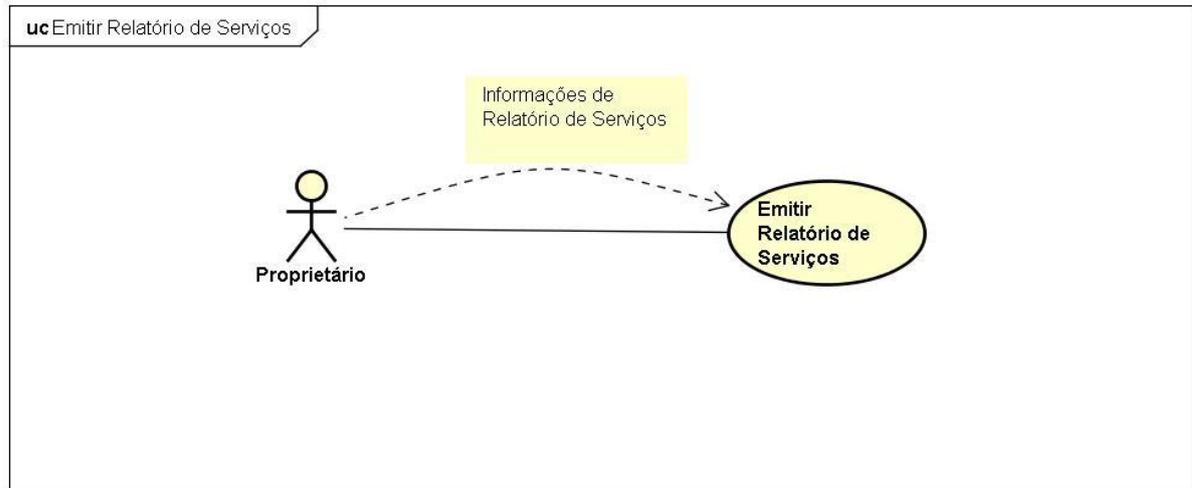


Figura 14. UC – Emitir Relatório de Serviços

Nome do Caso de Uso	Emitir Relatório de Serviços
Pré-Condição	O proprietário deve estar logado no sistema.
Ator(es)	Proprietário.
Cenário Principal	<p>1 - O sistema exibe na tela a opção de gerar relatórios.</p> <p>2 – O proprietário clica na opção.</p> <p>3 – O sistema pede para o proprietário escolher pelo menos um tipo de relatório.</p> <p>4 – O proprietário escolhe emitir relatório de Serviços.</p> <p>5 – O sistema exibe os tipos de serviços para que o proprietário escolha uma opção.</p> <p>6 – O proprietário clica no serviço desejado.</p> <p>5 – O sistema exibe todos os serviços prestados, materiais usados, horários da realização, frequência entre outras informações.</p>
Cenário Alternativo	O usuário cancela a operação antes da emissão do relatório.
Casos de Testes	O sistema verifica todos os serviços prestados por vários tipos de pesquisas e exibe com detalhes o relatório.

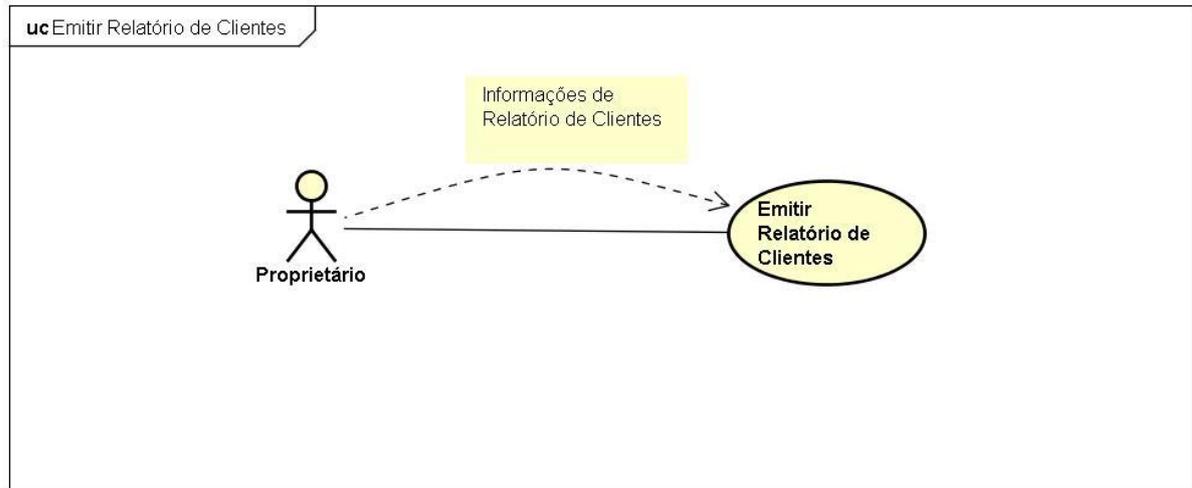


Figura 15. UC – Emitir Relatório de Clientes

Nome do Caso de Uso	Emitir Relatório de Clientes
Pré-Condição	O proprietário deve estar logado no sistema.
Ator(es)	Proprietário.
Cenário Principal	<p>1 - O sistema exibe na tela a opção de gerar relatórios.</p> <p>2 – O proprietário clica na opção.</p> <p>3 – O sistema pede para o proprietário escolher pelo menos um tipo de relatório.</p> <p>4 – O proprietário escolhe emitir relatório de Clientes.</p> <p>5 – O sistema exibe todos os clientes com filtragem para preferências, valores gastos, frequências, contas devedoras entre outras informações.</p>
Cenário Alternativo	O usuário cancela a operação antes da emissão do relatório.
Casos de Testes	O sistema faz todo um levantamento desde a primeira compra até a última e obtém detalhes variados dos clientes.

