

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
Tecnologia em Análise e Desenvolvimento de Sistemas

AITON LUIS DEMITO

Sistema Informatizado Aplicado na Produção e Manutenção de Equipamentos

TRABALHO DE DIPLOMAÇÃO

CORNÉLIO PROCÓPIO
2013

AITON LUIS DEMITO

Sistema Informatizado Aplicado na Produção e Manutenção de Equipamentos

Trabalho de Conclusão de Curso de Graduação, apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná - UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Fabrício Martins Lopes

CORNÉLIO PROCÓPIO
2013

Aos meus pais, Mauro e Hilda, irmãos Anderson, Angélica e namorada Ana Carolina, por estarem do meu lado sempre, além do incentivo constante.

Agradecimentos

Agradeço primeiramente aos meus pais, pelo constante incentivo desde minha criação até minha formação acadêmica e meu avô José Trevisan(*in memoriam*) pelo exemplo de vida a ser seguido.

À minha namorada Ana Carolina pelo companheirismo, paciência, carinho e ânimo nos momentos difíceis.

Agradeço também ao Prof. Dr. Fabrício Martins Lopes pela orientação deste trabalho; ao analista Guilherme Luiz Frufrek e Sônia Maria Rodrigues por me acolherem como estagiário na COGETI e Grupo de Dança respectivamente, e a todo o setor administrativo pelo aprendizado adquirido durante dois anos e oito meses de convívio diário.

Gostaria de agradecer, também, a todo o corpo docente do curso Tecnologia em Análise e Desenvolvimento de Sistemas da UTFPR, campus Cornélio Procópio, que me proporcionaram uma formação de qualidade.

Sumário

Lista de Figuras	iv
Lista de Tabelas	v
Lista de Siglas	vi
Resumo	viii
Abstract	ix
1 Introdução	10
1.1 A EMPRESA	11
1.2 ORGANIZAÇÃO DO TRABALHO	11
2 JUSTIFICATIVA	13
2.1 FLUXO OPERACIONAL ATUAL	13
2.2 FLUXO OPERACIONAL PROPOSTO	14
3 OBJETIVOS	16
3.1 VISÃO DA SOLUÇÃO - OBJETIVOS GERAIS	16
3.2 ESCOPO DA SOLUÇÃO - OBJETIVOS ESPECÍFICOS	16
4 MATERIAIS E MÉTODOS	19
4.1 TECNOLOGIAS	19
4.1.1 Cascading Style Sheets	19
4.1.2 Enterprise Java Beans	19
4.1.3 Hibernate e JPA	19
4.1.4 Java EE	20
4.1.5 JavaServer Faces	21
4.1.6 Primefaces	21
4.1.7 HTML	21

4.2	RECURSOS DE SOFTWARE	21
4.2.1	Dia	21
4.2.2	Eclipse IDE	21
4.2.3	Glassfish <i>Open Source Edition</i>	22
4.2.4	PostgreSQL, versão 9.0	22
4.2.5	Subversion	22
4.2.6	Ubuntu, versão 10.10	22
4.2.7	Umbrello 1.5.71	23
4.3	RECURSOS DE HARDWARE	23
4.4	METODOLOGIA E DESENVOLVIMENTO	24
4.4.1	Introdução às Metodologias Ágeis	24
4.4.2	Metodologia Adotada	25
4.4.3	Especificação	26
4.4.4	Projeto	27
4.4.5	Implementação	28
4.4.6	Testes	28
5	DESCRIÇÃO DO SISTEMA	32
5.1	DESCRIÇÃO DOS USUÁRIOS DO SISTEMA	32
5.2	LIMITES E RESTRIÇÕES DO SISTEMA	32
5.3	ANÁLISE DE PONTOS POR FUNÇÃO	33
5.4	CRONOGRAMA	35
5.4.1	Cronograma Proposto	35
5.4.2	Cronograma Oficial	35
6	CONCLUSÃO	36
6.1	PROBLEMAS ENCONTRADOS	36
6.2	TRABALHOS FUTUROS	37
	Referências	38
	Índice Remissivo	40
	Apêndice A - Casos de Uso	41

Lista de Figuras

2.1	Fluxo operacional atual	14
2.2	Fluxo operacional proposto	15
4.1	Estrutura do Desenvolvimento Guiado por Funcionalidades	25
4.2	Diagrama Entidade Relacionamento	27
4.3	Cadastro de Clientes	29
4.4	Listagem de Clientes	30
4.5	Edição de Cliente	31
A.1	Caso de Uso Geral	41

Lista de Tabelas

3.1	Funcionalidades	17
5.1	Qualificação de Complexidades	34
5.2	Cronograma Proposto	35
5.3	Cronograma Oficial	35
A.1	Caso de Uso - Manter Operadores	42
A.2	Caso de Uso - Manter Relacionamentos Externos	42
A.3	Caso de Uso - Manter Clientes	43
A.4	Caso de Uso - Manter Unidades	44
A.5	Caso de Uso - Manter Máquinas	44
A.6	Caso de Uso - Abertura de Ocorrência	45
A.7	Caso de Uso - Histórico de Manutenções	46
A.8	Caso de Uso - Emitir Relatórios	46

Lista de Siglas

A4	A4 é um tamanho de papel, definido pela norma ISO 216, com as dimensões de 210 mm de largura e 297 mm de altura.
API	Interface de Programação de Aplicativos (<i>Application Programming Interface</i>)
CDDL	Licença para código aberto que parte da <i>Mozilla Public License</i> (MPL) e a torna reutilizável sem modificações (<i>Common Development and Distribution License</i>)
CLP	Controlador Lógico Programável
CNC	Controle numérico Computadorizado (<i>Computer Numeric Control</i>)
COGETI	Coordenação de Tecnologia da Informação
CSS	Linguagem de apresentação ou folhas de estilo (<i>Cascading Style Sheets</i>)
CVS	Sistema de versionamento (<i>Concurrent Versions System</i>)
DAO	Padrão para persistência de dados que permite separar regras de negócio das regras de acesso a banco de dados (<i>Data Access Object</i>)
JEE	Java Edicao Empresarial (<i>Enterprise Edition</i>)
EJB	<i>Container</i> do servidor de aplicacoes Java EE (<i>Enterprise JavaBeans</i>)
FDD	Desenvolvimento Guiado por Funcionalidades (<i>Feature Driven Development</i>)
FUNC	Funcionalidade
GB	Unidade de medida de dados (<i>Gigabytes</i>)
GPL	Licença para software livre (<i>General Public License</i>)
HD	Dispositivo de armazenamento de dados (<i>Hard Disk</i>)
IBGE	Instituto Brasileiro de Geografia e Estatística
IDE	Ambiente de Desenvolvimento Integrado (<i>Integrated Development Environment</i>)
JDBC	Responsável por conexões com bancos SQL (<i>Java Database Connectivity</i>)
JDK	Kit de Desenvolvimento Java (<i>Java Development Kit</i>)
JPA	Conjunto de bibliotecas para persistência de dados em java (<i>Java Persistence API</i>)

JRE	Ambiente de Tempo de Execução Java (<i>Java Runtime Enviroment</i>)
JVM	Máquina Virtual Java (<i>Java Virtual Machine</i>)
JSP	Tecnologia de desenvolvimento web em java (<i>JavaServer Pages</i>)
Ltda	Sociedade limitada
PDF	Formato de Documento Portável (<i>Portable Document Format</i>)
PPF	Pontos Por Função
MVC	Desenvolvimento em camadas (<i>Model, View, Controller</i>)
RAM	Memória de acesso aleatório (<i>Random-access memory</i>)
SE	<i>Standard Edition</i>
S.O.	Sistema Operacional
SVN	Sistema de controle de versão Apache Subversion
TD	Trabalho de Diplomação
UML	Linguagem de Modelagem Unificada (<i>Unified Modeling Language</i>)
XHTML	Reformulação da linguagem HTML (<i>eXtensible Hypertext Markup Language</i>)

Resumo

Este trabalho apresenta o desenvolvimento do processo de informatização dos atendimentos na manutenção dos equipamentos da empresa Coolseed. A empresa Coolseed atua na área industrial produzindo equipamentos para pós-colheita de grãos. Nesse contexto, o aplicativo trouxe diversas melhorias como agilizar o atendimento a clientes, vendedores e técnicos, iniciando o atendimento de pós venda já com os dados e relatórios de atendimentos anteriores prestados aos clientes. O desenvolvimento do aplicativo foi iniciado pela modelagem do sistema, utilizando o método UML e sua codificação contou com a tecnologia Java e a persistência dos dados baseada no banco de dados objeto-relacional PostgreSQL. O estudo do fluxo de informações dentro da empresa, sua modelagem e codificação contribuíram para o aprendizado do aluno, e conseqüentemente, levaram a construção do aplicativo de controle. Pode-se concluir que o desenvolvimento do aplicativo teve resultados muito positivos para o estudante, contribuindo decisivamente para sua formação e também para a empresa, que passou a gerenciar com mais eficiência e precisão os chamados de manutenção das máquinas. Além disso, o uso do aplicativo foi fundamental para descobrir correlações entre as chamadas de manutenção e outros eventos como a queda de energia elétrica, aquecimento das máquinas, mal uso, entre outras através do registro de chamados e relatórios precisos sobre a manutenção dos equipamentos.

Palavras-chave: atendimentos, controle; decisões; desenvolvimento; modelagem; sistema.

