

EDER RIBEIRO DA SILVA

SOFTWARE PARA ASSISTÊNCIA TÉCNICA DE INFORMÁTICA

Assis

2010

EDER RIBEIRO DA SILVA

SOFTWARE PARA ASSISTÊNCIA TÉCNICA DE INFORMÁTICA

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

Orientador: Fernando César de Lima.

Área de Concentração: Desenvolvimento de Sistemas.

Assis

2010

FICHA CATALOGRÁFICA

RIBEIRO, Eder

Software para assistência técnica de Informática / Eder Ribeiro. Fundação Educacional do Município de Assis – FEMA –Assis, 2010.

26p.

Orientador: Fernando César de Lima.

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

1. Software Web. 2. Assistência Técnica

CDD:001.61

Biblioteca da FEMA

SOFTWARE PARA ASSISTÊNCIA TÉCNICA DE INFORMÁTICA

EDER RIBEIRO DA SILVA

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: Fernando César de Lima.

ANALISADOR (1): Alex Sandro Romeo de Souza Poletto.

Assis

2010

AGRADECIMENTOS

Ao Professor, Fernando César de Lima, pela orientação e pelo constante estímulo transmitido durante o trabalho.

Ao amigo, Gabriel Fernandes Rios e a todos que colaboraram direta ou indiretamente, na execução deste trabalho.

Aos familiares, João Ribeiro da Silva e Francisca Célia Tibúrcio da Silva (pais), Elis Regina da Silva e Regina Marta da Silva (irmãs), sem a ajuda e motivação deles seria impossível o término deste trabalho.

O que prevemos raramente ocorre.
O que menos esperamos geralmente acontece.

Benjamin Disraeli
(1804 – 1881)

RESUMO

Este trabalho consiste no desenvolvimento de um software web para agendamento de serviços de informática on-line, onde tanto o cliente quanto o funcionário da loja poderão fazer agendamentos, bem como outros serviços oferecidos pelo sistema, como cadastros, históricos de visitas técnicas, dentre outros.

O Sistema conta com cadastro de: Cidade, Cliente, Funcionário, Produto, Serviço, Agendamento, Ordem de Serviço e relatórios. Foi desenvolvido para empresa de pequeno porte, mas está apto a atender também empresas de médio porte.

Palavra-chave: Sistema, Web, Informática, Assistência.

ABSTRACT

This work have the objective develop a software for web scheduling information technology services online, where the client and the store clerk can not only schedule as other services offered by the system as entries, historical visits techniques among other.

The system includes records of: City, Customer, Employee, Product, Service Scheduling, Service Order and reports. It was developed for small business, but is

able to meet also medium-sized businesses.

Keyword: System, Web, Computer Assistance.

LISTA DE ABREVIATURAS E SIGLAS

IDE	Integrated Development Environment.
HSQldb	Hyperthreaded Structured Query Language Database
JUDE	Java and UML Developers' Environment
EE	Enterprise Edition
J2EE	Java 2 Enterprise Edition
SE	Standard Edition
RPC	Remote Procedure Call
CORBA	Common Object Request Broker Architecture
IDE	Integrated Development Environment
IBM	International Business Machines
SWT	Standard Widget Toolkit
SGBD	Sistema Gerenciador de Banco de Dados
SQL	Structured Query Language
JDBC	Java Database Connectivity
UML	Unified Modeling Language
XML	eXtensible Markup Language
JSP	Java Server Pages
ASP	Active Server Pages
MVC	Model-view-controller

SUMÁRIO

1. INTRODUÇÃO.....	13
2. OBJETIVO DO TRABALHO.....	13
3. PUBLICO ALVO.....	13
4. JUSTIFICATIVAS.....	13
5. MÉTODO DE DESENVOLVIMENTO.....	14
5.1 Tecnologia (Java EE)	14
5.2 Eclipse (IDE).....	15
5.3 HSQLDB (Banco de Dados)	15
5.4 Jude (UML)	16
5.5 Hibernate	16
5.6 Java Server Pages (JSP)	16
5.7 Java Server Faces (JSF)	17
5.8 Model View Controller (MVC)	18
6. ESTRUTURA DE DESENVOLVIMENTO DO SISTEMA.....	19
6.1 Mapa Mental	20
6.2 Diagrama de Caso de Uso: Manter	21
6.3 Diagrama de Caso de Uso: Relatório.....	21
6.4 Diagrama de Caso de Uso: Login.....	22
6.5 Diagrama de Classe	22
6.6 Diagrama de Atividade Cliente	23
6.7 Diagrama de Atividade Funcionário: Manter	23

6.8 Diagrama de Atividade Funcionário: Relatório	24
6.9 Tela de Cadastro de Cliente	25
6.10 Tela de Cadastro de Produto.....	25
6.11 Tela de Consulta de Cliente	26
6.12 Tela de Consulta de Cidade	26
6.13 Código de Consulta de Cidade.....	26
7. CRONOGRAMA DA ESTRUTURA DE DESENVOLVIMENTO.....	37
8. CONCLUSÃO.....	38
9. REFERÊNCIAS.....	39

LISTA DE ILUSTRAÇÕES

Figura 01 – Mapa Mental	20
Figura 02 – Diagrama de Caso de Uso: Manter	21
Figura 03 - Diagrama de Caso de Uso: Relatório	21
Figura 04 - Diagrama de Caso de Uso: Login	22
Figura 05 - Diagrama de Classe	22
Figura 06 - Diagrama de Atividade Cliente.....	23
Figura 07 - Diagrama de Atividade Funcionário: Manter.....	23
Figura 08 - Diagrama de Atividade Funcionário: Relatório.....	24
Figura 09 – Tela de Cadastro de Cliente.....	25
Figura 10 – Tela de Cadastro de Produto	25
Figura 11 – Tela de Consulta de Cliente.....	26
Figura 12 – Tela de Consulta de Cidade.....	26
Figura 13 – Cronograma	37

1. INTRODUÇÃO

Desde a criação, o comércio eletrônico tem sido absorvido em grande escala. Com seu crescimento, muitos ramos da economia agora são ligados ao comércio eletrônico.

No seu início, somente era comercializado produtos, mas com os seus avanços, uma nova tendência surgiu para o comércio on-line, a venda de serviços.

A empresa Vision Infocenter, situada na Rua Ângelo Pipolo, 731 – Cândido Mota/SP, é nova no mercado e pretende oferecer um diferencial para seus clientes, agendamento on-line de serviços de informática.

