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

GABRIEL VIEIRA GALLI

SISTEMA PARA GESTÃO DE AUTOPEÇAS E OFICINA

ASSIS
2013

GABRIEL VIEIRA GALLI

SISTEMA PARA GESTÃO DE AUTOPEÇAS E OFICINA

Trabalho de Conclusão de Curso no Curso em Analise em Desenvolvimento de Sistemas do Instituto Educacional do Munícipio de Assis – IMESA e a Fundação Educacional do Munícipio de Assis – FEMA, como requisito à obtenção do Certificado de Conclusão.

Orientador: Célio Desiró.

ASSIS
2013

FICHA CATALOGRÁFICA

GALLI, Gabriel Vieira.

Sistema para Gestão de Autopeças e Oficina / Gabriel Vieira Galli. Fundação Educacional do Munícipio de Assis – FEMA -- Assis, 2013.

42 p.

Orientador: Célio Desiró.

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

1. Tecnologia em Analise e Desenvolvimento de Sistemas.
2. Java.

CDD: 001.61

Biblioteca da FEMA.

Sistema para Gestão de Autopeças e Oficina

Gabriel Vieira Galli

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: Celio Desiró

Analisador: Douglas Sanches da Cunha.

ASSIS
2013

DEDICATÓRIA

Dedico aos meus familiares, pais e avós que me incentivaram, me apoiaram e me encorajaram dando o apoio necessário nos momentos difíceis na longa caminhada que enfrentei até agora para o término desse projeto e conclusão da faculdade.

AGRADECIMENTO

Agradeço primeiramente e DEUS, pois sem ele nada seria possível nesse momento, pela minha vida e saúde.

Aos meus colegas de curso, que me ajudaram e apoiaram nas dificuldades durante a jornada no curso.

Ao meu professor Orientador Célio Desiró que me instruiu neste trabalho dando sempre o auxílio necessário.

RESUMO

Este projeto aborda o desenvolvimento do software “Sistema para Gestão de Autopeças e Oficina”.

Este software tem por objetivo informatizar a empresa, auxiliando na tomada de decisão e na redução de tempo e custos, tanto para a empresa quanto para os clientes. Foi desenvolvido com a linguagem de programação Java, simples porém completa, utilizando o banco de dados MySql, muito usado no mundo todo, de fácil manipulação, além de ser livre.

ABSTRACT

This project addresses the development of the "Management System for Auto Parts and Workshop"

This software aims at computerising the company, assisting in decision making and reduce time and costs for both the company and to customers. It was developed with the Java, programming simple, but complete, using the MySQL database, widely used worldwide, easy handling as well as free.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

SGBD	Sistema Gerenciador de Banco de Dados.
SQL	Structure Query Language.
JVM	Java Virtual Machine.
IDE	Ambiente Integrado de Desenvolvimento.
UML	Unites Modeling Language.
IBM	International Business Machines
JFC	Java Foundation Classes
JSF	Java Server Faces

SUMÁRIO

1 - INTRODUÇÃO

A Autopeça Galli LTDA e Oficina Mecânica Galli ME trabalham em conjunto, uma com a venda de peças e a outra com a venda de serviços, e possuem os mesmos proprietários.

A Empresa Oficina Mecânica Galli ME é uma empresa fundada no ano de 1987, já a empresa Autopeças Galli LTDA é uma empresa fundada no ano de 2010 e foi criada através da ideia de expandir os negócios, para que os serviços da oficina fossem realizados mais rapidamente, não dependendo da entrega de peças de outras autopeças. Com um estoque próprio é mais fácil identificar as peças no próprio balcão do que ir a outras lojas em busca das mesmas. Atualmente as empresas contam com uma carteira com cerca de 700 clientes de Assis e Região.

O presente trabalho, denominado de SISTEMA PARA GESTÃO DE AUTOPEÇAS E OFICINA terá como principal objetivo informatizar o gerenciamento de controle de estoque, cadastro de cliente e fornecedores, contas a pagar e receber e ordens de serviços executados em veículos. Com a automação do sistema fica muito mais fácil um cliente saber todos os serviços realizados em seu veículo e as peças utilizadas, bem como as datas dos serviços realizados.

O sistema visa atender todas as necessidades de uma autopeça e oficina, visando um atendimento mais eficaz, possibilitando possíveis atualizações no sistema, emissão de relatórios entre outros procedimentos.

1.2 - OBJETIVO

Este sistema tem por objetivo ajudar no processo de desenvolvimento da Auto Peça, auxiliando na manutenção, gerenciamento do sistema e na tomada de decisões, economizando tempo e dinheiro.

Em geral o sistema visa o gerenciamento de uma Autopeça, cadastrando produtos, clientes, fornecedores, contas a pagar e receber etc.
Com o atual crescimento em torno de tecnologias e desenvolvimento de softwares, muitas autopeças já investem em tecnologias, muitos catálogos deixaram de serem

impressos e passaram à digital, com isso, este projeto visa ajudar o setor que neste momento esta passando por esta evolução. Empresas que não estão pensando em se desenvolver estão ficando para trás.

1.3 - PÚBLICO ALVO

O sistema proposto irá auxiliar as empresas na área de atuação de autopeças e oficinas mecânicas, visando a melhora do atendimento aos clientes e a praticidade de acesso as informações necessárias, com mais agilidade e segurança nas informações fornecidas aos clientes e aos proprietários.

1.4 - JUSTIFICATIVA

Com o sistema é esperada uma otimização dos custos e tempo de serviços em relação aos profissionais que utilizarão o sistema, uma busca detalhada dos produtos cadastrados com um menor tempo, um relatório de vendas detalhado, uma fácil compreensão do sistema.

1.5 - METODOLOGIA DE DESENVOLVIMENTO

Na primeira etapa do projeto serão levantados os requisitos do sistema junto ao cliente por meio de visitas ao local onde este será implantado. Com uma conversa detalhada com os usuários do sistema será possível colher dados importantes para se conhecer um pouco mais das necessidades para elaboração do projeto.

Na segunda etapa do projeto será desenvolvido o software com base no que foi relatado na pesquisa da etapa anterior.

1.6 – ESTRUTURA E DESENVOLVIMENTO DO SISTEMA

Para o desenvolvimento do software será necessário uma organização dos dados e tarefas visando economia de tempo e obtenção dos resultados desejados.

1.6.1 - Metodologias de Análise

- Declaração de Objetivos;
- Diagrama de Casos de Uso;
- Diagrama de Classe;

1.6.2 - Linguagem de Implementação (Java)

- Historia Java

Em 1991, a Sun Microsystems, iniciou um projeto chamado Green Project, o berço do Java uma linguagem de programação orientada a objetos. Os mentores do projeto eram Patrick Naughton, Mike Sheridan e James Gosling. O objetivo do projeto não era a criação de uma nova linguagem de programação, mas antecipar e planejar a próxima onde do mundo digital. Eles acreditavam que em algum tempo haveria uma convergência dos computadores com os equipamentos e eletrodomésticos comumente usados pelas pessoas no seu dia a dia.

Para provar a viabilidade desta ideia, treze pessoas trabalharam arduamente durante dezoito meses. No verão de 1992 eles emergiram de um escritório de Sand Hill Road no Menlo Park com uma demonstração funcional da ideia inicial. O protótipo se chamava *7(StarSeven), um controle remoto com uma interface gráfica touchscreen. Para *7 foi criado um mascote, hoje amplamente conhecido no mundo Java o Duke. O trabalho do Duke no *7 era ser uma guia virtual ajudando e ensinando o usuário a utilizar o equipamento. O *7 tinha a habilidade de controlar diversos dispositivos e aplicações. James Gosling especificou uma nova linguagem de programação para o *7. Goslling decidiu batiza-la de “Oak”, que quer dizer carvalho, uma arvore que ele podia observar quando olhava pela sua janela, tempos mais tarde, descobriu-se que já existia uma linguagem de programação com esse nome. Com isso, em uma visita da equipe de desenvolvimento a uma cafeteria local, o nome Java surgiu (nome do café) e logo denominada o da linguagem.