Abstract

This paper presents the development process of computerising the equipment care maintenance of the Coolseed Company. The company operates in the industrial sector and produces agricultural equipments for post-harvest grain. In this context, the application of the software brought several improvements to speed up customer service, salespeople and technicians, starting the post sale service with data and reports of previous services provided to the clients. The development of the software was initiated by the modeling of the system using the UML method, its codification with Java technology and the persistence of data based on the PostgreSQL object-relational database system. The study of the information flow within the company, its modeling and coding contributed to the student learning process and, consequently, led to the construction of the control application. It can be concluded that the development application had very positive results for the student, highly contributing to their professional development and also to the Company, which now manages more efficiently and accurately its maintenance calls of the machines. Furthermore, the use of software was critical to find correlations between maintenance calls and other events such as power failure, heating of the machines, misuse, among others through recording calls and accurate reports on equipment maintenance.

Key-words: calls; control; decisions; development; modeling; software.

1 Introdução

A informação ocupa hoje em dia uma posição de destaque no que diz respeito a tomada de decisões nas mais diferentes áreas. Em particular, informações precisas e recuperadas de forma eficiente são essenciais para os mais diversos fins, em particular na produção e manutenção de equipamentos. Este trabalho aborda essa questão em parceria com a empresa Coolseed Indústria e Comércio de Equipamentos de Aeração Condicionada Ltda.

Atualmente a empresa produz equipamentos de aeração, produzindo desde os cortes e dobras das placas de metal através do uso de equipamentos de Controle Numérico Computadorizado até os painéis de Controlador Lógico Programável que controlam digitalmente os componentes elétricos dos equipamentos.

Quando a empresa negocia um produto com um cliente, possui em sua filosofia a política de garantia de assistência gratuita durante o primeiro ano e sem custos para assegurar o bom funcionamento do equipamento e a satisfação do cliente. Nesse contexto, uma das dificuldades enfrentadas pela empresa é que com o crescimento do parque de máquinas, torna-se difícil acompanhar os problemas de campo e as soluções implementadas. Também, pela falta de um sistema que permita registrar esses eventos perde-se a oportunidade de identificar problemas de fabricação ou pontos fracos destes equipamentos, assim como de abrir o histórico de cada um destes equipamentos no momento da chamada de um cliente. A imagem do que normalmente acontece durante uma consulta médica quando o médico tem todo o histórico do paciente. Atualmente este controle se faz de forma manual, pelo preenchimento por parte dos técnicos, de planilhas, em folhas tipo A4 que são arquivadas em pastas normais.

De acordo com o (IBGE, 2012) , em 2010, o segmento industrial que possuía de 50 a 499 colaboradores, 98,7% delas possuíam computador, e 98,1% destas, faziam uso da internet. Com base nestes dados, a conversão de sistemas manuais para informatizados é justificado por si só, não somente para o ramo industrial, mas também o comercial, que apresenta os maiores índices de uso de internet por computador.

1.1 A EMPRESA

A armazenagem de grãos sempre foi um desafio, não há registros do período em que o homem começou a armazenar seus alimentos, mas no Brasil, nos anos de 40 a 60 iniciaram a construção de armazéns convencionais de fundo plano, onde os grãos eram armazenados em sacarias, especialmente café e arroz. A partir da década de 60 e 70, começou a surgir a granelização, onde os grãos eram armazenados em silos graneleiros com fundo plano ou em “V” e contavam com equipamentos de transporte a granel para carga e descarga e a secagem contínua dos grãos. Pois grãos secos e frios mantêm melhor a qualidade original do produto, tendo o teor de umidade como o fator mais importante no controle do processo de deterioração de grãos armazenados. Se a umidade puder ser mantida a níveis baixos, os demais fatores terão seus efeitos diminuídos. Condições de armazenamento que promovem um aumento da intensidade da respiração dos grãos são prejudiciais porque produzem mudanças nas suas propriedades físicas e químicas que os tornam inúteis para o consumo “*in natura*” ou processamento industrial.

Pensando nisso, no ano de 1999 uma equipe de técnicos liderada pelo engenheiro Francisco Ayala Barreto, desenvolveu um equipamento robusto para reduzir a temperatura e umidade de grãos de soja armazenados em silos. Superadas as dificuldades iniciais, em 2001 nasce a Coolseed Ind. e Com. Ltda. primeiramente em Foz do Iguaçu - Paraná e mais tarde, em 2007 transferiu seu parque tecnológico para Santa Tereza do Oeste - Paraná, onde atualmente encontra-se em operação.

Desde sua elaboração, a Cool Seed está em sintonia com o meio ambiente, suas tecnologias evitam o uso de praguicidas e reduzem a queima de combustíveis fósseis ou lenha, utilizando apenas energia elétrica, fonte renovável e limpa.

Além de diminuir o armazenamento a base de praguicidas, seus equipamentos oferecem a qualidade de alimentos mais limpos e seguros para nutrição tanto animal quanto humana.

A empresa hoje conta com mais de 300 equipamentos distribuídos entre 8 países da América Latina e América Central, apoia projetos de várias áreas tecnológicas e consolidou-se como referência de armazenagem limpa de grãos.

1.2 ORGANIZAÇÃO DO TRABALHO

Neste trabalho são descritas todas as etapas de elaboração e desenvolvimento do sistema apresentado como trabalho de conclusão de curso para Tecnólogo

em Análise e Desenvolvimento de Sistemas.

A partir do Capítulo 2 é apresentada a justificativa da escolha do tema e no Capítulo 3 os objetivos a serem alcançados com o desenvolvimento do sistema. No quarto capítulo são descritos os materiais e métodos utilizados para a implementação dos requisitos levantados no terceiro. Já na quinta parte do trabalho, são descritos os usuários, limites e restrições do sistema, além da contagem de pontos por função e metodologia de desenvolvimento adotadas. E por fim a conclusão, que descreve a experiência do aluno com o desenvolvimento do trabalho, tais como problemas encontrados e os trabalhos futuros a serem realizados para um melhor aproveitamento do sistema.

2 JUSTIFICATIVA

A informatização de processos é sem dúvida essencial a qualquer sistema, seja ele já existente ou que poderá ser desenvolvido juntamente com a informatização.

Neste caso, o sistema já é existente e operante, porém de maneira manual e não informatizada, fazendo com que qualquer tipo de relatório para tomada de decisões seja um processo trabalhoso e lento. Tratando-se de uma grande quantidade de clientes, o atual processo torna o registro dos atendimentos e a busca por atendimentos anteriores uma tarefa cansativa e suscetível a erros.

A proposta é informatizar e melhorar os processos no setor, com o acesso web o sistema ficará disponível a partir de qualquer computador que esteja ligado à rede local, podendo ser estendido o acesso também à rede externa posteriormente, conforme a necessidade.

A informatização trará à equipe de pós-venda agilidade na revisão de máquinas no momento em que estiverem em contato com o cliente, facilitando assim o diagnóstico de possíveis causas para o atual problema com o equipamento operante em campo. Os benefícios trazidos a equipe de engenharia serão ainda maiores, com base nos relatórios rapidamente obtidos pela equipe de pós-venda, terão acesso a dados para auxiliar na tomada de decisões em nível de projeto das máquinas.

2.1 FLUXO OPERACIONAL ATUAL

O fluxo operacional atual consiste em receber a ligação do cliente, vendedor ou técnico responsável e registrar a situação em formulários A4 que após análise e conclusão da situação, serão armazenados nos fichários correspondentes a cada cliente. Este processo foi estudado para ser melhorado com a implantação de uma ferramenta informatizada para eliminação das fichas manuais e maior agilidade na identificação do problema.

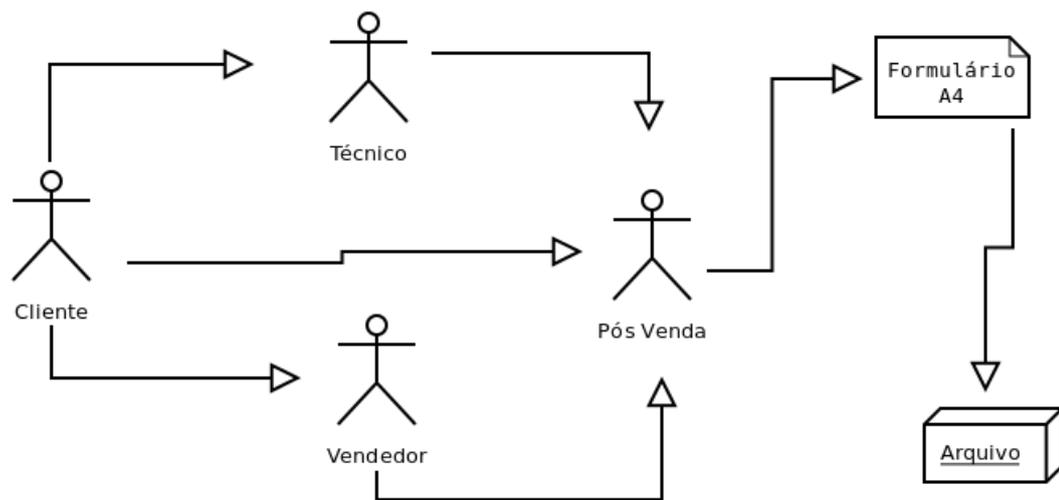


Figura 2.1: Fluxo operacional atual

2.2 FLUXO OPERACIONAL PROPOSTO

O fluxo operacional proposto consiste em atividades semelhantes às já utilizadas e mais os melhoramentos propostos, analisando o histórico de atendimentos ao cliente já no momento da ligação telefônica e registrando os dados da ligação já no banco de dados, através de chamados, que mais tarde podem ser finalizados ou adicionados trâmites, visto que nem todas as situações são resolvidas em apenas uma ligação.

