

**DANIELLE MATIAS MATUDA**

**SISTEMA WEB PARA GERENCIAMENTO DE SALÃO DE BELEZA DE  
PEQUENO E MÉDIO PORTE**

**Assis - SP  
2012**

**DANIELLE MATIAS MATUDA**

**SISTEMA WEB PARA GERENCIAMENTO DE SALÃO DE BELEZA DE  
PEQUENO E MÉDIO PORTE**

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

Orientanda: Danielle Matias Matuda  
Orientador: Prof. Dr. Luiz Carlos Begosso

**Assis - SP**  
2012

## FICHA CATALOGRÁFICA

MATIAS MATUDA, Danielle

SISTEMA WEB PARA GERENCIAMENTO DE SALÃO DE BELEZA DE PEQUENO E MÉDIO PORTE / Danielle Matias Matuda. Fundação Educacional do Município de Assis – FEMA – Assis, 2012.

51p.

Orientador: Luiz Carlos Begosso

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

1. Salão de beleza. 2. Sistema *Web*

CDD: 001.61  
Biblioteca da FEMA

# **SISTEMA WEB PARA GERENCIAMENTO DE SALÃO DE BELEZA DE PEQUENO E MÉDIO PORTE**

**DANIELLE MATIAS MATUDA**

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

Orientador: Luiz Carlos Begosso  
Avaliador: Domingos de Carvalho Villela Junior

**Assis - SP**  
2012

## DEDICATÓRIA

Com muita saudade e todo o meu amor,  
dedico este trabalho a vovó Dina, que tanto  
sonhou com esse momento.

## AGRADECIMENTOS

Para a realização desta conquista, foram tantos que contribuíram, e por eles tenho eterna gratidão.

A Deus, luz divina, que me guiou neste percurso.

Aos meus pais, que ao longo dos anos, me deram estrutura para trilhar todas as fases que a vida proporciona.

Aos meus irmãos, que sempre fazem os meus dias mais felizes.

Aos amigos de tempo.

Aos amigos que tive o prazer de conhecer durante esses três anos de faculdade, que foram verdadeiros companheiros e fonte de motivação.

A todos os mestres, pelos valiosos ensinamentos e dedicação.

Ao meu professor e orientador, Luiz Carlos Begosso, por toda compreensão, apoio e sabedoria transmitida, deixo minha admiração.

Ao meu professor, Domingos, pela constante paciência e desprendimento nas sabias e valiosas lições.

E a todos aqueles, que de alguma maneira, participaram na realização desse trabalho.

Com essas resumidas palavras, deixo minha felicidade, e o meu sincero agradecimento a cada um. Muito Obrigada!

"Ama-se mais o que se conquista com esforço."

Benjamin Disraeli

## RESUMO

O setor de tecnologia da informação, nos últimos tempos, cresce em grandes medidas. A utilização de *softwares* específicos tem salientado tal informação.

O ramo de salão de beleza, no Brasil, é um dos setores de maior potencial, visto que a preocupação com os padrões estéticos tendem a aumentar.

O sistema computacional *Bela Mais* foi desenvolvido com o intuito de suprir a necessidade deste novo setor emergente. Trata-se um *software* específico, que visa automatizar a rotina dos salões, com funcionalidades como: cadastros, agendamentos, encerramento de caixa e relatório diversos.

Para esta implementação, utilizou-se da tecnologia *Java Web*, o que proporcionou ao sistema mobilidade e independência.

Palavras - chave: Salão de beleza, Sistema *Java Web*

## **ABSTRACT**

The information technology sector in current times, grows in large measures. The use of specific software has emphasized such information.

The beauty salon industry in Brazil is one of the sectors with the greatest potential, since the concern with the aesthetic standards tend to increase.

The "Most Beautiful" computational system was developed in order to meet the need of this new emerging sector. This is a specific software, which aims to automate routine beauty saloons, with features such as: records, schedules, closing cash and several reports.

For this implementation, was used the Java Web technology, which provided the system mobility and independence.

Keywords - Keywords: Beauty, Java System Web

## LISTA DE ILUSTRAÇÕES

Figura 1. <i>Work Breakdown Structure</i> .....	24
Figura 2. Mapa Mental .....	25
Figura 3. Diagrama de Caso de Uso Geral .....	26
Figura 4. Diagrama de Classes .....	44
Figura 5. Diagrama Entidade e Relacionamento.....	45
Figura 6. Cronograma. ....	49

## LISTA DE TABELAS

Tabela 1. Efetuar Login .....	27
Tabela 2. Cadastrar Usuário .....	28
Tabela 3. Cadastrar Cliente.....	29
Tabela 4. Cadastrar Fornecedor .....	30
Tabela 5. Cadastrar Produto .....	31
Tabela 6. Cadastrar Serviço.....	32
Tabela 7. Cadastrar Forma de Pagamento .....	33
Tabela 8. Agendar Cliente.....	34
Tabela 9. Realizar Venda .....	35
Tabela 10. Encerramento de Caixa .....	36
Tabela 11. Gerar Relatório de Cliente.....	38
Tabela 12. Gerar Relatório de Produto .....	39
Tabela 13. Gerar Relatório de Serviço .....	40
Tabela 14. Gerar Relatório de Movimentação.....	42
Tabela 15. Custo Pessoal .....	47
Tabela 16. Custo Total .....	48

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>14</b>
1.1	Objetivo .....	14
1.2	Público Alvo .....	14
1.3	Justificativa .....	15
1.4	Estrutura do Trabalho.....	15
<b>2</b>	<b>TECNOLOGIAS E FERRAMENTAS DE DESENVOLVIMENTO.....</b>	<b>17</b>
2.1	Linguagem de programação: Java .....	17
2.2	Plataforma de desenvolvimento: NetBeans IDE 7.1 .....	17
2.3	Servidor de aplicação: Apache TomCat .....	18
2.4	Criação de relatórios: IReport .....	18
2.5	Banco de dados: MySQL.....	18
2.6	Hibernate .....	18
2.6	RIA .....	19
<b>3</b>	<b>TEC. E FERRAMENTAS PARA A PREPARAÇÃO DA ANÁLISE.....</b>	<b>21</b>
3.1	UML: Astah Community .....	21
3.2	UML: WBS Chart Pro.....	21
3.3	UML: iMindMap .....	22
3.4	UML: DB Designer.....	22
<b>4</b>	<b>ESTRUTURA DE DESENVOLVIMENTO DO SISTEMA.....</b>	<b>23</b>
4.1	Técnica para o levantamento de requisitos: Entrevista.....	23
4.2	Especificação dos requisitos .....	23
4.3	Validação dos requisitos .....	23
<b>5</b>	<b>MÉTODO DE DESENVOLVIMENTO .....</b>	<b>24</b>
<b>6</b>	<b>MAPA MENTAL .....</b>	<b>25</b>
<b>7</b>	<b>CASOS DE USO .....</b>	<b>26</b>
7.1	Especificação dos Casos de Uso .....	27
<b>8</b>	<b>DIAGRAMA DE CLASSES .....</b>	<b>44</b>
<b>9</b>	<b>DIAGRAMA DE ENTIDADE E RELACIONAMENTO.....</b>	<b>45</b>
<b>10</b>	<b>IMPLEMENTAÇÃO .....</b>	<b>46</b>

<b>11 ORÇAMENTO</b> .....	<b>47</b>
11.1 Recurso Humano .....	47
11.2 Custo Material .....	47
<b>12 CRONOGRAMA</b> .....	<b>49</b>
<b>13 CONCLUSÃO</b> .....	<b>50</b>
<b>REFERÊNCIAS</b> .....	<b>51</b>

## **1. INTRODUÇÃO**

Com a inclusão digital acontecendo cada vez mais rápido, as pessoas, tem a necessidade de ferramentas que proporcionem facilidade no decorrer do dia a dia. Os sistemas computacionais estão se firmando como uma importante ferramenta no auxílio das tarefas, sejam elas domésticas, ou profissionais.

Visando o expoente crescimento nos cuidados com os padrões estéticos, e conseqüentemente, o aumento de salões de beleza, o presente trabalho propõe a elaboração de um *software* que atenda as rotinas envolvidas em um salão de beleza. O *software* será denominado de “Bela Mais”.

O sistema enquadra-se no uso de empresas de pequeno e médio porte, que necessitam de um controle específico para o ramo.

O gerenciamento de informação dos salões, na sua maioria, é realizado de forma manual, sem qualquer tipo de automação. Ou seja, ficam sujeitos a diversos tipos de falhas, tem gasto de tempo elevado, e dados imprecisos.