O próximo passo era encontrar um mercado para o *7. A equipe achava que uma boa ideia seria controlar televisões de vídeo por demanda com o equipamento. Eles construíram um demo chamado MovieWood, mas infelizmente era muito cedo para

que o vídeo por demanda bem como as empresas de TV a cabo pudessem viabilizar o negocio. A ideia do *7 tentava vender, hoje já e realidade em programas interativos e também na televisão digital. A sorte e que o bom da Internet aconteceu, e rapidamente uma grande rede interativa estava se estabelecendo. Era este tipo de rede interativa que a equipe do *7 estava tentando vender para as empresas de TV a cabo. E da noite para o dia, não era mais necessário construir a infraestrutura para a rede, em um golpe de sorte, ela simplesmente estava lá. Gosling foi incumbido de adaptar o Oak para a Internet e em janeiro de 1995 foi lançada uma nova versão do Oak que foi rebatizado para Java. A tecnologia Java tinha sido projetada para se mover por meio das redes de dispositivos heterogêneos, redes com ao Internet. Agora aplicações poderiam ser executa\das dentro de Browsers nos Applets Java e tudo seria disponibilizado pela Internet instantaneamente. Foi o estático HTML dos Browsers que promoveu a rápida disseminação da dinâmica tecnologia Java. A velocidade dos acontecimentos seguintes foi assustadora, o numero de usuários cresceu rapidamente, grandes players, como a IBM anunciaram suporte para a tecnologia Java.

Desde seu lançamento , em maio de 1995, a plataforma Java foi adotada mais rapidamente do que qualquer outra linguagem de programação na historia da computação. Em 2003 Java atingiu a marca de quatro milhões de desenvolvedores em todo mundo. Java continuou continua crescendo e hoje e com certeza um padrão para o mercado oferecendo qualidade, e desempenho e segurança ainda sem nenhum competidor a altura. Java tornou-se popular pelo sue uso na Internet e hoje possui seu ambiente de execução em web browsers, mainframes, SOs, celulares, palmtops e cartões inteligentes , entre outros.(CAELUM FJ11, p.03-06)

1.6.3 - Máquina Virtual Java (JVM)

É um mecanismo que permite executar códigos Java em qualquer plataforma. Segundo a Sun, a JVM pode ser entendida como uma máquina imaginaria, implementada via software ou hardware que executa instruções vindas de bytecodes. Quando compilamos um programa e criado um arquivo .exe, executável, que roda em contato direto com o Sistema Operacional do computador do usuário.

Assim, se o .exe foi compilado no Windows não roda no Linux. Por isso que os programas precisam ter versões, uma para o Linux e outra para o Windows e assim por diante.

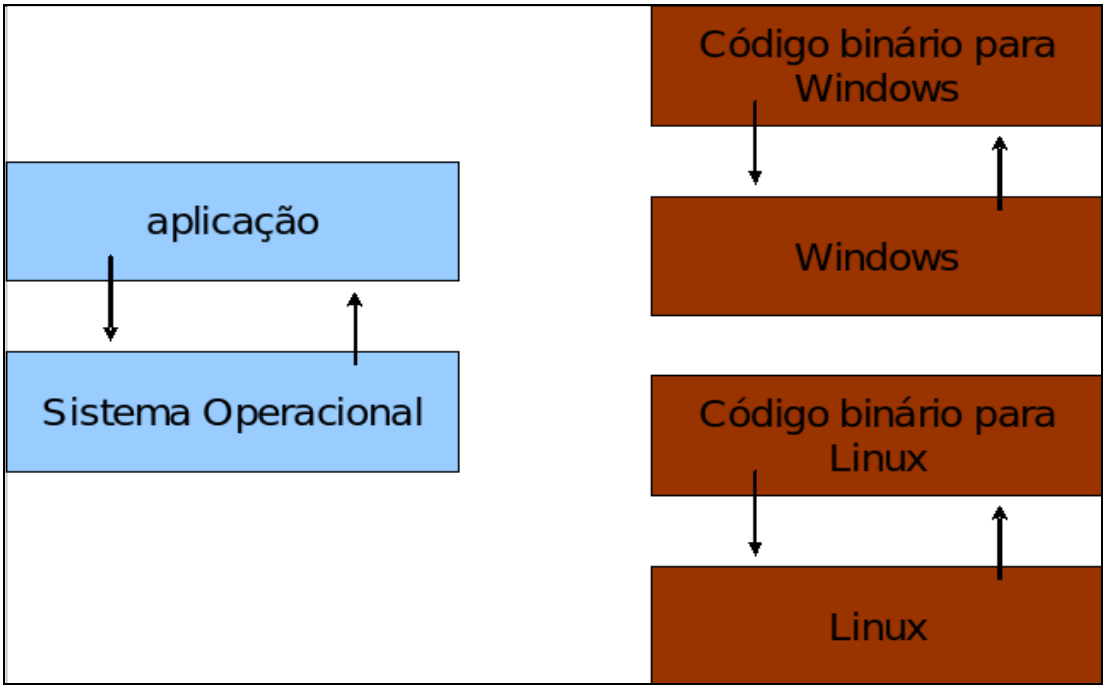

Figura01: Código executado em um determinado sistema operacional. (CAELUM, 2013)

Quando você faz um programa em Java e o compila, o compilador gerará bytecodes desse programa. Bytecode é uma espécie de codificação que traduz tudo o que foi escrito no programa para um formato que JVM entenda e seja capaz de executar. Assim, se você fizer um programa em Java no Linux, ele será capaz de rodar em Windows ou em qualquer outro sistema operacional que tenha JVM. Isso ocorre porque não existe bytecodes diferentes, isto é, os bytecodes dos programas em Java compilador no Windows são constituídos da mesma forma que bytecodes gerados se a compilação fosse feita no Linux ou em qualquer outro sistema operacional.

Quando um código em Java e compilado, um arquivo com a extensão .class é gerado. Esse tipo de arquivo e o bytecode. Assim quando o programa Infowester.java for compilado, um arquivo chamado infowester.class dever ser executado. (CAELUM FJ11, p.04-07)

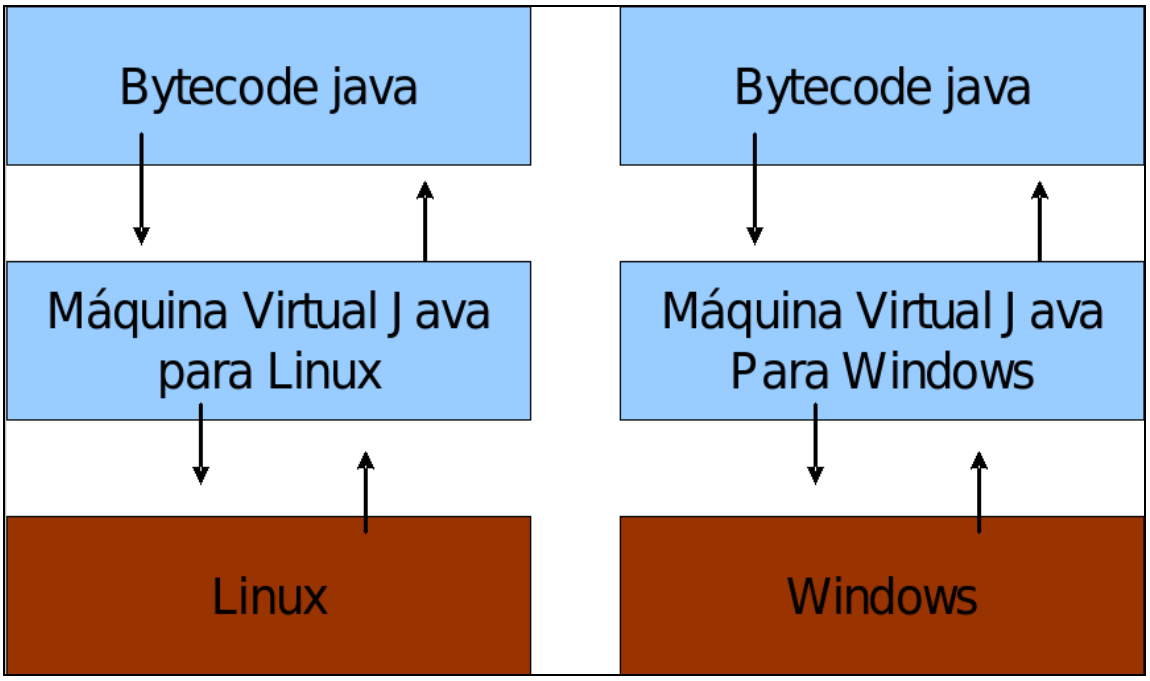

Figura02: Código executado em Java. (CAELUM, 2013)

1.6.4 - Fundamentos da Orientação Objetos

A orientação objetos e uma tecnologia que enxerga os sistemas como sendo coleção de objetos integrantes. Permite melhorar a reutilização dos softwares.