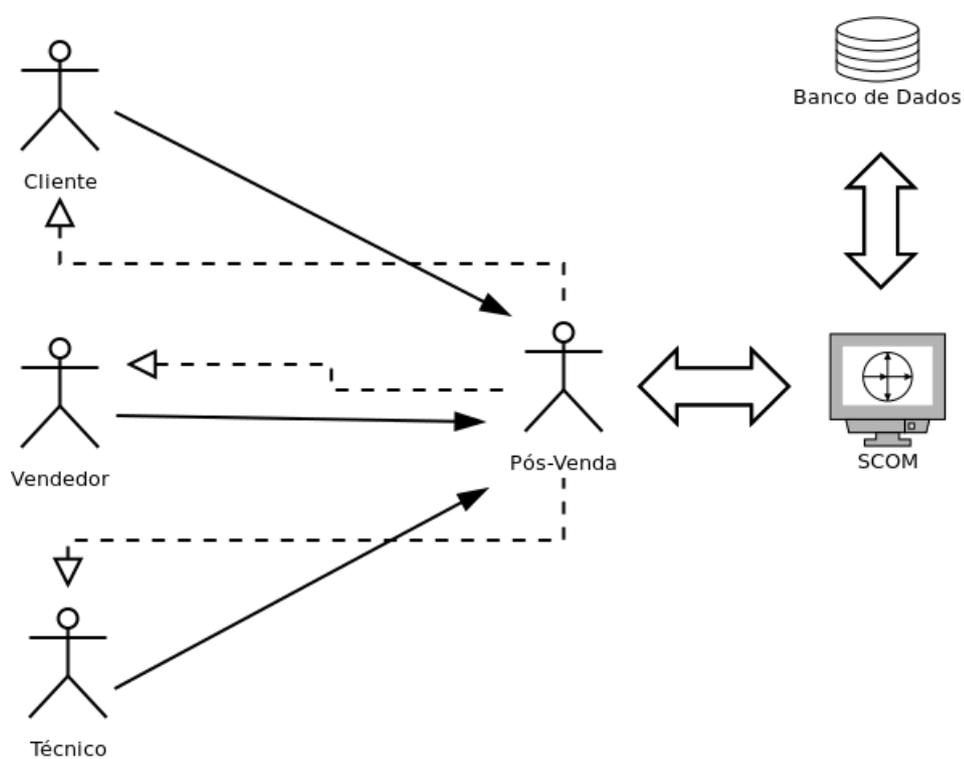


Figura 2.2: Fluxo operacional proposto

3 OBJETIVOS

O presente trabalho tem como objetivo colocar em prática as habilidades adquiridas ao decorrer do curso superior de Tecnologia em Análise e Desenvolvimento de Sistemas para solucionar situações do mercado de trabalho como a retratada aqui.

Com o desenvolvimento de uma solução personalizada, iniciada e finalizada seguindo metodologias ensinadas durante a graduação, o aluno aprofunda o conhecimento e desenvolve habilidades necessárias para entrada definitiva no mercado corporativo ou institucional.

3.1 VISÃO DA SOLUÇÃO - OBJETIVOS GERAIS

A empresa necessita de um sistema informatizado que permita ao gerente de pós-venda acessar em tempo real todo o histórico de uma máquina no momento da chamada de algum cliente ou na liberação da equipe para assistência em campo. Este sistema deve arquivar todas as informações relacionadas ao requerido equipamento, como número de série, versão do programa do CLP e alguma outra informação importante sobre a última atualização tecnológica do equipamento. Este sistema deve ter condições de emissão de relatórios por parte da pós-venda para que o controle de qualidade possa interferir na engenharia ou nos processos e assim evitar a repetição dos problemas.

3.2 ESCOPO DA SOLUÇÃO - OBJETIVOS ESPECÍFICOS

Com base no estudo de caso sobre a rotina de trabalho do setor de pós-venda da empresa, foram levantados requisitos essenciais, importantes e desejáveis que deveriam ser informatizados. Para que a rotina pudesse ser completamente informatizada, seria necessário registrar em bases de dados o cadastro dos clientes e suas respectivas máquinas operantes em campo, assim como os vendedores e os técnicos responsáveis pelo cliente. Com os dados iniciais populados, já seria possível registrar

os atendimentos e suas soluções, até sua conclusão total. Para mais tarde, gerar relatórios que pudessem auxiliar a equipe de engenharia na tomada de decisões. Segue abaixo um quadro com as necessidades levantadas:

Tabela 3.1: Funcionalidades

	Necessidade	Categoria
1	Manter Clientes	Essencial
2	Manter Modelos de Máquinas	Essencial
3	Manter Vendedores	Essencial
4	Manter Técnicos	Essencial
5	Abrir Chamado para Manutenção	Essencial
6	Editar Chamado para Manutenção	Essencial
7	Concluir Manutenção	Essencial
8	Gerar Relatórios de Manutenção	Importante
9	Imprimir Relatórios e Manutenções	Desejável
10	Manter Usuários do Sistema	Importante

Com base na Tabela 3.1, temos como objetivos específicos as necessidades citadas acima, que consistem em ter um cadastro de clientes, no qual serão armazenadas as seguintes informações: nome fantasia da empresa cliente, razão social, endereço do escritório, telefone, numero de serie da máquina e versão do CLP e um número identificador gerado automaticamente. O cadastro de máquinas armazena informações como modelo e um numero identificador gerado automaticamente.

O cadastro de técnicos, descrito como quinta funcionalidade do sistema, tem por objetivo manter um cadastro de todos os técnicos da empresa Coolseed que atuam em campo para identificar, relatar e solucionar problemas, tendo como dados de cadastro no sistema nome completo, CPF e o identificador único.

Os vendedores são os responsáveis por atender os clientes, cada vendedor tem sua região e público específico.

A etapa de manutenção engloba as necessidades 5, 6 e 7, que são necessárias pela abertura do chamado, onde são identificados o cliente e a localização da máquina, na abertura do chamado é possível realizar uma busca de manutenções anteriores que a máquina foi submetida, contendo data de manutenção, descrição do problema, resolução do problema e se estava ou não no período de garantia, que é de um ano a partir da venda da máquina.

A emissão de relatórios têm como objetivo principal auxiliar a equipe de engenharia no desenvolvimento e aperfeiçoamento das máquinas, gerando relatórios precisos que indicam quais os equipamentos apresentam maior reincidência de chamados, ou problemas de operação das máquinas, informação que passa a ser importante

também a equipe de treinamento e capacitação de operadores das máquinas.

Os relatórios poderão ser impressos fisicamente ou para arquivos .pdf para envio a equipes externas ou colaboradores que estarão atuando em campo.

4 MATERIAIS E MÉTODOS

4.1 TECNOLOGIAS

4.1.1 Cascading Style Sheets

Cascading Style Sheets (CSS) são folhas de estilo utilizadas para atribuir cor e forma a páginas web (html). Tem seu uso altamente recomendado (praticamente obrigatório) por separar a formatação do conteúdo do código da aplicação, melhorando a legibilidade e organização dos códigos. Com o aparecimento de uma grande variedade de navegadores de 2001 a 2006, os desenvolvedores diminuíram o ritmo de atualizações, até que todos os navegadores se modernizassem. Hoje encontra-se na versão 3 (CSS3) (SILVA, 2007).

4.1.2 Enterprise Java Beans

Enterprise JavaBeans (EJB) é um componente executado no *container* do servidor de aplicação, neste caso o seu objetivo é simplificar o desenvolvimento de aplicações Java altamente distribuídas e portáteis (MANZONI, 1997). Neste projeto, seu papel é elevar o nível de aprendizado do aluno.

4.1.3 Hibernate e JPA

O Hibernate é um *framework* de persistência de objetos, desenvolvido em Java e baseado no mapeamento objeto/relacional (DOC2004,). A especificação JPA *Java Persistence API* foi criada com o objetivo de padronizar as ferramentas ORM e diminuir a complexidade de desenvolvimento.

O Hibernate é dividido em três camadas:

- gerencia as conexões com o banco de dados.
- executa uma ou mais sentenças no banco de dados.

- São geradas as *queries* apropriadas para execução de operações como inserção, atualização, remoção e recuperação dos dados.

O JPA possui especificações de classes e métodos que, no nosso caso, o Hibernate deve implementar; ou seja, o JPA não implementa nenhum código, apenas especifica como fazer. Isto significa dizer que, qualquer outra ferramenta de implementação, como por exemplo EclipseLink, OpenJPA, TopLink, entre outros, poderá ser utilizado, justamente porque a JPA especifica como estas ferramentas devem trabalhar.

Para (KEITH MIKE; SCHINCARIOL, 2008), a introdução do *Java Persistence API* (JPA) foi a maneira de preencher a lacuna entre os modelos orientados a objetos e os sistemas de banco de dados relacionais.

4.1.4 Java EE

A linguagem Java, lançada em 1995, hoje é uma das principais linguagens de desenvolvimento, seja para web, *desktop* e dispositivos portáteis. Mantida atualmente pela Oracle, possui grande variedade de IDEs (*Integrated Development Environment*) e colaboradores ao redor do mundo. Para os desenvolvedores é necessária a instalação da JDK (*Java Development Kit*), já para os clientes é necessária apenas a JVM (*Java Virtual Machine*) e a JRE (*Java Runtime Environment*) (LEMAY LAURA; PERKINS, 1997).

O JEE (Java Enterprise Edition) é a plataforma Java voltada para redes, internet e semelhantes que contém bibliotecas desenvolvidas para o acesso a servidores, a banco de dados, entre outras características. Graças a essas características, o JEE foi desenvolvido para suportar uma grande quantidade de usuários simultâneos.

A plataforma JEE contém uma série de especificações, cada uma com suas funcionalidades distintas, a saber:

- JDBC (*Java Database Connectivity*), utilizado no acesso e conexão ao banco de dados;
- JSP (*Java Server Pages*), uma espécie de servidor Web (Servidores Web são as aplicações que permitem a você acessar um site na internet);
- Servlets, que vem a ser o funcionamento dos servidores Web, permitindo a geração de conteúdo dinâmico nos sites.

