

JABES FELIPE CUNHA

## SISTEMA DE CONTROLE E GERENCIAMENTO DE ORÇAMENTOS

Assis

2009

# SISTEMA DE CONTROLE E GERENCIAMENTO DE ORÇAMENTOS

JABES FELIPE CUNHA

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: Ms. Douglas Sanches da Cunha

Analisador: Dr. Almir Rogério Camolesi

Assis  
2009

JABES FELIPE CUNHA

## SISTEMA DE CONTROLE E GERENCIAMENTO DE ORÇAMENTOS

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: Ms. Douglas Sanches da Cunha

Área de Concentração: Desenvolvimento de Sistemas

Assis

2009

## DEDICATÓRIA

Dedico este trabalho para os meus Pais, meu Irmão, e todos os Amigos que estiveram do meu lado nesta longa jornada.

## AGRADECIMENTOS

Agradeço a Deus primeiramente que me capacitou por chegar até aqui, dando-me força, saúde e sabedoria, e sempre me abençoou nos momentos mais difíceis que enfrentei.

Agradeço aos meus Pais, por ter acreditado na minha capacidade, e sempre estiveram me ajudando e apoiando em todos os momentos. Aos meus Familiares que confiaram e acreditaram no meu potencial.

Ao meu orientador Ms. Douglas Sanches da Cunha, pela suas orientações, idéias, opiniões e por sempre me incentivar durante este trabalho.

A todos os Amigos do curso de TPD, aos Amigos que conviveram comigo durante meu estágio no CEPEIN e todos os Amigos que me ajudaram e acreditaram no meu potencial.

## RESUMO

O sistema desenvolvido tem por finalidade automatizar a etapa do serviço de cotação e orçamento de compra de produtos, oferecendo mais controle e gerenciamento dos pedidos de compras e os orçamentos realizados, que poderá ser utilizado em organizações de diversos segmentos. O software é uma aplicação desenvolvida para web, utilizando a linguagem Java e alguns *Frameworks Open Source*, como *JavaServer Faces*, *Hibernate* e o banco de dados *MySQL*. Por meio dessas tecnologias produtivas, o software proporciona um controle total das cotações dos produtos solicitados a diversos fornecedores, de forma prática e dinâmica, proporcionando uma forma ágil de realizar e acompanhar com transparência e segurança os orçamentos e pedidos de compra.

**Palavras - chave:** Java. Frameworks. Open Source. JavaServer Faces. Hibernate. MySQL.

## ABSTRACT

The developed application aims on automating purchase quotations and budgets, offering a better control and management of purchase orders and realized budgets, which may be used in organizations of different segments. The software is a web application, developed in Java with MySQL database, using some open source frameworks like JavaServer Faces and Hibernate. By means of these productive technologies, the software offers a total control of requested products quotations made to different suppliers, in a pratic and dynamic manner, offering a fast way to realize and monitor with transparency and security the budgets and purchase orders.

**Keywords:** Java. Frameworks. Open Source. JavaServer Faces. Hibernate. MySQL.

## LISTA DE ILUSTRAÇÕES

<b>Figura 1 - Caso de Uso: Realizar Login.....</b>	<b>32</b>
<b>Figura 2 - Caso de Uso: Gerenciar Área Fornecedor.....</b>	<b>36</b>
<b>Figura 3 - Caso de Uso: Gerenciar Área Cliente.....</b>	<b>37</b>
<b>Figura 4 - Caso de Uso: Editar Perfil.....</b>	<b>38</b>
<b>Figura 5 - Caso de Uso: Manter Categoria.....</b>	<b>39</b>
<b>Figura 6 - Caso de Uso: Manter SubCategoria.....</b>	<b>41</b>
<b>Figura 7 - Caso de Uso: Manter Produto.....</b>	<b>43</b>
<b>Figura 8 - Caso de Uso: Fazer Orçamento.....</b>	<b>44</b>
<b>Figura 9 - Caso de Uso: Publicar Orçamento.....</b>	<b>50</b>
<b>Figura 10 - Caso de Uso: Consultar Orçamento.....</b>	<b>51</b>
<b>Figura 11 - Caso de Uso: Cadastrar Preço.....</b>	<b>53</b>
<b>Figura 12 - Caso de Uso: Finalizar Orçamento.....</b>	<b>56</b>
<b>Figura 13 - Caso de Uso: Emitir Pedido de Compra.....</b>	<b>57</b>
<b>Figura 14 - Caso de Uso: Consultar Pedido.....</b>	<b>59</b>
<b>Figura 15 - Caso de Uso: Emitir Relatório.....</b>	<b>60</b>
<b>Figura 16 - Diagrama de Caso de Uso Global.....</b>	<b>62</b>
<b>Figura 17 - Diagrama de Classes de Negócio.....</b>	<b>63</b>
<b>Figura 18 - Diagrama de Sequência: Realizar Login.....</b>	<b>64</b>
<b>Figura 19 - Diagrama de Sequência: Manter Fornecedor.....</b>	<b>65</b>
<b>Figura 20 - Diagrama de Sequência: Manter Cliente.....</b>	<b>66</b>
<b>Figura 21 - Diagrama de Sequência: Manter Usuário.....</b>	<b>67</b>
<b>Figura 22 - Diagrama de Sequência: Manter Produto.....</b>	<b>68</b>
<b>Figura 23 - Diagrama de Sequência: Manter Categoria.....</b>	<b>69</b>
<b>Figura 24 - Diagrama de Sequência: Manter SubCategoria.....</b>	<b>70</b>
<b>Figura 25 - Diagrama de Sequência: Movimentação Orçamentos... </b>	<b>71</b>



<b>Figura 26 - Diagrama de Sequência: Emitir Pedido de Compra.....</b>	<b>72</b>
<b>Figura 27 - Diagrama de Atividade: Cliente.....</b>	<b>73</b>
<b>Figura 28 - Diagrama de Atividade: Fornecedor.....</b>	<b>74</b>
<b>Figura 29 - Diagrama WBS.....</b>	<b>75</b>
<b>Figura 30 - Sequenciamento das Atividades Definidas.....</b>	<b>77</b>
<b>Figura 31 - Cronograma das Atividades Definidas.....</b>	<b>78</b>
<b>Figura 32 - Arquitetura da Aplicação.....</b>	<b>82</b>
<b>Figura 33 - Estrutura de Aplicações Web no Netbeans.....</b>	<b>83</b>
<b>Figura 34 - Páginas Web no Netbeans.....</b>	<b>84</b>
<b>Figura 35 - Arquivos de Configuração Web no Netbeans.....</b>	<b>85</b>
<b>Figura 36 - Código Fonte com os Pacotes no Netbeans.....</b>	<b>86</b>
<b>Figura 37 - Tela de Login.....</b>	<b>93</b>
<b>Figura 38 - Tela de Cadastro de Fornecedores.....</b>	<b>93</b>
<b>Figura 39 - Tela de Clientes Cadastrados.....</b>	<b>94</b>
<b>Figura 40 - Tela de Cadastro de Produtos.....</b>	<b>94</b>
<b>Figura 41 - Tela Inicial do Administrador.....</b>	<b>95</b>
<b>Figura 42 - Tela para Escolha de Idiomas.....</b>	<b>95</b>
<b>Figura 43 - Tela Orçamento I.....</b>	<b>96</b>
<b>Figura 44 - Tela Orçamento II.....</b>	<b>96</b>
<b>Figura 45 - Tela Orçamentos Cadastrados.....</b>	<b>97</b>
<b>Figura 46 - Tela Emitir Pedido de Compra.....</b>	<b>97</b>
<b>Figura 47 - Tela Pedidos Cadastrados.....</b>	<b>98</b>
<b>Figura 48 - Tela Pedido de Compra.....</b>	<b>98</b>

## LISTA DE TABELAS

<b>Tabela 1 - Orçamento do Projeto.....</b>	<b>81</b>
---	-----------

## LISTA DE ABREVIATURAS E SIGLAS

UML	Unified Modeling Language
OMG	Object Management Group
ORM	Object Relational Mapping
JSF	Java ServerFaces
MVC	Model View Controller
HTML	Hyper Text Markup Language
AJAX	Asynchronous JavaScript and XML
PDF	Portable Document Format
CSS	Cascading Style Sheets
IDE	Integrated Development Environment
JUDE	Java and UML Developers' Environment
JRE	Java Runtime Environment
SGBD	Sistema de Gerenciamento de Banco de Dados
DDL	Data Definition Language
DML	Data Manipulation Language
WBS	Work Breakdown Structure
UC	Use Case
JEE	Java Enterprise Edition
EJB	Enterprise JavaBeans