A tecnologia orientada a objetos é fundamentada no que, coletivamente, chamamos de modelo de objetos, que engloba os princípios da abstração, hierarquização, encapsulamento, classificação, modularização, relacionamento, simultaneidade e persistência.

Os programas Orientados a objetos são programas estruturados em módulos que agrupam um estado e operações sobre este estado. Apresentam ênfase em reutilização de código.

1.6.4.1 - Objetos

Um objeto e um elemento computacional que representa, no domínio da solução, alguma entidade (abstrata ou concreta) do domínio de interesse do problema sob a análise. Objetos similares são agrupados em classe.

1.6.4.2 – Classes

É um gabarito para a definição de objetos. Através da definição de uma classe, descreve-se que propriedades ou atributos o objeto terá.

1.6.4.3 Métodos

Um conjunto de funcionalidades da classe, com restrições que ele manipula apenas suas variáveis locais e os atributos que foram definidos para um objeto desse tipo.

1.6.4.4 Herança

Permite a reutilização da estrutura e do comportamento de uma classe ao se definir novas classes. A classe que herda o comportamento é chamada de subclasse e a que definiu o comportamento superclasse.

1.6.5 - Ambiente de Desenvolvimento Eclipse IDE

Eclipse é um IDE desenvolvido em Java, seguindo o modelo open source de desenvolvimento de software. O projeto Eclipse foi iniciado na IBM que desenvolveu a primeira versão do produto e doou-o com software livre para a comunidade. O gasto inicial da IBM no produto foi demais de 40 milhões de dólares. Hoje, o Eclipse é uma das IDES Java mais utilizadas no mundo. Possui como característica marcante o uso de SWT e não do Swing como biblioteca gráfica, a forte orientação ao desenvolvimento baseado em plug-ins e o amplo suporte ao desenvolvedor com centenas de plug-ins que procuram atender as diferentes necessidades de diferentes programadores.

1.6.5.1 - Swing

Swing é o nome dado ao pacote de classes desenvolvidas pelo "Projeto Swing" – parte de um contexto maior chamado de JFC – Java Foundation Classes. O Swing consiste em um conjunto de componentes gráficos (extensões dos componentes AWT e novos componentes como representação de árvores e painéis tabulados),

que agregam o conceito de look and feel (L&F), ou seja, a capacidade de um mesmo componente assumir aparências diferentes sem a necessidade de mudanças no seu código. Por exemplo, você pode criar uma aplicação Java que se adapte à aparência gráfica do sistema operacional em que for executado – assumindo um aspecto semelhante às demais janelas Windows, Linux, Solaris, Macintosh, etc. Os componentes Swing são totalmente criados em Java (100% pure Java) e foram desenvolvidos no conceito de interface peso-leve do usuário. A ideia é codificar apenas a funcionalidade do componente e a sua relação com o modelo de dados ao qual está associado, deixando a sua aparência a cargo do gerenciador de interface do usuário (UI Manager), um novo recurso incorporado às máquinas virtuais a partir da versão 1.3 do ambiente de desenvolvimento Java (jdk1.3.1).Nesta aula não iremos detalhar a manipulação dos componentes Swing. Ao invés disso, focaremos a atenção no conceito de Modelo-Visão-Controle, o padrão de projeto adotado pela maioria dos componentes Swing. Aconselha-se fortemente ao aluno que leia a excelente documentação sobre Swing e analise os exemplos que acompanham o jdk.

1.6.6 JSF (JAVA SERVER FACES)

O Java Server Faces é uma tecnologia que permite a criação de aplicações WEB utilizando componentes visuais, com características de um framework MVC (MODEL-VIEW-CONTROLLER) para WEB. Por se basear no padrão MVC uma de suas principais vantagens é a de separar a aplicação em camadas: MODEL, VIEW e CONTROLLER. (GOMES, Yuri Marx P. p.10-16)

MODEL: A camada model e a responsável por fornecer o acesso aos dados ao controlador e manter o estado da aplicação.

VIEW: A camada view representa a interface com o usuário, como os dados serão apresentados e encaminhados ao controlador.

CONTROLLER: a camada controller é responsável por fazer a comunicação entre a view e a model, e interpretar as ações do usuário enviando os dados pro model realizar as mudanças e gerar uma visualização apropriada.

1.6.7 Apache Tomcat

É um servidor Web Java, um container de Servlets, que permite a execução de aplicações Web, desenvolvido pela Fundação Apache, é de código livre e escrito em Java para implementação nas tecnologias Java Servlet e JavaServer Pages (JSF). Possui características de servidor de aplicação, porém não podemos dizer que é um servidor por não preenche todas as características de um servidor.

2 - LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos foi feito com base em entrevistas realizadas com os envolvidos que farão uso do sistema no dia-a-dia da empresa, com o propósito de se obter o maior número de informações necessárias para o desenvolvimento do sistema.

A seguir é apresentada a pesquisa padrão realizada.

Analista: Por que a empresa em questão necessita de um sistema informatizado?

Cliente: A empresa precisa de mais controle em seu funcionamento, porque ainda utilizamos métodos ultrapassados para se guardar as informações necessárias da empresa.

Analista: O sistema ajudará em que na empresa?

Cliente: Ajudará na tomada de decisões, nos investimentos e no gerenciamento do escritório.

Analista: Quantos computadores utilizarão o sistema previsto?

Cliente: Três computadores?

Analista: Quantas Pessoas farão uso do sistema?

Cliente: Quatro Pessoas.

Analista: O que o sistema precisa para satisfazer as necessidades da empresa?

Cliente: O sistema precisa ter um cadastro de clientes, fornecedores, veículos, contas a receber e a pagar, efetuar consultas, emitir relatórios de vendas, produtos em estoque, serviços realizados, orçamentos.

2.1 - ANÁLISE DOS REQUISITOS

2.1.1 - Classificação dos Requisitos

Conforme entrevista efetuada, foram levantados os seguintes requisitos:

- Manter Cliente;

- Manter Fornecedores;

- Manter Produtos;

- Manter Ordem de Serviço;

- Manter Produtos Ordem Serviço.