4.1.5 JavaServer Faces

JavaServer Faces(JSF) é um *framework* MVC de aplicações web em Java que visa agilizar o desenvolvimento da aplicação. A versão utilizada neste trabalho é a JSF 2, amplamente adotada em aplicações de grande porte (GONCALVES, 2007).

4.1.6 Primefaces

Primefaces é um conjunto de bibliotecas desenvolvidas para melhorar e valorizar a apresentação de uma aplicação web através de uma vasta variedade de funções pré-definidas.

4.1.7 HTML

HTML relaciona-se diretamente com web, sem dúvida um dos termos mais lembrados considerando um ambiente web. HTML é uma linguagem de marcação para desenvolvimento de páginas web a serem acessadas via navegadores (SILVA, 2007). Formalizada em 1990 atualmente encontra-se em sua versão 5(HTML5) e continua sendo um dos pilares da internet.

4.2 RECURSOS DE SOFTWARE

4.2.1 Dia

O é uma ferramenta para geração de *layouts*, fluxogramas, modelagem UML e diagramas em geral. Desenvolvido para auxiliar os analistas e desenvolvedores de sistemas na integração dos diagramas da UML com a lógica. Além da modelagem para informática, possui recursos para projetos hidráulicos, elétricos e de redes (WILLIAMS, 2009).

4.2.2 Eclipse IDE

O Eclipse é um IDE de desenvolvimento, desenvolvido inicialmente pela IBM e posteriormente doado à comunidade de software livre. Possui uma marcante característica de desenvolvimento baseado em *plugins* que agilizam o trabalho do programador.

4.2.3 Glassfish *Open Source Edition*

O Glassfish é um servidor de aplicações Java que assim como os outros softwares utilizados neste trabalho, também é *open source*, mantido pela Oracle tem sua versão proprietária nomeada como *Oracle Glassfish Server*. Escolhido para este trabalho por seu total suporte a API Java EE e confiabilidade de execução (SAMPAIO, 2011).

Um Servidor de Aplicações é um servidor que disponibiliza um ambiente para a instalação e execução de certas aplicações, centralizando e dispensando a instalação nos computadores clientes. Os servidores de aplicação também são conhecidos por *middleware*.

4.2.4 PostgreSQL, versão 9.0

Sistema gerenciador de banco de dados objeto relacional de código aberto. Pode ser instalado em praticamente todos os sistemas operacionais conhecidos, de uso gratuito e mantido pela comunidade de software livre (POSTGRESQL.ORG, 2011).

4.2.5 Subversion

Sistema de controle de versões , lançado em agosto de 2001 em alternativa ao CVS, é um gerenciador de versões de arquivos e diretórios que mantém o histórico de alterações da estrutura dos arquivos, além de possibilitar o resgate de versões anteriores dos arquivos. Foi utilizado diariamente neste trabalho, em forma de *plugin* para o Eclipse, por permitir o desenvolvimento a partir de computadores distintos e o resgate de versões anteriores dos arquivos, além de servir também como *backup* para o código (COLLINS BEN; FITZPATRICK, 2007).

4.2.6 Ubuntu, versão 10.10

O Ubuntu é um sistema operacional baseado em GNU/Linux desenvolvido pela comunidade Ubuntu. E voltado para *notebooks*, *desktops* e servidores. Pode ser utilizado por usuários domésticos, instituições de ensino e ate mesmo em empresas. O Ubuntu é e sempre será livre, não existindo encargos de licença. É possível baixar, usar e compartilhar livremente (UBUNTU.ORG, 2007).

4.2.7 Umbrello 1.5.71

Umbrello é uma ferramenta de ambiente gráfico para documentação de sistemas. Pode ser usado para a criação da maioria dos diagramas necessários para uma documentação UML. É um software livre de código fonte aberto mantido por Gopala Krishna e sua atual versão é a 2.0 (KRISHNA, 2008).

4.3 RECURSOS DE HARDWARE

Segundo (Gonçalves, Edson, 2007) a configuração de hardware para trabalhar com aplicações escritas em Java utilizando as IDEs Eclipse e NetBeans seria no mínimo:

- Processador: Pentium 4 ou similar
- Memória: 768 MB de RAM
- HD: 10GB livres
- Monitor: 17 polegadas (devido ao arranjo das telas das IDEs)

Para o desenvolvimento deste trabalho foram utilizadas as seguintes configurações de hardware:

- Processador Intel Core 2 Duo 1.83 Gigahertz
- Memória RAM 3 Gibabytes
- Disco Rígido 500 Gigabytes
- Tela 15,3 polegadas com resolução 1280x800 pixels

Porém, conforme testes realizados durante a fase final do desenvolvimento, se o sistema for codificado com servidores distintos para aplicação e banco de dados, as configurações mínimas para desenvolvimento passam a ser extremamente modestas, deixando o código acessível em máquinas antigas ou de baixo poder de processamento.

4.4 METODOLOGIA E DESENVOLVIMENTO

4.4.1 Introdução às Metodologias Ágeis

A indústria do software sempre contou com métodos cujos processos de desenvolvimento eram baseados em um conjunto de atividades predefinidas, descritas como processos prescritivos (AMBLER, 2004), onde nas quais o trabalho começava com o levantamento completo de um conjunto de requisitos, seguido por um projeto de alto nível, de uma implementação, de uma validação e, por fim, de uma manutenção (SOMMERVILLE, 2003).

A partir da década de 90, começaram a surgir novos métodos sugerindo uma abordagem de desenvolvimento ágil. Nessa abordagem os processos adotados tentam se adaptar às mudanças, apoiando a equipe de desenvolvimento no seu trabalho (BECK, 2001). Estes novos métodos surgiram como uma reação aos métodos tradicionais de desenvolvimento de sistemas, ganhando com o passar dos anos um número cada vez maior de adeptos.

Com o intuito de definir um manifesto para encorajar melhores meios de desenvolver sistemas, em fevereiro de 2001 um grupo inicial de 17 metodologistas, entre eles Robert C. Martin, Jim Highsmith, Kent Beck e outros, formaram a Aliança para o Desenvolvimento Ágil de Software, que formularam um manifesto contendo um conjunto de princípios que definem os critérios para os processos de desenvolvimento ágil de sistemas (AMBLER, 2004). Os doze princípios (BECK, 2001), aos quais os métodos ágeis devem se adequar são:

1. A prioridade é satisfazer ao cliente através de entregas contínuas e frequentes;
2. Receber bem as mudanças de requisitos, mesmo em uma fase avançada do projeto;
3. Entregas com frequência, sempre na menor escala de tempo.
4. As equipes de negócio e de desenvolvimento devem trabalhar juntas diariamente;
5. Manter uma equipe motivada fornecendo ambiente, apoio e confiança necessários;
6. A maneira mais eficiente da informação circular através é de uma conversa face-a-face;
7. Ter o sistema funcionando é a melhor medida de progresso;
8. Processos ágeis promovem o desenvolvimento sustentável;

9. Atenção contínua à excelência técnica e a um bom projeto aumentam a agilidade;
10. Simplicidade é essencial;
11. As melhores arquiteturas, requisitos e projetos provêm de equipes organizadas;
12. Em intervalos regulares, a equipe deve refletir sobre como se tornar mais eficaz.

4.4.2 Metodologia Adotada

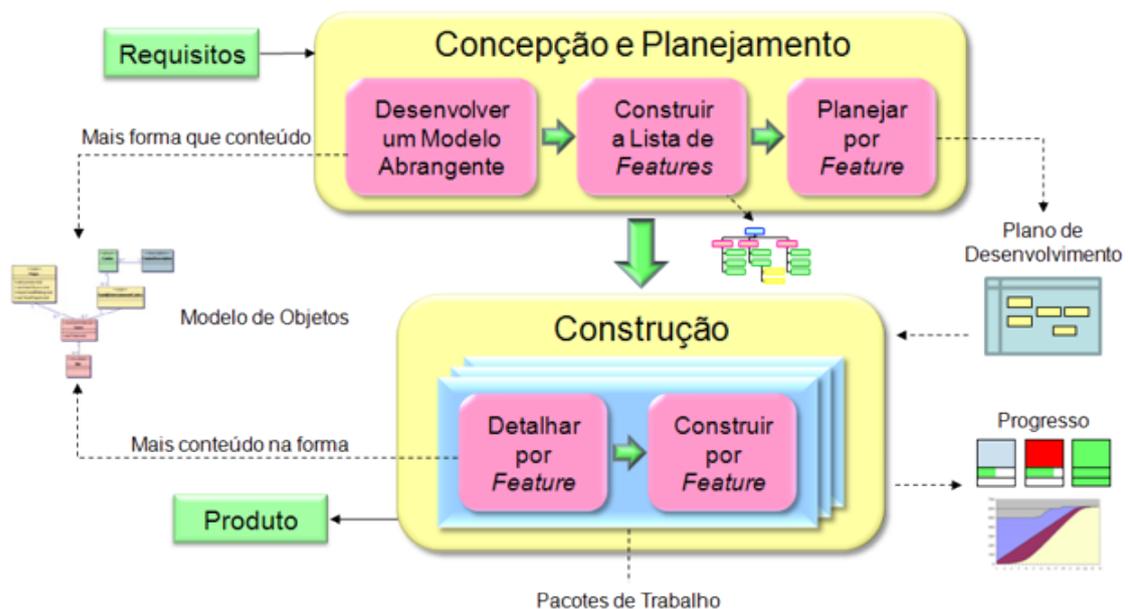


Figura 4.1: Estrutura do Desenvolvimento Guiado por Funcionalidades

Fonte: <http://www.heptagon.com.br/fdd-estrutura> Acesso em: 10 mai 2011

A metodologia adotada para o desenvolvimento do trabalho de conclusão de curso é a FDD - *Feature Driven Development*, ou Desenvolvimento Guiado por Funcionalidades. Se trata de uma metodologia ágil para desenvolvimento e gerenciamento de software. Tem por característica a combinação de boas práticas para gerenciamento ágil de projetos com uma abordagem completa para a Engenharia de Software Orientada a Objetos.