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>15</b>
1.1 OBJETIVO.....	15
1.2 JUSTIFICATIVA E MOTIVAÇÃO.....	16
1.3 PÚBLICO ALVO.....	16
1.4 ESTRUTURA DO TRABALHO.....	17
<b>2 METODOLOGIA DA ANÁLISE DE ORIENTAÇÃO A OBJETOS.....</b>	<b>18</b>
2.1 CLASSES.....	18
2.2 OBJETO.....	19
2.3 ABSTRAÇÃO.....	19
2.4 ATRIBUTOS.....	19
2.5 MÉTODOS.....	19
2.6 ENCAPSULAMENTO.....	20
2.7 CONSTRUTOR.....	20
2.8 HERANÇA.....	20
2.9 POLIMORFISMO.....	21
2.10 AMBIENTE DE DESENVOLVIMENTO.....	21
<b>2.10.1 Ferramentas Utilizadas.....</b>	<b>21</b>
2.10.1.1 Java.....	21
2.10.1.2 Hibernate.....	22
2.10.1.3 JavaSever Faces.....	22
2.10.1.4 IceFaces.....	24
2.10.1.5 Jasper Reports .....	24
2.10.1.6 CSS.....	24
2.10.1.7 Netbeans.....	25
<b>2.10.2 Ferramentas para Análise.....</b>	<b>25</b>
<b>2.10.3 Banco de Dados.....</b>	<b>25</b>
<b>3 PROJETO E ANÁLISE DO SISTEMA.....</b>	<b>27</b>
3.1 LEVANTAMENTO DOS REQUISITOS.....	27

<b>3.1.1 Forma de Levantamento Adotada.....</b>	<b>27</b>
<b>3.1.2 Análise dos Requisitos.....</b>	<b>28</b>
3.1.2.1 Problemas Potenciais.....	28
3.1.2.2 Restrições do Sistema.....	28
<b>3.1.3 Validação dos Requisitos.....</b>	<b>29</b>
<b>3.1.4 Especificação dos Requisitos.....</b>	<b>30</b>
3.1.4.1 Escopo.....	30
3.1.4.2 Lista de Eventos.....	31
<b>3.2 ANÁLISE ORIENTADA A OBJETOS.....</b>	<b>32</b>
<b>3.2.1 Especificação dos Casos de Usos.....</b>	<b>32</b>
3.2.1.1 Realizar Login.....	32
3.2.1.2 Cadastrar Fornecedor.....	33
3.2.1.3 Cadastrar Cliente.....	34
3.2.1.4 Gerenciar Área Fornecedor.....	36
3.2.1.5 Gerenciar Área Cliente.....	37
3.2.1.6 Editar Perfil.....	38
3.2.1.7 Manter Categoria.....	39
3.2.1.8 Manter SubCategoria.....	41
3.2.1.9 Manter Produto.....	43
3.2.1.10 Fazer Orçamento.....	44
3.2.1.11 Salvar Orçamento.....	46
3.2.1.12 Cancelar Orçamento.....	47
3.2.1.13 Incluir Produto.....	48
3.2.1.14 Incluir Fornecedor.....	49
3.2.1.15 Publicar Orçamento.....	50
3.2.1.16 Consultar Orçamento.....	51
3.2.1.17 Cadastrar Preço.....	53
3.2.1.18 Salvar Cotação.....	54
3.2.1.19 Cancelar Cotação.....	55
3.2.1.20 Finalizar Orçamento.....	56
3.2.1.21 Emitir Pedido de Compra.....	57
3.2.1.22 Consultar Pedido.....	59
3.2.1.23 Emitir Relatório.....	60
<b>3.2.2 Diagrama de Caso de Uso Global.....</b>	<b>62</b>

<b>3.2.3 Diagrama de Classe.....</b>	<b>63</b>
<b>3.2.4 Diagrama de Sequência.....</b>	<b>64</b>
3.2.4.1 Diagrama de Sequência: Realizar Login.....	64
3.2.4.2 Diagrama de Sequência: Manter Fornecedor.....	65
3.2.4.3 Diagrama de Sequência: Manter Cliente.....	66
3.2.4.4 Diagrama de Sequência: Manter Usuário.....	67
3.2.4.5 Diagrama de Sequência: Manter Produto.....	68
3.2.4.6 Diagrama de Sequência: Manter Categoria.....	69
3.2.4.7 Diagrama de Sequência: Manter SubCategoria.....	70
3.2.4.8 Diagrama de Sequência: Movimentação Orçamentos.....	71
3.2.4.9 Diagrama de Sequência: Emitir Pedido de Compra.....	72
<b>3.2.5 Diagrama de Atividades.....</b>	<b>73</b>
3.2.5.1 Diagrama de Atividade Cliente.....	73
3.2.5.2 Diagrama de Atividade Fornecedor.....	74
<b>3.3 PLANEJAMENTO DO PROJETO.....</b>	<b>75</b>
<b>3.3.1 WBS – Work Breakdown Structure.....</b>	<b>75</b>
<b>3.3.2 Definição das Atividades.....</b>	<b>76</b>
<b>3.3.3 Sequenciamento das Atividades Definidas.....</b>	<b>77</b>
<b>3.3.4 Cronograma das Atividades Definidas.....</b>	<b>78</b>
<b>3.3.5 Recursos Necessários para o Desenvolvimento do Projeto.....</b>	<b>78</b>
<b>3.3.6 Estimativa de Custos.....</b>	<b>79</b>
<b>3.3.7 Orçamento do Projeto.....</b>	<b>81</b>
<b>4 IMPLEMENTAÇÃO.....</b>	<b>82</b>
<b>5 CONCLUSÃO.....</b>	<b>88</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>89</b>
<b>REFERÊNCIAS ELETRÔNICAS.....</b>	<b>91</b>
<b>ANEXO – INTERFACE DO SISTEMA.....</b>	<b>93</b>

## 1 INTRODUÇÃO

Com a idéia de sistematizar um serviço, intenciona-se torná-lo, ágil, seguro e eficiente na execução daquilo a que se propõe. O serviço de comercializar produtos, por ser um mercado significativamente competitivo, exige que determinadas ações sejam controladas de modo que o proprietário do negócio em questão, possua meios de conquistar um conhecimento de possibilidades para assegurar que seu serviço/produto tenha um lugar de destaque no mercado e sua empresa consiga os objetivos a que se propõe, enquanto atividade comercial.

Portanto este sistema tem o propósito de controlar e gerenciar de forma ágil, segura e eficiente os orçamentos realizados por empresas de comércio, indústria e qualquer empresa que necessite de realizar orçamentos em escalas diversas.

Para atender o quesito de mobilidade o sistema será desenvolvido de forma que sua aplicação esteja disponível para a web, e a linguagem utilizada será Java e seus *Frameworks*.

O geral é que, a empresa de posse do gerenciamento proposto, possa controlar de forma dinâmica suas atividades e consequentemente abrir horizontes no ramo de sua atividade.

### 1.1 OBJETIVO

O software a ser desenvolvido permitirá o controle e gerenciamento de orçamentos, oferecendo alguns benefícios, como agilidade, segurança e facilidade na criação de qualquer orçamento a ser realizado. Este sistema poderá ser usado em empresas que atuam na área da indústria, comércio, entre outras, independente do seu porte, ou seja, qualquer empresa que necessita fazer orçamentos poderá implantar o software.

O objetivo geral é que, por meio deste sistema as empresas que necessitam fazer orçamentos, possa ter total gerenciamento dos mesmos, de uma forma prática, rápida e dinâmica, e com consequências adquirir melhor controle no acompanhamento dos orçamentos, possibilitando mais transparência para os serviços realizados.

## 1.2 JUSTIFICATIVA E MOTIVAÇÃO

A burocracia por partes de algumas empresas para entregar uma cotação é constante, e respectivamente o cliente fica insatisfeito, e grande parte deste problema é o modo como é feito o controle e gerenciamento das cotações, algumas empresas fazem este processo manualmente e encaminha para o cliente via fax, outras passam verbalmente, outras acabam até esquecendo-se de entregar, algumas até possuem um sistema, porém o mesmo não possui o devido controle que deveria ser feito.

Em meio a este problema, a implantação de um sistema que possa fazer o gerenciamento automatizado destes orçamentos e ainda que possa ficar disponível na web, pode ser uma solução alternativa, não para resolver completamente o problema, mas sim, amenizar e muito este processo de cotações de materiais e serviços.

## 1.3 PÚBLICO ALVO

As empresas que ainda não adotaram a informatização nos seus negócios, que operam com grande volumes de dados, que gastam tempo para elaborar, consultar ou localizar os orçamentos e que a cada dia aumenta a quantidade de papéis que são usados para fazer a cotação, podem implantar o sistema para obter uma melhora significativa em termos de organização.



Em geral, o foco do sistema é controlar a gestão de orçamentos, independente do tipo dos materiais a serem cotados, ou seja, os fornecedores poderão cadastrar seus produtos e os clientes poderão realizar suas cotações.

## 1.4 ESTRUTURA DO TRABALHO

Este trabalho foi estruturado em 5 capítulos, sendo que o primeiro capítulo mostra o assunto a ser tratado, quais os objetivos do sistema, a justificativa e motivação para desenvolver o sistema e qual é o público alvo que poderá usar o sistema.