2.2 - DIAGRAMAS DE CASO DE USO

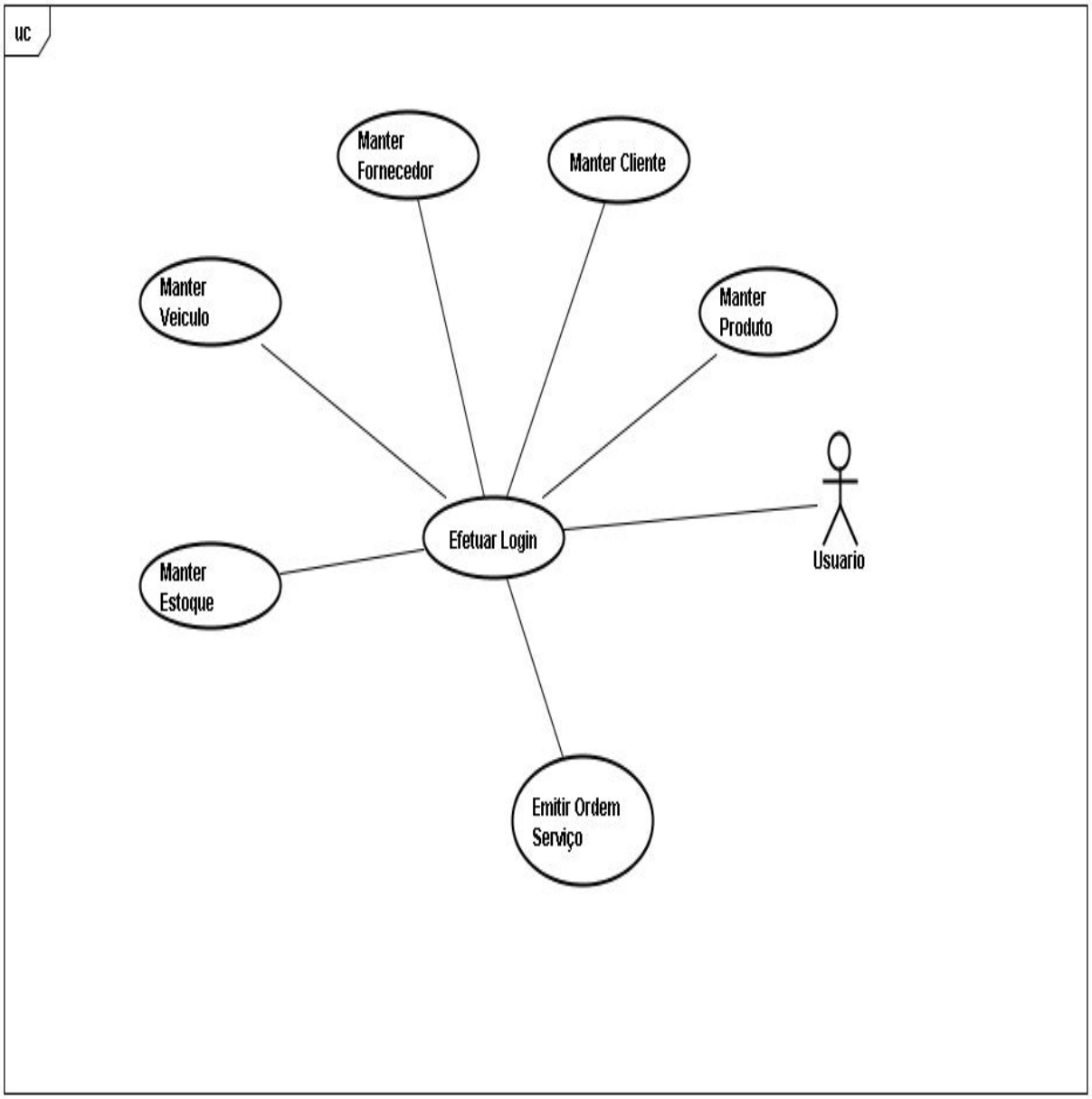

Figura03: Diagrama Caso de Uso

2.2.1 MANTER CLIENTE

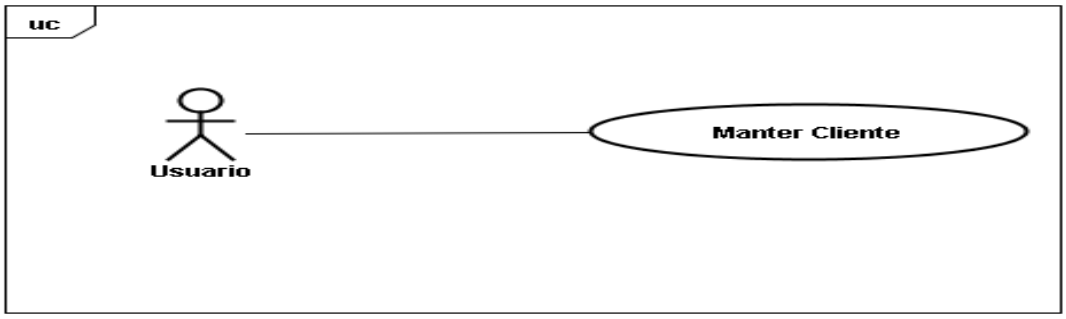

Figura04: Caso de Uso Manter Cliente.

1 – FINALIDADE

Permitir ao funcionário cadastrar clientes.

2 – ATORES

Usuário.

3 – PRÉ-CONDIÇÕES

O funcionário deve estar autenticado no sistema.

4 – EVENTO INICIAL

O usuário deverá selecionar a opção Cadastro de Cliente.

5 – FLUXO PRINCIPAL

5.1 O sistema solicita os parâmetros para o cadastro de clientes.

5.2 O usuário informa os dados do cliente. (6.1)

5.3 O sistema verifica os dados digitados pelo usuário.

5.4 O sistema retorna a mensagem que o cliente foi cadastrado com sucesso.

5.5 O caso de uso é encerrado.

6 – FLUXO ALTERNATIVO

6.1 Cancelar Operação.

6.1.1 O usuário cancela a operação.

6.1.2 O sistema sai do módulo de cadastro de cliente.

7 – PÓS-CONDIÇÕES

O usuário fecha o ambiente de cadastro de cliente.

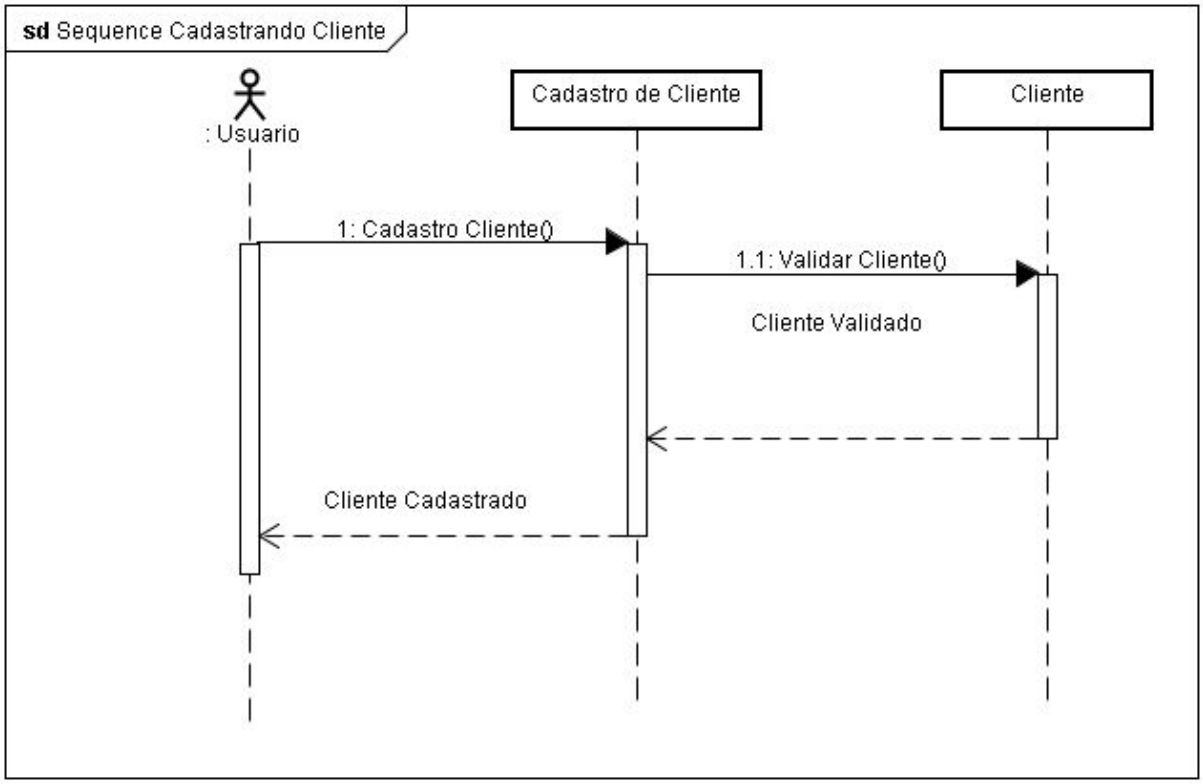

Figura05: Diagrama Sequencia Cadastrar Cliente.