De acordo com a figura 4.1 inicia-se com a definição do esboço dos requisitos, os já conhecidos são listados, definidos e documentados através de casos de uso ou *users stories*. Sugere-se também que sejam construídos um diagrama de classes UML e diagrama(s) de sequência UML com o objetivo de auxiliar na compreensão do projeto (HIGHSMITH, 2002).

Na atribuição dos requisitos às iterações, as características (requisitos) são agrupadas e priorizadas de acordo com a sua importância e dependência dando origem aos conjuntos de características que serão desenvolvidos durante as várias iterações (HIGHSMITH, 2002).

De acordo com (HIGHSMITH, 2002), (LUCA, 2002) e (ABRAHAMSSON et al., 2002), para projetar a arquitetura do sistema, o FDD sugere que seja construído um diagrama de classes da UML para representar a arquitetura do sistema. Para complementar o diagrama de classes também poderão ser gerados diagramas de sequência da UML.

No desenvolvimento de incrementos do sistema, a implementação das classes e métodos correspondentes às características que serão desenvolvidas durante as iterações é antecedida pelas atividades de análise da documentação existente, geração de diagramas de sequência para o conjunto de características, refinamento do modelo gerado durante as atividades de definição dos requisitos e do projeto da arquitetura do sistema (HIGHSMITH, 2002).

Segundo Abrahamsson (ABRAHAMSSON et al., 2002), os testes acontecem ao final de cada iteração, onde o conjunto de características implementadas durante a mesma é testado através de testes de unidade pelos proprietários de classe - os programadores.

A integração do conjunto de características implementadas durante a iteração corrente com os outros conjuntos acontece ao final de cada iteração, após os testes e inspeções (HIGHSMITH, 2002).

E finalmente o FDD sugere que todos os conjuntos de características devem passar pelas etapas de projeto e construção até que o sistema esteja pronto para ser entregue (HIGHSMITH, 2002), (ABRAHAMSSON et al., 2002).

4.4.3 Especificação

Na fase de especificação foram realizadas reuniões com o diretor da empresa e os responsáveis pelo setor em análise para o desenvolvimento do sistema informatizado. Sendo possível assim, identificar os atores e as principais funcionalidades a serem implementadas.

Após as reuniões, foi apresentado como poderia ser o sistema, como ambiente de execução, itens de segurança da informação e uma apresentação sobre as tecnologias a serem utilizadas.

A princípio a proposta de construir um sistema *desktop* pareceu a melhor opção, pois somente os colaboradores ligados aos setores que utilizariam o sistema o

teriam instalado em suas máquinas. Porém para acompanhar a evolução tecnológica na programação e expandir o conhecimento e habilidade do aluno, a proposta foi alterada para ser desenvolvido um sistema em ambiente web, alternativa que agradou a todos por não haver a necessidade de instalação do sistema nos computadores e, principalmente o acesso de fora da empresa, podendo este ser feito a partir de computadores de mesa, *notebooks*, celulares ou *tablets*.

4.4.4 Projeto

De acordo com a documentação registrada na especificação, foram gerados os artefatos que descrevem as funcionalidades, como casos de uso e diagrama entidade-relacionamento, para uma melhor análise do comportamento individual de cada funcionalidade descrita nos casos de uso.

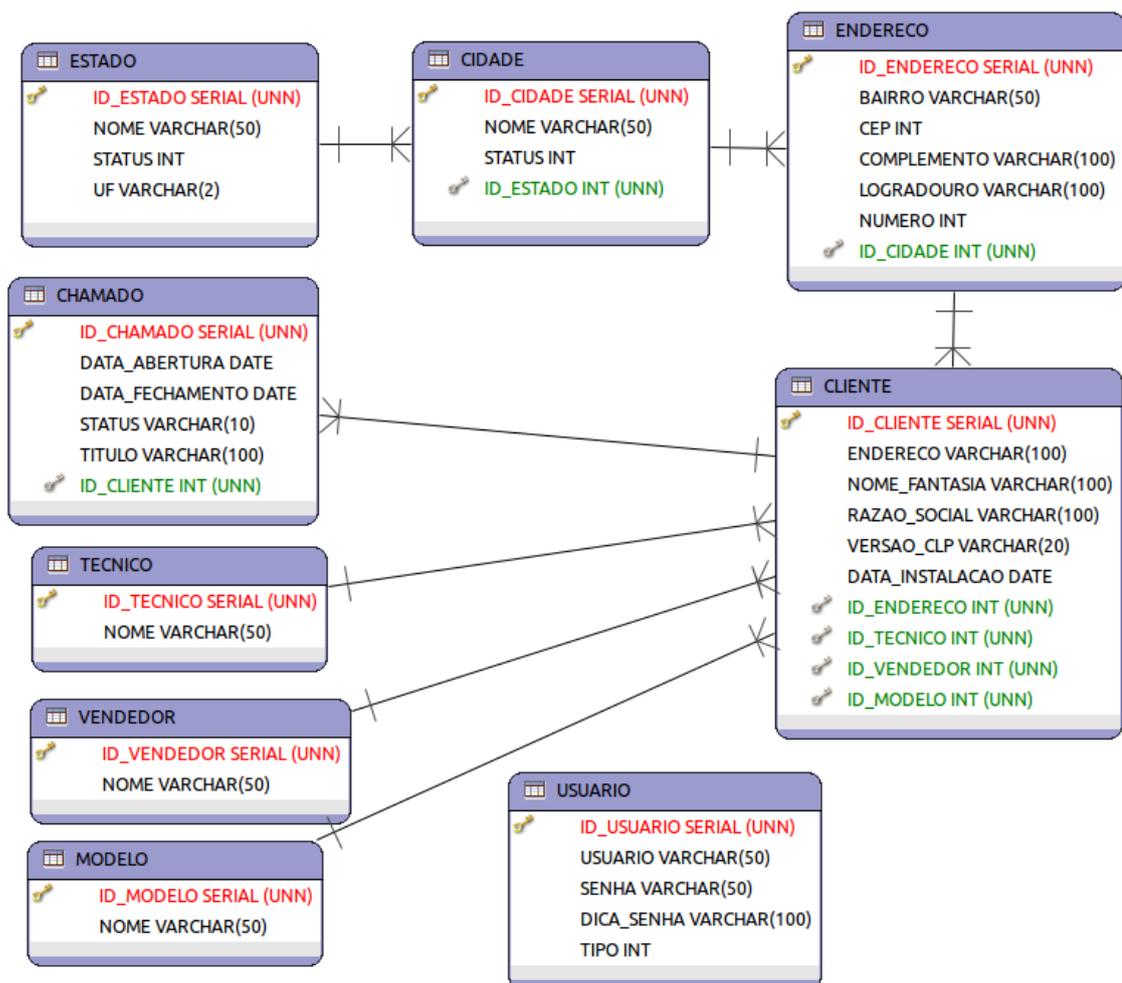


Figura 4.2: Diagrama Entidade Relacionamento

No diagrama de entidade-relacionamento, estão documentados os objetos a

serem persistidos na base de dados, assim como o relacionamento entre eles.

4.4.5 Implementação

A implementação da aplicação segue as funcionalidades descritas nos casos de uso, com código fonte implementado de forma integrada, sem a necessidade de módulos independentes.

A camada de visão foi representada pelos arquivos XHTML (*eXtensible Hypertext Markup Language*), que interagem com a parte servidor da aplicação através dos ManagedBeans, que representam a camada de controle. Estes são POJOs anotados com a *annotation* @ManagedBean definida na especificação do JSF 2, registrando automaticamente a classe anotada como um componente do JSF, definindo seu nome de acordo com o nome do parâmetro designado.

Conforme ilustrado no trecho de código abaixo, a declaração da classe BaseDAO.java, que é responsável pelas operações de CRUD (*Create, Read, Update, Delete*) da entidade Projeto, exemplifica a utilização da anotação @ManagedBean.

```
@ManagedBean(name = "projetoMBean")  
  
@SessionScoped  
  
public class ProjetoMBean implements Serializable{.....}
```

A classe anotada é responsável pela parte de controle da aplicação, que por meio de injeção de dependências é conectada automaticamente ao EJB responsável pela camada de modelo e faz a chamada da persistência, controlada pelo JPA (PANDA, 2009). O código abaixo representa um trecho em que é utilizada a anotação @EJB para a configuração da injeção de dependências.

```
@EJB  
  
Private GerenciarProjeto projetoBean
```

4.4.6 Testes

Os testes foram executados de forma unitária, em que cada funcionalidade foi testada separadamente.

Segundo (MARTINS, 2007), o teste de caixa preta especifica como testar um Caso de Uso ou um cenário específico de um Caso de Uso. O teste verifica o resultado da interação entre os atores e o sistema, seguindo os passos descritos no caso de uso, visando testar as funcionalidades do sistema externamente e que, portanto, este tipo

de teste não avalia o código desenvolvido e sim os resultados esperados à execução de cada funcionalidade.

Como documentação dos testes são gerados os Documentos de Evidência, em que são capturadas as telas do sistema e expostas no documento seguindo os passos descritos nas especificações dos casos de uso, conforme exemplificado nas imagens abaixo.

Para a exemplificação de um documento de evidência será utilizada a especificação de caso de uso da funcionalidade de Incluir Cliente.