O segundo capítulo é referente aos conceitos da análise de orientação a objetos, descreve as tecnologia e ferramentas utilizadas.

O terceiro capítulo explica os estudos de casos, mostra o levantamento dos requisitos, descrevendo a forma de levantamento adotada, a análise, validação e especificação dos requisitos, também ilustra os diagramas da análise de orientação a objetos, tais como: diagrama de Caso de Uso, diagrama de Classe, diagrama de Atividade e diagrama de Sequência e por fim o planejamento do projeto, que contém a definição e o cronograma das atividades definidas, a estimativa de custos, o orçamento do projeto, entre outros descritivos.

Já o quarto capítulo aborda como o sistema foi implementado, descrevendo o padrão MVC e como foi estruturado o projeto.

Concluindo, o último capítulo descreve as considerações finais, que contém os resultados do trabalho e os objetivos que foram alcançados.

## 2 METODOLOGIA DA ANÁLISE DE ORIENTAÇÃO A OBJETOS

A análise orientada a objeto surgiu na década de 1990 e após atingir sua maturidade, surge a Linguagem de Modelagem Unificada (UML). A criação desta linguagem teve vários contribuintes, entre os principais criadores foram *Grady Booch*, *James Rumbaugh* e *Ivar Jacobson*. Em 1997, a UML foi aprovada como padrão pelo *Object Management Group* (OMG) que é uma organização internacional que aprova padrões abertos para aplicações orientadas a objetos (BEZERRA, 2002).

“A UML é uma linguagem visual para modelar sistemas orientados a objetos. Isso quer dizer que a UML é uma linguagem constituída de elementos gráficos (visuais) utilizados na modelagem que permitem representar os conceitos do paradigma da orientação objetos” (BEZERRA, 2002, p. 14).

O sistema seguirá o paradigma de programação orientada à objetos, esta forma de programação permite à construir classes e criar objetos destas classes, fazendo que este trabalhem em conjunto para que os propósitos de um sistema sejam alcançados.

Esta maneira de programar, embora mais complexa, é muito mais apropriada para o desenvolvimento de sistemas pois permite trabalhar com os problemas de maneira semelhante ao nosso entendimento de mundo ao invés de dirigir a solução para as características de funcionamento dos computadores, tal como faz o paradigma da programação procedural (JÚNIOR, 1999).

### 2.1 CLASSES

Classe é um tipo de objeto definido pelo programador que possuem características e comportamentos que descreve uma entidade real ou abstrata. De uma forma mais

computacional, as classes representam um modelo, tipo de uma estrutura, que possuem os atributos e métodos, respectivamente, variáveis que armazenam os dados e as funções que manipulam os dados (JÚNIOR, 1999).

## 2.2 OBJETO

Objeto é a representação de uma classe que contém seus atributos e métodos ou seja, estado e comportamento.

A partir das classes, pode-se criar os objetos, denominando este recurso da orientação a objetos de instanciação, que nada mais é do que criar uma instância da classe, através da instanciação pode se criar vários objetos de uma mesma classe, entretanto com características diferentes (JÚNIOR, 1999).

## 2.3 ABSTRAÇÃO

Abstração é uma maneira de focar nos objetivos principais do negócio, deixando de lado características não tão importantes.

## 2.4 ATRIBUTOS

As classes possuem características, isto vem a ser os atributos, ou seja, são as variáveis que estão destinada a armazenar algum tipo de informação (JÚNIOR, 1999).

## 2.5 MÉTODOS

As classes possuem comportamentos, isto vem a ser os métodos, que servem para

realizar operações sobre os atributos de uma classe. Através dos métodos os objetos são capazes de enviar e receber mensagem, proporcionando o estado desejado da aplicação (JÚNIOR, 1999).

## 2.6 ENCAPSULAMENTO

A acessibilidade de uma classe, método ou atributo de uma classe é a forma que tal elemento pode ser visto e utilizado por outras classes, este conceito é chamado de encapsulamento, através deste conceito, pode-se controlar de forma mais robusta o acesso a atributos ou métodos, possibilitando total proteção.

O encapsulamento oferece várias vantagens, sendo, definir quais ou quando os atributos podem ser alterados, permitir o acesso aos atributos pelos métodos e validar informação a serem armazenadas no atributos (JÚNIOR, 1999).

## 2.7 CONSTRUTOR

Os construtores são utilizados para criar instâncias de objetos de uma classe, isto é, estruturas de dados e operações que representam um objeto (JÚNIOR, 1999).

## 2.8 HERANÇA

A orientação à objetos permite o reaproveitamento do código, e o conceito de herança é poder utilizar atributos e métodos definidos por outras classes especificada como seu ancestral, portanto a herança é um mecanismo de especialização (JÚNIOR, 1999).

A herança permite que novas classes sejam criadas e adicionadas a uma hierarquia sem a necessidade de qualquer modificação do código existente das classes e das

aplicações que utilizam estas classes pois cada classe define estritamente seus próprios métodos e atributos (JÚNIOR, 1999).

## 2.9 POLIMORFISMO

O polimorfismo é um dos recursos mais importantes da orientação objetos, significando que uma classe base é derivada várias vezes, ou seja várias formas para a classe base. O polimorfismo também pode ser definido como a capacidade de um mesmo código ser executado, de maneira real ou aparente, sobre valores de diferentes tipos (JÚNIOR, 1999).

## 2.10 AMBIENTE DE DESENVOLVIMENTO

### 2.10.1 Ferramentas Utilizadas

#### 2.10.1.1 Java

O código fonte do sistema será escrito em Java, que é uma linguagem de programação, introduzida no mercado pela *Sun Microsystem*, que provocou e ainda provoca excitação e entusiasmo em programadores, analistas e projetistas de software (JÚNIOR, 1999).

Java é resultado de um trabalho consistente de pesquisa e desenvolvimento de mais do que uma simples linguagem de programação, mas de todo um ambiente de desenvolvimento e execução de programas que exhibe as facilidades proporcionadas pela orientação à objetos, pela extrema portabilidade do código, pelas características de segurança que esta plataforma oferece e finalmente pela facilidade de sua integração aos outros ambientes, destacando-se a internet (JÚNIOR, 1999, p. 9).

Java é uma linguagem que possui características importantes, como: Orientação à

Objetos, Independência de Plataforma, Desempenho, Segurança e Robustez. Através desta poderosa linguagem, o sistema fará o uso de alguns audaciosos *Frameworks* que são descritos logo mais (JÚNIOR, 1999).

#### 2.10.1.2 Hibernate

*Hibernate* é um poderoso *Framework* escrito em Java que se relaciona com o banco de dados, onde este relacionamento é conhecido como mapeamento objeto/relacional (ORM) para Java. Este *Framework* procura ter uma completa solução para o gerenciamento de dados persistentes em Java (GONÇALVES, 2008).

#### 2.10.1.3 JavaSever Faces

*JavaServer Faces* é um *framework* para desenvolvimento de aplicações web, a vantagem prometida pelo JSF é trazer o desenvolvimento rápido de interfaces de usuário para *java-side* (GEARY; HORSTMAN, 2007).

O JSF possui as principais partes:

- Um conjunto de componentes pré-fabricados de IU (Interface de usuário);
- Um modelo de programação orientada a eventos;
- Um modelo de componentes que permite a desenvolvedores independentes fornecerem componentes adicionais.

As aplicações web possuem duas partes: a camada de apresentação e a lógica de negócios. A camada de apresentação refere-se à aparência da aplicação, e a lógica de negócios é implementado no código Java, que determina o comportamento da aplicação.

No contexto do JSF, o código das aplicações está localizado nos *beans* e o design está localizado nas páginas web. Um *Bean* é uma classe que possui propriedades e eventos a um ambiente como o JSF, usa-se os *beans* para todos os dados que devem ser acessíveis a partir de uma página. Os *beans* são os canais entre a interface de usuário e o *backend* da aplicação (GEARY; HORSTMAN, 2007).

O JSF implementa a clássica arquitetura MVC (modelo-visão-controlador), ele conecta a visão ao modelo, como por exemplo, um componente de visão pode ser ligado a uma propriedade *bean* de um objeto modelo (GEARY; HORSTMAN, 2007). O processamento do *Framework* JSF é bem complexa, também chamada de ciclo de vida, a sua especificação é definida em seis fases distintas (GEARY; HORSTMAN, 2007):

- Restaurar Visão: recupera a árvore de componentes para página requisitada;
- Aplicar Valores de Requisição: JSF faz iterações sobre os objetos componentes da árvore de componentes;
- Processar Validações: A *String* de valores enviada é primeiramente convertida para valores locais, que podem ser objetos de qualquer tipo;
- Atualizar Valores do Modelo: os valores locais são usados para atualizar os *beans* vinculados aos componentes;
- Invocar Aplicação: São executados os métodos de ação;
- Renderizar Resposta: Codifica a resposta e envia- para o navegador.