2. OBJETIVO DO TRABALHO

O trabalho tem por objetivo o desenvolvimento de um sistema Web para o controle e agendamento on-line de assistência técnica, no qual o cliente tem a possibilidade de fazer seus próprios agendamentos no sistema.

3. PÚBLICO ALVO

O sistema poderá ser implantado em empresas de pequeno e médio porte que prestam serviços de informática.

4. JUSTIFICATIVAS

Com o crescimento da empresa, o aumento de clientes e conseqüentemente de agendamentos de serviços aumentarão. Assumindo essa situação, surgiu a ideia de oferecer para o cliente mais essa opção de agendamento, agilizando e facilitando tanto para o cliente quanto para a empresa, além de não existir esse tipo de serviço na região.

Com esse projeto, pretende-se aprender a linguagem e posteriormente atuar como programador Web no mercado.

5. MÉTODO DE DESENVOLVIMENTO

O sistema será desenvolvido na linguagem Java que é orientada a objetos. Um sistema orientado a objetos é composto por um conjunto de classes e objetos bem definidos que interagem entre si, de modo a gerar o resultado esperado.

A ferramenta para o desenvolvimento do sistema será a IDE Eclipse.

O banco de dados que será utilizado no sistema é o HSQLDB, escolha esta feita por ser uma solução simples, que utiliza poucos recursos e que possui bom desempenho.

A ferramenta para preparação da análise será o Jude. O programa é de uso fácil e intuitivo e também desenvolvido na linguagem Java.

5.1 Tecnologia (Java EE)

Java EE é uma plataforma de programação para servidores na linguagem de programação Java.

O nome J2EE era usado nas versões mais antigas, até a 1.4. Hoje, o nome correto é Java EE, que atualmente encontra-se na versão 6.

A Plataforma Java EE difere-se da Plataforma Java SE pela adição de bibliotecas que fornecem funcionalidades para implementar software Java distribuído, tolerante a falhas e multi-camada, baseada amplamente em componentes modulares executando em um servidor de aplicações. A plataforma Java EE é considerada um padrão de desenvolvimento já que o fornecedor de software nesta plataforma deve seguir determinadas regras se quiser declarar os seus produtos como compatíveis

com Java EE. Ela contém bibliotecas desenvolvidas para o acesso a base de dados, RPC, CORBA, etc. Devido a essas características, a plataforma é utilizada principalmente para o desenvolvimento de aplicações corporativas.

5.2 Eclipse (IDE)

Eclipse é uma IDE desenvolvida em Java, com código aberto para a construção de programas de computador. O projeto Eclipse foi iniciado na IBM, que desenvolveu a primeira versão do produto e doou-o como software livre para a comunidade. O gasto inicial da IBM no produto foi de mais de 40 milhões de dólares. Hoje, o Eclipse é a IDE Java mais utilizada no mundo. Possui como características marcantes o uso da SWT e não do Swing como biblioteca gráfica, a forte orientação ao desenvolvimento baseado em plugins e o amplo suporte ao desenvolvedor com centenas de plugins que procuram atender as diferentes necessidades de diferentes programadores. Atualmente o Eclipse é controlado por uma organização independente sem fins lucrativos, a Eclipse Foundation. (BURNETTE, 2006)

5.3 HSQLDB (Banco de Dados)

O HSQLDB é um servidor de banco de dados (SGBD), de código aberto, escrito totalmente na linguagem Java.

Não é possível compará-lo, em termos de robustez e segurança com outros servidores SGBD, como Oracle ou Microsoft SQL Server, entretanto, o HSQLDB é uma solução simples, que utiliza poucos recursos e que possui bom desempenho. Devido a essas características, ele é bastante utilizado em aplicações que são executadas em desktops e que necessitam interagir com uma camada de persistência através da linguagem SQL.

Principais características:

- Suporte à linguagem SQL básica, incluindo junções, triggers e visões.
- Portabilidade em virtude de sua implementação ser feita em Java.

- Repositórios acessíveis através de tecnologia JDBC.
- Criação de bancos de dados em arquivo texto, banco de dados e em memória.
- Recurso de dump para backups facilitados.
- Ocupa pouco espaço em disco.
- Praticamente dispensa configurações para operar. (<http://www.hsqldb.org/>)

5.4 Jude (UML)

JUDE é uma das ferramentas grátis para UML mais poderosas disponíveis atualmente. Com características que não são encontradas nas outras ferramentas grátis, como adicionar métodos no diagrama de sequência e a alteração se refletir no diagrama de classes.

5.5 Hibernate

O Hibernate é um framework para o mapeamento objeto-relacional escrito na linguagem Java. Esta camada facilita o mapeamento dos atributos entre uma base tradicional de dados relacionais e o modelo objeto de uma aplicação, mediante o uso de arquivos XML para estabelecer esta relação.

O objetivo do Hibernate é diminuir a complexidade entre os programas Java, baseado no modelo orientado a objeto, que precisam trabalhar com um banco de dados do modelo relacional. Em especial, no desenvolvimento de consultas e atualizações dos dados.

Sua principal característica é a transformação das classes em Java para tabelas de dados (e dos tipos de dados Java para os da SQL). O Hibernate gera as chamadas SQL e libera o desenvolvedor do trabalho manual da conversão dos dados resultante, mantendo o programa portátil para quaisquer bancos de dados SQL,

porém causando um pequeno aumento no tempo de execução.
(<http://javafree.uol.com.br/wiki/hibernate>)

5.6 Java Server Pages (JSP)

Java Server Pages é uma tecnologia utilizada no desenvolvimento de aplicações para Web, similar às tecnologias ASP da Microsoft. Por ser baseada na linguagem de programação Java, tem a vantagem da portabilidade de plataforma, que permite a sua execução em diversos sistemas operacionais, como o Windows, Unix e Linux. Esta tecnologia permite ao desenvolvedor de páginas para Internet produzir aplicações que acessem o banco de dados, manipulem arquivos no formato texto, capturem informações a partir de formulários e capturem informações sobre o visitante e sobre o servidor.

Uma página criada com a tecnologia JSP, após instalada em um servidor de aplicação compatível com a tecnologia Java EE, é transformada em um Servlet.

Tomcat e o Glassfish são exemplos de servidores compatíveis com a tecnologia JSP. (<http://java.sun.com/products/jsp/>)

5.7 Java Server Faces (JSF)

Java Server Faces é um framework MVC para o desenvolvimento de aplicações Web, que permite o desenvolvimento de aplicações para a internet de forma visual, ou seja, arrastando e soltando os componentes na tela (JSP), definindo propriedades dos mesmos.