2.2.2 MANTER FORNECEDOR

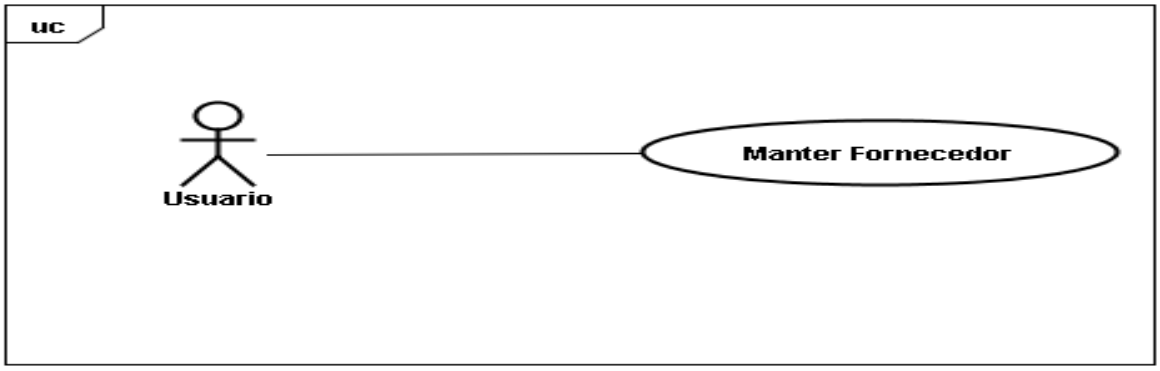

Figura06: Caso de Uso Manter Fornecedor.

1 – FINALIDADE

Permitir ao funcionário cadastrar fornecedores.

2 – ATORES

Usuário.

3 – PRÉ-CONDIÇÕES

O funcionário deve estar autenticado no sistema.

4 – EVENTO INICIAL

O usuário deverá selecionar a opção Cadastro de Fornecedor.

5 – FLUXO PRINCIPAL

5.1 O sistema solicita os parâmetros para o cadastro de fornecedor.

5.2 O usuário informa os dados do fornecedor. (6.1)

5.3 O sistema verifica os dados digitados pelo usuário.

5.4 O sistema retorna a mensagem que o fornecedor foi cadastrado com sucesso.

5.5 O caso de uso é encerrado.

6 – FLUXO ALTERNATIVO

6.1 Cancelar Operação.

6.1.1 O usuário cancela a operação.

6.1.2 O sistema sai do módulo de cadastro de fornecedor.

7 – PÓS-CONDIÇÕES

O usuário fecha o ambiente de cadastro de fornecedor.

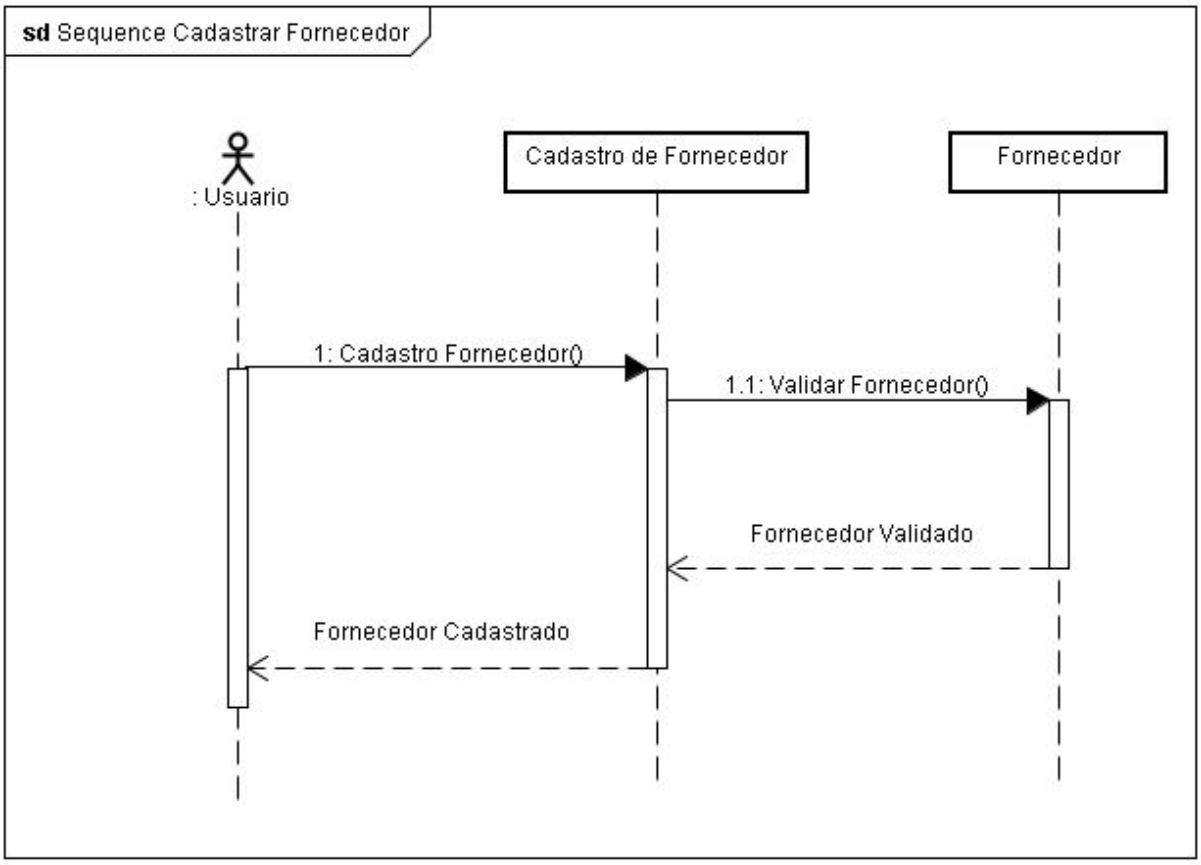

Figura07: Diagrama Sequencia Cadastro de Fornecedor.

2.2.3 MANTER PRODUTOS

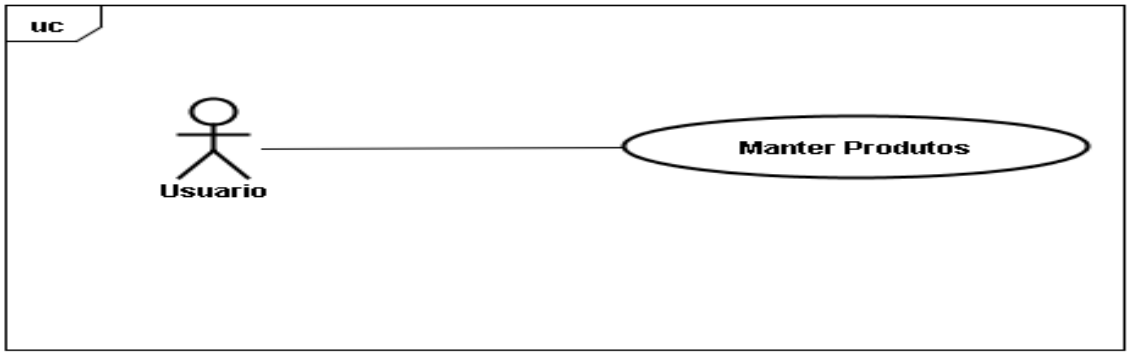

Figura08: Caso de Uso Manter Produtos.

1 – FINALIDADE

Permitir ao funcionário cadastrar Produtos.

2 – ATORES

Usuário.

3 – PRÉ-CONDIÇÕES

O funcionário deve estar autenticado no sistema.

4 – EVENTO INICIAL

O usuário deverá selecionar a opção Produtos.

5 – FLUXO PRINCIPAL

5.1 O sistema solicita os parâmetros para o cadastro de Produtos.

5.2 O usuário informa os dados do Produto. (6.1)

5.3 O sistema verifica os dados digitados pelo usuário.

5.4 O sistema retorna a mensagem que o Produto foi cadastrado com sucesso.

5.5 O caso de uso é encerrado.

6 – FLUXO ALTERNATIVO

6.1 Cancelar Operação.

6.1.1 O usuário cancela a operação.

6.1.2 O sistema sai do módulo de cadastro de Produtos.

7 – PÓS-CONDIÇÕES

O usuário fecha o ambiente de cadastro de produtos.