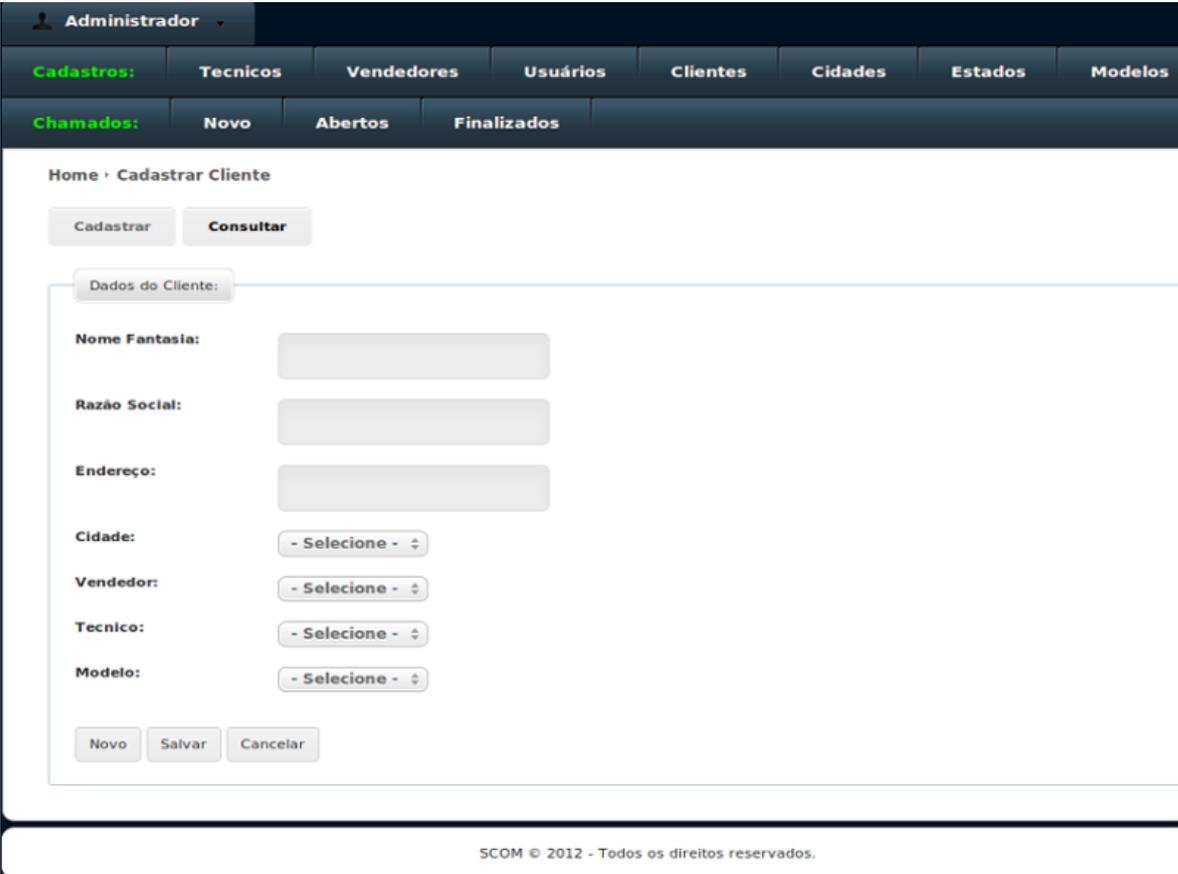


Figura 4.3: Cadastro de Clientes

1. Usuário informa o nome fantasia, razão social e endereço.
2. Usuário seleciona a cidade, vendedor, técnico e modelo de máquina já existentes no sistema.
3. Usuário clica em Salvar.
4. Sistema valida os dados e persiste os dados.

Após a persistência dos dados, o usuário pode visualizar o cadastro na tela de lista de clientes, como exhibe a figura 4.4.

Nome Fantasia	Razão Social	Endereço	Cidade	Vendedor	Tecnico	Modelo	Ações
Cliente 01	Cliente 01 Ltda.	Rua dois, 34	Londrina	Joao Vendedor	Zeh Tecnico	PCS 12	[Pencil] [X]
Cliente 03	Cliente 03 Ltda.	Rua do Cliente 03	Londrina	Joao Vendedor	Zeh Tecnico	PCS 12	[Pencil] [X]
Cliente 04	Cliente 04 e Cia	Av. do Cliente 04	Londrina	Joao Vendedor	Zeh Tecnico	PCS 12	[Pencil] [X]
Cliente View	Views S/A.	Avenida Carabatinga, 90	Salvador	Joao Vendedor	Zeh Tecnico	PCS 15	[Pencil] [X]
Clientão	Cliente Um Ltda.	Rua Brasil, 89	Londrina	Joao Vendedor	Zeh Tecnico	PCS 12	[Pencil] [X]

Figura 4.4: Listagem de Clientes

Para que seja alterado algum dos dados, basta clicar no indicador de edição e os dados serão carregados novamente na tela para que seja feita a alteração. A 4.5 exhibe a edição de cliente, também utilizando o mesmo caso de uso.

Os procedimentos citados a pouco seguem na mesma sequência para os demais casos de uso do sistema, por assim manter um padrão e facilitar a operação por parte dos colaboradores na empresa.

The screenshot shows a web application interface for editing a client record. The interface is organized as follows:

- Navigation Menu:** A dark blue header with a user profile icon and the name "Administrador". Below it, a horizontal menu contains several items: "Cadastros:", "Técnicos", "Vendedores", "Usuários", "Clientes", "Cidades", "Estados", and "Modelos". A second row of menu items includes "Chamados:", "Novo", "Abertos", and "Finalizados".
- Breadcrumb Trail:** Below the navigation menu, the path "Home > Cadastrar Cliente" is displayed.
- Form Controls:** Two buttons, "Cadastrar" and "Consultar", are positioned above the main form area.
- Form Fields:** A section titled "Dados do Cliente:" contains the following fields:
 - Nome Fantasia:** Text input field containing "Cliente 03".
 - Razão Social:** Text input field containing "Cliente 03 Ltda."
 - Endereço:** Text input field containing "Rua do Cliente 03".
 - Cidade:** Dropdown menu with "Londrina" selected.
 - Vendedor:** Dropdown menu with "Joao Vendedor" selected.
 - Tecnico:** Dropdown menu with "Zeh Tecnico" selected.
 - Modelo:** Dropdown menu with "PCS 12" selected.
- Form Actions:** At the bottom of the form, there are three buttons: "Novo", "Salvar", and "Cancelar".

At the bottom of the page, a footer contains the text: "SCOM © 2012 - Todos os direitos reservados."

Figura 4.5: Edição de Cliente

5 DESCRIÇÃO DO SISTEMA

5.1 DESCRIÇÃO DOS USUÁRIOS DO SISTEMA

Para justificar a separação do sistema em níveis de acesso, a taxonomia dos usuários deve possuir pelo menos dois níveis. Neste caso, os atores são divididos em três níveis, sendo dois de usuários e outro de relacionamentos externos, os operadores tem a necessidade de uso total do sistema desde cadastros até relatórios, excluindo-se apenas a opção de manutenção de usuários, atividade esta restrita aos administradores. Segue abaixo o detalhamento dos usuários:

- Administrador: Além das opções de operador, terá acesso a manutenção de usuários registrados no sistema, adicionando e removendo contas de operadores.
- Operador: Funcionários ligados ao setor de pós-venda que atendem os clientes e registram as manutenções realizadas. Estes também são responsáveis por emitir os relatórios para o setor de engenharia para estudarem possíveis atualizações nas máquinas produzidas.
- Relacionamentos Externos: Pessoas ligadas indiretamente ao sistema de manutenção, necessárias para que o processo de manutenção seja realizado, são elas:
 - Cliente;
 - Responsável técnico (colaborador da empresa);
 - Vendedor responsável pelo equipamento;

5.2 LIMITES E RESTRIÇÕES DO SISTEMA

Tratando-se de uma aplicação em ambiente web, o usuário necessita apenas do navegador. Sendo acessível através de computador, *notebook*, *tablet* ou celular.

Todo o processamento é deixado por conta do servidor de aplicação e o manuseio dos dados para o servidor de banco de dados.

Para o servidor de aplicação é necessário ter instalado a Máquina Virtual Java (JVM) e uma instância do servidor de aplicações *Glassfish Community 3.1* ou superior com as devidas configurações ajustadas, assim como também é possível a utilização de outros servidores de aplicação, como Jboss.

O servidor de banco de dados necessita apenas da base de dados, neste caso o PostgreSQL, em sua versão 9.1 ou superior.

Toda a aplicação, desde seu desenvolvimento até o uso, baseia-se em Java, o que a torna livre de restrições relacionadas a sistema operacional, portanto, pode ser atualizada e alterada tanto em Linux quanto em Windows ou Mac. Já o uso por parte do cliente aceita uma maior diversidade de sistemas operacionais, incluindo sistemas para dispositivos móveis, como Android ou IOS.

Quanto aos recursos de hardware para os servidores, seja de aplicação, hospedagem do banco de dados ou ambos na mesma máquina, deve ter como especificação mínima equivalente ou superior a:

- Processador Intel Pentium 4
- 1 GB de memória RAM
- HD de 160 GB
- Conexão com a rede local

A mesma recomendação de hardware dos servidores se repetem para os computadores *desktop* ou *notebook* que terão acesso ao sistema.

5.3 ANÁLISE DE PONTOS POR FUNÇÃO

Utilizando as diretrizes da métrica de Análise de Pontos por Função (PPF) foi possível medir o tamanho funcional do sistema desenvolvido, porém, seguindo estes indicadores, não é possível definir qual o tempo ou o esforço necessário para a construção do projeto.