As principais funcionalidades apresentadas pelo sistema computacional “Bela Mais”, serão: cadastro de clientes, cadastro de produtos e serviços, agenda de horários, fechamento de caixa, bem como apresentação de dados por meio de relatórios gerenciais.

O desenvolvimento do sistema *web* “Bela Mais”, proporcionará a administração do fluxo de informações, contribuindo com o crescimento da empresa.

### **1.1 OBJETIVO**

O principal foco, do sistema proposto, é a automatização das etapas presentes na rotina de um salão de beleza. Visando a organização dos processos, bem como, tornando-os mais ágeis, reduzindo despesas, e inibindo a ocorrência de possíveis falhas humanas. Assim, contribuindo de maneira direta para a lucratividade da organização.

### **1.2 PÚBLICO ALVO**

O sistema computacional “Bela Mais”, destina-se a empresas de pequeno e médio porte, que atuam no setor de salões de beleza.

### **1.3 JUSTIFICATIVA**

Nos dias atuais, faz-se visível a preocupação dos indivíduos, de todas as classes sociais, com a própria imagem ou aparência. Em um tempo moderno, e competitivo, no qual vivenciamos, é primordial o cuidado com o exterior. As pessoas estão cada vez mais conscientes disso, fato que indica o crescimento volumoso de salões de beleza.

De acordo com estudos, o segundo segmento mais vislumbrado por novos empreendedores no país é o de salão de beleza. A capital São Paulo, conta com aproximadamente 11 (onze) mil salões de beleza, o que significa uma média de 150 (cento e cinquenta) por bairro.

Apesar de um mercado aquecido, ainda existem lacunas nesse setor, tal como programas computacionais específicos, para gerir a demanda de informações presente neste ramo.

Com base nessas informações, verificou-se a necessidade de um *software* que atendesse com clareza e simplicidade os requisitos, e as restrições contidas na demanda de um salão de beleza.

### **1.4 ESTRUTURA DO TRABALHO**

O presente trabalho tem contido em sua estrutura doze capítulos, conforme apresentados a seguir.

O primeiro capítulo é a Introdução do sistema Bela Mais, este possui quatro seções: objetivo, público alvo, justificativa e este narrando à estrutura do trabalho.

O segundo capítulo, diz respeito a Tecnologias e ferramentas de desenvolvimento. Ele está dividido em sete seções: linguagem de programação, plataforma de desenvolvimento, servidor de aplicação, criação de relatórios, banco de dados, Hibernate, e por fim, RIA.

O terceiro capítulo é composto pelas tecnologias e ferramentas para a realização de análise. Nele, estão quatro seções: UML: Astah Community, UML: WBS Chart Pro, UML: iMindMap, e UML: DBDesigner.

No capítulo de número quatro, está a estrutura de desenvolvimento do sistema. Este possui três seções: técnica para o levantamento de requisitos, especificação dos requisitos e validação dos requisitos.

O quinto capítulo é dedicado ao método de desenvolvimento.

O sexto capítulo é voltado ao mapa mental.

No capítulo de número sete são apresentados os casos de uso. E em uma única seção, a especificação dos mesmos.

O oitavo capítulo, apresenta o diagrama de classes.

Em relação ao capítulo nove, nele é exposto o diagrama Entidade Relacionamento.

O décimo capítulo menciona sobre a implementação do sistema.

No décimo primeiro capítulo, é apresentado o cronograma do desenvolvimento do sistema.

E por fim, o capítulo doze apresenta as conclusões, e considerações finais sobre o desenvolvimento do sistema.

## 2. TECNOLOGIAS E FERRAMENTAS DE DESENVOLVIMENTO

Abaixo, serão apresentadas as tecnologias, e ferramentas utilizadas no desenvolvimento do sistema Bela Mais.

### 2.1 Linguagem de programação: Java

O sistema *web* proposto, será desenvolvido na linguagem de programação de alto nível Java. Dentre os fatores que contribuíram para a escolha desta tecnologia, está o fato de sua licença ser livre, ou seja, não é necessário pagar pela sua utilização.

A linguagem de programação Java, segundo Gonçalves (2007), surgiu em maio do ano de 1995, proposta pela empresa Sun Microsystems.

A tecnologia Java, que é orientada a objetos, possui como características principais:

- Portabilidade - Não depende de plataforma para execução;
- Recursos de rede - Facilita a colaboração com protocolos;
- Grande conjunto de bibliotecas;
- Concorrência - permite o uso de threads múltiplos e mecanismos explícitos de concorrência;
- Tratamento de eventos e exceções - Permite a programação orientada a eventos e a alternativa de escrever sistemas que respondem a falhas de execução

De acordo com Horstmann e Cornell (2001), autores da linguagem Java organizaram um documento, resumindo as principais características propostas pela linguagem. São elas: simples, orientada a objetos, distribuída, robusta, segura, neutra em relação à arquitetura, portátil, interpretada, alto desempenho, múltiplas linhas de execução e dinâmica.

### 2.2 Plataforma de desenvolvimento: NetBeans IDE 7.1

De acordo com Gonçalves (2008), NetBeans é um ambiente de desenvolvimento integrado IDE, escrito em linguagem Java. Caracteriza-se por ser multiplataforma, ter código aberto, e ser gratuito. Possui um vasto conjunto de bibliotecas e API's. Esta versão também permite ao desenvolvedor criar páginas dinâmicas utilizando JSP, *Servlets*, JSTL, padrões de desenvolvimento MVC e DAO, além de ser flexível quanto ao suporte de outras linguagens de programação, como *Ajax* e *Ruby*.

### **2.3 Servidor de aplicação: Apache TomCat**

O servidor Apache (ou servidor HTTP apache) foi desenvolvido no ano de 1995, pela Fundação Apache. Destaca-se pela excelente comunicação com aplicações *web*. Além de estar centrado na tecnologia Java, o TomCat está presente especificamente nas *Servlets* e Java Server Pages (JSP). Sua distribuição é gratuita.

### **2.4 Criação de relatórios: IReport**

De acordo com Gonçalves (2008), a ferramenta para elaboração de relatórios IReport é um *software Open Source*. Tem como parâmetro a criação de relatórios para aplicações Java, no formato da biblioteca JasperReports .

### **2.5 Banco de dados: MySQL**

O sistema gerenciador de banco de dados (SGDB) MySQL, foi criado na Suécia por volta de 1979, por Michael Widenius, na companhia suíça TcX. No entanto, sua primeira versão foi lançada apenas em maio de 1995.

O MySQL é um servidor multitarefa, com singularidade com o padrão SQL (*Structured Query Language* - Linguagem de Consulta Estruturada). Essa linguagem é vastamente utilizada para a manipulação de dados em Repositórios de Dados Relacionais.

As características que podem justificar a abrangência desta tecnologia são:

- Implementação leve - Significa maior velocidade as instruções SQL;
- Facilidade de ligação com servidor Web, e fácil integração com linguagens de programação;
- Estabilidade;
- Funcionalidades simples.

### **2.6 Hibernate**

O *framework* denominado de Hibernate tem como principal objetivo realizar o mapeamento objeto/relacional para a linguagem de programação Java, facilitando o

desenvolvimento de aplicações e proporcionando independência do banco de dados.

O OMR (mapeamento objeto/relacional) é a persistência automatizada dos objetos em uma aplicação Java para as tabelas de um banco de dados relacional.

O Hibernate detém de sua própria linguagem de *Structured Query Language* (SQL), conhecida como *Hibernate Query Language* (HQL). Onde a HQL é convertida para SQLs específicas de cada banco de dados.

De acordo com Sauv  (2007), o Hibernate possui as seguintes particularidades:

- O Hibernate persiste objetos Java comuns (POJO);
- Usa reflex o para acessar as propriedades persistentes de um objeto;
- As classes persistentes s o definidas (descritas) em documentos de mapeamento;
- Arquivos XML s o usados para descrever os campos, associa es e subclasses persistentes;
- Os Mapeamentos s o "compilados" na inicializa o da aplica o;
- Podem ser usados tamb m para opera es de suporte como: Gera o de esquemas do banco de dados e gera o de c digo-fonte Java.

Utilizando o Hibernate, o desenvolvedor de sistemas pode reduzir o trabalho, e economizar in meras repeti es de c digos, como "*inserts*", "*selects*", "*updates*", e "*deletes*" no banco de dados.

## **2.7 RIA (Ritch Internet Application): PrimeFaces**

O conceito de "Internet Rica", est  diretamente ligado com a interface *website* - usu rio. Um de seus principais objetivos,   tornar a intera o homem-m quina mais intuitiva e simples.