O JSF é atualmente considerado pela comunidade Java como a última palavra em termos de desenvolvimento de aplicações Web utilizando Java, resultado da experiência e maturidade adquiridas com o JSP/Servlet (Model1), Model2 (MVC) e Struts.

Características:

- Permite que o desenvolvedor crie UIs através de um conjunto de componentes UIs pré-definidos.
- Fornece um conjunto de tags JSP para acessar os componentes.
- Reutiliza componentes da página.
- Associa os eventos do lado cliente com os manipuladores dos eventos do lado do servidor (os componentes de entrada possuem um valor local representando o estado no lado servidor).
- Fornece separação de funções que envolvem a construção de aplicações **Web**.
- Utiliza Ajax em alguns de seus componentes tornando alguns processos mais rápidos e eficientes. (<http://www.oracle.com/technetwork/java/javasee/javaserverfaces-139869.html>)

5.8 Model view controller (MVC)

MVC é um padrão de arquitetura de software que visa separar a lógica de negócio da lógica de apresentação, permitindo o desenvolvimento, teste e manutenção isolado de ambos.

Com o aumento da complexidade das aplicações desenvolvidas, torna-se relevante a separação entre os dados e a apresentação das aplicações. Desta forma, alterações feitas no *layout* não afetam a manipulação de dados, e estes poderão ser reorganizados sem alterar do mesmo.

Esse padrão resolve este problema por meio da separação das tarefas de acesso aos dados e lógica de negócio, lógica de apresentação e de interação com o utilizador, introduzindo um componente entre os dois, o controlador. (MACORATTI, 2010)

6. ESTRUTURA DE DESENVOLVIMENTO DO SISTEMA

O Trabalho será desenvolvido da seguinte forma:

- Levantamento de requisitos.
- Análise do Sistema.
- Desenvolvimento do Sistema.
- Testes.
- Implantação do Sistema.
- Conclusão.