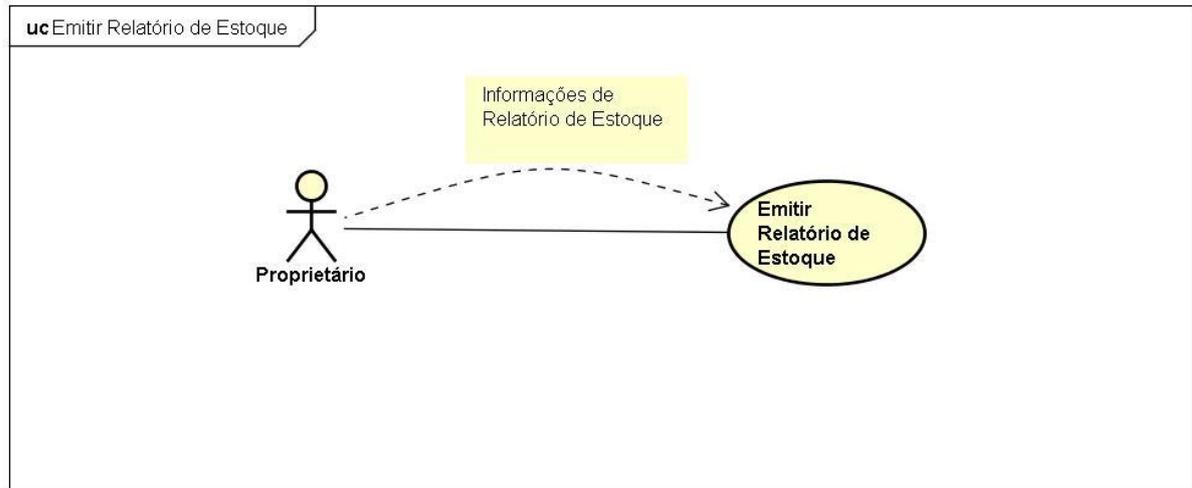


Figura 16. UC – Emitir Relatório de Estoque

Nome do Caso de Uso	Emitir Relatório de Estoque
Pré-Condição	O proprietário deve estar logado no sistema.
Ator(es)	Proprietário.
Cenário Principal	<p>1 - O sistema exibe na tela a opção de gerar relatórios.</p> <p>2 – O proprietário clica na opção.</p> <p>3 – O sistema pede para o proprietário escolher pelo menos um tipo de relatório.</p> <p>4 – O proprietário escolhe emitir relatório de Estoque.</p> <p>5 – O sistema detalha com filtragem de datas, estoque mínimo, validade entre outras informações.</p>
Cenário Alternativo	O usuário cancela a operação antes da emissão do relatório.
Casos de Testes	O sistema é capaz de produzir relatórios de estoque em tempo real e avisar quando um produto está com quantidade mínima em estoque.

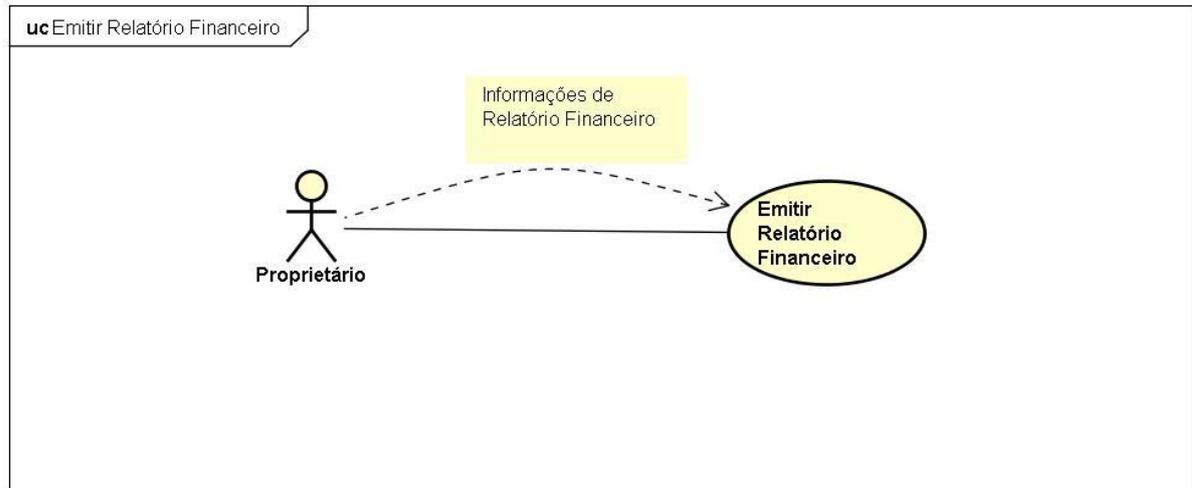


Figura 17. UC – Emitir Relatório Financeiro

Nome do Caso de Uso	Emitir Relatório Financeiro
Pré-Condição	O proprietário deve estar logado no sistema.
Ator(es)	Proprietário.
Cenário Principal	<p>1 - O sistema exibe na tela a opção de gerar relatórios.</p> <p>2 – O proprietário clica na opção.</p> <p>3 – O sistema pede para o proprietário escolher pelo menos um tipo de relatório.</p> <p>4 – O proprietário escolhe emitir relatório de Financeiro.</p> <p>5 – O sistema exibe todas as oscilações e fluxos de caixa entre um período desejado de tempo além de fazer um balanço geral do negócio.</p>
Cenário Alternativo	O usuário cancela a operação antes da emissão do relatório.
Casos de Testes	O sistema