O padr o JSF (Java Server Faces), faz uso da descri o de componentes para a elabora o de interfaces de sistemas *web*. As *taglibs* presentes no JSF encapsulam o uso de *tags* HTML, proporcionando assim, o desenvolvedor de sistemas trabalhar em um n vel mais alto de abstra o.

O JSF também possui uma robusta API (*Application Programming Interface*), que encapsula o uso de AJAX nas páginas de uma aplicação. A partir destes conceitos de componentes, é que surgiram os *frameworks*. Estes, visam enriquecer e contribuir ainda mais o uso do JSF.

O PrimeFaces é um *framework* desenvolvido pela empresa turca Prime Teknoloji. Esse *framework* oferece um conjunto de componentes ricos para o JSF.

Os componentes foram desenvolvidos para trabalhar com AJAX por "*default*", ou seja, não é preciso que o desenvolvedor de sistemas tenha que realizar qualquer esforço extra para realização de chamadas assíncronas ao servidor.

### **3. TECNOLOGIAS E FERRAMENTAS PARA A PREPARAÇÃO DA ANÁLISE**

O sistema computacional proposto, na etapa de análise, utilizará a linguagem padrão de modelagem de dados orientado a objetos UML (*Unified Modeling Language*).

Segundo Guedes (2009, p.19), a UML:

“é uma linguagem visual utilizada para modelar softwares baseados no paradigma de orientação a objetos. É uma linguagem de modelagem de propósito geral que pode ser aplicada a todos os domínios de aplicação. Essa linguagem tornou-se, nos últimos anos, a linguagem padrão de modelagem adotada internacionalmente pela indústria de engenharia de software”.

É uma linguagem que abrange a modelagem de um simples sistema, até um de alta complexidade. Não é um método de desenvolvimento, e sim um suporte para a visualização do desenho e a comunicação entre objetos.

A UML permite ao desenvolvedor a visualização de seu produto, em forma de diagramas padronizados (representações gráficas do modelo parcial de um sistema), expandindo as visões sobre o programa computacional.

Abaixo, serão apresentadas as tecnologias e ferramentas utilizadas para a análise do sistema computacional *Bela Mais*.

#### **3.1 UML: Astah Community**

O Astah Community é uma ferramenta utilizada em todo o processo de modelagem de dados. Com ela, pode-se obter de forma fácil e eficaz, a elaboração de diversos diagramas, tais como: Diagrama de Caso de Uso, Diagrama de Classes, Diagrama de Sequência, Diagrama de Atividades, entre muitos outros.

#### **3.2 UML: WBS Chart Pro**

WBS Chart Pro é uma ferramenta utilizada para a criação de diagramas de WBS (*Work Breakdown Structure* – Estrutura Analítica do Projeto). Trata-se de uma representação gráfica, que possibilita o analista de sistemas visualizar de maneira expandida os processos contidos ao longo do projeto.

### **3.3 UML: iMindMap**

O programa computacional iMindMap trata-se de uma poderosa ferramenta de criação de Mapas Mentais. A sua utilização, permite ao analista de sistemas, através dos Mapas, abranger e conceber detalhes do projeto. Além de possibilitar a inserção de imagens.

### **3.4 UML: DB Designer**

O DB Designer é uma ferramenta utilizada para a criação do Diagrama Entidade Relacionamento (DER). Nele é desenvolvido o relacionamento entre as classes, permitindo maior agilidade ao programador no desenvolvimento do sistema.

## **4. ESTRUTURA DE DESENVOLVIMENTO DO SISTEMA**

Abaixo, serão apresentadas as etapas de estruturação do sistema computacional *Bela Mais*.

### **4.1 Técnica para o levantamento de requisitos: Entrevista.**

A técnica de apoio escolhida para auxiliar na elicitação dos requisitos foi a de entrevista. Deste modo, realizando perguntas sobre o sistema a ser desenvolvido, para o usuário, como por exemplo, a importância do *software* para a empresa, os benefícios esperados, os desejos, as restrições e funcionalidades. As respostas obtidas são os requisitos.

### **4.2 Especificação dos requisitos.**

Nesta etapa, os requisitos obtidos anteriormente são descritos e especificados com nível de detalhamento adequado ao sistema.

### **4.3 Validação dos requisitos.**

Após a especificação dos requisitos, a etapa de validação consiste em verificar se os requisitos obtidos estão de acordo com os propostos no começo do processo. Se tudo estiver coerente, a validação é realizada junto ao cliente, ou usuário.

## 5. MÉTODO DE DESENVOLVIMENTO

O desenvolvimento do programa computacional “Bela Mais”, ocorrerá por meio de etapas. Para melhor entendimento, as fases referidas serão apresentadas através do diagrama chamado *Work Breakdown Structure* (WBS). A WBS, ou Estrutura Analítica do Projeto, conforme tradução do PMBOK (*Project Management Body of Knowledge*), é uma ferramenta de separação do trabalho existente nos projetos em partes manejáveis. É caracterizado por níveis hierárquicos, formado pelos pacotes de trabalho que integram o projeto.

As fases para o desenvolvimento do sistema proposto, ilustrada na figura 1, visam exemplificar a organização do trabalho.

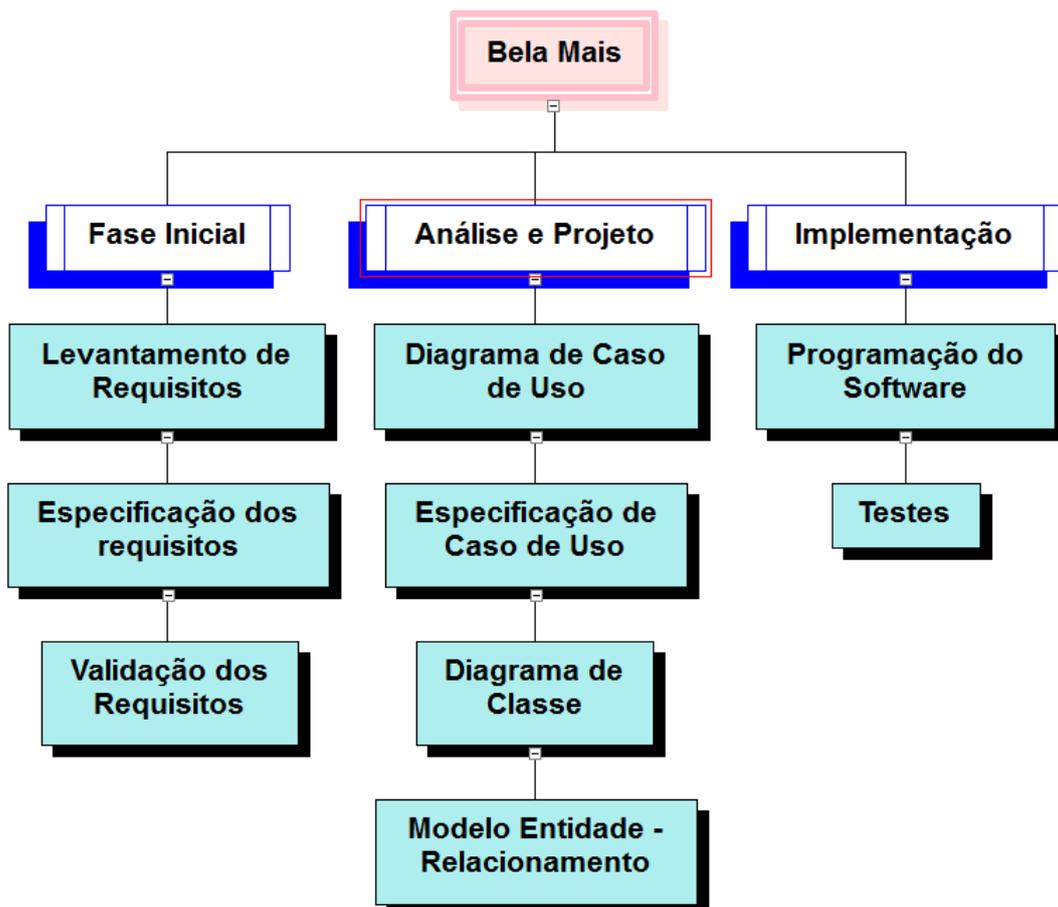


Figura 1 - *Work Breakdown Structure*

## 6. MAPA MENTAL

Para melhor entendimento e visualização do sistema é apresentado o Mapa Mental. O surgimento desta ferramenta deu-se na década de 1960, pelo inglês Tony Buzan.