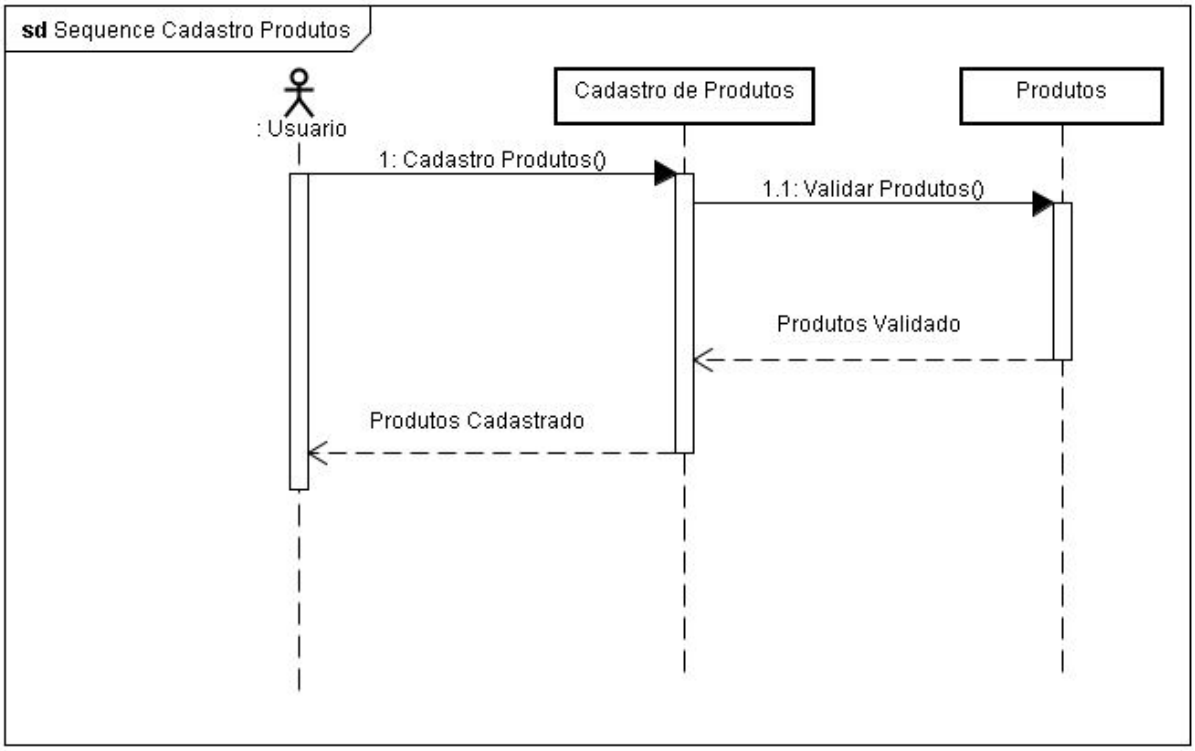

Figura09: Diagrama Sequencia Cadastrar Produtos.

2.2.4 MANTER ORDEM DE SERVIÇO

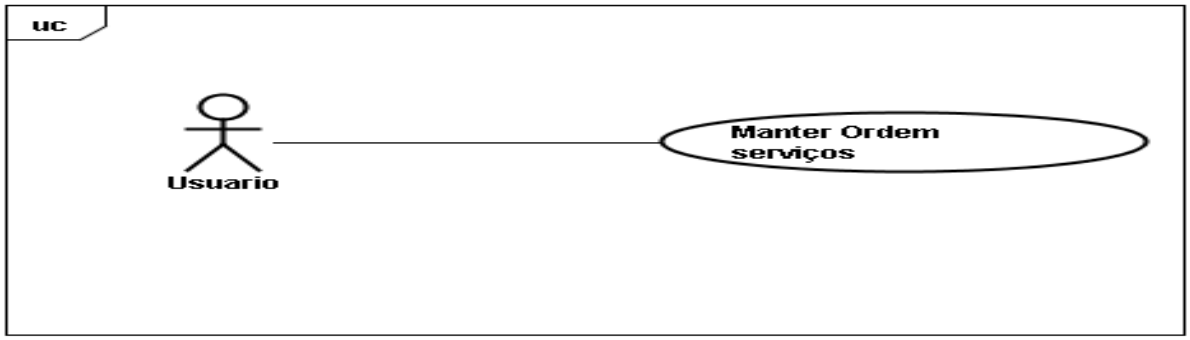

Figura10: Caso de Uso Manter Ordem de Serviço.

1 – FINALIDADE

Permitir ao funcionário poder cadastrar Ordem de Serviço.

2 – ATORES

Usuário.

3 – PRÉ-CONDIÇÕES

O funcionário deve estar autenticado no sistema.

4 – EVENTO INICIAL

O usuário deverá selecionar a opção Ordem de Serviço.

5 – FLUXO PRINCIPAL

5.1 O sistema solicita os parâmetros para o cadastro da Ordem de Serviço.

5.2 O usuário informa os dados do Cliente. (6.1)

5.3 O usuário informa os dados do veículo.

5.4 O usuário informa os dados do serviço a ser realizado.

5.5 O usuário informa os produtos.

5.6 O sistema verifica os dados digitados pelo usuário.

5.7 O sistema retorna a mensagem que a Ordem de Serviço foi cadastrada com sucesso.

5.8 O sistema retorna a mensagem se deseja imprimir a Ordem de Serviço.

5.9 O caso de uso é encerrado.

6 – FLUXO ALTERNATIVO

6.1 Cancelar Operação.

6.1.1 O usuário cancela a operação.

6.1.2 O sistema sai do módulo de cadastro de Ordem de Serviço.

7 – PÓS-CONDIÇÕES

O usuário fecha o ambiente de cadastro de Ordem de Serviço.

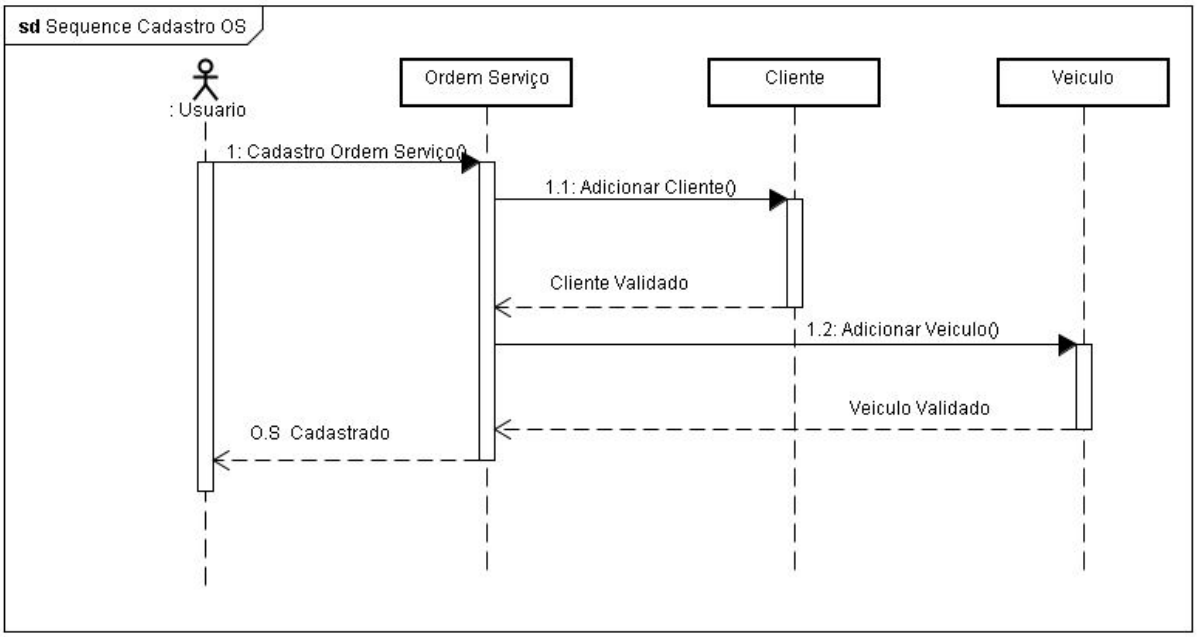

Figura11: Diagrama Sequencia Cadastrar Ordem de Serviço.

2.2.5 MANTER PRODUTO ORDEM SERVIÇO

Figura12: Caso de Uso Manter Estoque.