7. DIAGRAMA DE CLASSES

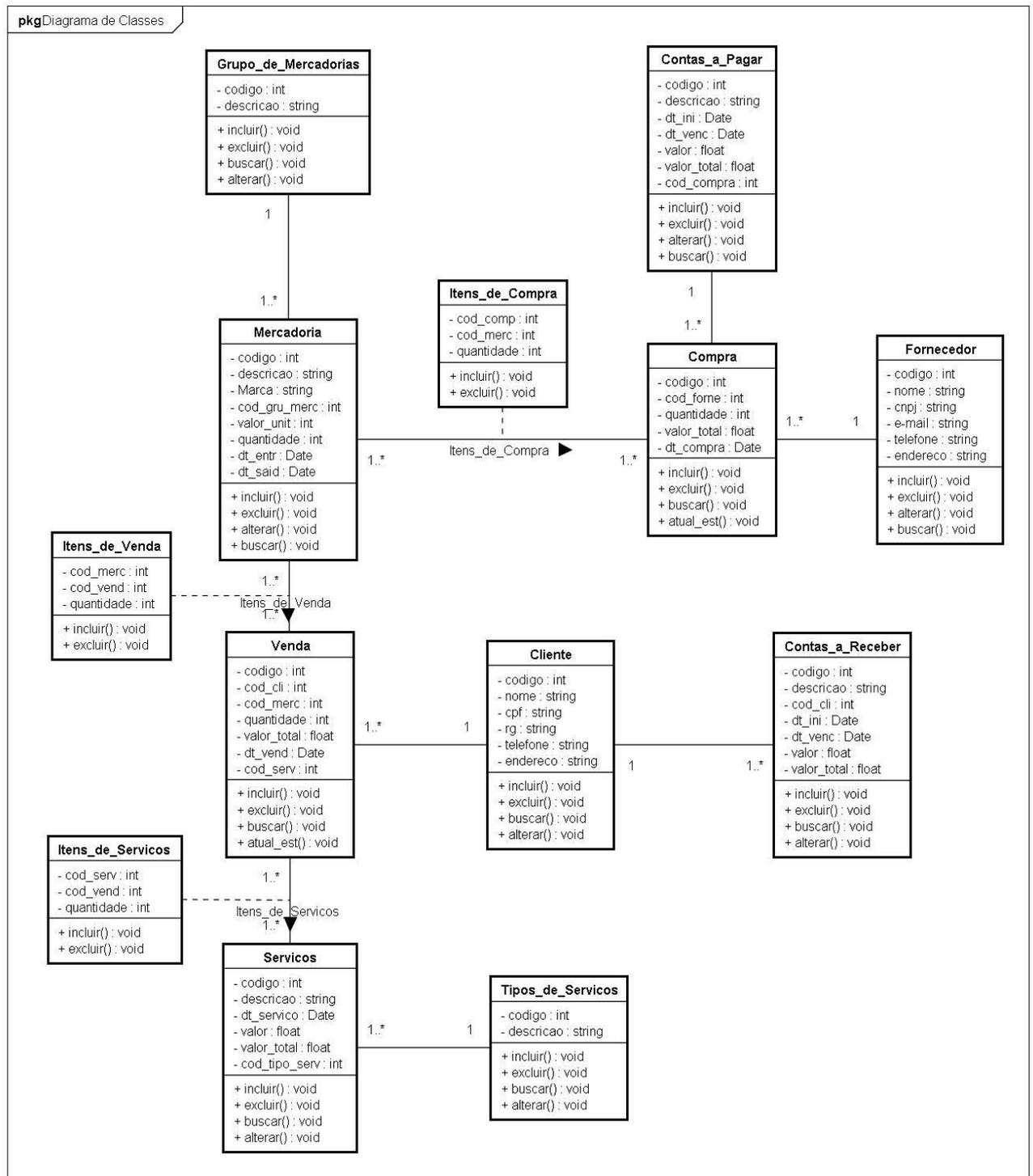


Figura 18 – Diagrama de Classes

8. DIAGRAMAS DE SEQUÊNCIA

FORNECEDOR

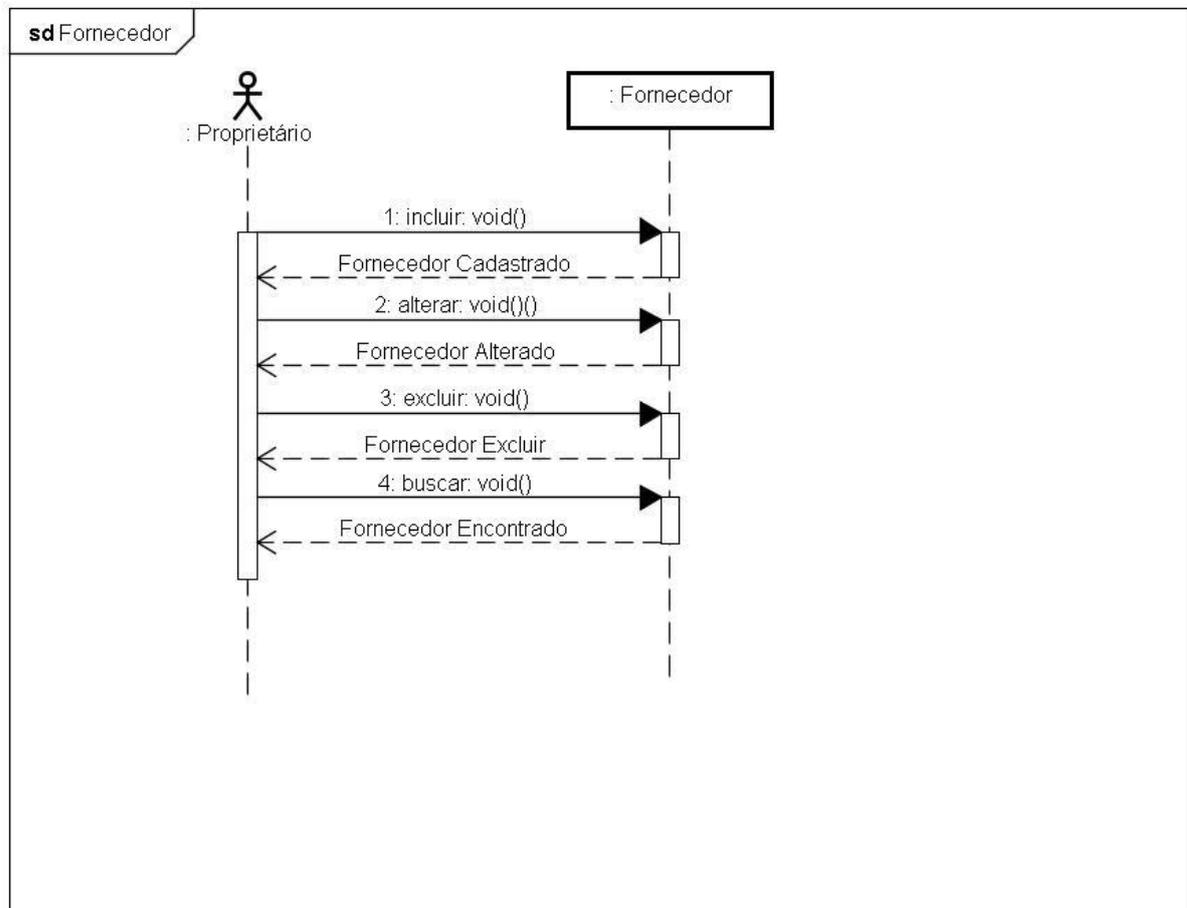


Figura 19. Diagrama de Sequência – Fornecedor

SERVIÇOS

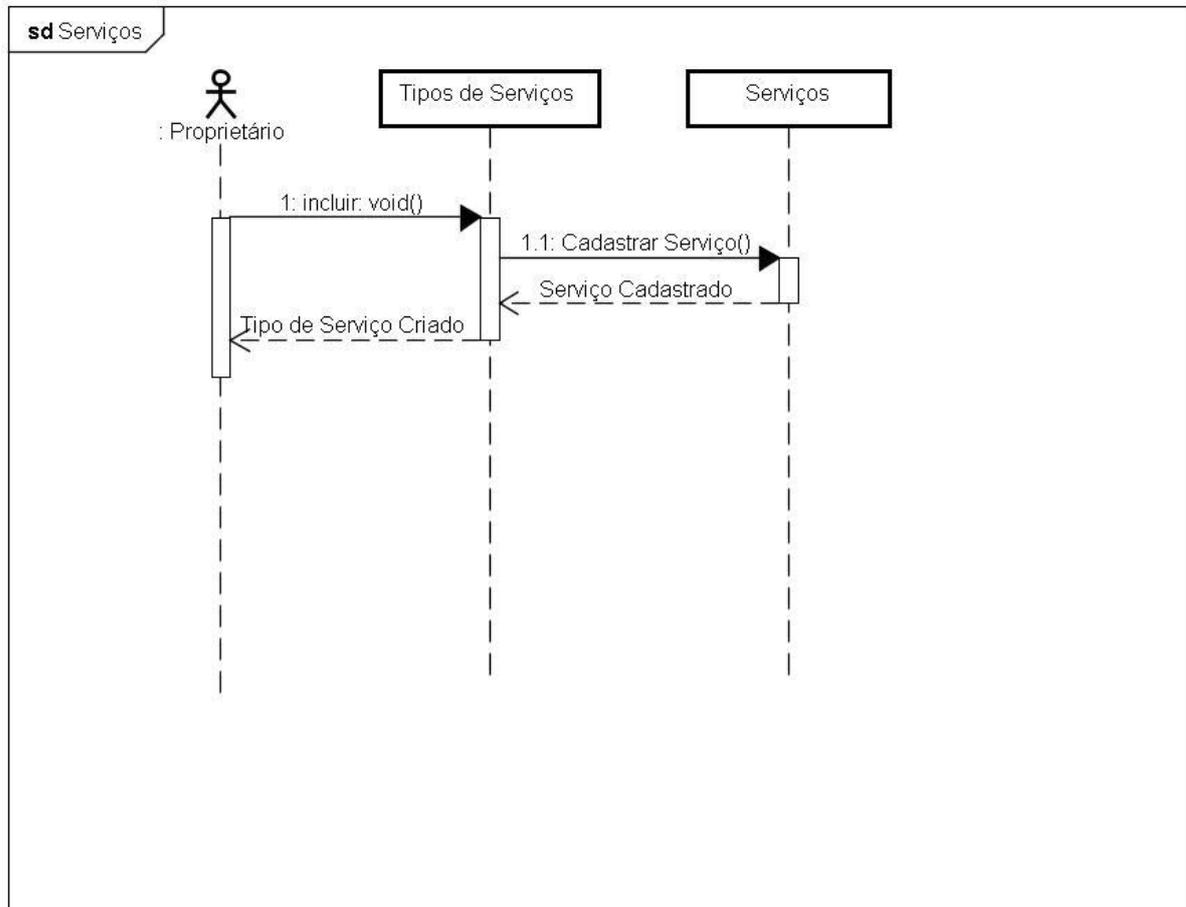


Figura 20. Diagrama de Sequência - Serviços

CLIENTES

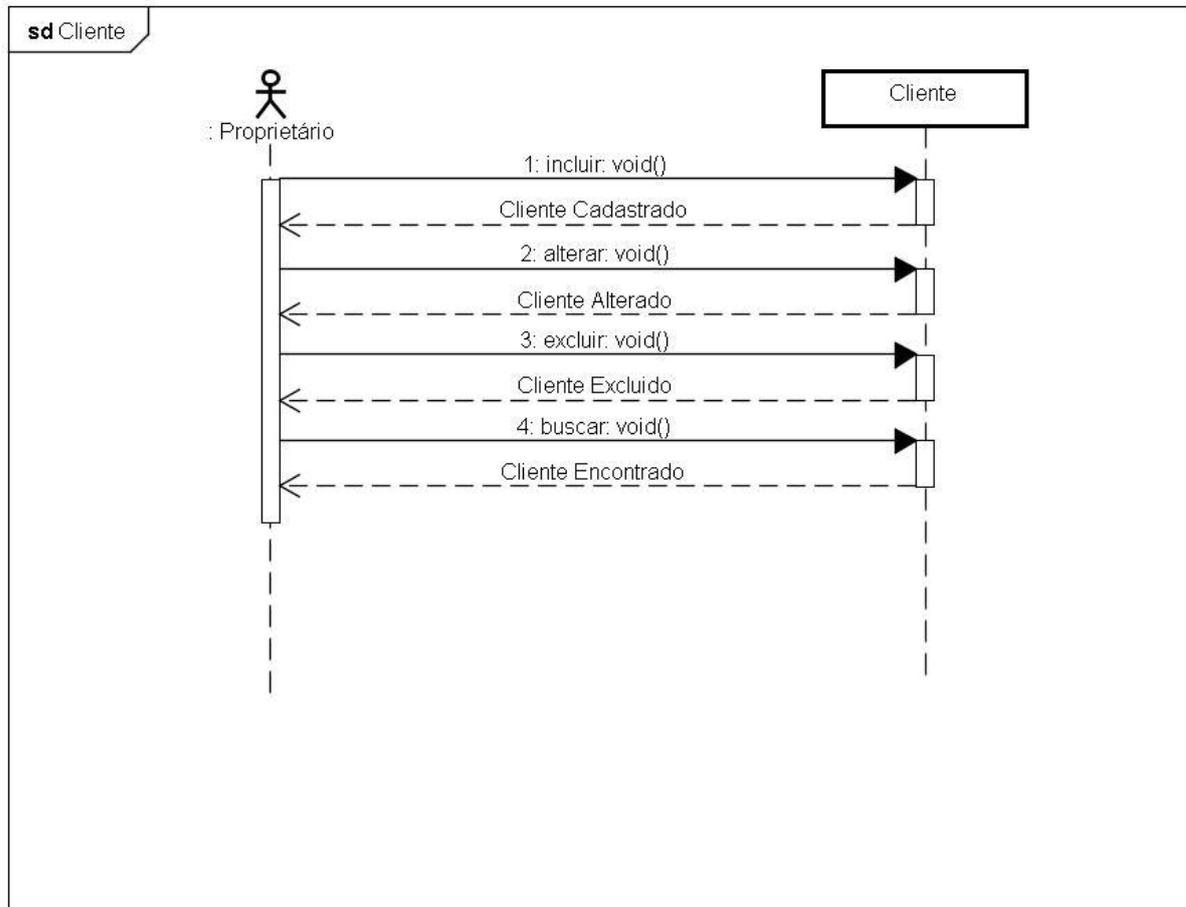


Figura 21. Diagrama de Sequência – Clientes

CONTAS A PAGAR

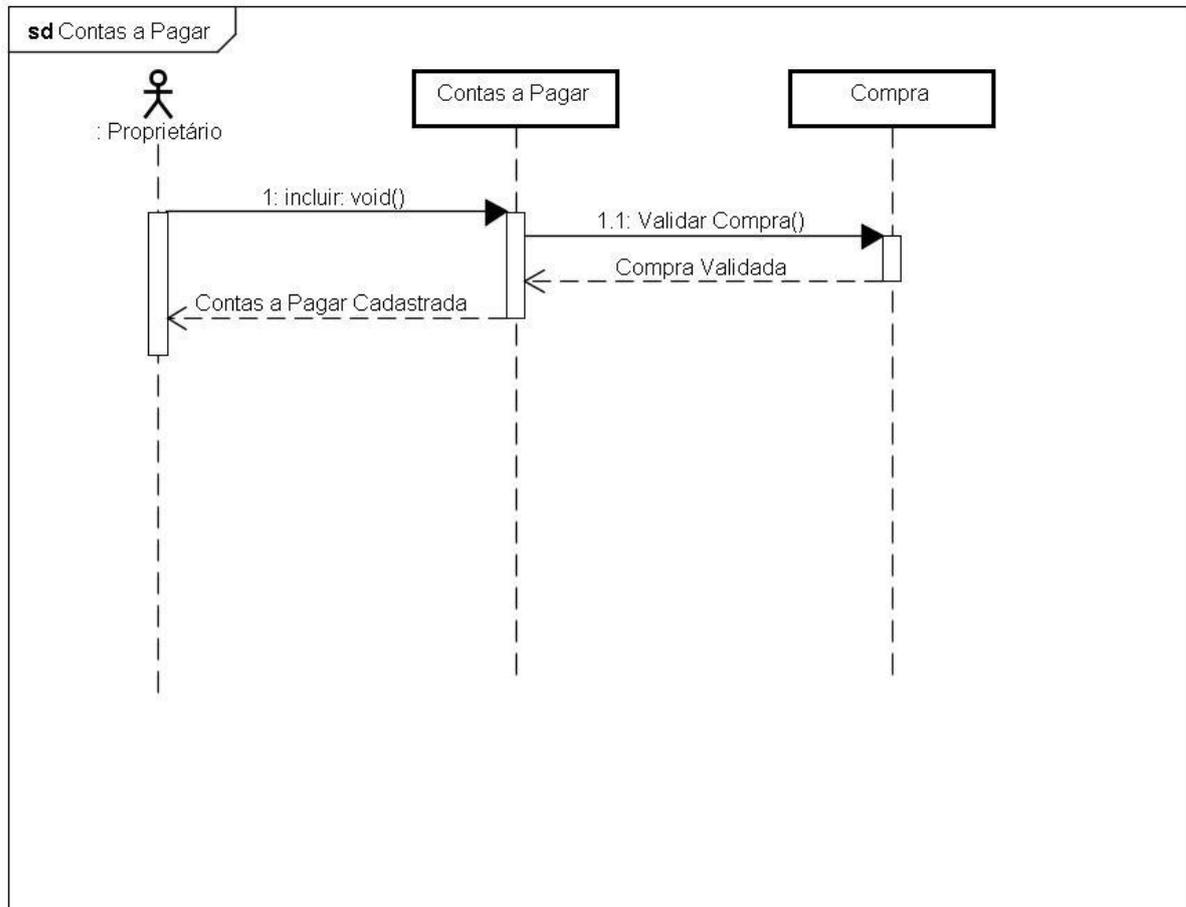