Segundo Buzan (2009, p. 10):

“Os Mapas Mentais são um método de armazenar, organizar e priorizar informações (em geral no papel), usando Palavras-Chave e Imagens-Chave, que desencadeiam lembranças específicas e estimulam novas reflexões e ideias. Cada ativador da memória em um Mapa Mental é uma chave que dá acesso a fatos, ideias e informações, além de liberar o verdadeiro potencial da mente, de modo que possamos nos tornar o que quisermos ser”.

O método tem início com uma ideia central, que irradia novas outras ideias relacionadas, a figura 2 ilustra o Mapa Mental do sistema proposto.

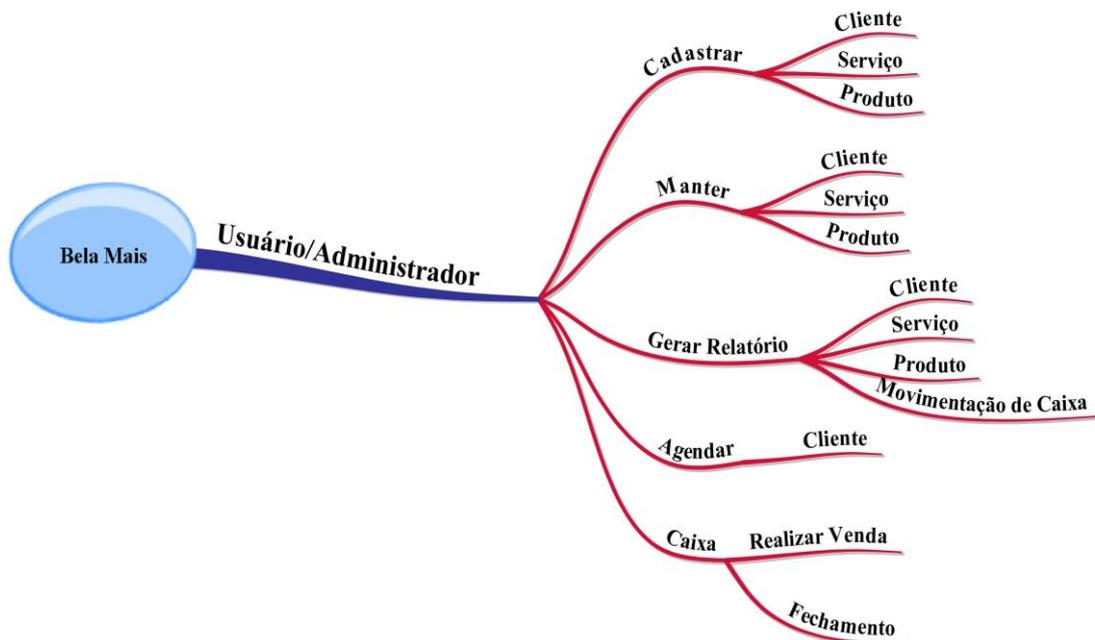


Figura 2 – Mapa Mental do sistema *Bela Mais*

## 7. CASOS DE USO

Como elemento básico da UML, utiliza-se Caso de Uso. Casos de Uso são representações, em diagramas, de cenários que visam um objetivo do usuário. Tais representações permite que o analista de sistemas, conheça melhor a interação desejada pelo usuário com o *software*. E que também possa compreender com maior segurança, os requisitos funcionais do sistema.

Segundo Pádua (2009), os casos de usos são empregados como descrições de funcionalidades específicas. Em sua elaboração são utilizadas figuras, elipses, relacionamentos, entre outros.

A ilustração abaixo exemplifica o caso de uso comum do sistema *Bela Mais*.

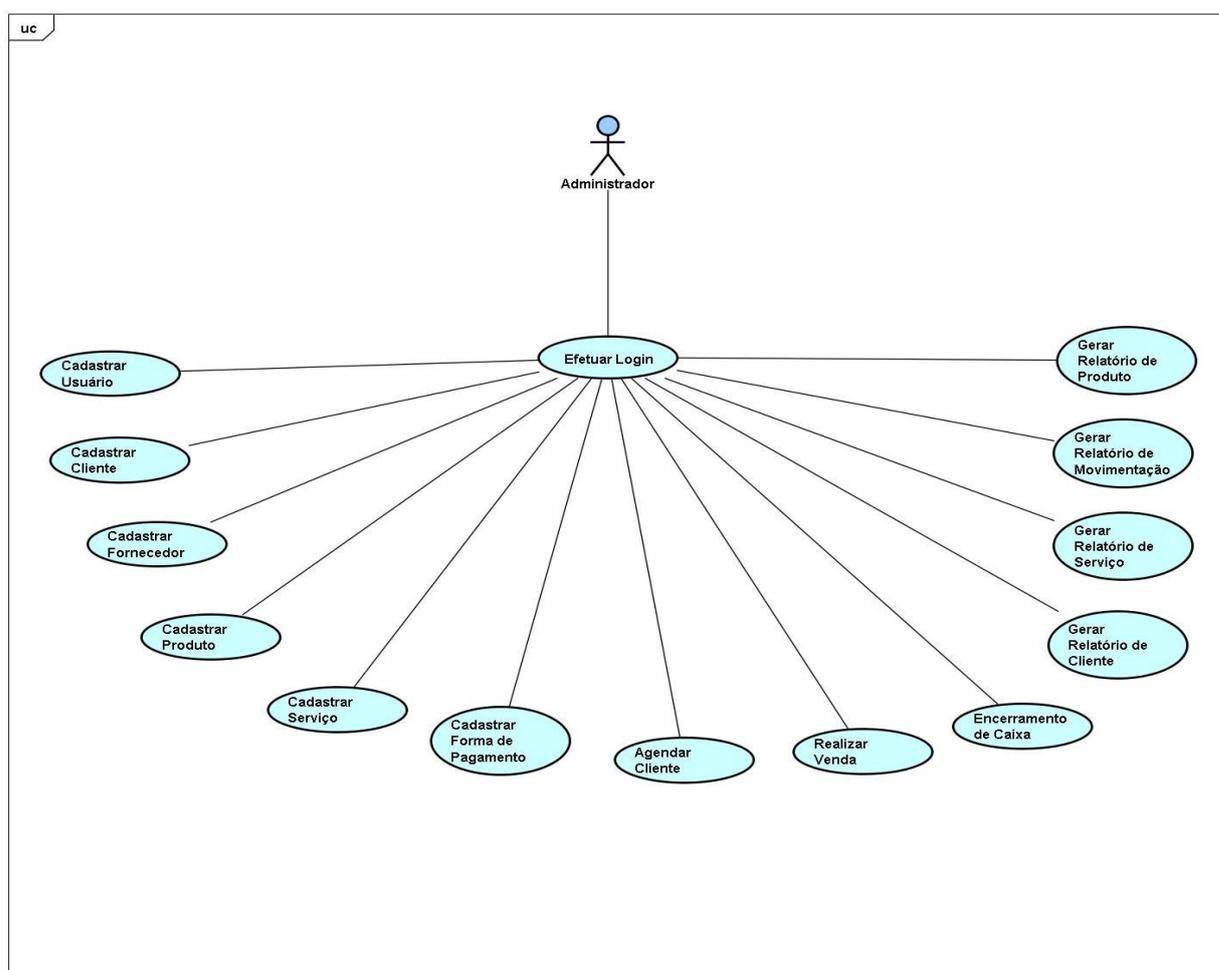


Figura 3 – Diagrama de Caso de Uso Geral do sistema computacional *Bela Mais*

## 7.1 Especificação dos Casos de Uso

<b>1. Finalidade/ Objetivo:</b>	Realizar login no sistema
<b>2. Ator (es):</b>	Administrador
<b>3. Pré – Condição:</b>	Obter senha de acesso
<b>4. Evento Inicial:</b>	O administrador entra com o login
<b>5. Fluxo Principal:</b>	<ul style="list-style-type: none"><li>a. O Administrador efetua login no sistema; (E1)</li><li>b. O caso de uso é encerrado</li></ul>
<b>6. Fluxo Alternativo:</b>	-
<b>7. Fluxo de Exceção:</b>	E1- Login Inválido <ul style="list-style-type: none"><li>a. A senha de acesso do administrador é inválida;</li><li>b. O caso de uso é encerrado.</li></ul>
<b>8. Pós – condições:</b>	<ul style="list-style-type: none"><li>a. O administrador poderá realizar cadastros;</li><li>b. O administrador poderá gerar relatórios;</li><li>c. O administrador poderá realizar agendamentos;</li><li>d. O administrador poderá efetuar vendas;</li><li>e. O administrador poderá finalizar o caixa.</li></ul>