O *JavaServer Faces* oferece vários recursos, tais como:

- *Managed Bean*: que realiza a separação entre a apresentação e a lógica de negócios;
- Regras de Navegação: que diz à implementação JSF qual página deve ser mandada de volta para o navegador após o envio de um formulário ou com uma ação de botão;
- Componentes: possui bibliotecas de *tags* fundamental e HTML. A biblioteca fundamental contém as *tags* que são independentes da tecnologia de renderização, como por exemplo: *converter*, *validator*, etc. As *tags* HTML do JSF representam os seguintes tipos de componentes: *inputs*, *outputs*, comandos, etc;
- Tabela de Dados: representa um par componente / renderizador adequado,

ela faz iterações sobre dados para criar uma tabela HTML;

- Conversores e Validadores: As *tags* padrão fornecidas pelo JSF para conversão e validação, atendem à maior parte das necessidades comuns, como conversores de números e datas, validadores de tamanho. Também é possível implementar *tags* customizadas, isto é, conversores e validadores configuráveis pelos autores da página;
- Manipulação de Eventos: Aplicações web precisam responder a eventos de usuários, três tipos de eventos são usados no JSF, eventos do ciclo de vida, eventos de alteração de valor e eventos de ação.

#### 2.10.1.4 IceFaces

O *IceFaces* é um projeto *Open Source* desenvolvido pela empresa *IceSoft*, o *IceFaces* possui um conjunto de componentes usando a tecnologia JSF e Ajax de forma nativa, todos os componentes JSF do *IceFaces* dão suporte ao AJAX (GONÇALVES, 2008).

#### 2.10.1.5 Jasper Reports

O *JasperReports* é um poderoso *Framework* gratuito, escrito em Java, que é usado para geração de relatórios, permitindo gerar dinamicamente relatórios em vários formatos, como: PDF, HTML, entre outros. Para facilitar a formatação e criação do *layout* dos relatórios é usado a ferramenta *iReport* (FURUTANI, 2009).

#### 2.10.1.6 CSS

A definição do visual do sistema está por parte do *Cascading Style Sheets (CSS)*, que traduzindo seria Folha de Estilos em Cascata, que é uma linguagem para definir



a aparência e estruturação dos sistemas desenvolvidos para web. Através do CSS pode-se controlar cores, fontes, margens, alturas, larguras, posicionamento e muito mais.

#### 2.10.1.7 Netbeans

O NetBeans IDE *Integrated Development Environment* é um ambiente integrado para desenvolvimento de software, escrito em Java, é usado por muito desenvolvedores que utilizam esta linguagem. Esta ferramenta pode ser adquirida gratuitamente e auxiliam os programadores a escrever, debugar, compilar e executar o programa.

#### 2.10.2 Ferramentas para Análise

O JUDE é uma IDE feita em Java, que pode ser usada em qualquer sistema operacional que possui a JRE instalada. Através do JUDE, é possível criar desde simples até os mais complexos diagramas da UML, os mais comuns são: os diagramas de casos de uso, classes, sequência, atividades e colaboração. Após ter elaborada a modelagem, o JUDE possibilita que os diagramas podem ser exportados para arquivo HTML ou arquivo de Imagem.

#### 2.10.3 Banco de Dados

O *MySQL* é um sistema gerenciador de banco de dados relacional (SGBD), é um dos banco de dados mais popular que se encontra no mercado, utiliza a linguagem de definição de dados (DDL) para criar, alterar e remover tabelas, também utiliza a linguagem de manipulação de dados (DML) para fazer instruções de *insert*, *update*, *delete* e *select*.

O *MySQL* é um banco de dados extremamente leve, que não depende de muitos recursos computacionais para sua execução e funciona nos mais populares sistemas operacionais que existem no mercado.

Algumas das principais e mais importantes características do *MySQL* são:

- Funciona em diversas plataformas;
- Suporte a operações e funções nas partes *Select* e *Where* das consultas;
- Um sistema de privilégios flexível;
- Suporte total para vários conjuntos de caracteres;
- O servidor pode apresentar mensagem de erros aos clientes em diversas línguas.

## 3 PROJETO E ANÁLISE DO SISTEMA

### 3.1 LEVANTAMENTO DOS REQUISITOS

#### 3.1.1 Forma de Levantamento Adotada

O levantamento dos requisitos é uma das partes mais importantes no desenvolvimento do sistema, é neste momento que se deve absorver o máximo de informação do cliente para poder facilitar os processos posteriores que englobam no desenvolvimento do sistema.

Ramo das empresas

R: Em geral, as empresas que atuam na área de indústria, comércio, entre outras, independente do seu segmento poderá implantar o sistema.

Descrição do problema

R: De um modo geral, algumas empresas processam seus orçamentos de forma manual ou possuem algum sistema que não atende aos requisitos necessários, em meio a este problema, não consegue obter um controle necessário para poder gerenciar os antigos, novos e atuais orçamentos que são feitos, com consequência começam surgir problemas nos negócios da empresa.

Resultado esperado

R: Com a implantação do sistema, a empresa que elabora os orçamentos, poderá obter total controle e gerenciamento dos serviços realizados e também os

fornecedores poderão cadastrar seus produtos, deixando de amostra para os clientes que fazem os orçamentos.

### **3.1.2 Análise dos Requisitos**

#### **3.1.2.1 Problemas Potenciais**

Cadastrar clientes e fornecedores;

Cadastrar usuários;

Cadastrar produtos com sua devida categoria e subcategoria;

Controlar os orçamentos;

Controlar os pedidos de compras;

Emitir relatórios dos orçamentos, produtos, pedidos e etc...

#### **3.1.2.2 Restrições do Sistema**

Por ser uma aplicação desenvolvida para a web, não necessitará de muitos recursos de hardware, independente de sistema operacional, tendo como exigência ter conexão com a internet e um navegador seguro.

### 3.1.3 Validação dos Requisitos

Os requisitos estão completos?

Sim( )

Não( )

Os objetivos estão concretos?

Sim( )

Não( )

Os dados levantados atende as necessidades?

Sim( )

Não( )

O desenvolvimento foi atingido dentro das restrições?

Sim( )

Não( )

Descrição de inconsistências, redundâncias, omissões e falta de clareza:

Informações adicionais:

---

Responsável pela Validação

### 3.1.4 Especificação dos Requisitos

O sistema a ser desenvolvido irá controlar e gerenciar os orçamentos da empresa, possibilitando que o cliente e o fornecedor realize seus serviços de uma forma dinâmica e segura.

#### 3.1.4.1 Escopo

As categorias e subcategorias dos produtos devem ser cadastradas pelo fornecedor.

Os produtos ou serviços devem ser cadastrados pelo fornecedor.

Os clientes poderão realizar seus orçamentos, incluindo quantos produtos e fornecedores forem necessários.

Ao final de um orçamento, o cliente tem a possibilidade de gerar um pedido.

Diversos relatórios serão disponibilizados para os clientes e fornecedores.

Por questão de segurança o sistema contará com um recurso de autenticação de usuários.

Para evitar dados inconsistentes, o sistema terá um recurso de validação e conversão de dados.

Uma versão demo será disponibilizada aos clientes e fornecedores para poderem testar o sistema.

O sistema em si não faz *backup*, mas a base de dados sempre terá uma cópia de segurança, feita no mínimo três vezes por semana.

### 3.1.4.2 Lista de Eventos

Manter Usuário

Manter Cliente

Manter Fornecedor

Manter Produto

Manter Categoria

Manter SubCategoria

Movimentar Orçamento

Emitir Pedido de Compra

Emitir Relatório de Produtos

Emitir Relatório de Produtos por Fornecedor

Emitir Relatório de Fornecedores por Cidade

Emitir Relatório de Clientes por Cidade

Emitir Relatório de Categorias

Emitir Relatório de SubCategorias

Emitir Relatório de Orçamentos por Cliente

Emitir Relatório de Orçamentos por Fornecedor

Emitir Relatório de Orçamentos

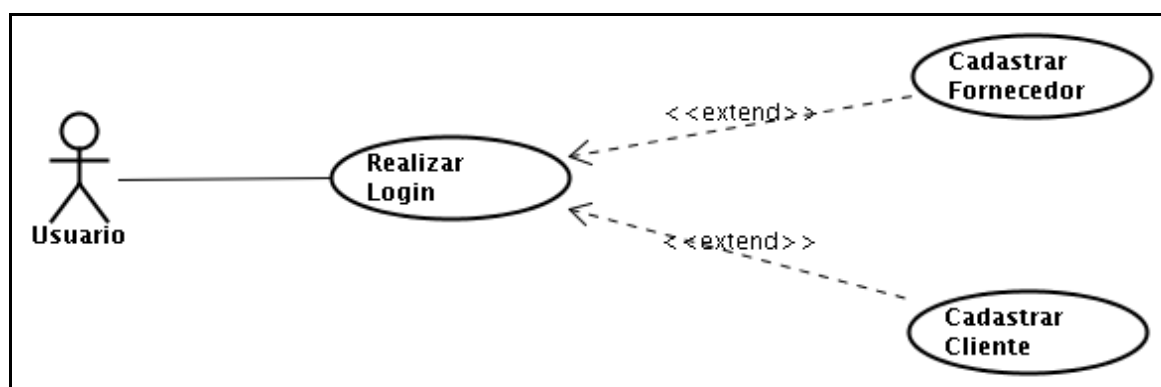
Emitir Relatório de Pedidos de Clientes

Emitir Relatório de Pedidos de Fornecedores

## 3.2 ANÁLISE ORIENTADA A OBJETOS

### 3.2.1 Especificação dos Casos de Usos

#### 3.2.1.1 Realizar Login



**Figura 1 - Caso de Uso: Realizar Login**

#### **Finalidade/Objetivo:**

Permitir que o usuário acesse o sistema para gerenciar os orçamentos.