1 – FINALIDADE

Permitir ao funcionário Manter Produtos Ordem Serviço.

2 – ATORES

Usuário.

3 – PRÉ-CONDIÇÕES

O funcionário deve estar autenticado no sistema.

4 – EVENTO INICIAL

O usuário deverá selecionar a opção Produtos O.S.

5 – FLUXO PRINCIPAL

5.1 O sistema solicita o Numero da Ordem de Serviço.

5.2 O usuário informa os dados da Ordem Serviço. (6.1)

5.3 O usuário informa o Código do Produto.

5.4 O usuário informa a quantidade de Produtos.

5.5 O sistema verifica os dados digitados pelo usuário.

5.6 O sistema Mostra os dados em uma Tabela.

5.7 O sistema Valida os Dados.

5.8 O Caso de Uso e encerrado.

6 – FLUXO ALTERNATIVO

6.1 Cancelar Operação.

6.1.1 O usuário cancela a operação.

6.1.2 O sistema sai do módulo Produtos O.S.

7 – PÓS-CONDIÇÕES

O usuário fecha o ambiente Produtos O.S.

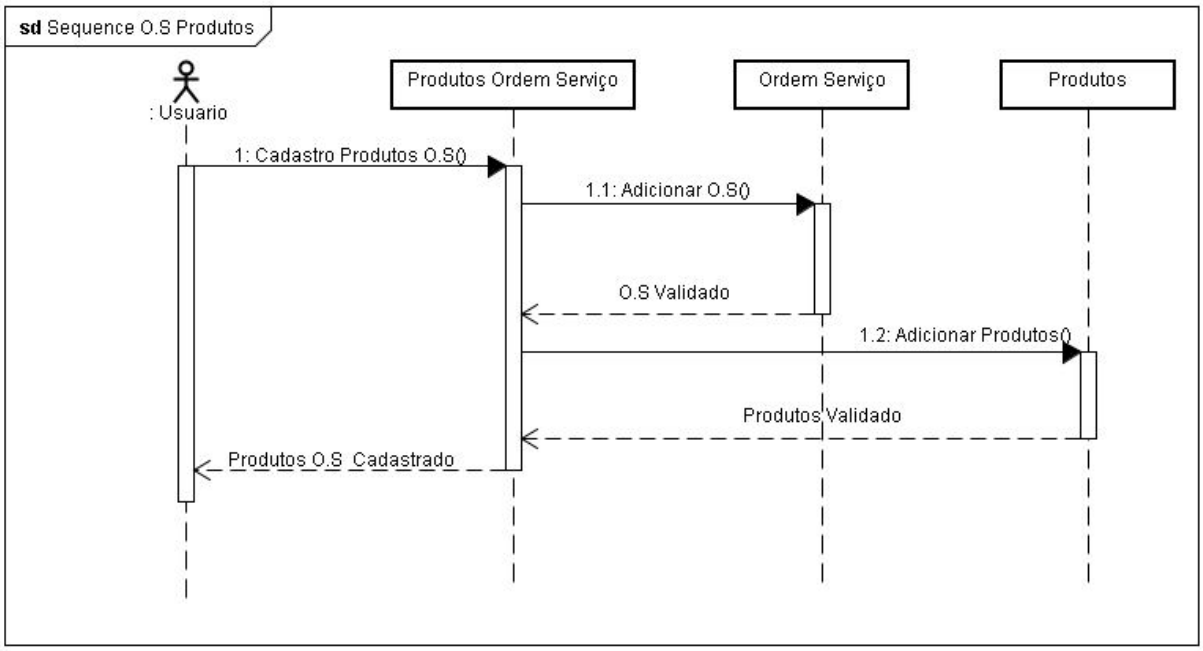

Figura13: Diagrama Sequencia Produtos Ordem Serviço.