<b>9. Casos de testes:</b>	Tentar o acesso ao sistema com senha errônea.
----------------------------	---

**Tabela 1 – Especificação de caso de uso - Efetuar Login**

<b>1. Finalidade/ Objetivo:</b>	Realizar o cadastro de usuário
<b>2. Ator (es):</b>	Administrador
<b>3. Pré - Condição:</b>	Efetuar login no sistema
<b>4. Evento Inicial:</b>	O Administrador escolhe a opção "Cadastrar Usuário"
<b>5. Fluxo Principal:</b>	<p>a. O administrador efetua login no sistema; (E1)</p> <p>b. O administrador escolhe a opção "Cadastrar Usuário", e efetua o cadastro; (A1)</p>
<b>6. Fluxo Alternativo:</b>	<p>A1 - Cancela a operação:</p> <p>a. O administrador cancela a opção;</p> <p>b. O sistema retorna ao passo 5.a podendo ou não cadastrar;</p> <p>c. O caso de uso é encerrado</p>
<b>7. Fluxo de exceção:</b>	<p>E1 - Login inválido:</p> <p>a. A senha de acesso do</p>

	administrador é inválida;  b. O caso de uso é encerrado
<b>8. Pós - condições:</b>	O usuário poderá utilizar o sistema
<b>9. Casos de teste:</b>	-

**Tabela 2 – Especificação de caso de uso - Cadastrar Usuário**

<b>1. Finalidade/ Objetivo:</b>	Realizar o cadastro de cliente
<b>2. Ator (es):</b>	Administrador
<b>3. Pré – Condição:</b>	Efetuar login no sistema
<b>4. Evento Inicial:</b>	O administrador escolhe a opção “Cadastrar Cliente”
<b>5. Fluxo Principal:</b>	<ul style="list-style-type: none"> <li>a. O administrador efetua login no sistema; (E1)</li> <li>b. O administrador escolhe a opção “Cadastrar Cliente”, e efetua o cadastro; (A1)</li> </ul>
<b>6. Fluxo Alternativo:</b>	<p>A1 – Cancela a operação:</p> <ul style="list-style-type: none"> <li>a. O administrador cancela a operação;</li> <li>b. O sistema retorna ao passo 5.a, podendo ou não cadastrar;</li> <li>c. O caso de uso é encerrado</li> </ul>
<b>7. Fluxo de exceção:</b>	<p>E1 – Login inválido:</p> <ul style="list-style-type: none"> <li>a. A senha de acesso do administrador é inválida;</li> </ul>

	b. O caso de uso é encerrado
<b>8. Pós – condições:</b>	-
<b>9. Casos de teste:</b>	Tentar realizar o cadastro duplicado de um mesmo cliente.

**Tabela 3 – Especificação de caso de uso – Cadastrar Cliente**

<b>1. Finalidade/ Objetivo:</b>	Realizar o cadastro de fornecedor
<b>2. Ator (es):</b>	Administrador
<b>3. Pré - Condição:</b>	Efetuar login no sistema
<b>4. Evento Inicial:</b>	O Administrador escolhe a opção "Cadastrar Fornecedor"
<b>5. Fluxo Principal:</b>	<p>a. O administrador efetua login no sistema; (E1)</p> <p>b. O administrador escolhe a opção "Cadastrar Fornecedor", e efetua o cadastro; (A1)</p>
<b>6. Fluxo Alternativo:</b>	<p>A1 - Cancela a operação:</p> <p>a. O administrador cancela a opção;</p> <p>b. O sistema retorna ao passo 5.a podendo ou não cadastrar;</p> <p>c. O caso de uso é encerrado</p>

<b>7. Fluxo de exceção:</b>	E1 - Login inválido:  a. A senha de acesso do administrador é inválida;  b. O caso de uso é encerrado
<b>8. Pós - condições:</b>	-
<b>9. Casos de teste:</b>	Tentar realizar o cadastro duplicado de um mesmo fornecedor.

**Tabela 4 – Especificação de caso de uso – Cadastrar Fornecedor**

<b>1. Finalidade/ Objetivo:</b>	Realizar o cadastro de produto
<b>2. Ator (es):</b>	Administrador
<b>3. Pré – Condição:</b>	Efetuar login no sistema
<b>4. Evento Inicial:</b>	O administrador escolhe a opção “Cadastrar Produto”
<b>5. Fluxo Principal:</b>	a. O administrador efetua login no sistema; (E1)  b. O administrador escolhe a opção “Cadastrar Produto”, e efetua o cadastro; (A1)
<b>6. Fluxo Alternativo:</b>	A1 – Cancela a operação:  a. O administrador cancela a operação;  b. O sistema retorna ao passo 5.a, podendo ou não cadastrar;

	c. O caso de uso é encerrado
<b>7. Fluxo de exceção:</b>	E1 – Login inválido:  a. A senha de acesso do administrador é inválida;  b. O caso de uso é encerrado
<b>8. Pós – condições:</b>	a. O produto poderá ser vendido
<b>9. Casos de teste:</b>	Tentar realizar o cadastro duplicado de um mesmo produto.

**Tabela 5 – Especificação de caso de uso – Cadastrar Produto**

<b>1. Finalidade/ Objetivo:</b>	Realizar o cadastro de serviço
<b>2. Ator (es):</b>	Administrador
<b>3. Pré – Condição:</b>	Efetuar login no sistema
<b>4. Evento Inicial:</b>	O administrador escolhe a opção “Cadastrar Serviço”
<b>5. Fluxo Principal:</b>	a. O administrador efetua login no sistema; (E1)  b. O administrador escolhe a opção “Cadastrar Produto”, e efetua o cadastro; (A1)
<b>6. Fluxo Alternativo:</b>	A1 – Cancela a operação:  a. O administrador cancela a operação;  b. O sistema retorna ao passo 5.a,

	<p>podendo ou não cadastrar;</p> <p>c. O caso de uso é encerrado</p>
<b>7. Fluxo de exceção:</b>	<p>E1 – Login inválido:</p> <p>a. A senha de acesso do administrador é inválida;</p> <p>b. O caso de uso é encerrado</p>
<b>8. Pós – condições:</b>	-
<b>9. Casos de teste:</b>	Tentar realizar o cadastro duplicado de um mesmo serviço.

**Tabela 6 – Especificação de caso de uso – Cadastrar Serviço**

<b>1. Finalidade/ Objetivo:</b>	Realizar o cadastro de Forma de Pagamento
<b>2. Ator (es):</b>	Administrador
<b>3. Pré - Condição:</b>	Efetuar login no sistema
<b>4. Evento Inicial:</b>	O ator escolhe a opção "Cadastrar Forma de Pagamento"
<b>5. Fluxo Principal:</b>	<p>a. O administrador efetua login no sistema; (E1)</p> <p>b. O administrador escolhe a opção "Cadastrar Forma de Pagamento e efetua o cadastro; (A1)</p>

<b>6. Fluxo Alternativo:</b>	<p>A1 - Cancela a operação:</p> <p>a. O administrador cancela a opção;</p> <p>b. O sistema retorna ao passo 5.a podendo ou não cadastrar;</p> <p>c. O caso de uso é encerrado</p>
<b>7. Fluxo de exceção:</b>	<p>E1 - Login inválido:</p> <p>a. A senha de acesso do administrador é inválida;</p> <p>b. O caso de uso é encerrado</p>
<b>8. Pós - Condições:</b>	Utilizar a nova forma de pagamento cadastrada
<b>9. Casos de Teste:</b>	-

**Tabela 7 – Especificação de caso de uso – Cadastrar Forma de Pagamento**

<b>1. Finalidade/ Objetivo:</b>	Realizar o agendamento de um cliente
<b>2. Ator (es):</b>	Administrador
<b>3. Pré – Condição:</b>	Efetuar login no sistema
<b>4. Evento Inicial:</b>	O administrador escolhe a opção “Agendar cliente”
<b>5. Fluxo Principal:</b>	<p>a. O administrador efetua login no sistema; (E1)</p> <p>b. O administrador escolhe a opção</p>