#### **Atores:**

Cliente e Fornecedor.

#### **Evento Inicial:**

O usuário fornece seu nome de usuário e senha.

#### **Fluxo Principal:**

O usuário deve informar seus dados na tela de login;

Será feita a autenticação dos dados que o usuário informou na base de dados do sistema;

Depois de passar pela autenticação a área de controle e gerenciamento do sistema



fica disponível para o usuário.

**Fluxos Alternativos:**

Se for um novo usuário é necessário fazer o cadastro;

Se o usuário for um cliente, ele acessa sua área de controle;

Se o usuário for um fornecedor, ele acessa sua área de controle.

**Fluxos de Exceção:**

Se o usuário entrar com os dados incorretos, o sistema não abre e é necessário realizar o processo de autenticação novamente.

**Pós-Condições:**

O usuário tem total acesso a sua área principal e consequentemente pode realizar seus serviços.

**Casos de Testes:**

Validar os dados do usuário e verificar se o mesmo está cadastrado no sistema.

### 3.2.1.2 Cadastrar Fornecedor

**Finalidade/Objetivo:**

Permitir que o fornecedor crie seu login e se cadastre no sistema, e consequentemente terá a possibilidade de usar o sistema.

**Atores:**

Fornecedor.

**Pré-Condições:**

Para o fornecedor fazer o cadastro é necessário informar ao administrador do

sistema, sendo assim, é liberada a página para fazer o cadastro.

**Evento Inicial:**

O fornecedor deve preencher corretamente os seus dados na tela de cadastro.

**Fluxo Principal:**

O fornecedor preenche todos os campos da tela com seus dados;

O fornecedor seleciona o botão enviar;

Se o cadastro for feito com sucesso, o fornecedor já pode usar o sistema.

**Fluxos Alternativos:**

O fornecedor pode cancelar o cadastro.

**Fluxos de Exceção:**

Se o fornecedor informar os dados incorretos, o cadastro não é feito e é necessário fazer as correções.

**Pós-Condições:**

O fornecedor pode realizar o login e acessar o sistema para fazer suas tarefas.

**Casos de Testes:**

Validar os dados do cadastro.

### 3.2.1.3 Cadastrar Cliente

**Finalidade/Objetivo:**

Permitir que o cliente crie seu login e se cadastre no sistema, e consequentemente oferece a possibilidade de usar o sistema.

**Atores:**

Cliente.

**Pré-Condições:**

Para o cliente fazer o cadastro é necessário informar ao administrador do sistema, sendo assim, é liberada a página para fazer o cadastro.

**Evento Inicial:**

O cliente deve preencher corretamente os seus dados na tela de cadastro.

**Fluxo Principal:**

O cliente preenche todos os campos da tela com seus dados;

O cliente seleciona o botão enviar;

Se o cadastro for feito com sucesso, o sistema já pode ser usado pelo cliente.

**Fluxos Alternativos:**

O cliente pode cancelar o cadastro.

**Fluxos de Exceção:**

Se o cliente informar os dados incorretos, o cadastro não é feito e é necessário corrigir os dados.

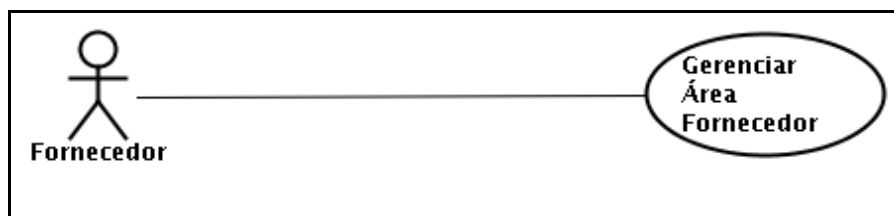
**Pós-Condições:**

O cliente pode realizar o login e acessar o sistema para fazer suas tarefas.

**Casos de Testes:**

Validar os dados do cadastro.

### 3.2.1.4 Gerenciar Área Fornecedor



**Figura 2 - Caso de Uso: Gerenciar Área Fornecedor**

**Finalidade/Objetivo:**

Permitir que o fornecedor gerencie e controle o sistema.

**Atores:**

Fornecedor.

**Pré-Condições:**

É necessário estar cadastrado e fazer autenticação no login.

**Evento Inicial:**

Exibição da área principal do sistema.

**Fluxo Principal:**

O fornecedor deve fazer a autenticação no login;

Depois de autenticado, a área principal do sistema fica disponível para o fornecedor, podendo gerenciar e controlar seus serviços.

**Fluxos Alternativos:**

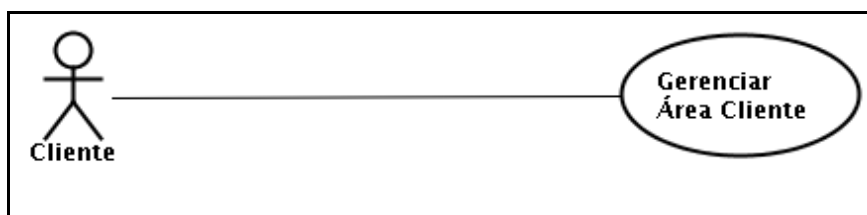
O fornecedor pode sair do sistema.

**Pós-Condições:**

O fornecedor pode cadastrar subcategorias, categoria e produto, poderá fazer a

alteração do seu cadastro, consultar relatórios, orçamento, pedidos e incluir preços nos produtos orçados.

#### 3.2.1.5 Gerenciar Área Cliente



**Figura 3 - Caso de Uso: Gerenciar Área Cliente**

#### **Finalidade/Objetivo:**

Permitir que o cliente gerencie e controle o sistema.

#### **Atores:**

Cliente

#### **Pré-Condições:**

É necessário estar cadastrado e fazer autenticação no login.

#### **Evento Inicial:**

Exibição da área principal do sistema.

#### **Fluxo Principal:**

O cliente deve realizar a autenticação no login;

Logo após, a área principal do sistema fica disponível, podendo assim, gerenciar e controlar seus serviços.

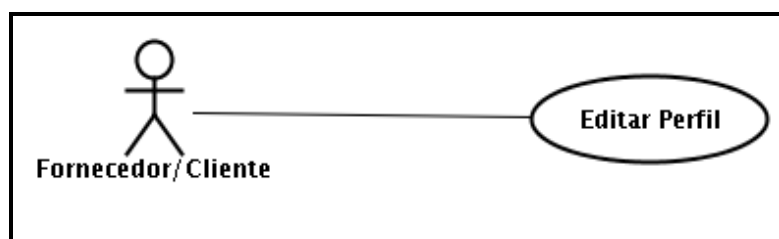
#### **Fluxos Alternativos:**

O cliente pode sair do sistema.

**Pós-Condições:**

O cliente pode visualizar e fazer seus orçamentos, alterar seu cadastro e consultar os relatórios e pedidos.

3.2.1.6 Editar Perfil



**Figura 4 - Caso de Uso: Editar Perfil**

**Finalidade/Objetivo:**

Permitir que o fornecedor ou cliente altere os dados do seu cadastro.

**Atores:**

Fornecedor e Cliente.

**Pré-Condições:**

É necessário fazer a autenticação no login.

**Evento Inicial:**

O fornecedor ou cliente deve preencher corretamente os seus dados que vão ser alterados na tela de cadastro.

**Fluxo Principal:**

Após acessar sua área de gerenciamento e controle;

O fornecedor ou cliente escolhe a opção de alteração do cadastro;

Pode fazer as alterações que forem necessárias;

Deve selecionar o botão salvar;

A tela principal do sistema é retornada.

### **Fluxos Alternativos:**

O fornecedor ou cliente pode cancelar a alteração do cadastro.

### **Fluxos de Exceção:**

Se o fornecedor ou cliente informar os dados incorretos a alteração no cadastro não é feita, e é necessário corrigi-la para concluir a alteração.