6.1 Mapa Mental

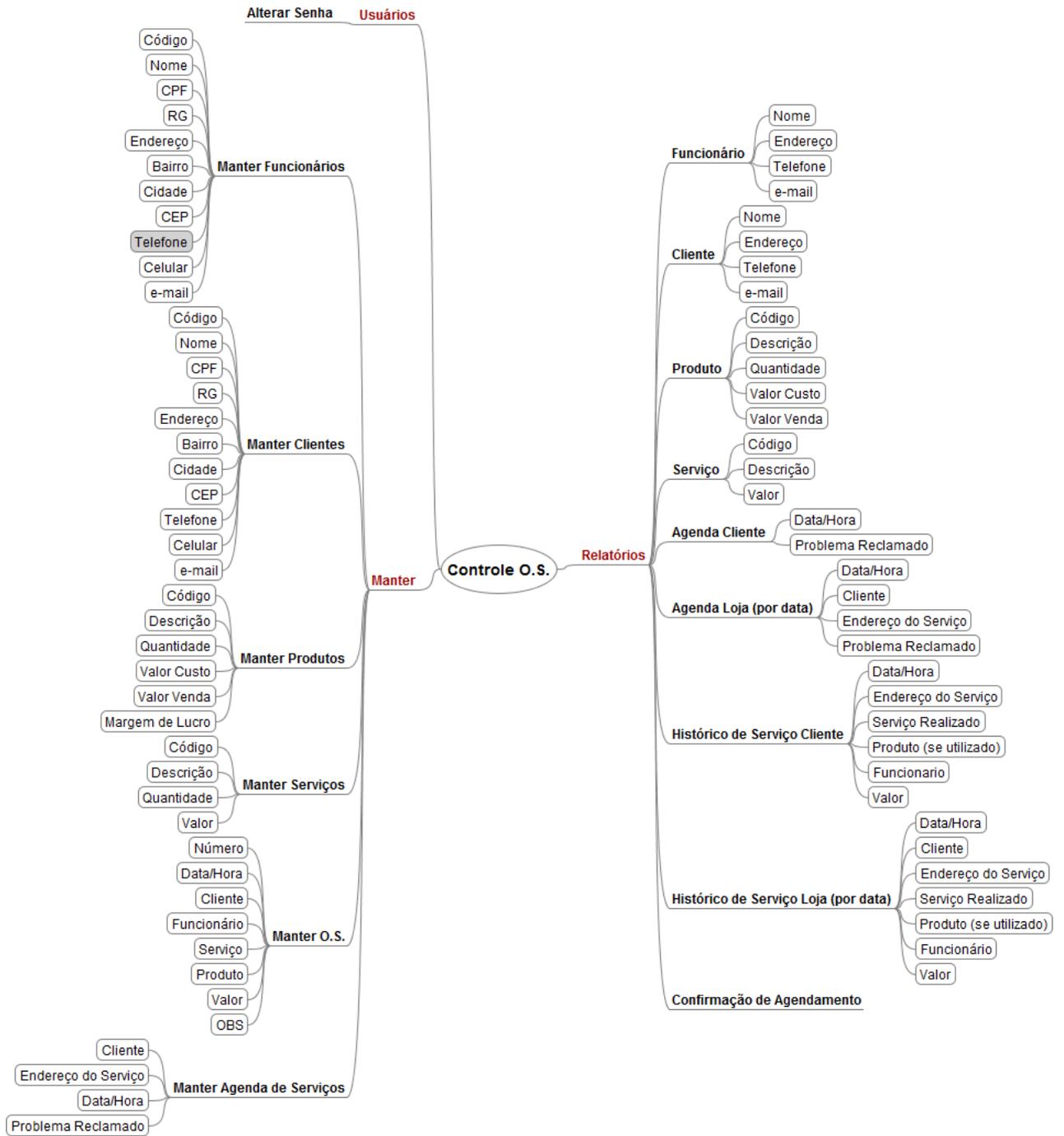


Figura 1º: Mapa Mental

6.2 Diagrama de Caso de Uso: Manter

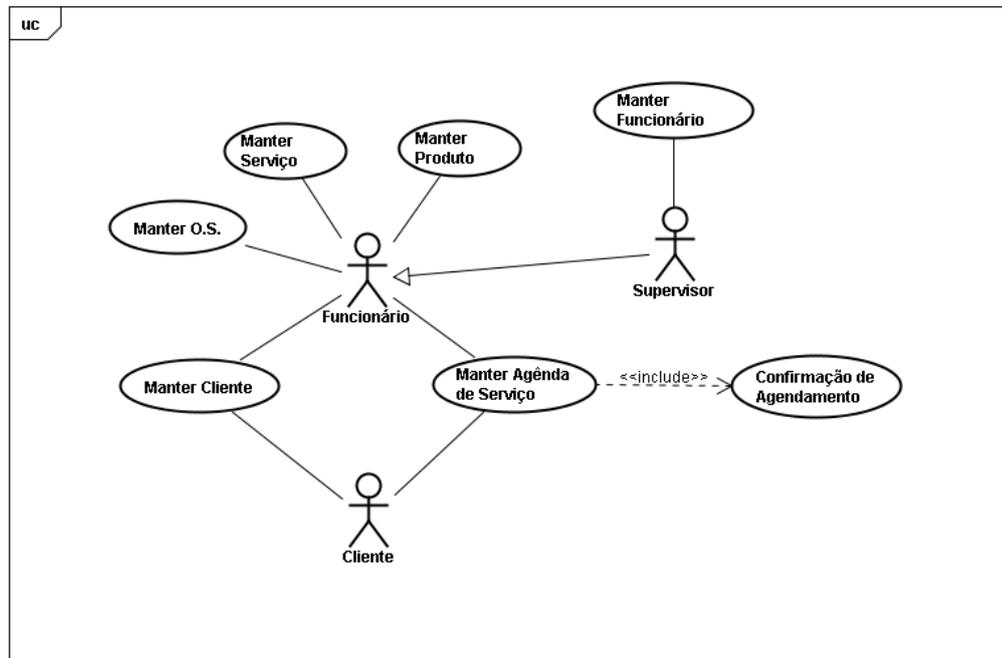


Figura 2º: Diagrama de Caso de Uso: Manter

6.3 Diagrama de Caso de Uso: Relatório

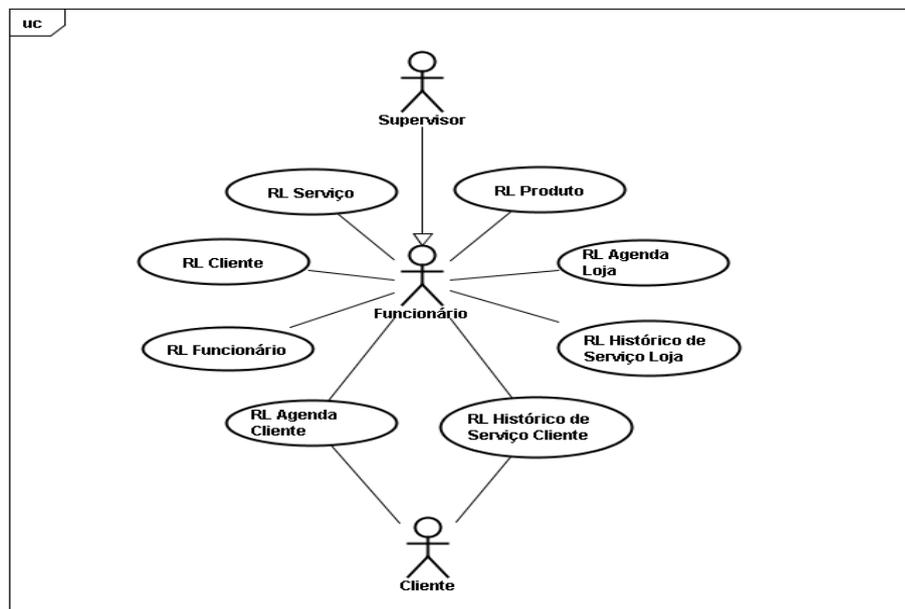


Figura 3º: Diagrama de Caso de Uso: Relatório

6.4 Diagrama de Caso de Uso: Login

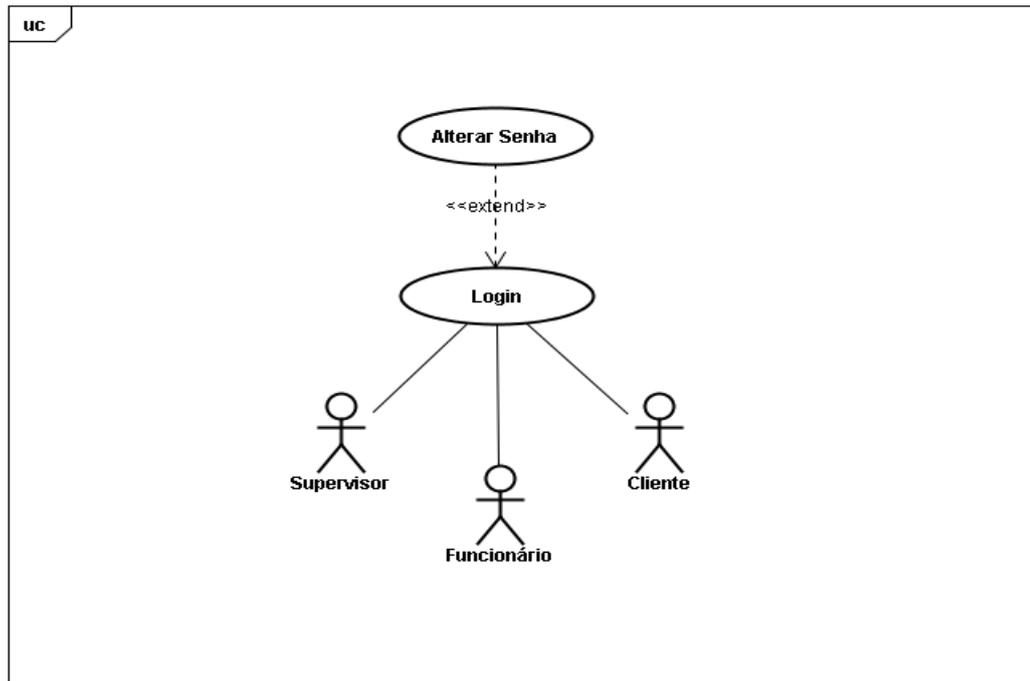


Figura 4º: Diagrama de Caso de Uso: Login

6.5 Diagrama de Classe

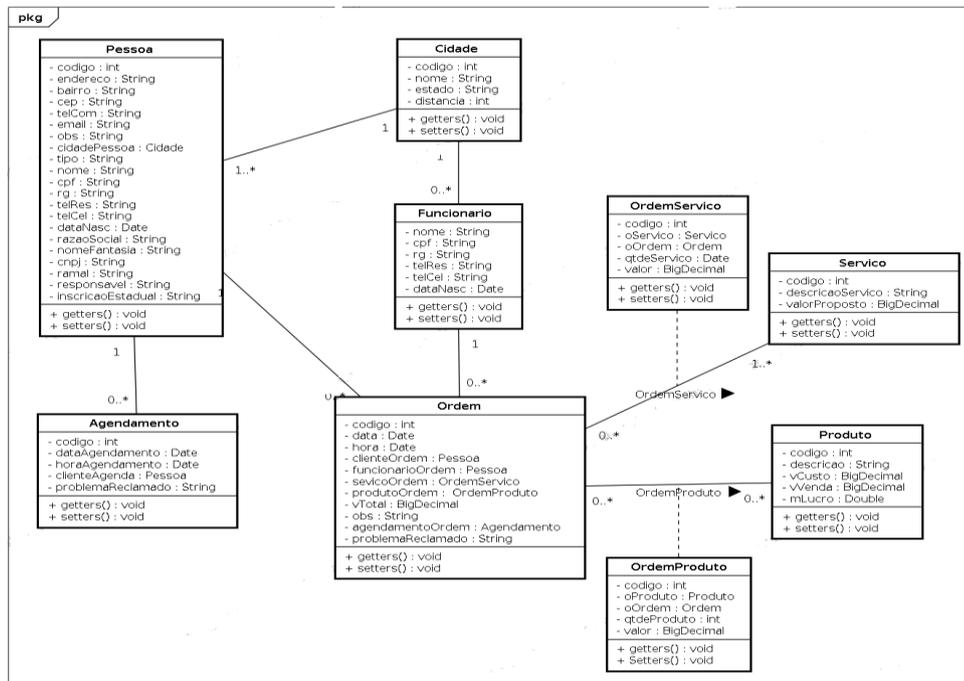


Figura 5º: Diagrama de Classe

6.6 Diagrama de Atividade Cliente

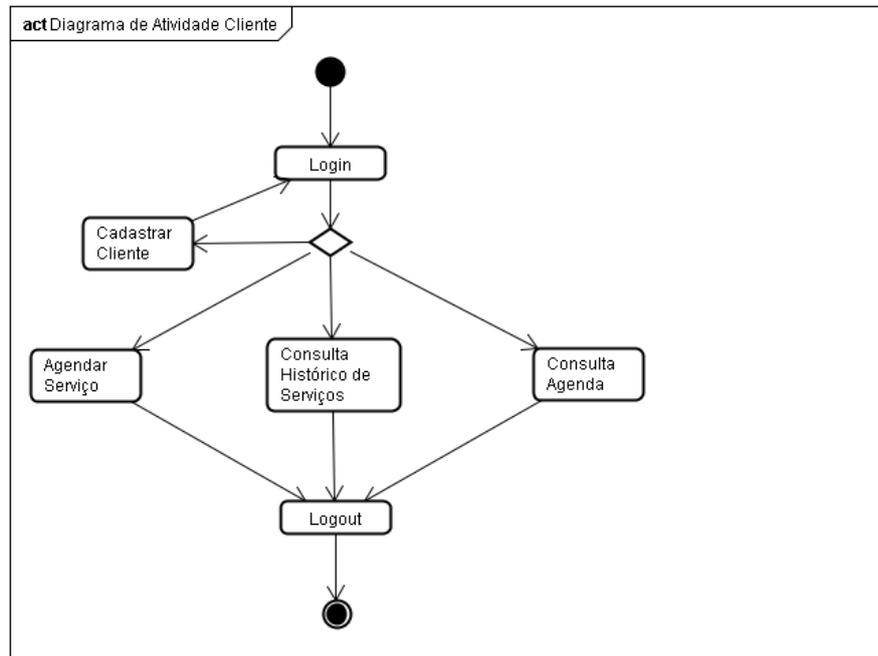


Figura 6º: Diagrama de Atividade Cliente

6.7 Diagrama de Atividade Funcionário: Manter

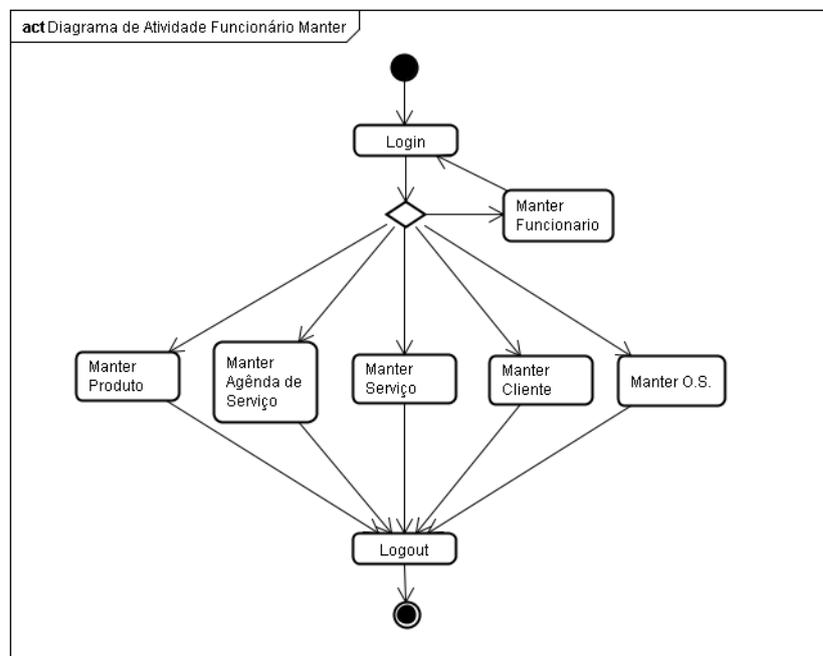


Figura 7º: Diagrama de Atividade Funcionário: Manter

6.8 Diagrama de Atividade Funcionário: Relatório

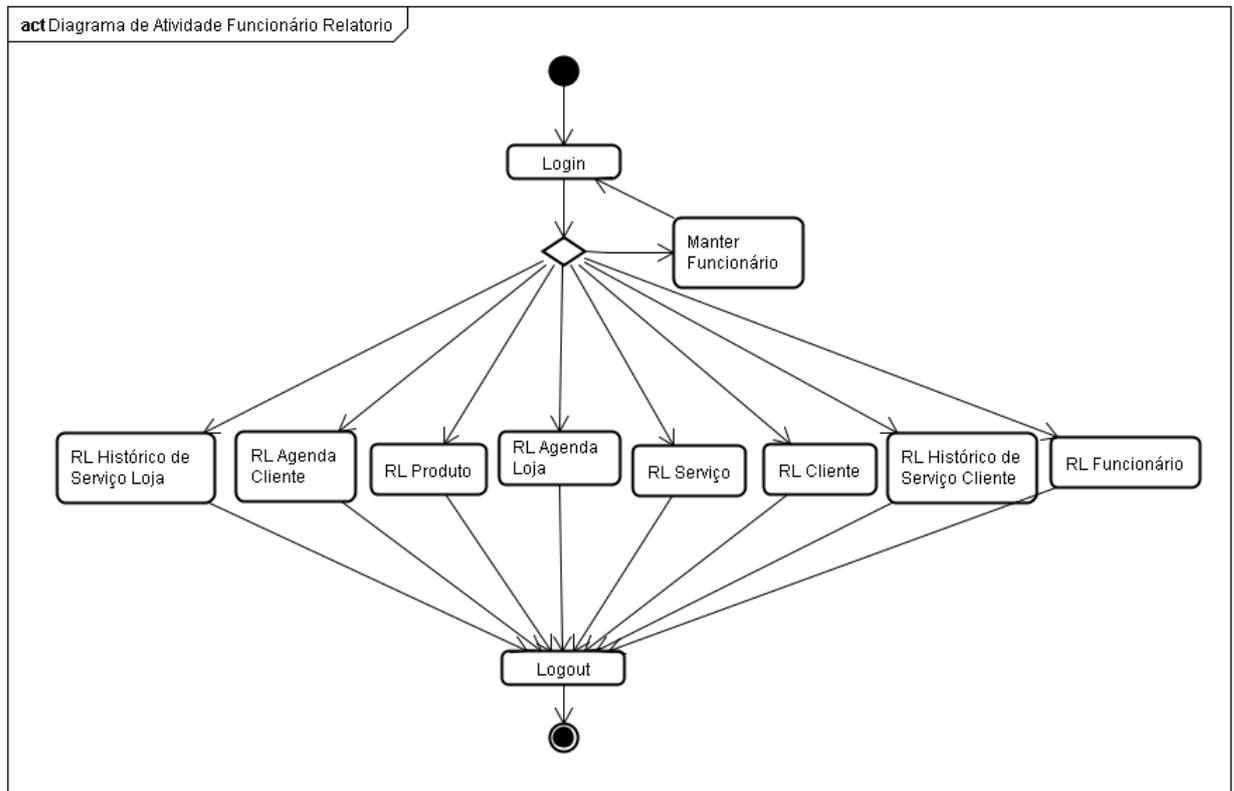