	“Agendar Cliente”, e efetua o agendamento (A1)
<b>6. Fluxo Alternativo:</b>	A1 – Cancela a operação: <ul style="list-style-type: none"> <li>a. O administrador cancela a operação;</li> <li>b. O sistema retorna ao passo 5.a, podendo ou não agendar;</li> <li>c. O caso de uso é encerrado</li> </ul>
<b>7. Fluxo de exceção:</b>	E1 – Login inválido: <ul style="list-style-type: none"> <li>a. A senha de acesso do administrador é inválida;</li> <li>b. O caso de uso é encerrado</li> </ul>
<b>8. Pós – condições:</b>	-
<b>9. Casos de teste:</b>	Tentar o agendamento de mais de um cliente no mesmo horário, e com o mesmo profissional (cabeleireiro /manicure)

**Tabela 8 – Especificação de caso de uso – Agendar Cliente**

<b>1. Finalidade/ Objetivo:</b>	Efetuar a realização de uma venda
<b>2. Ator (es):</b>	Administrador
<b>3. Pré – Condição:</b>	<ul style="list-style-type: none"> <li>a. Efetuar login no sistema;</li> <li>b. Ter o produto ou serviço cadastrados no sistema</li> </ul>

<b>4. Evento Inicial:</b>	O administrador escolhe a opção “Realizar Venda”
<b>5. Fluxo Principal:</b>	<ul style="list-style-type: none"> <li>a. O administrador efetua login no sistema; (E1)</li> <li>b. O administrador escolhe a opção “Realizar Venda”, e efetua a venda; (A1)</li> </ul>
<b>6. Fluxo Alternativo:</b>	<p>A1 – Cancela a operação:</p> <ul style="list-style-type: none"> <li>a. O administrador cancela a operação;</li> <li>b. O sistema retorna ao passo 5.a, podendo ou não realizar a venda;</li> <li>c. O caso de uso é encerrado</li> </ul>
<b>7. Fluxo de exceção:</b>	<p>E1 – Login inválido:</p> <ul style="list-style-type: none"> <li>a. A senha de acesso do administrador é inválida;</li> <li>b. O caso de uso é encerrado</li> </ul>
<b>8. Pós – condições:</b>	-
<b>9. Casos de teste:</b>	-

**Tabela 9 – Especificação de caso de uso – Realizar Venda**

<b>1. Finalidade/ Objetivo:</b>	Realizar o encerramento de caixa
<b>2. Ator (es):</b>	Administrador

<b>3. Pré – Condição:</b>	Efetuar login no sistema
<b>4. Evento Inicial:</b>	O administrador escolhe a opção “Encerramento de Caixa”
<b>5. Fluxo Principal:</b>	<ul style="list-style-type: none"> <li>a. O administrador efetua login no sistema; (E1)</li> <li>b. O administrador escolhe a opção “Encerramento de Caixa”, e efetua o encerramento; (A1)</li> </ul>
<b>6. Fluxo Alternativo:</b>	<p>A1 – Cancela a operação:</p> <ul style="list-style-type: none"> <li>a. O administrador cancela a operação;</li> <li>b. O sistema retorna ao passo 5.a, podendo ou não realizar o encerramento de caixa;</li> <li>c. O caso de uso é encerrado</li> </ul>
<b>7. Fluxo de exceção:</b>	<p>E1 - Login inválido:</p> <ul style="list-style-type: none"> <li>a. A senha de acesso do administrador é inválida;</li> <li>b. O caso de uso é encerrado</li> </ul>
<b>8. Pós – condições:</b>	<ul style="list-style-type: none"> <li>a. O administrador poderá gerar relatório de movimentação de caixa</li> </ul>
<b>9. Casos de teste:</b>	-

**Tabela 10 – Especificação de caso de uso – Encerramento de Caixa**

<b>1. Finalidade/ Objetivo:</b>	Gerar relatório contendo os clientes cadastrados
<b>2. Ator (es):</b>	Administrador
<b>3. Pré – Condição:</b>	<ul style="list-style-type: none"> <li>a. Efetuar login no sistema;</li> <li>b. Ter cliente cadastrado no sistema</li> </ul>
<b>4. Evento Inicial:</b>	O administrador escolhe a opção “Gerar Relatório de Cliente”
<b>5. Fluxo Principal:</b>	<ul style="list-style-type: none"> <li>a. O administrador efetua login no sistema; (E1)</li> <li>b. O administrador escolhe a opção “Gerar Relatório de Cliente”; (E2)</li> <li>c. O administrador confirma a operação; (A1)</li> </ul>
<b>6. Fluxo Alternativo:</b>	<p>A1 – Cancela a operação:</p> <ul style="list-style-type: none"> <li>a. O administrador cancela a operação;</li> <li>b. O sistema retorna ao passo 5.a, podendo ou não gerar o relatório de cliente;</li> <li>c. O caso de uso é encerrado</li> </ul>
<b>7. Fluxo de exceção:</b>	<p>E1 - Login inválido:</p> <ul style="list-style-type: none"> <li>a. A senha de acesso do administrador é inválida;</li> <li>b. O caso de uso é encerrado</li> </ul> <p>E2 – Indisponibilidade de dados:</p> <ul style="list-style-type: none"> <li>a. O sistema alerta que não existem</li> </ul>

	<p>dados para gerar o relatório de cliente;</p> <p>b. O caso de uso é encerrado</p>
<b>8. Pós – condições:</b>	<p>a. O administrador poderá imprimir o relatório de cliente</p>
<b>9. Casos de teste:</b>	<p>a. Gerar relatório quando houver dados;</p> <p>b. Gerar relatório quando não houver dados</p>

**Tabela 11 – Especificação de caso de uso – Gerar Relatório de Cliente**

<b>1. Finalidade/ Objetivo:</b>	Gerar relatório contendo os produtos cadastrados
<b>2. Ator (es):</b>	Administrador
<b>3. Pré – Condição:</b>	<p>a. Efetuar login no sistema;</p> <p>b. Ter produto cadastrado no sistema</p>
<b>4. Evento Inicial:</b>	O administrador escolhe a opção “Gerar Relatório de Produto”
<b>5. Fluxo Principal:</b>	<p>a. O administrador efetua login no sistema; (E1)</p> <p>b. O administrador escolhe a opção “Gerar Relatório de Produto”; (E2)</p> <p>c. O administrador confirma a operação; (A1)</p>

<p><b>6. Fluxo Alternativo:</b></p>	<p>A1 – Cancela a operação:</p> <ul style="list-style-type: none"> <li>a. O administrador cancela a operação;</li> <li>b. O sistema retorna ao passo 5.a, podendo ou não gerar o relatório de produto;</li> <li>c. O caso de uso é encerrado</li> </ul>
<p><b>7. Fluxo de exceção:</b></p>	<p>E1 - Login inválido:</p> <ul style="list-style-type: none"> <li>a. A senha de acesso do administrador é inválida;</li> <li>b. O caso de uso é encerrado</li> </ul> <p>E2 – Indisponibilidade de dados:</p> <ul style="list-style-type: none"> <li>a. O sistema alerta que não existem dados para gerar o relatório de produto;</li> <li>b. O caso de uso é encerrado</li> </ul>
<p><b>8. Pós – condições:</b></p>	<ul style="list-style-type: none"> <li>a. O administrador poderá imprimir o relatório de produto</li> </ul>
<p><b>9. Casos de testes:</b></p>	<ul style="list-style-type: none"> <li>a. Gerar relatório quando houver dados;</li> <li>b. Gerar relatório quando não houver dados</li> </ul>

**Tabela 12 – Especificação de caso de uso – Gerar Relatório de Produto**

<p><b>1. Finalidade/ Objetivo:</b></p>	<p>Gerar relatório contendo os serviços</p>
--	---

	cadastrados
<b>2. Ator (es):</b>	Administrador
<b>3. Pré – Condição:</b>	<ul style="list-style-type: none"> <li>a. Efetuar login no sistema;</li> <li>b. Ter serviço cadastrado no sistema</li> </ul>
<b>4. Evento Inicial:</b>	O administrador escolhe a opção “Gerar Relatório de Serviço”
<b>5. Fluxo Principal:</b>	<ul style="list-style-type: none"> <li>a. O administrador efetua login no sistema; (E1)</li> <li>b. O administrador escolhe a opção “Gerar Relatório de Serviço”; (E2)</li> <li>c. O administrador confirma a operação; (A1)</li> </ul>
<b>6. Fluxo Alternativo:</b>	<p>A1 – Cancela a operação:</p> <ul style="list-style-type: none"> <li>a. O administrador cancela a operação;</li> <li>b. O sistema retorna ao passo 5.a, podendo ou não gerar o relatório de serviço;</li> <li>c. O caso de uso é encerrado</li> </ul>
<b>7. Fluxo de exceção:</b>	<p>E1 - Login inválido:</p> <ul style="list-style-type: none"> <li>a. A senha de acesso do administrador é inválida;</li> <li>b. O caso de uso é encerrado</li> </ul> <p>E2 – Indisponibilidade de dados:</p> <ul style="list-style-type: none"> <li>a. O sistema alerta que não existem dados para gerar o relatório de</li> </ul>