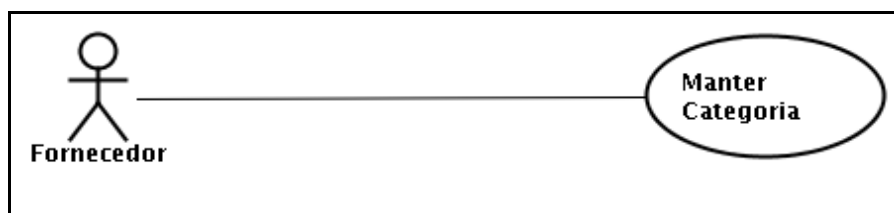
### **Pós-Condições:**

O fornecedor ou cliente poderá usar normalmente o sistema após as alterações.

### **Casos de Testes:**

Validar os dados que vão ser modificados.

#### 3.2.1.7 Manter Categoria



**Figura 5 - Caso de Uso: Manter Categoria**

**Finalidade/Objetivo:**

Permitir que o fornecedor cadastre e gerencie as categorias dos produtos.

**Atores:**

Fornecedor.

**Pré-Condições:**

É necessário fazer a autenticação no login.

**Evento Inicial:**

É mostrado as categorias que o sistema possuem.

**Fluxo Principal:**

O fornecedor deve acessar sua área de gerenciamento;

Selecionar a opção de categorias;

Deve preencher os dados da nova categoria;

Selecionar o botão salvar e é feito o cadastro da categoria.

**Fluxos Alternativos:**

Pode cancelar o cadastro.

Os dados da categoria podem ser alterados pelo fornecedor.

O fornecedor pode excluir a categoria.

Pode realizar busca de categoria para não cadastrar as mesmas com o mesmo nome.

**Fluxos de Exceção:**

Se o fornecedor informar os dados incorretos a categoria não é cadastrada ou



alterada, e é preciso fazer as correções.

Pode ocorrer que a busca não encontre a categoria desejada.

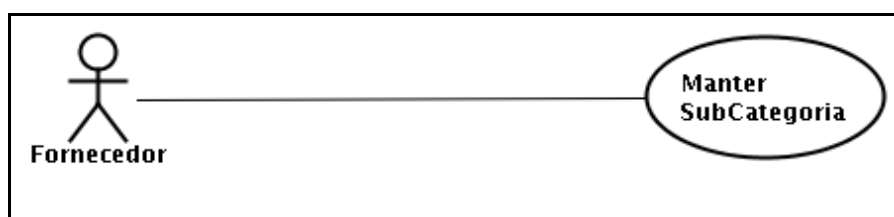
**Pós-Condições:**

O fornecedor poderá cadastrar produtos que necessitam dessa categoria.

**Casos de Testes:**

Validar os dados que vão ser cadastrados e alterados.

3.2.1.8 Manter SubCategoria



**Figura 6 - Caso de Uso: Manter SubCategoria**

**Finalidade/Objetivo:**

Permitir que o fornecedor cadastre e gerencie as subcategorias dos produtos.

**Atores:**

Fornecedor.

**Pré-Condições:**

É necessário fazer a autenticação no login.

**Evento Inicial:**

É exibido as subcategorias que estão cadastradas no sistema.

**Fluxo Principal:**

O fornecedor deve acessar sua área de controle;

Selecionar a opção de subcategorias;

Deve preencher os dados da nova subcategoria;

Selecionar o botão salvar e é feito o cadastro da subcategoria;

**Fluxos Alternativos:**

Pode cancelar o cadastro.

Os dados da subcategoria podem ser alterados pelo fornecedor.

O fornecedor pode excluir a subcategoria.

Pode realizar busca de subcategoria para não cadastrar as mesmas com o mesmo nome.

**Fluxos de Exceção:**

Se o fornecedor informar os dados incorretos a subcategoria não é cadastrada ou alterada, e é preciso fazer as correções.

Pode ocorrer que a busca não encontre a subcategoria desejada.

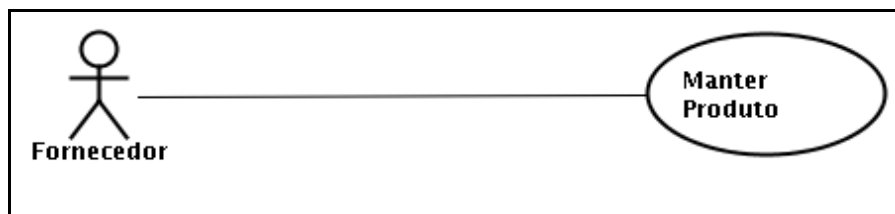
**Pós-Condições:**

O fornecedor poderá cadastrar produtos que necessitam dessa subcategoria.

**Casos de Testes:**

Validar os dados que vão ser cadastrados ou alterados.

### 3.2.1.9 Manter Produto



**Figura 7 - Caso de Uso: Manter Produto**

**Finalidade/Objetivo:**

Permitir que o fornecedor cadastre e controle os produtos que serão orçados pelos clientes.

**Atores:**

Fornecedor.

**Pré-Condições:**

É necessário fazer a autenticação no login.

**Evento Inicial:**

Os produtos cadastrados no sistema são exibidos.

**Fluxo Principal:**

Acessar sua área de gerenciamento;

O fornecedor escolhe a opção de produtos;

Deve preencher todos os dados do produto da tela de cadastro;

Selecionando o botão salvar é feito o cadastro do novo produto;

**Fluxos Alternativos:**

Pode cancelar o cadastro do produto.

As informações do produto podem ser alteradas pelo fornecedor.

O fornecedor pode excluir o produto.

Pode realizar busca de produtos para não cadastrar os mesmos com o mesmo nome.

### Fluxos de Exceção:

Se o fornecedor informar as informações incorretas o produtos não é cadastrado, e é preciso fazer as correções para que o mesmo seja alterado.

Pode ocorrer que a busca não encontre o produto desejado.

### Pós-Condições:

O cliente poderá orçar os produtos que foram cadastrados.

### Casos de Testes:

Validar as informações que irão ser cadastradas ou alteradas.

#### 3.2.1.10 Fazer Orçamento

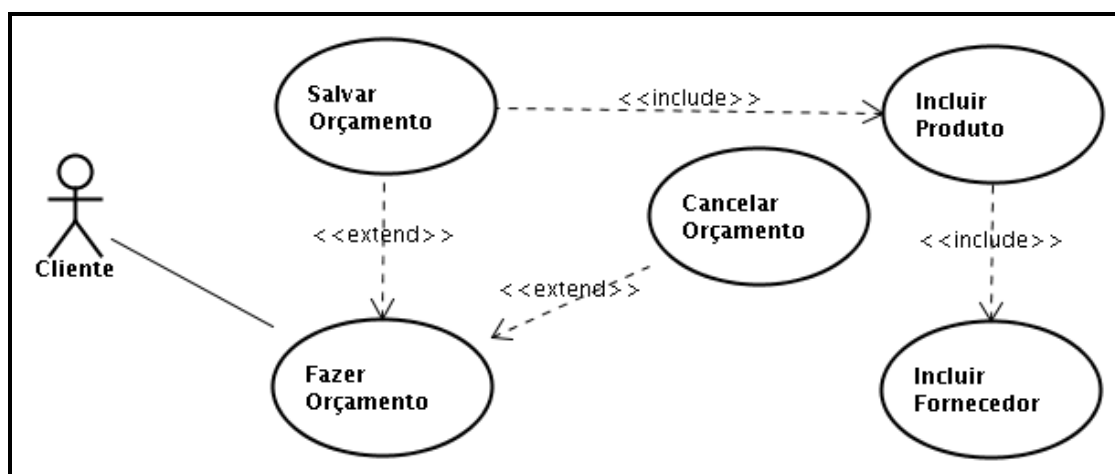


Figura 8 - Caso de Uso: Fazer Orçamento

**Finalidade/Objetivo:**

Permitir que o cliente faça os orçamentos dos produtos e consequentemente pode escolher os fornecedores que irão orçar os produtos.

**Atores:**

Cliente.

**Pré-Condições:**

É necessário fazer a autenticação no login.

**Evento Inicial:**

O cliente deve preencher as informações necessárias para fazer o orçamento.

**Fluxo Principal:**

Acessar sua área de gerenciamento e controle;

O cliente escolhe a opção de orçamento;

A tela cadastro fica disponível para a criação de um novo orçamento;

Preencher os campos do orçamento corretamente.

**Fluxos Alternativos:**

O cliente pode cancelar o orçamento.

**Fluxos de Exceção:**

Se o cliente não preencher as informações corretamente, o orçamento não é concluído e é preciso corrigi-las.

**Pós-Condições:**

O cliente poderá escolher os produtos e fornecedores que irão compor no orçamento.

**Casos de Testes:**

Verificar e validar se as informações do orçamento estão corretas.

**3.2.1.11 Salvar Orçamento****Finalidade/Objetivo:**

Permitir que o cliente possa salvar seu orçamento definitivamente.

**Atores:**

Cliente.

**Pré-Condições:**

É necessário informar os dados do orçamento corretamente.

**Evento Inicial:**

O cliente escolhe a opção de salvar o orçamento.

**Fluxo Principal:**

Acessar sua área de gerenciamento e controle;

Fazer o orçamento;

Selecionar o botão salvar.