Figura 8º: Diagrama de Atividade Funcionário: Relatório

6.9 Tela de Cadastro de Cliente.

Cadastros Agendamentos O.S. Relatórios

CADASTRO DE PESSOA FÍSICA

Identificação

Nome:

Cpf:

Rg:

Telefone Residencial:

Telefone Celular:

Data de Nascimento:

Login

E-mail:

Senha:

Confirmação da Senha:

Endereço

Endereço:

Bairro:

Cep:

Cidade:

Obs:

Figura 9º: Tela de Cadastro de Cliente.

6.10 Tela de Cadastro de Produto.

Locais Sistema

produto

localhost:8080/InfoOnLine/faces/Produto.jsp

Cadastros Agendamentos O.S. Relatórios

Cadastro de Produto

Descrição:

Valor de Custo:

Valor de Proposto:

Margem de Lucro:

Figura 10º: Tela de Cadastro de Produto.

6.11 Tela de Consulta de Cliente.

Pesquisar Cliente

Selecione o Tipo do Cliente: Todos

Selecione o Tipo da Pesquisa: Nome Pesquisar

Clientes Cadastrados

Codigo	Nome / Razão Social	Tipo	Telefone Comercial	Tel. Celular	Endereço	Cidade	Editar	Excluir
1	EDER RIBEIRO DA SILVA	Pessoa Física	(18) 33411996	(18) 96059270	RUA JOSÉ TEODORO RAMOS, 456	ASSIS	Editar	Excluir
2	TORRESMO E TORRESMO COM. DE TORRESMO LTDA	Pessoa Jurídica	(18) 33411997		RUA DOS TORRESMOS, 1000	IBIRAREMA	Editar	Excluir
3	JANAINA SALATINE	Pessoa Física	(18) 33411414	(18)81188181	RUA BLÁ BLÁ BLÁ, 345	ASSIS	Editar	Excluir

Voltar Novo

Figura 11º: Tela de Consulta de Cliente.

6.12 Tela de Consulta de Cidade.

Pesquisar Cidades

Pesquisar por: Nome Pesquisar

Cidades Cadastradas

Nome	Estado	Distância	Editar	Excluir
ASSIS	SP	12	Editar	Excluir
PALMITAL	SP	30	Editar	Excluir
CANDIDO MOTA	SP	0	Editar	Excluir
TARUMÃ	SP	40	Editar	Excluir
FRUTAL DO CAMPO	SP	20	Editar	Excluir
NOVA ALEXANDRIA	SP	13	Editar	Excluir
IBIRAREMA	SP	50	Editar	Excluir
ECHAPORÃ	SP	45	Editar	Excluir

Voltar Novo

Figura 12º: Tela de Consulta de Cidade.