	<p>serviço;</p> <p>b. O caso de uso é encerrado</p>
<b>8. Pós – condições:</b>	<p>a. O administrador poderá imprimir o relatório de serviço</p>
<b>9. Casos de testes:</b>	<p>a. Gerar relatório quando houver dados;</p> <p>b. Gerar relatório quando não houver dados</p>

**Tabela 13 – Especificação de caso de uso – Gerar Relatório de Serviço**

<b>1. Finalidade/ Objetivo:</b>	Gerar relatório da movimentação do caixa
<b>2. Ator (es):</b>	Administrador
<b>3. Pré – Condição:</b>	<p>a. Efetuar login no sistema;</p> <p>b. Ter alguma movimentação lançada no caixa</p>
<b>4. Evento Inicial:</b>	O administrador escolhe a opção “Gerar Relatório de Movimentação”
<b>5. Fluxo Principal:</b>	<p>a. O administrador efetua login no sistema; (E1)</p> <p>b. O administrador escolhe a opção “Gerar Relatório de Movimentação”; (E2)</p> <p>c. O administrador confirma a operação; (A1)</p>

<p><b>6. Fluxo Alternativo:</b></p>	<p>A1 – Cancela a operação:</p> <ul style="list-style-type: none"> <li>a. O administrador cancela a operação;</li> <li>b. O sistema retorna ao passo 5.a, podendo ou não gerar o relatório de movimentação;</li> <li>c. O caso de uso é encerrado</li> </ul>
<p><b>7. Fluxo de exceção:</b></p>	<p>E1 - Login inválido:</p> <ul style="list-style-type: none"> <li>a. A senha de acesso do administrador é inválida;</li> <li>b. O caso de uso é encerrado</li> </ul> <p>E2 – Indisponibilidade de dados:</p> <ul style="list-style-type: none"> <li>a. O sistema alerta que não existem dados para gerar o relatório de movimentação;</li> <li>b. O caso de uso é encerrado</li> </ul>
<p><b>8. Pós – condições:</b></p>	<ul style="list-style-type: none"> <li>a. O administrador poderá imprimir o relatório de movimentação</li> </ul>
<p><b>9. Casos de testes:</b></p>	<ul style="list-style-type: none"> <li>a. Gerar relatório quando houver dados;</li> <li>b. Gerar relatório quando não houver dados;</li> </ul>

**Tabela 14 – Especificação de caso de uso – Gerar Relatório de Movimentação**

## 8. DIAGRAMA DE CLASSES

Segundo Macoratti (2004), os diagramas de classe são os principais diagramas estruturais presentes na UML. Eles ilustram as classes, interfaces e relacionamento entre elas.

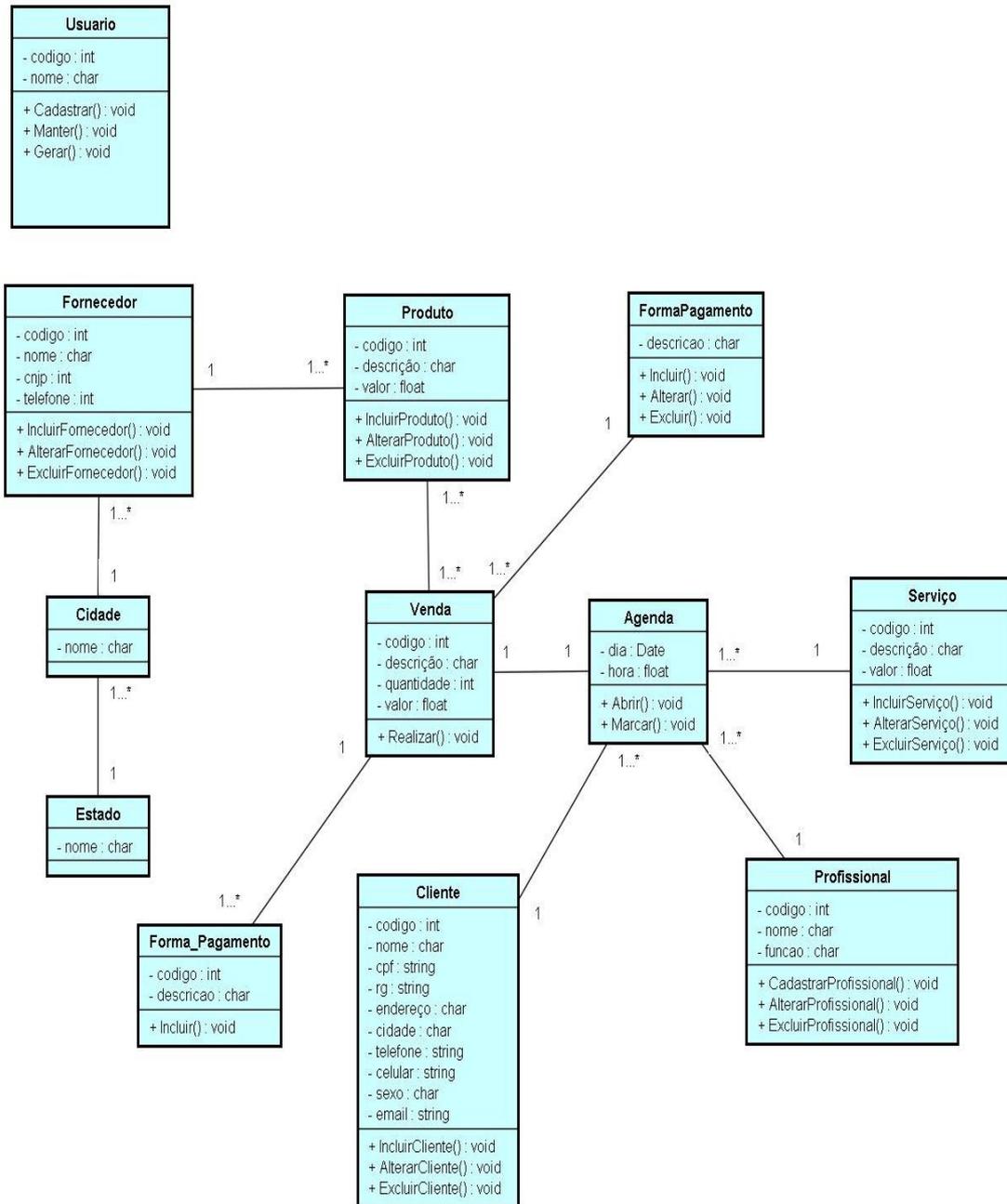


Figura 4 – Diagrama de Classes do sistema *Bela Mais*

## 9. DIAGRAMA DE ENTIDADE E RELACIONAMENTO (D.E.R.)

No Diagrama E – R são determinadas as dependências, e o relacionamento das entidades que estão instituídas no sistema. É uma representação da estrutura total do banco de dados.