**Fluxos Alternativos:**

O cliente pode cancelar o orçamento.

**Pós-Condições:**

O cliente pode cadastrar os produtos e fornecedores no orçamento.

### 3.2.1.12 Cancelar Orçamento

**Finalidade/Objetivo:**

Permitir que o cliente cancela o orçamento.

**Atores:**

Cliente.

**Pré-Condições:**

É necessário ter feito o orçamento.

**Evento Inicial:**

O cliente escolhe a opção de cancelar o orçamento.

**Fluxo Principal:**

Na tela de cadastro de orçamento, o cliente tem a opção de cancelar o cadastro de um novo orçamento;

É voltado para a tela de inicio do sistema.

**Fluxos Alternativos:**

O cliente pode salvar o orçamento.

**Pós-Condições:**

O orçamento é cancelado e retorna-se na tela principal do sistema.

**Casos de Testes:**

Verificar se o cliente deseja realmente cancelar o orçamento.

### 3.2.1.13 Incluir Produto

**Finalidade/Objetivo:**

Permitir que o cliente possa incluir os produtos no seu orçamento.

**Atores:**

Cliente.

**Pré-Condições:**

É necessário fazer o orçamento.

**Evento Inicial:**

O cliente pode incluir os produtos que precisam ser orçados.

**Fluxo Principal:**

Acessar sua área de gerenciamento;

Fazer primeiramente o orçamento;

Selecionar o orçamento desejado;

Buscar e salvar os produtos;

**Fluxos Alternativos:**

O cliente pode cancelar a operação e retornar na tela de orçamento.

Pode realizar uma busca para encontrar os produtos.

**Fluxos de Exceção:**

Pode não haver o produto que o cliente necessite que seja orçado.

**Pós-Condições:**

O cliente poderá escolher os fornecedores que irão orçar os produtos escolhidos.



**Casos de Testes:**

Verificar se constam produtos para ser orçado.

**3.2.1.14 Incluir Fornecedor****Finalidade/Objetivo:**

Permitir que o cliente possa escolher os fornecedores que irão orçar os produtos do orçamento.

**Atores:**

Cliente.

**Pré-Condições:**

É necessário fazer o orçamento e escolher os produtos.

**Evento Inicial:**

O cliente pode incluir os fornecedores no orçamento.

**Fluxo Principal:**

Acessar sua área de gerenciamento e controle;

Fazer primeiramente o cadastro do orçamento;

Incluir os produtos do orçamento;

Cadastrar os fornecedores que irão orçar cada produto e salvar;

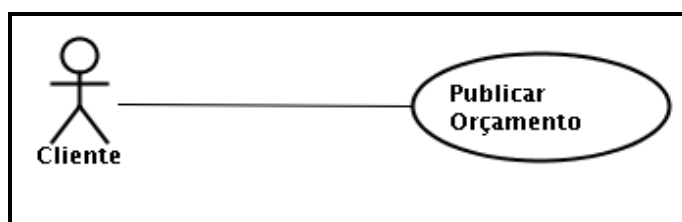
**Fluxos Alternativos:**

O cliente pode cancelar o orçamento e retornar na tela de inclusão de produtos.

Pode realizar uma busca no sistema para escolher fornecedores.

**Pós-Condições:**

O cliente pode publicar o orçamento.

**3.2.1.15 Publicar Orçamento**

**Figura 9 - Caso de Uso: Publicar Orçamento**

**Finalidade/Objetivo:**

Permitir que o cliente publique o orçamento para os fornecedores.

**Atores:**

Cliente.

**Pré-Condições:**

É necessário que os produtos e os fornecedores estejam cadastrados no orçamento.

**Evento Inicial:**

O cliente tem a opção de publicar orçamento.

**Fluxo Principal:**

Consultar os orçamentos;

Selecionar um único orçamento;

Selecionar o botão publicar o orçamento;

É enviado um e-mail para os fornecedores que foram escolhidos.

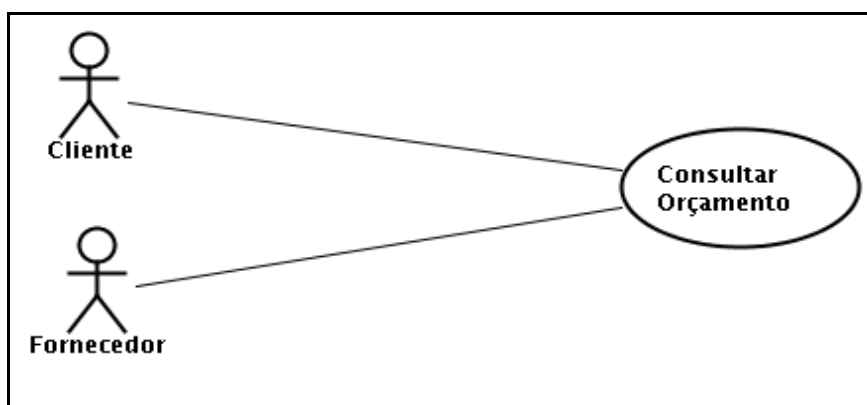
#### **Fluxos Alternativos:**

O cliente pode retornar para tela de consulta de orçamentos.

#### **Pós-Condições:**

O cliente fica no aguardo dos fornecedores, para colocarem os preços nos produtos que foram pedidos para serem orçados.

#### 3.2.1.16 Consultar Orçamento



**Figura 10 - Caso de Uso: Consultar Orçamento**

#### **Finalidade/Objetivo:**

Permitir que o cliente consulte os orçamentos que ele fez, podendo finalizar os orçamentos que estejam pendentes, também permitir que o fornecedor consulte os orçamentos que lhe foram pedidos e consequentemente pode incluir os valores dos produtos que estão no orçamento.

#### **Atores:**

Cliente e Fornecedor.

**Pré-Condições:**

É necessário que exista o orçamento.

**Evento Inicial:**

O cliente e o fornecedor pode consultar os orçamentos.

**Fluxo Principal:**

Se o usuário for o fornecedor são mostrados os orçamentos que lhe foram pedidos.

Se o usuário for o cliente é exibido os orçamentos que ele fez.

**Fluxos Alternativos:**

O fornecedor e o cliente pode realizar busca dos orçamentos.

**Fluxos de Exceção:**

Pode não haver orçamentos para os fornecedores.

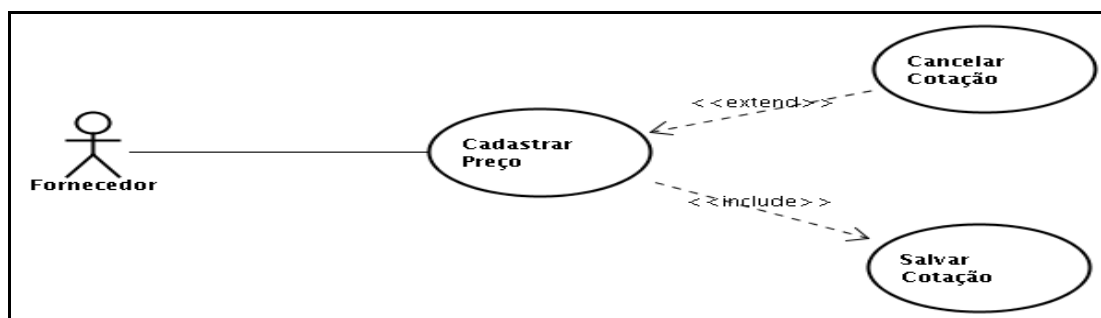
**Pós-Condições:**

O fornecedor poderá selecionar um orçamento e incluir os valores dos produtos que foram pedidos para serem orçados.

O cliente poderá finalizar o orçamento.

O cliente poderá escolher com quais fornecedores ele deseja finalizar o orçamento e emitir um pedido de compra dos produtos.

### 3.2.1.17 Cadastrar Preço



**Figura 11 - Caso de Uso: Cadastrar Preço**

#### **Finalidade/Objetivo:**

Permitir que o fornecedor informe os valores dos produtos no orçamento que o cliente fez.

#### **Atores:**

Fornecedor.

#### **Pré-Condições:**

É preciso consultar os orçamentos.

#### **Evento Inicial:**

O fornecedor tem a oportunidade de escolher os melhores preços para os produtos.

#### **Fluxo Principal:**

O fornecedor deve consultar os orçamentos;

Selecionar o orçamento que está publicado;

Selecione o produto e cadastra o valor do produto, valor do IPI, prazo de entrega e alguma observação.

#### **Fluxos Alternativos:**

O fornecedor pode não querer informar os preços para aquele cliente.

**Fluxos de Exceção:**

O fornecedor não responde o orçamento.

**Pós-Condições:**

Após informar os dados dos produtos, o fornecedor pode salvar a cotação.

**Casos de Testes:**

Verificar se os dados informados são válidos.

### 3.2.1.18 Salvar Cotação

**Finalidade/Objetivo:**

Permitir que o fornecedor salve definitivamente a cotação que lhe foi pedido.