Para a contagem dos pontos é necessário entender alguns conceitos:

- Arquivo Lógico Interno (ALI): É a unidade de armazenamento lógico interna da aplicação, por exemplo, tabelas no banco de dados.

- Arquivo Lógico Externo (AIE): Unidades de armazenamento lógico externas à aplicação.
- Entrada Externa (EE): Entrada de dados externa à aplicação, por exemplo, um formulário de inclusão de dados preenchido pelo usuário.
- Consulta Externa (CE): Saída de dados disparada através de uma solicitação de um usuário ou uma aplicação qualquer.
- Saída Externa (SE): Relatórios gerados pela requisição de um usuário ou aplicação.

Com base nestes conceitos, é necessária então a análise e a contagem dos requisitos funcionais do sistema, calculando assim a complexidade de cada item, classificando-as entre baixa, média ou alta.

Tabela 5.1: Qualificação de Complexidades

Componente	Complexidade	Pesos
ALI	Baixa	7
	Média	10
	Alta	15
AIE	Baixa	5
	Média	7
	Alta	10
IE	Baixa	3
	Média	4
	Alta	6
OE	Baixa	4
	Média	5
	Alta	7
CE	Baixa	3
	Média	4
	Alta	6

Após a definição da complexidade, deve-se contar a quantidade de itens em cada categoria e multiplicar de acordo com a tabela acima chegando à quantidade de PPF não ajustados, que no caso deste trabalho é 201.

É importante ressaltar que o cálculo de PPF metrifica apenas o tamanho funcional do sistema, e não o esforço ou quantidade de tempo necessário para a conclusão do sistema, em que fatores como nível da equipe de desenvolvimento, linguagem utilizada e quantidade de pessoas envolvidas na construção influenciam fortemente.

5.4 CRONOGRAMA

5.4.1 Cronograma Proposto

Tabela 5.2: Cronograma Proposto

Fases	07/11				08/11				09/11				10/11				11/11			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Reuniões			x				x				x			x						
Desenhar FUNC	x	x		x		x		x		x		x	x		x	x	x			
Contruir FUNC	x		x	x		x		x		x		x		x	x	x	x			
Testes															x	x	x	x		
Implantacao																			x	x
Escrita - TCC	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	X		

5.4.2 Cronograma Oficial

Tabela 5.3: Cronograma Oficial

Fases	10/12				11/12				12/12				01/13				02/13			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Reuniões	x					x						x			x					
Desenhar FUNC		x	x	x		x		x		x		x	x		x	x	x			
Contruir FUNC		x	x	x		x		x		x		x		x	x	x	x			
Testes														x	x	x	x	x		
Implantacao																x	x	x	x	x
Escrita - TCC	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

6 CONCLUSÃO

Após o período de testes na empresa, foi constatado que a duração dos atendimentos reduziu consideravelmente, uma vez que o cliente já foi cadastrado, assim como os modelos de equipamentos, entre outros dados, o atendente ganha tempo ao apenas localizar o cadastro e iniciar um novo chamado. Além disso, no momento de localizar os atendimentos, utilizando as opções de busca, a inquietação e indagação do cliente em relação ao atendimento diminuiu notavelmente.

Por parte dos usuários não houve rejeições e o projeto atualmente continua em testes para que no momento em que estiver totalmente dentro das necessidades do cliente, seja implantado no setor.

Em relação ao aprendizado, proporcionou o aprofundamento nas tecnologias que estão em evidência no mercado, como JSF 2, JPA e EJB. E também grande expansão de visão de negócio, estudo de rotinas de trabalho nas empresas, negociação com o cliente, implantação e suporte.

Outro ponto importante foi a eliminação de folhas de papel para registro dos atendimentos, não aumentando assim, o volume dos arquivos físicos da empresa e poupando o meio ambiente, uma vez que não se utiliza mais papéis nem tintas e todos os computadores envolvidos no processo já permaneciam ligados devido a outras tarefas, não aumentando também o consumo de energia na empresa.

6.1 PROBLEMAS ENCONTRADOS

Um dos problemas encontrados a se destacar, foi a mudança do *framework* de desenvolvimento durante a implementação da aplicação e o ambiente de execução do sistema. Durante os primeiros testes, foi constatado que o sistema *desktop* não seria interessante para a empresa, visto que poderia haver rotatividade de computadores e não saberiam se alguns computadores poderiam executar o sistema ou não.

Assim surgiu a alternativa do desenvolvimento de uma aplicação web, baseada nos mesmos requisitos levantados para a aplicação *desktop* e foi implementada

de acordo com o as necessidades e conforto do cliente, vide cores e formatos em ambiente web.

6.2 TRABALHOS FUTUROS

O projeto se estenderá futuramente com o maior detalhamento dos relatórios, conforme já solicitado pela equipe de engenharia da empresa, sendo possível obter dados com maior nível de detalhamento e possibilidade de cruzamento de informações, auxiliando na evolução constante na produção das máquinas projetadas pela empresa.

Além do detalhamento dos relatórios, será implementada a replicação de dados para que as informações não fiquem centralizadas em apenas um servidor, atualmente sujeito de falhas e falta de acesso ao sistema.

Referências

- ABRAHAMSSON, P. et al. *Agile software development methods - Review and analysis*. Boston, MA, USA, 2002.
- AMBLER, S. W. *Modelagem Agil - Praticas Eficazes para a Programacao Extrema e o Processo Unificado*. [S.l.]: Bookman, 2004.
- BECK, K. *Agile Manifesto*. 2001. <http://www.agilemanifesto.org/>. [Online; acessado 16-Maio-2011].
- COLLINS BEN; FITZPATRICK, B. *Controle de Versao com Subversion*. 2007. <http://svnbook-pt-br.googlecode.com/svn/snapshots/1.4/index.html/>. [Online; acessado 11-Janeiro-2013].
- DOC2004. *Hibernate Documentation*. <http://www.hibernate.org/>. [Online; acessado 12-Novembro-2012].
- GONCALVES, E. *Desenvolvendo aplicacoes web com jsp, servlets, javaserver faces, hibernate, ejb 3 persistence e ajax*. Rio de Janeiro, RJ: Ciencia Moderna, 2007.
- HIGHSMITH, J. *Agile Software Development Ecosystems*. Boston, MA,: Addison-Wesley Longman Publishing Co., Inc., 2002.
- IBGE. *Pesquisa sobre o Uso das Tecnologias de Informacao e Comunicacao nas Empresas 2010*. 2012. <http://www.ibge.gov.br/home/presidencia/noticias/>. [Online; acessado 16-Janeiro-2013].
- KEITH MIKE; SCHINCARIOL, M. *EJB 3 Professional: Java Persistence API*. [S.l.]: Editora Ciencia Moderna, 2008. ISBN 8573936967.
- KRISHNA, G. *Umbrello UML Modeller*. 2008. <http://uml.sourceforge.net/>. [Online; acessado 14-Maio-2011].
- LEMAY LAURA; PERKINS, C. L. *Aprenda em 21 dias java, 4. ed.* Rio de Janeiro, RJ: Campus, 1997.
- LUCA, J. D. *Feature-Driven Development (FDD) Overview Presentation*. 2002. <http://www.nebulon.com/articles/fdd/download/fddoverview.pdf/>. [Online; acessado 16-Maio-2011].
- MANZONI, L. V. *Tecnologia Enterprise JavaBeans*. Santa Maria, RS: [s.n.], 1997. http://www.inf.ufrgs.br/gppd/disc/cmp167/trabalhos/mp2000-1/Lisandra/Artigo_EJB.html/. [Online; acessado 20-Janeiro-2012].
- MARTINS, J. C. C. *Gerenciando Projetos de Desenvolvimento de Software com PMI, RUP e UML. 4. ed.* Rio de Janeiro, RJ: Brasport, 2007.

PANDA, D. *EJB 3 em Acao. 2. ed.* Rio de Janeiro, RJ: Alta Books, 2009.

POSTGRESQL.ORG. *About PostgreSQL.* 2011. <http://www.postgresql.org/about/>. [Online; acessado 14-Maio-2011].

SAMPAIO, C. *Java Enterprise Edition 6: desenvolvendo aplicacoes corporativas.* Rio de Janeiro, RJ: Brasporta, 2011. ISBN 9788574524603.

SILVA, M. S. *Construindo Sites com CSS e (X)HTML.* [S.l.]: Novatec, 2007. ISBN 9788575221396.

SOMMERVILLE, I. *Engenharia de software.* Sao Paulo, SP: Addison Wesley Longman Publishing Co., Inc., 2003.

UBUNTU.ORG. *O que e o Ubuntu?* 2007. <http://www.ubuntu-br.org/ubuntu/>. [Online; acessado 14-Maio-2011].

WILLIAMS, G. *DIA: Charts and Diagrams.* 2009. http://www.togaware.com/linux/survivor/DIA_Charts.html/. [Online; acessado 22-Novembro-2012].