6.13 Código de Consulta de Cidade.

Classe: CidadeMB

```
package br.com.vision.navegacao;
```

```
import java.io.Serializable;
```

```
import java.util.ArrayList;
```

```
import java.util.List;

import javax.faces.event.ActionEvent;
import javax.faces.model.SelectItem;
import br.com.vision.beans.Cidade;
import br.com.vision.dao.CidadeDao;
import br.com.vision.util.Constantes;
import br.com.vision.util.FacesUtility;

public class CidadeMB implements Serializable{
    private static final long serialVersionUID = 1L;
    private Cidade cidade;
    private List<Cidade> cidades = new ArrayList<Cidade>();
    private String textoPesquisa;
    private List<SelectItem> comboPesquisa = new ArrayList<SelectItem>();
    private String pesquisaSelecionada;
    private List<SelectItem> comboEstado = new ArrayList<SelectItem>();
    private String estadoSelecionado;
    private boolean inclusao;
    private String detalheErro;

    public String prepararConsulta(){
        try {
            cidades = new CidadeDao().lista();
            listaComboPesquisa();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return "consultaCidade";
    }

    public String prepararCadastro() throws Exception{
```

```
        cidade = new Cidade();
        listaComboEstado();
        inclusao = true;
        return "cadastroCidade";
    }

    public void pesquisar(ActionEvent evt){
        try {
            if (Constantes.NOME.equals(pesquisaSelecionada)){
                cidades = new
CidadeDao().cidadePorNome(textoPesquisa);
            }else {
                cidades = new
CidadeDao().cidadePorEstado(textoPesquisa);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        textoPesquisa = "";
    }

    public String salvar(){
        try {
            cidade.setEstado(estadoSelecionado);
            if (inclusao) {
                new CidadeDao().salvar(cidade);
                cidades.add(cidade);
            }
            else{
                new CidadeDao().atualizar(cidade);
            }
        }
```

```

        } catch (Exception e) {
            e.printStackTrace();
        }
        return "consultaCidade";
    }
    public String editar(){
        cidade = (Cidade) FacesUtility.recuperaLinha("cidade");
        listaComboEstado();
        estadoSelecionado = cidade.getEstado();
        inclusao = false;
        return "cadastroCidade";
    }
    public void excluir(ActionEvent evt) {
        cidade = (Cidade) FacesUtility.recuperaLinha("cidade");
        try {
            new CidadeDao().deletar(cidade);
            cidades.remove(cidade);

        } catch (Exception e) {
            setDetalleErro("Não foi possível excluir: " + e.getMessage());
            e.printStackTrace();
        }
    }
    private void listaComboPesquisa(){
        comboPesquisa = new ArrayList<SelectItem>();
        comboPesquisa.add(new SelectItem(Constants.NOME));
        comboPesquisa.add(new SelectItem(Constants.ESTADO));
    }
    private void listaComboEstado(){
        comboEstado.clear();
        comboEstado.add(new SelectItem(""));
        comboEstado.add(new SelectItem(Constants.AC));
    }

```

```
        comboEstado.add(new SelectItem(Constants.AL));
        comboEstado.add(new SelectItem(Constants.AM));
        comboEstado.add(new SelectItem(Constants.AP));
        comboEstado.add(new SelectItem(Constants.BA));
        comboEstado.add(new SelectItem(Constants.CE));
        comboEstado.add(new SelectItem(Constants.DF));
        comboEstado.add(new SelectItem(Constants.ES));
        comboEstado.add(new SelectItem(Constants.GO));
        comboEstado.add(new SelectItem(Constants.MA));
        comboEstado.add(new SelectItem(Constants.MT));
        comboEstado.add(new SelectItem(Constants.MS));
        comboEstado.add(new SelectItem(Constants.MG));
        comboEstado.add(new SelectItem(Constants.PA));
        comboEstado.add(new SelectItem(Constants.PB));
        comboEstado.add(new SelectItem(Constants.PR));
        comboEstado.add(new SelectItem(Constants.PE));
        comboEstado.add(new SelectItem(Constants.PI));
        comboEstado.add(new SelectItem(Constants.RJ));
        comboEstado.add(new SelectItem(Constants.RN));
        comboEstado.add(new SelectItem(Constants.RS));
        comboEstado.add(new SelectItem(Constants.RO));
        comboEstado.add(new SelectItem(Constants.RR));
        comboEstado.add(new SelectItem(Constants.SC));
        comboEstado.add(new SelectItem(Constants.SP));
        comboEstado.add(new SelectItem(Constants.SE));
        comboEstado.add(new SelectItem(Constants.TO));
    }

    public Cidade getCidade() {
        return cidade;
    }

    public void setCidade(Cidade cidade) {
        this.cidade = cidade;
    }
}
```

```
}  
public List<Cidade> getCidades() {  
    return cidades;  
}  
public void setCidades(List<Cidade> cidades) {  
    this.cidades = cidades;  
}  
public String getTextoPesquisa() {  
    return textoPesquisa;  
}  
public void setTextoPesquisa(String textoPesquisa) {  
    this.textoPesquisa = textoPesquisa;  
}  
public List<SelectItem> getComboPesquisa() {  
    return comboPesquisa;  
}  
public void setComboPesquisa(List<SelectItem> comboPesquisa) {  
    this.comboPesquisa = comboPesquisa;  
}  
public String getPesquisaSelecionada() {  
    return pesquisaSelecionada;  
}  
public void setPesquisaSelecionada(String pesquisaSelecionada) {  
    this.pesquisaSelecionada = pesquisaSelecionada;  
}  
public List<SelectItem> getComboEstado() {  
    return comboEstado;  
}  
public void setComboEstado(List<SelectItem> comboEstado) {  
    this.comboEstado = comboEstado;  
}  
public String getEstadoSelecionado() {
```

```

        return estadoSelecionado;
    }
    public void setEstadoSelecionado(String estadoSelecionado) {
        this.estadoSelecionado = estadoSelecionado;
    }
    public String getDetalheErro() {
        return detalheErro;
    }
    public void setDetalheErro(String detalheErro) {
        this.detalheErro = detalheErro;
    }
}

```

Classe: CidadeConsulta.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="f" uri="http://java.sun.com/jsf/core"%>
<%@ taglib prefix="h" uri="http://java.sun.com/jsf/html"%>
<%@ taglib prefix="t" uri="http://myfaces.apache.org/tomahawk"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Clientes Cadastrados</title>
</head>
<body>
<f:view>

```

```

<h:form>
    <t:saveState value="#{clienteMB}" />
    <t:saveState value="#{agendamentoMB}" />
    <t:saveState value="#{ordemServicoMB}" />
    <jsp:include page="/Menu.jsp"></jsp:include>
    <div class="formCentralizado">
        <fieldset>
            <legend>Pesquisar Cliente</legend>
            <TABLE>
                <tr>
                    <TD> <h:outputLabel value="Selecione o Tipo do
Cliente:"></h:outputLabel> </TD>
                    <TD>
                        <h:selectOneMenu
value="#{clienteMB.tipoClienteSelecioneado}" style="width: 120px">
                            <f:selectItems
value="#{clienteMB.comboTipoCliente}" />
                        </h:selectOneMenu>
                    </TD>
                </tr>
                <tr>
                    <td> <h:outputLabel value="Selecione o Tipo da
Pesquisa:"></h:outputLabel> </td>
                    <TD>
                        <h:selectOneMenu
value="#{clienteMB.pesquisaSelecioneada}" style="width: 80px" >
                            <f:selectItems
value="#{clienteMB.comboPesquisa}" />
                    </TD>
                </tr>
            </TABLE>
        </fieldset>
    </div>
</h:form>

```