Figura 22. Diagrama de Sequência – Contas a Pagar

CONTAS A RECEBER

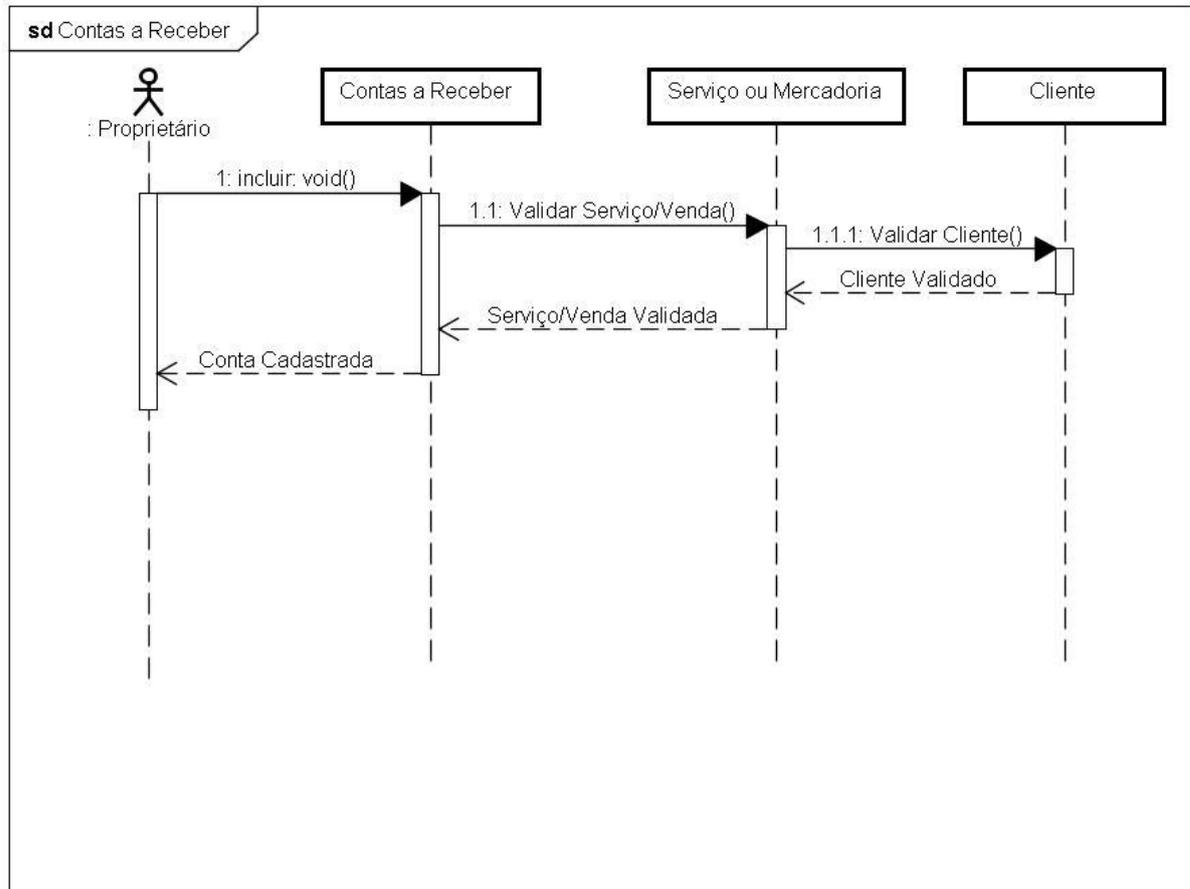


Figura 23. Diagrama de Sequência – Contas a Receber

MERCADORIA

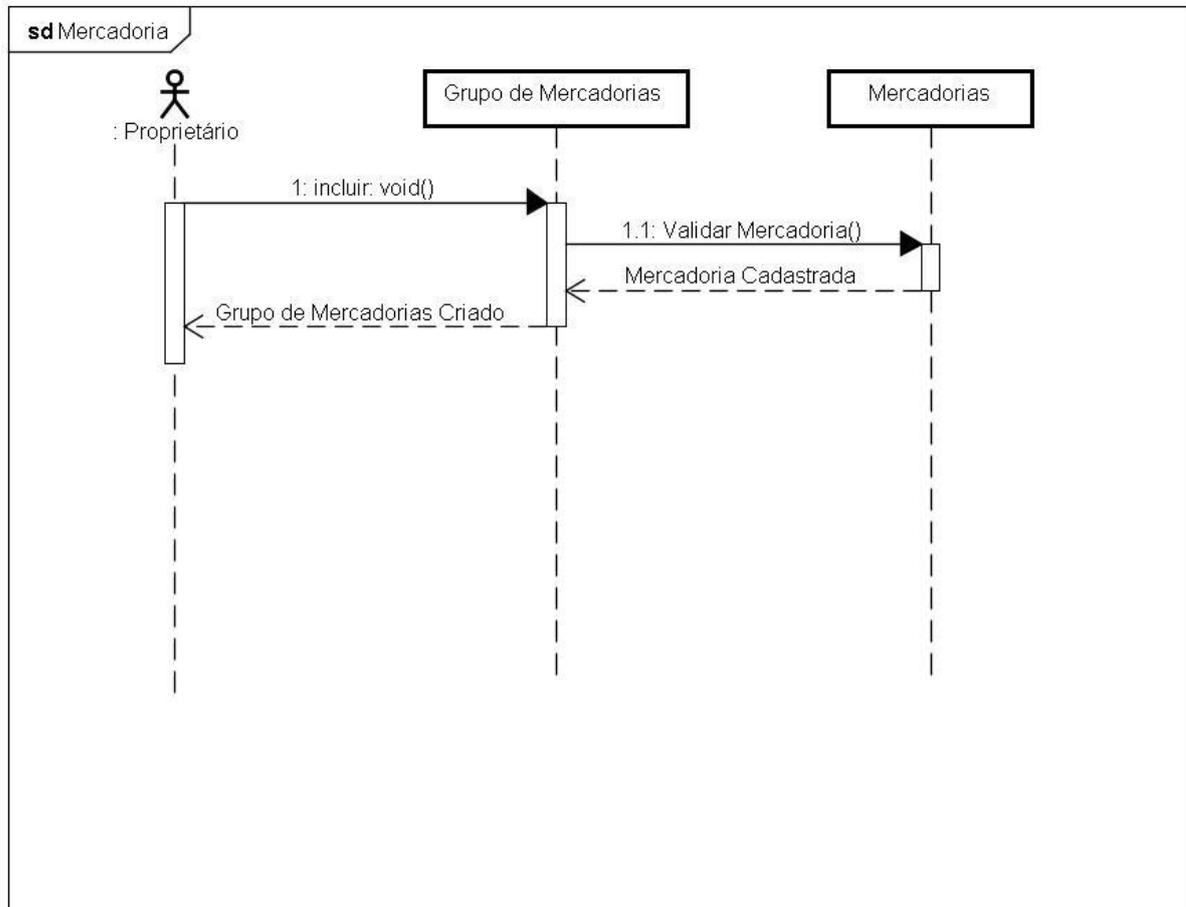


Figura 24. Diagrama de Sequência – Mercadoria

9. DIAGRAMA DE ATIVIDADES

COMPRA

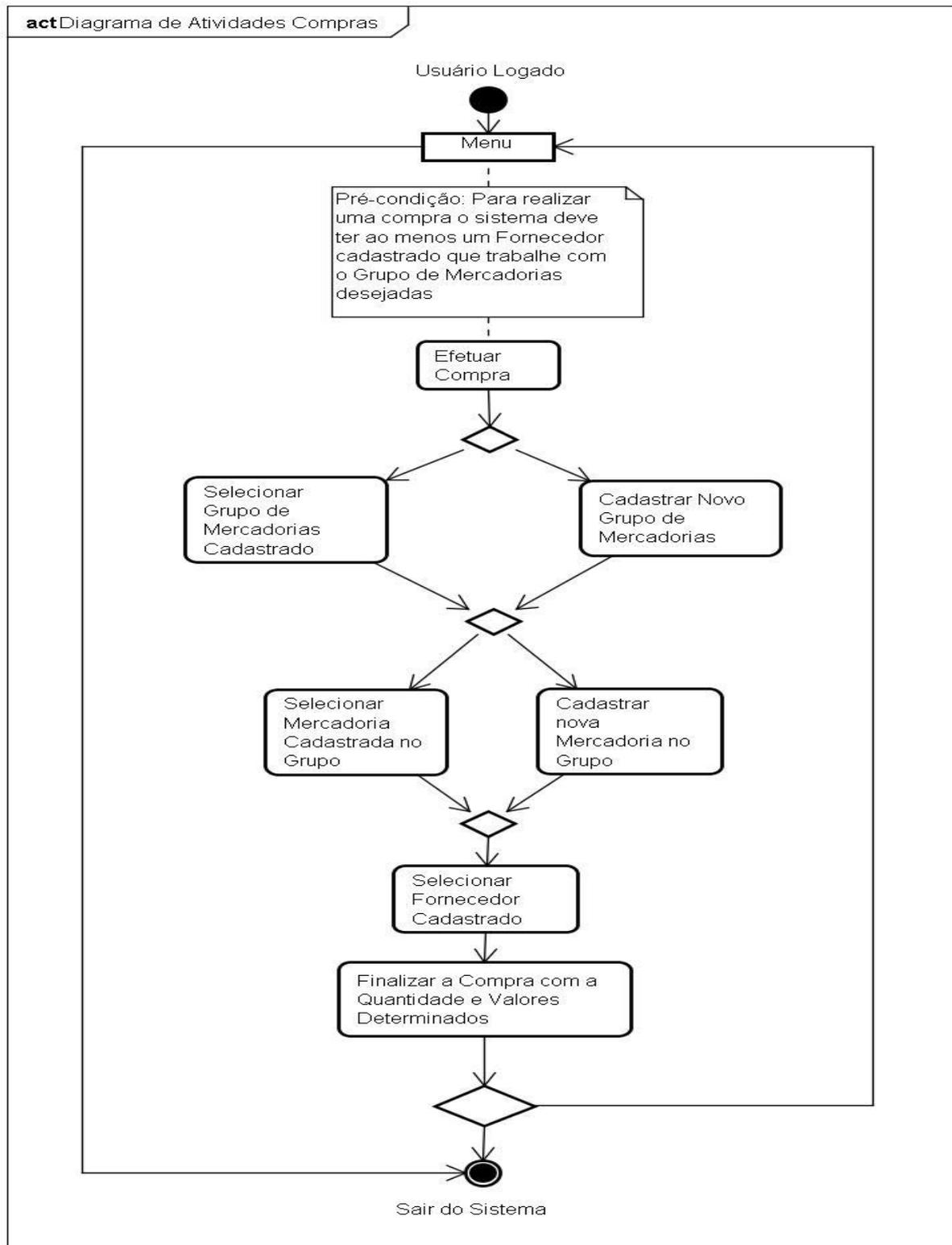


Figura 25. Diagrama de Atividades – Compra

VENDA

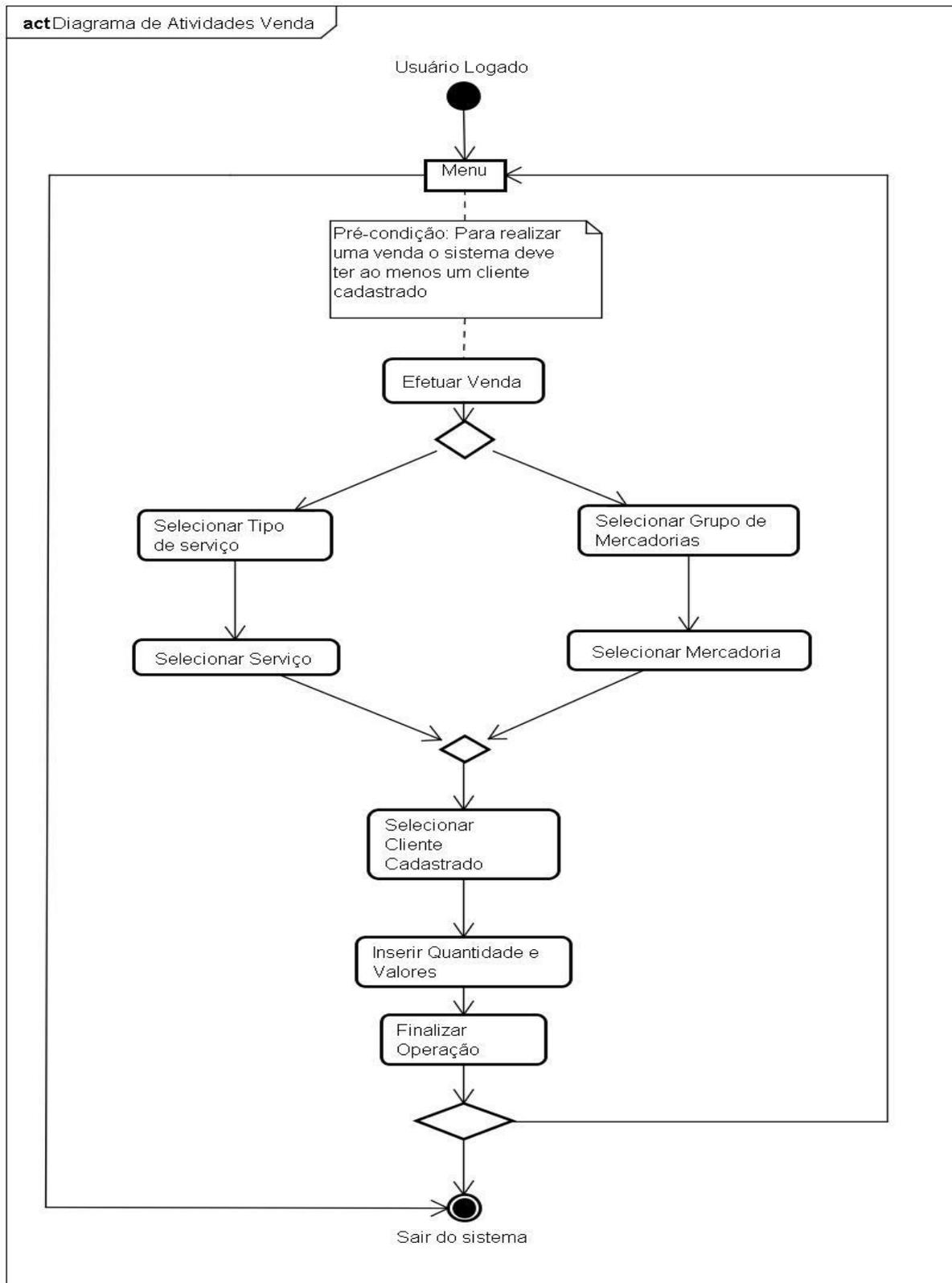


Figura 26. Diagrama de Atividades – Venda

CRONOGRAMA