Índice Remissivo

- ágil, 24
- AIE, 34
- ALI, 33
- aperfeiçoamento, 17
- CE, 34
- CLP, 16
- compartilhar, 22
- Conexões, 19
- corporativo, 16
- CVS, 22
- decisões, 17
- desenvolvimento, 17
- DIA, 21
- diagnóstico, 13
- dificuldades iniciais, 11
- Eclipse, 21
- EclipseLink, 20
- EE, 34
- EJB, 19
- engenharia, 17
- específicos, 17
- especificação, 26
- estilo, 19
- FDD, 25
- fluxo, 13
- fluxogramas, 21
- formatação, 19
- futuramente, 37
- Glassfish, 19, 22
- GNU/Linux, 22
- Hibernate, 19
- HTML, 21
- identificador, 17
- informatização, 13
- institucional, 16
- Java, 20
- JDBC, 20
- JDK, 20
- JPA, 19
- JRE, 20
- JSP, 20
- JVM, 20
- legibilidade, 19
- Mapeamento, 20
- metodologias, 16
- metodologistas, 24
- MVC, 21
- OpenJPA, 20
- ORM, 19
- público, 17
- PPF, 33
- praguicidas, 11
- problemas, 36
- proposta, 13
- proposto, 14
- relatórios, 17
- requisitos, 16
- SE, 34
- Servlets, 20
- silos graneleiros, 11
- Simplicidade, 25
- tomada, 13
- TopLink, 20
- trâmites, 14
- Transações, 19
- Ubuntu, 22
- Umbrello, 23
- XHTML, 28

Apêndice A - Casos de Uso

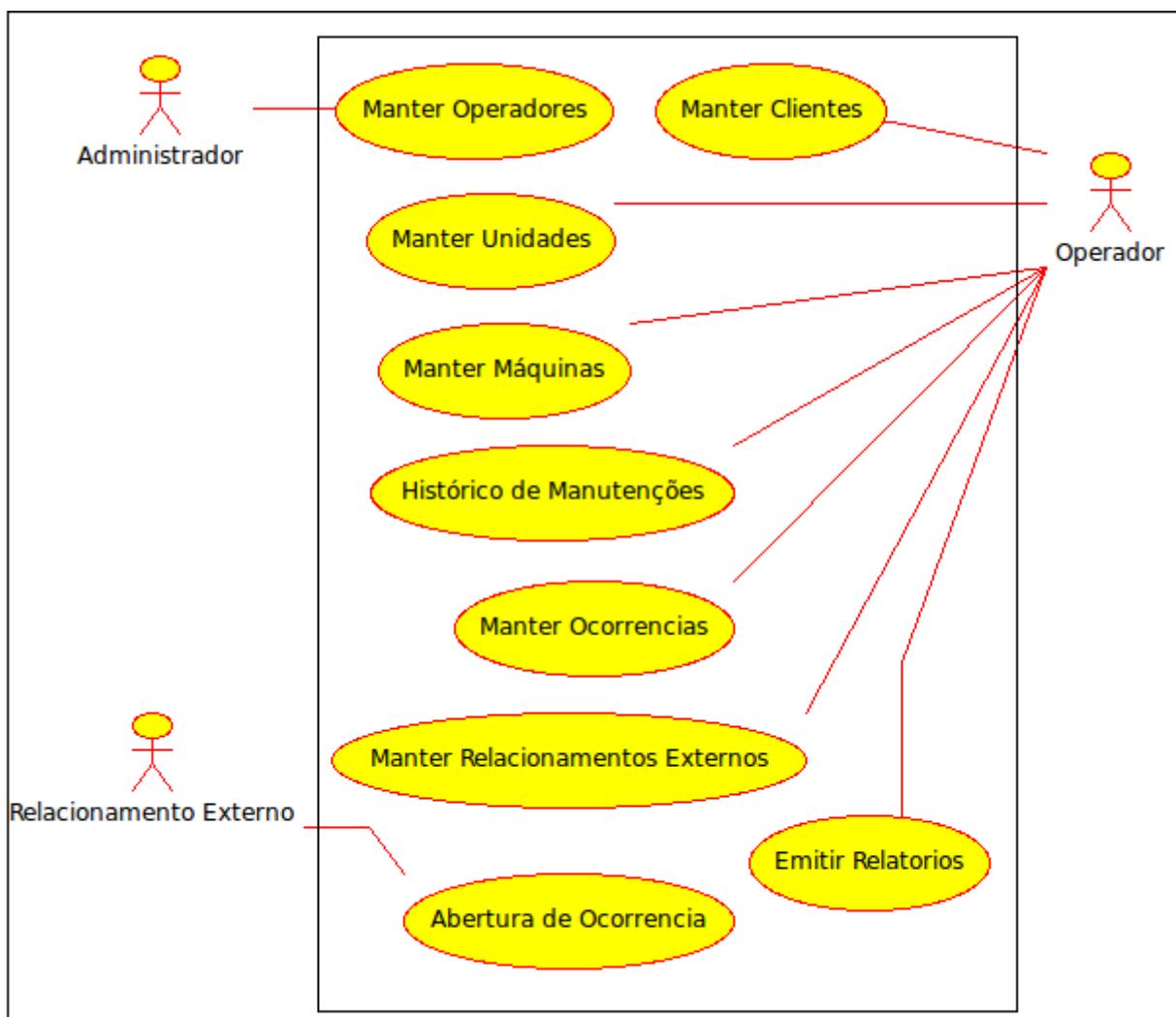


Figura A.1: Caso de Uso Geral

Tabela A.1: Caso de Uso - Manter Operadores

Nome do Caso de Uso	Manter Operadores
Ator Principal	Administrador
Resumo	Este caso de uso descreve as possíveis alterações no cadastro de operadores do sistema, permitindo incluir, alterar, consultar ou desativar operadores.
Pré-Condições	Um administrador cadastrado no sistema.
Pós-Condições	O novo operador deve alterar sua senha antes de poder iniciar o uso do sistema.
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Informar nome completo do operador	
	2. Caso o operador já esteja cadastrado, mostrar opções de edição do usuário.
	3. Se não retornar nenhum cadastro, mostrar opção de novo usuário.
Restrições/Validações	1. O login será sempre o primeiro nome ponto ultimo nome 2. um login é único para cada operador

Tabela A.2: Caso de Uso - Manter Relacionamentos Externos

Nome do Caso de Uso	Manter Relacionamentos Externos
Ator Principal	Operador
Resumo	Gerencia o cadastro pessoas relacionadas com o processo de manutenção das máquinas, ou seja, os técnicos responsáveis, os responsáveis na empresa e os vendedores.
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Informar nome completo da pessoa	
	2. Verificar se o cadastro já existe
	3. Mostrar opções de acordo com sua situação
4. Informar qual o tipo de pessoa	
Restrições/Validações	1. Cada relacionamento pode ser de apenas um tipo

Tabela A.3: Caso de Uso - Manter Clientes

Nome do Caso de Uso	Manter Clientes
Ator Principal	Operador
Resumo	Formulário de gerencia de clientes, para posteriormente serem adicionadas as unidades de cada cliente
Pré-Condições	Possuir em mãos os dados do cliente
Pós-Condições	Devem ser adicionadas unidades para cada cliente
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Informar a Razão Social e o Nome Fantasia do Cliente, juntamente com o telefone do escritório e endereço.	
	2. Verificar se o cadastro já existe
	3. Mostrar opções de acordo com sua situação
4. Confirmar os dados ou cancelar	
Restrições/Validações	1. Todos os campos devem ser de preenchimento obrigatório

Tabela A.4: Caso de Uso - Manter Unidades

Nome do Caso de Uso	Manter Unidades
Caso de Uso Geral	Manter Clientes
Ator Principal	Operador
Resumo	Gerencia as unidades de cada cliente, podendo ser apenas uma ou várias.
Pré-Condições	O cliente estar previamente cadastrado
Pós-Condições	Escolher o tipo de máquina utilizada na unidade
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Informar o cliente que possui a unidade	
	2. Verifica os clientes e retorna o informado ou semelhante
	3. Caso o cliente não esteja cadastrado, dar opção para cadastrar.
4. Cadastrar novo cliente ou escolher existente	
	5. Mostra a tela para gerencia da unidade
6. Informa os dados da unidade	

Tabela A.5: Caso de Uso - Manter Máquinas

Nome do Caso de Uso	Manter Máquinas
Ator Principal	Operador
Resumo	Gerencia os modelos de máquinas comercializados atualmente
Pré-Condições	A máquina ser comercializada pela empresa
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Informar o modelo da maquina e possíveis anotações textuais	
	2. Cadastra a nova maquina ou substitui a existente

Tabela A.6: Caso de Uso - Abertura de Ocorrência

Nome do Caso de Uso	Abertura de Ocorrência
Ator Principal	Relacionamento Externo
Ator Secundário	Operador
Resumo	Abertura de ocorrência para ocorrências atípicas de atividade da maquina
Pré-Condições	Haver uma máquina na unidade que esta iniciando a ocorrência
Pós-Condições	A ocorrência deve ser reaberta futuramente para informar o estado da mesma. Se foi concluída, não concluída ou está em andamento.
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Informa os dados para abertura da ocorrência	
	2. O sistema gera uma lista de ocorrências passadas da maquina, sendo possível ver detalhes de cada ocorrência
3. Caso não retorne ocorrências, iniciar a primeira.	
	4. O sistema grava a ocorrência e já disponibiliza para consultas futuras.
Restrições/Validações	1. Cliente, Unidade, Responsável Técnico, Responsável na Empresa e vendedor responsável deve ser obrigatoriamente preenchido.

Tabela A.7: Caso de Uso - Histórico de Manutenções

Nome do Caso de Uso	Histórico de Manutenções
Caso de Uso Geral	Abertura de ocorrência
Ator Principal	Operador
Resumo	Lista as informações resumidamente sobre as ocorrências que a máquina já teve, sendo possível expandir todas as ocorrências para análise em tempo de execução.
Pré-Condições	A máquina já ter alguma ocorrência
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Busca a máquina pelo campo desejado	
	2. Retorna os resultados compatíveis
3. Analisa os itens com opção de expandir os resultados	

Tabela A.8: Caso de Uso - Emitir Relatórios

Nome do Caso de Uso	Emitir Relatórios
Caso de Uso Geral	Histórico de Manutenções
Ator Principal	Operador
Resumo	Emite relatórios de acordo com as necessidades da equipe de engenharia, para auxílio na tomada de decisões.
Pré-Condições	Haver ocorrência cadastrada
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Seleciona o tipo de relatório	
	2. Retorna o relatório de acordo com a escolha do operador
3. Repassa o relatório à equipe de engenharia.	