2.3 – DIAGRAMA ENTIDADE RELACIONAMENTO

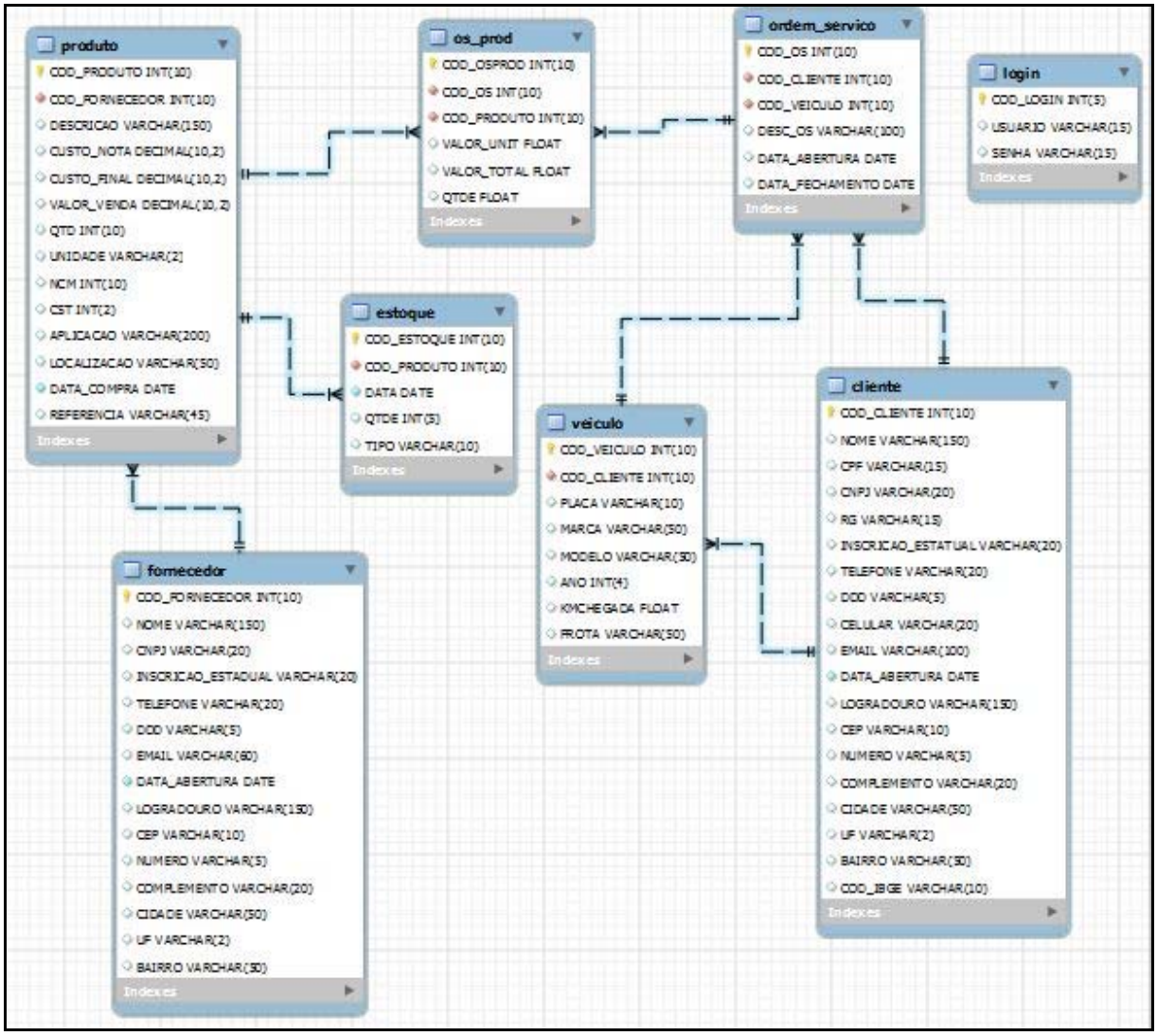

Figura:14 Diagrama Entidade Relacionamento

2.4 DIAGRAMA DE CLASSE

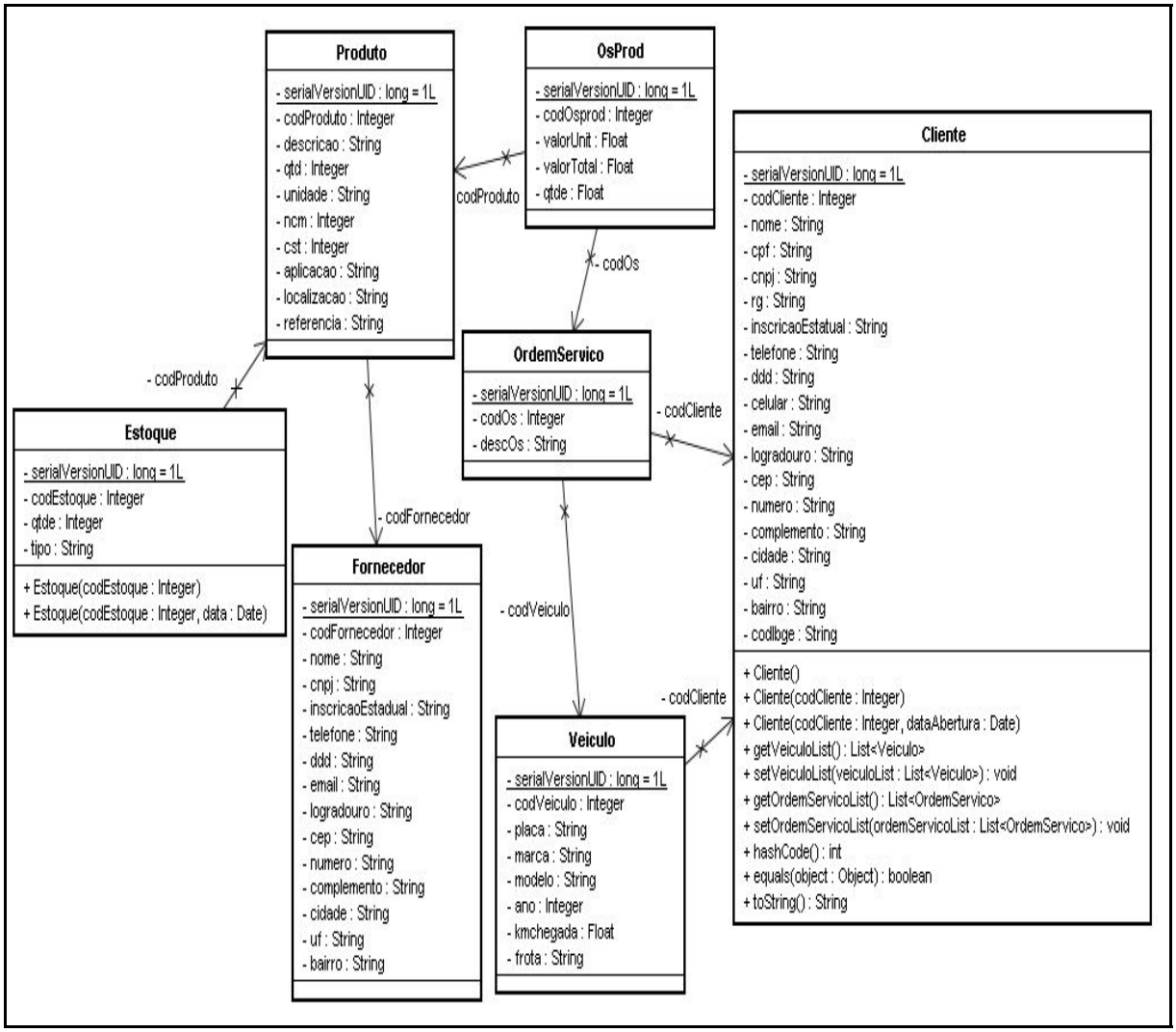

Figura15: Diagrama de Classe

3 - CRONOGRAMA

Cronograma	Ma	Abr	Mai	Ju	Jul	Ag	Set	Out	No
Pré-projeto									
Análise dos Requisitos									
Levantamento de Requisitos									
Modelagem dos requisitos									
Documentação									
Qualificação									
Desenvolvimento do Sistema									
Fase de Teste									
Conclusão									

Tabela 01: Cronograma de Desenvolvimento

4 - ESPECIFICAÇÃO DOS CUSTOS

Recursos Necessários ao Desenvolvimento

Serão necessários para o desenvolvimento do Projeto, os recursos descritos abaixo:

-Recursos Físicos

Número de Pessoas: 01 Analista/Programador;

Equipamentos:

- 01 Microcomputador;
- 01 Impressora Jato de Tinta;

-Software:

- Linguagem: IDE Eclipse e Java (Freeware);
- Modelagem: ASTAH Professional;
- Banco de Dados: Oracle 10g Express (Freeware);
- Ferramenta de Manutenção do Banco: MYSQL (Freeware);

Orçamento do Projeto = Estimativa de custos para as atividades + Estimativa de custos para os recursos.

Estimativa de custos para as atividades:

Prazo de dias para o desenvolvimento: a quantidade de dias foi estipulada no cronograma.

Pessoal:

Analista/Programador	**Quantidade de Dias**		**Custo/dia (R$)**	**Total (R$)**
Gabriel Vieira Galli	Análise	110	25,00	2.750,00
	Desenvolvimento	160	25,00	4.000,00
Total Custo Pessoal				**6.750,00**

Tabela 02: Custos do Programador

Equipamento

- **01 computador**

  Valor unitário = R$1.500,00

  Depreciação (2 anos) = R$1.500,00 / 24 = R$ 62,50/mês

  Custo por dia = R$62,50 / 26 (dias) = R$2,40 (ao dia)

  Custo do computador = R$2,40 * 270 = R$648,30.

- **01 impressora**

  Valor = R$300,00

  Depreciação = R$300,00 / 24 = R$12,50

  Custo dia = R$12,50 / 26 = R$0,49

  Custo impressora = R$0,49 * 270 = R$132,30

- **Custo Total Equipamento = R$648,30 + R$132,30 = R$780,60**

Software

ASTAH Professional = R$150,00

Total = R$150,00

- Depreciação = R$150,00 / 24 meses = R$6,25 mês
- Custo por dia = R$ 6,25 / 26 dias = R$0,25
- **Custo do Software = R$0,25 * 270 = R$67,50**

Custo Total do Projeto = R$6.750,00 + R$780,80 + R$67,50 = **R$7.598,30**

5 - CONCLUSÃO

A proposta do trabalho foi de desenvolver um sistema de gestão para autopeças e oficina com o objetivo de informatizar a empresa, eliminando o montante de papel que era gerado, obtendo um maior número de informações sobre os clientes, produtos e ordens de serviços, e um resumo das atividades da empresa. Foi utilizado frameworks como JSF para o desenvolvimento e criação de interfaces do usuário, assim como o JPA para a padronização e mapeamento dos objetos das classes Java.

O sistema proporciona diversas atividades para facilitar as atividades do dia-a-dia da empresa permitindo um maior controle das informações sobre os dados cadastrais dos clientes, produtos, fornecedores e ordens de serviços armazenando um histórico dessas informações.

REFERÊNCIAS

CAELUM. **FJ11 Java e Orientação Objetos.** Disponível em <http://www.caelum.com.br/apostilas/>. Acesso em 15/05/2013.

CAELUM. **FJ21 Java para Desenvolvimento Web.** Disponível em <http://www.caelum.com.br/apostilas/>. Acesso em 01/06/2013.

FERRARI, Fabricio Augusto, **Crie banco de dados em MySQL** São Paulo: Digerati Books, 2007.

GOMES, Yuri Marx P. **Java na Web com JSF, Spring, Hibernate e Netbeans6** Rio de Janeiro: Editora Ciência Moderna Ltda, 2008.

GUEDES, Gilleanes T. A, **UML2: uma abordagem prática** São Paulo: Novatec Editora, 2011.

KATHY Sierra, Bert Bates, **Use a Cabeça Java 2ª Edição**, Kathy Sierra, Bert Bates, Alta Books, 2005.

MACHADO, Delipe Nery Rodrigues, **Projeto de Banco de Dados: uma visão prática** Felipe Nery Rodrigues Machado, Mauricio Pereira de Abreu, São Paulo: Erica, 1996.

MYSQL Ab, **Manual de Referências do MySQL,** 1997-2003.

SANTOS, Fabiano Gonçalves**, Fundamentos da Linguagem Java,** 2002-2003.