```

        </h:selectOneMenu>
    </TD>
    <td style="width: 30px"></td>
    <td><h:inputText style="width: 250px"
value="#{clienteMB.textoPesquisa}"></h:inputText></td>
    <TD> <h:commandButton value="Pesquisar" style="width:
80px" actionListener="#{clienteMB.pesquisar}"></h:commandButton> </TD>
</tr>
</TABLE>
</fieldset>
<br>
<fieldset><legend>Clientes Cadastrados</legend>
    <h:dataTable value="#{clienteMB.clientes}" var="cliente" border="0"
width="100%" styleClass="tabela" headerClass="cabecalho" rowClasses="linha1,
linha2">
        <h:column id="column1">
            <f:facet name="header">
                <h:outputText value="Codigo"></h:outputText>
            </f:facet>
            <h:outputLabel
value="#{cliente.codigo}"></h:outputLabel>
        </h:column>
        <h:column id="column2">
            <f:facet name="header">
                <h:outputText value="Nome / Razão
Social"></h:outputText>
            </f:facet>
            <h:outputLabel value="#{cliente.nome}"></h:outputLabel>

```

```

</h:column>
<h:column id="column9">
    <f:facet name="header">
        <h:outputText value="Tipo"></h:outputText>
    </f:facet>
    <h:outputLabel value="#{cliente.tipo}"></h:outputLabel>
</h:column>
<h:column id="column4">
    <f:facet name="header">
        <h:outputText value="Telefone"></h:outputText>
    </f:facet>
    <h:outputLabel value="#{cliente.telRes}"></h:outputLabel>
</h:column>
<h:column id="column5">
    <f:facet name="header">
        <h:outputText value="Endereço"></h:outputText>
    </f:facet>
    <h:outputLabel
value="#{cliente.endereco}"></h:outputLabel>
</h:column>
<h:column id="column3">
    <f:facet name="header">
        <h:outputText value="Bairro"/>
    </f:facet>
    <h:outputLabel value="#{cliente.bairro}"/>
</h:column>

```

```

<h:column id="column6">
    <f:facet name="header">
        <h:outputText value="Cidade"></h:outputText>
    </f:facet>
    <h:outputLabel
value="#{cliente.cidadeCliente.nome}"></h:outputLabel>
</h:column>
<h:column id="column7">
    <f:facet name="header">
        <h:outputText value="Editar"></h:outputText>
    </f:facet>
    <h:commandLink value="Editar"
action="#{clienteMB.editar}"></h:commandLink>
</h:column>
<h:column id="column8">
    <f:facet name="header">
        <h:outputText value="Excluir"></h:outputText>
    </f:facet>
    <h:commandLink value="Excluir"
actionListener="#{clienteMB.excluir}" onclick="return confirm('Confirma a
exclusão?');" ></h:commandLink>
</h:column>
</h:dataTable></fieldset>
<div>
    <h:commandButton id="btVoltar" value="Voltar"
action="inicial"></h:commandButton>

```

```

        <h:commandButton id="btNovo" value="Novo"
action="#{clienteMB.prepararCadastro}"></h:commandButton>

        <h:outputLabel value="#{clienteMB.detalheErro}"
></h:outputLabel>

    </div>

</div>

</h:form>

</f:view>

</body>

</html>

```

7. CRONOGRAMA DA ESTRUTURA DE DESENVOLVIMENTO

Cronograma											
Tarefa	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
Definição do orientador	■										
Definição do sistema		■									
Levantamento de requisitos			■								
Desenvolvimento do Pré-projeto			■	■							
Entrega do Pré-projeto				■							
Análise do Sistema				■	■	■	■	■	■	■	■
Desenvolvimento do Sistema							■	■	■	■	■
Periodo de Testes										■	
Entrega da Monografia									■		
Conclusão e Defesa do TCC										■	
Implantação do Sistema											■

Figura 13º: Cronograma

8. CONCLUSÃO

Com o término do sistema, obtiveram-se resultados que permite o desenvolvedor apresentar a seguinte conclusão:

O sistema atende plenamente o que foi proposto, controle e agendamento de serviços on-line, está apto a atender empresas de menor porte e com poucas modificações, atenderá empresas de médio porte.

A decisão do desenvolvimento deste software foi da maneira mais pertinente no momento, pois não só queria desenvolver o sistema, mas também aprender especificamente as tecnologias envolvidas no mesmo.

9. REFERÊNCIAS

BURNETTE, Ed. Eclipse IDE: Guia de Bolso, 1ª ed. João Tortello, Porto Alegre: Bookman, 2006.

FREEMAN, Eric; FREEMAN, Elizabeth. Use a Cabeça! Padrões de Projetos, 2ª ed. Rio de Janeiro: Editora Alta Books, 2007.

HSQLDB. Disponível em: <<http://www.hsqldb.org/>>. Disponível em: 26 Jul. 2010.

JAVAFREE. Hibernate. Disponível em: <<http://javafree.uol.com.br/wiki/hibernate>> Disponível em: 26 Jul. 2010.

MACORATTI, J. Carlos. Padrões de Projeto : O modelo MVC - Model View Controller. Disponível em: <http://www.macoratti.net/vbn_mvc.htm>. Disponível em: 26 Jul. 2010.

MCLAUGHLIN, Brett. Use a Cabeça! Análise e Projeto Orientado ao Objeto, 2ª ed. Rio de Janeiro: Editora Alta Books, 2007.

ORACLE. JSF. Disponível em: <<http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>>. Disponível em: 26 Jul. 2010.

SIERRA, Kathy; BATES, Bert. Use a Cabeça! Java, 6ª Ed. São Paulo: Editora Alta Books, 2007.

SIERRA, Kathy; BASHAM, Bryan. Use a Cabeça! Servlets & JSP, 1ª Ed. Rio de Janeiro: Editora Alta Books, 2005.

SUN. JSP. Disponível em: <<http://java.sun.com/products/jsp/>>. Disponível em: 26 Jul. 2010.