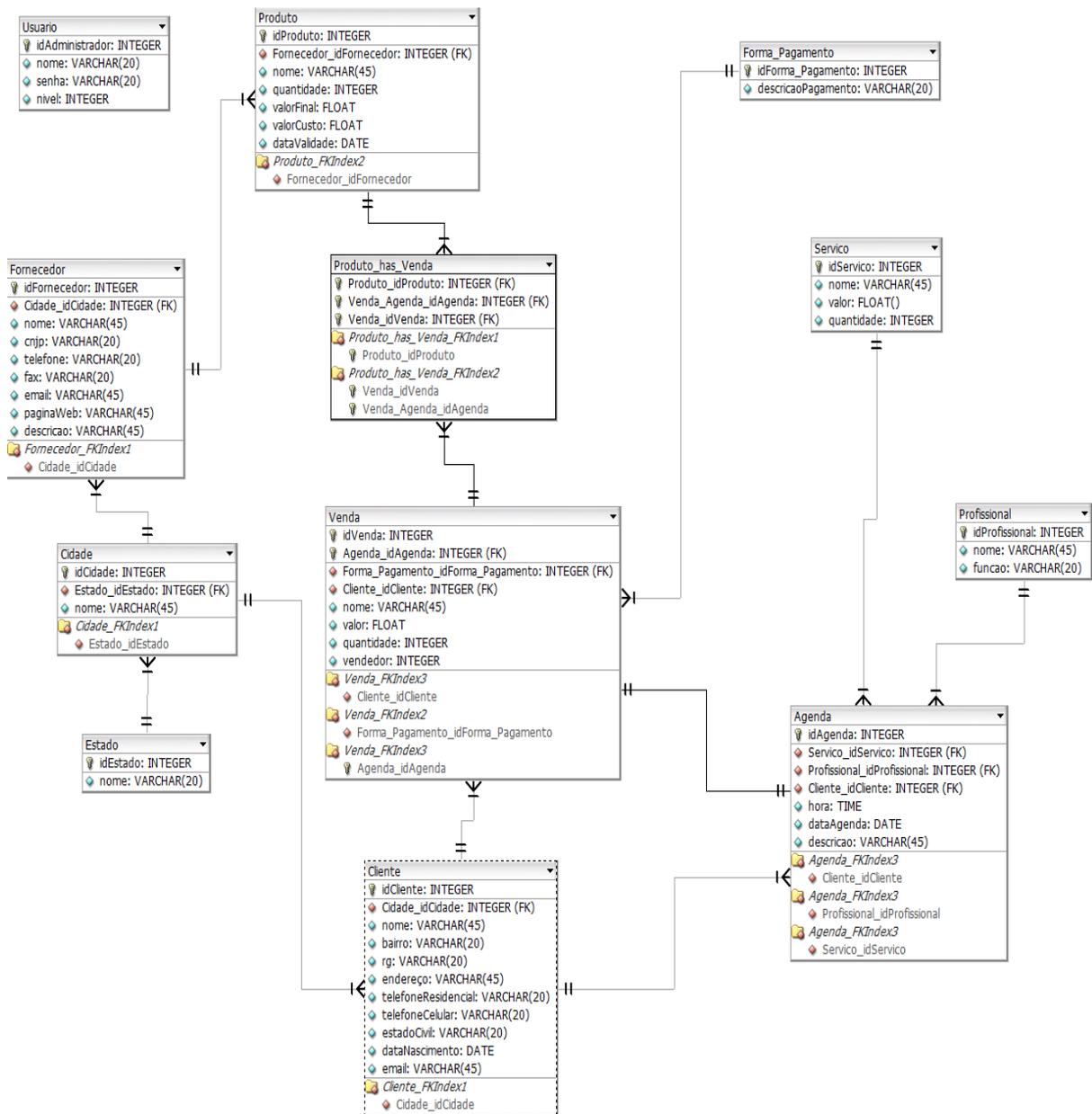


Figura 5 – Diagrama Entidade e Relacionamento do sistema Bela Mais

## 10. IMPLEMENTAÇÃO

Para a implementação do sistema Bela Mais foi utilizada a plataforma de desenvolvimento *Netbeans* com linguagem de programação Java, e *framework* RIA *Primefaces*. O mapeamento do banco de dados foi realizado através da ferramenta de persistência de dados *Hibernate*.

O desenvolvimento do *software* foi amparado ao Padrão de Projeto MVC (*Model - View - Controller*). Este padrão é composto por três tipos de objetos:

- *Model* ou Modelo: O Modelo é o objeto da aplicação, é utilizado para instituir e gerenciar o domínio da tarefa.
- *View* ou Visão: A partir dos dados gerados pelo Modelo, a Visão encarrega-se de manejar a apresentação visual. É a camada responsável por tornar presente os dados obtidos do Modelo ao usuário.
- *Controller* ou Controlador: O Controle define a forma como a interface do usuário reage às entradas do mesmo. Essa camada recebe a entrada de dados, e inicia a resposta ao usuário. Também é responsável pela validação da entrada de informações.

## 11. ORÇAMENTO

Os recursos necessários para a análise e o desenvolvimento do *software* Bela Mais são:

- 01 analista/ programador de sistemas;
- 01 Notebook;
- 01 Impressora jato de tinta

### 11.1 Recurso Humano

Analista/Programador	Quantidade de Horas	Custo do Dia	Total
Danielle Matuda	150 dias	R\$60,00	R\$9.000,00

Tabela 15 - Custo pessoal

### 11.2 Custo Material

01 Notebook:

Valor unitário: R\$3.000,00 (Três mil reais).

Dias de uso: 155 dias.

Depreciação: 3 anos - R\$3.000,00 / 36 (meses) = R\$83,33 (oitenta e três reais e trinta e três centavos) por mês.

Custo de 1 (um) dia: R\$83,33/ 30 (dias) = R\$2,77 (dois reais e setenta e sete centavos) por dia.

Custo de 155 (cento e cinquenta e cinco) dias: R\$2,77 \* 155 = R\$429,35.

01 Impressora Jato de tinta:

Valor unitário: R\$300,00 (Trezentos reais).

Dias de uso: 155 dias.

Depreciação: 3 anos - R\$300,00 / 36 (meses) = R\$8,33 (oito reais e trinta e três centavos) por mês.

Custo de 1 (um) dia: R\$8,33 / 30 (dias) = R\$0,27 (vinte e sete centavos) por dia.

Custo de 155 (cento e cinquenta e cinco) dias: R\$0,27\*155 = R\$41,85.

<b>RECURSOS UTILIZADOS</b>	<b>VALOR</b>
<b>Analista/Programador</b>	<b>R\$9.000,00</b>
<b>Notebook</b>	<b>R\$429,35</b>
<b>Impressora</b>	<b>R\$41,85</b>
	<b>TOTAL: R\$9.471,20</b>

**Tabela 16 - Custo total**

## 12. CRONOGRAMA

O cronograma é um instrumento de controle e planejamento. Nele, são detalhadas as atividades a serem executadas em um determinado período.

					FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV
TAREFAS														
Levantamento de Requisitos														
Especificação de Requisitos														
Validação dos Requisitos														
Diagrama de Caso de Uso														
Especificação de Caso de Uso														
Diagrama de Classe														
MER														
Qualificação														
Programação do Software														
Testes														
Escrita do relatório final														
Apresentação Final														

Figura 6 - Cronograma

### **13. CONCLUSÃO**

O crescimento na área de tecnologia da informação tem crescido cada vez mais no Brasil e no mundo. A necessidade de *softwares* específicos também acompanha esse expoente crescimento.

A escolha do desenvolvimento de um sistema computacional para o ramo de salões de beleza, deu-se pelo aumento da demanda em tal setor, e a carência de *softwares* que supram as necessidades envolvidas nesta atividade. O sistema Bela Mais apresenta funcionalidades específicas para o segmento de salões, o que o proporciona um diferencial.

Através da elaboração de diagramas UML, foi possível visualizar com maior amplitude as funcionalidades e detalhes presentes no sistema, assim, proporcionando maior simplicidade nos processos.

Com o uso do software, espera-se, que aumente a eficácia na empresa, proporcionando maior agilidade no gerenciamento das tarefas, e conseqüentemente, agregar mais qualidade aos serviços prestados.

## REFERÊNCIAS

APOLINÁRIO, Catharina. **Mercado de beleza do Brasil só perde para EUA e Japão**. São Paulo. Disponível em <<http://vilamulher.terra.com.br/mercado-de-beleza-do-brasil-so-perde-para-eua-e-japao-13-1-46-166.html>>. Acessado em: 22 mar. 2012.

BUZAN, Tony. **Mapas Mentais – Métodos criativos para estimular o raciocínio e usar ao máximo o potencial do seu cérebro**. Rio de Janeiro: Editora Sextante, 2009.

GONÇALVES, Edson. **Desenvolvendo Aplicações Web com JSP Servlets, JavaServer Faces, Hibernate, EJB3 Persistence e Ajax**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2007.

GONÇALVES, Edson. **Desenvolvendo Aplicações Web com NetBeans IDE 6.0**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2008.

GONÇALVES, Edson. **Dominando Relatórios JasperReports com iReport**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2008.

GUEDES, Guillanes T. A. **UML 2: Uma abordagem prática**. São Paulo : Editora Novatec Ltda, 2009.

HORSTMANN, Cays; CORNELL, Gary. **Java 2: Volume I – Fundamentos**. Tradução de João Eduardo Nóbrega Tortello. São Paulo: Editora Makron Books Ltda, 2001.

MACORATTI, José C. **UML – Diagrama de classes e objetos**. São Paulo. Disponível em <[http://www.macoratti.net/net\\_uml1.htm](http://www.macoratti.net/net_uml1.htm)>. Acessado em: 01 jun. 2012.

PÁDUA, Wilson P. F. **Engenharia de Software – Fundamentos, Métodos e Padrões**. Rio de Janeiro: Editora LTC, 2009.