Abaixo seguem os períodos com que serão executadas as etapas do projeto.

Atividade / Semana	Fevereiro				Março				Abril				Maio				Junho				Julho				Agosto				Setembro				Outubro				Novembro							
	1ª	2ª	3ª	4ª	1ª	2ª	3ª	4ª	1ª	2ª	3ª	4ª	1ª	2ª	3ª	4ª	1ª	2ª	3ª	4ª	1ª	2ª	3ª	4ª	1ª	2ª	3ª	4ª	1ª	2ª	3ª	4ª	1ª	2ª	3ª	4ª	1ª	2ª	3ª	4ª				
Levantamento de Requisitos	■	■	■	■																																								
Análise dos Requisitos					■	■																																						
Especificação dos Requisitos							■	■																																				
Especificação dos Casos de Uso									■	■																																		
Diagrama de Casos de Uso											■	■																																
Diagrama de Classes													■	■																														
Diagrama de Sequencia															■	■																												
Diagrama de Atividades																	■	■																										
Implantação do Banco de Dados																			■	■	■	■																						
Programação																																												
Testes																																												
Instalação																																												
Treinamento																																												
Apresentação																																												

Figura 27. Cronograma

REFERÊNCIAS

ANDRADE, Thiago Faria. Apostila Algaworks Treinamentos – Desenvolvimento Web com JavaServer faces. 2. Ed. Uberlândia, 2010.

GONÇALVES, Edson. Desenvolvendo Aplicações WEB com NetBeans IDE 6. 1. Ed. Rio de Janeiro: Editora Ciência Moderna, 2008.

MACORATTI, José Carlos. UML – Unified Modeling Language e Visual Modeler. Disponível em: <http://www.macoratti.net/uml_vb.htm>. Acesso em: 24 jun. 2012.

MIRANDA, Ana Paula. Aplicações Web Ricas - RIAs. Porto Alegre. Disponível em: <<http://www.anapaulaweb.com.br/2010/03/aplicacoes-web-ricas-rias.html>> Acesso em: 20 out. 2012.

MORIMOTO, Carlos. Bytecode. Disponível em: <<http://www.hardware.com.br/termos/bytecode>>. Acesso em: 24 jun. 2012.