**Atores:**

Fornecedor.

**Pré-Condições:**

É preciso consultar os orçamentos e informar os valores nos produtos.

**Fluxo Inicial:**

O fornecedor tem a opção de salvar a cotação.

**Fluxo Principal:**

O fornecedor deve consultar os orçamentos;

Selecionar o orçamento que está publicado;

Cadastrar os valores dos produtos;

Salvar a cotação;

É enviado um e-mail para o cliente, informando que o fornecedor já concluiu a cotação

**Fluxos Alternativos:**

O fornecedor pode cancelar a cotação.

**Pós-Condições:**

O cliente tem a possibilidade de ver os valores que os fornecedores informaram nos produtos dos orçamentos.

### 3.2.1.19 Cancelar Cotação

**Finalidade/Objetivo:**

Permitir que o fornecedor cancele definitivamente a cotação que lhe foi pedido.

**Atores:**

Fornecedor.

**Pré-Condições:**

É preciso consultar os orçamentos.

**Fluxo Inicial:**

O cliente escolhe a opção de cancelar a cotação.

**Fluxo Principal:**

Após ter seguido todos os passos para fazer a cotação, o fornecedor tem a opção de cancelar o cadastro dos preços nos produtos;

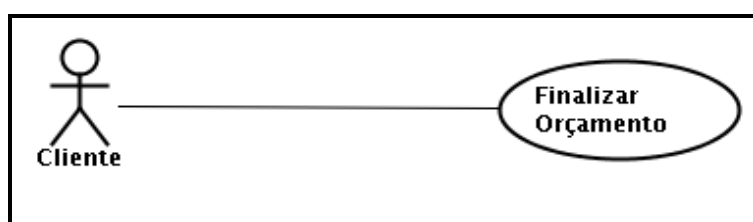
**Fluxos Alternativos:**

O cliente pode salvar a cotação.

**Pós-Condições:**

A cotação é cancelada e retorna-se na tela principal do sistema.

3.2.1.20 Finalizar Orçamento



**Figura 12 - Caso de Uso: Finalizar Orçamento**

**Finalidade/Objetivo:**

Permitir que o cliente finalize definitivamente o orçamento.

**Atores:**

Cliente.

**Pré-Condições:**

É necessário ter criado um orçamento.

**Evento Inicial:**

O cliente tem a opção de finalizar o orçamento.

**Fluxo Principal:**

Consultar os orçamentos;

Verificar se os fornecedores informaram os preços dos produtos;



Seleciona o botão finalizar o orçamento;

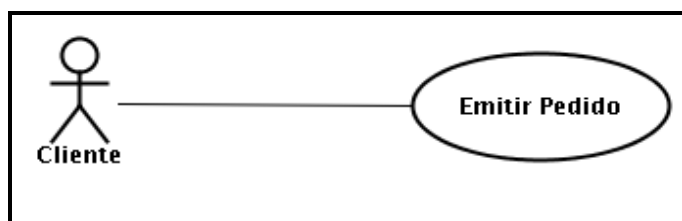
### **Fluxos Alternativos:**

O cliente pode cancelar o orçamento e retornar para área de gerenciamento.

### **Pós-Condições:**

Após ter finalizado o orçamento, o cliente pode gerar um pedido pelo sistema ou simplesmente negociar com o fornecedor por outro meio.

#### 3.2.1.21 Emitir Pedido de Compra



**Figura 13 - Caso de Uso: Emitir Pedido de Compra**

### **Finalidade/Objetivo:**

Permitir que o cliente emite um ou mais pedido de compra para o fornecedor, de acordo com o orçamento selecionado.

### **Atores:**

Cliente.

### **Pré-Condições:**

É necessário que o orçamento esteja finalizado;

É necessário que os fornecedores tenham informados os valores nos produtos.

### **Evento Inicial:**

O cliente tem a opção para emitir o pedido de compra para o fornecedor.

**Fluxo Principal:**

O cliente deve consultar os orçamentos;

O cliente deve finalizar o orçamento;

Selecionar o botão de emitir pedido de compra;

O cliente tem a opção de escolher os fornecedores que possuem o melhor preço ou prazo de entrega.

Selecionar o fornecedor que oferece a melhor proposta.

O cliente seleciona a botão para emitir o pedido de compra para o fornecedor escolhido.

É enviado um e-mail de confirmação para o fornecedor selecionado.

**Fluxos Alternativos:**

O cliente pode retornar para consultar de orçamentos.

**Fluxos de Exceção:**

Pode ocorrer que o fornecedor não tinha informado os preços dos produtos.

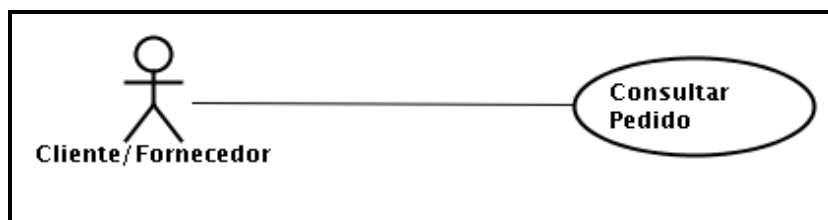
**Pós-Condições:**

O cliente e o fornecedor escolhido pode consultar o pedido de compra.

**Casos de Testes:**

Verificar se foi selecionado pelo menos um fornecedor único.

### 3.2.1.22 Consultar Pedido



**Figura 14 - Caso de Uso: Consultar Pedido**

**Finalidade/Objetivo:**

Permitir que o fornecedor e cliente consulte os pedidos de compras.

**Atores:**

Fornecedor e Cliente.

**Pré-Condições:**

É necessário que o cliente tenha gerado os pedidos.

**Evento Inicial:**

É mostrado os pedidos de compras que o fornecedor e o cliente possuem.

**Fluxo Principal:**

Acessar a área de gerenciamento do sistema;

Escolher a opção para consultar pedidos;

É mostrado os pedidos que o cliente e fornecedor possuem.

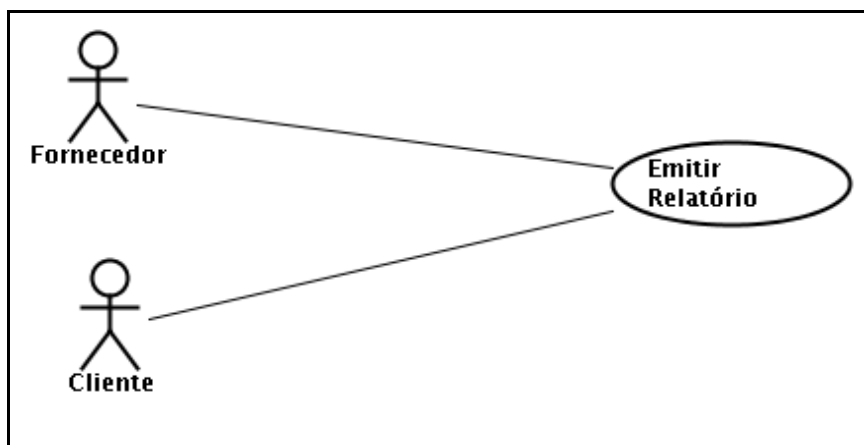
**Fluxos de Alternativo:**

Voltar para área de gerenciamento.

**Fluxos de Exceção:**

Pode não haver pedido.

### 3.2.1.23 Emitir Relatório



**Figura 15 - Caso de Uso: Emitir Relatório**

#### **Finalidade/Objetivo:**

Permitir que o cliente ou fornecedor consulte os relatórios dos orçamentos, produtos, pedidos, etc...

#### **Atores:**

Cliente e Fornecedor.

#### **Pré-Condições:**

É necessário que tenha produtos, clientes e fornecedores cadastrados e também orçamentos finalizados.

#### **Evento Inicial:**

Acessar a área principal do sistema;

O cliente ou fornecedor tem a opção de escolher quais os tipos de relatórios que desejam visualizar.

**Fluxo Principal:**

Após ter acessado a área de controle de orçamentos, o cliente pode gerenciar os seguintes relatórios do sistema:

Emitir Relatório de Produtos

Emitir Relatório de Produtos por Fornecedor

Emitir Relatório de Fornecedores por Cidade

Emitir Relatório de Orçamentos por Cliente

Emitir Relatório de Pedidos de Clientes

Após ter acessado a área de controle de orçamentos, o fornecedor pode gerenciar os seguintes relatórios do sistema:

Emitir Relatório de Produtos

Emitir Relatório de Produtos por Fornecedor

Emitir Relatório de Clientes por Cidade

Emitir Relatório de Categorias

Emitir Relatório de SubCategorias

Emitir Relatório de Orçamentos por Fornecedor

Emitir Relatório de Pedidos de Fornecedores

**Fluxos Alternativos:**

O cliente ou fornecedor pode voltar para a área de gerenciamento.

### 3.2.2 Diagrama de Caso de Uso Global

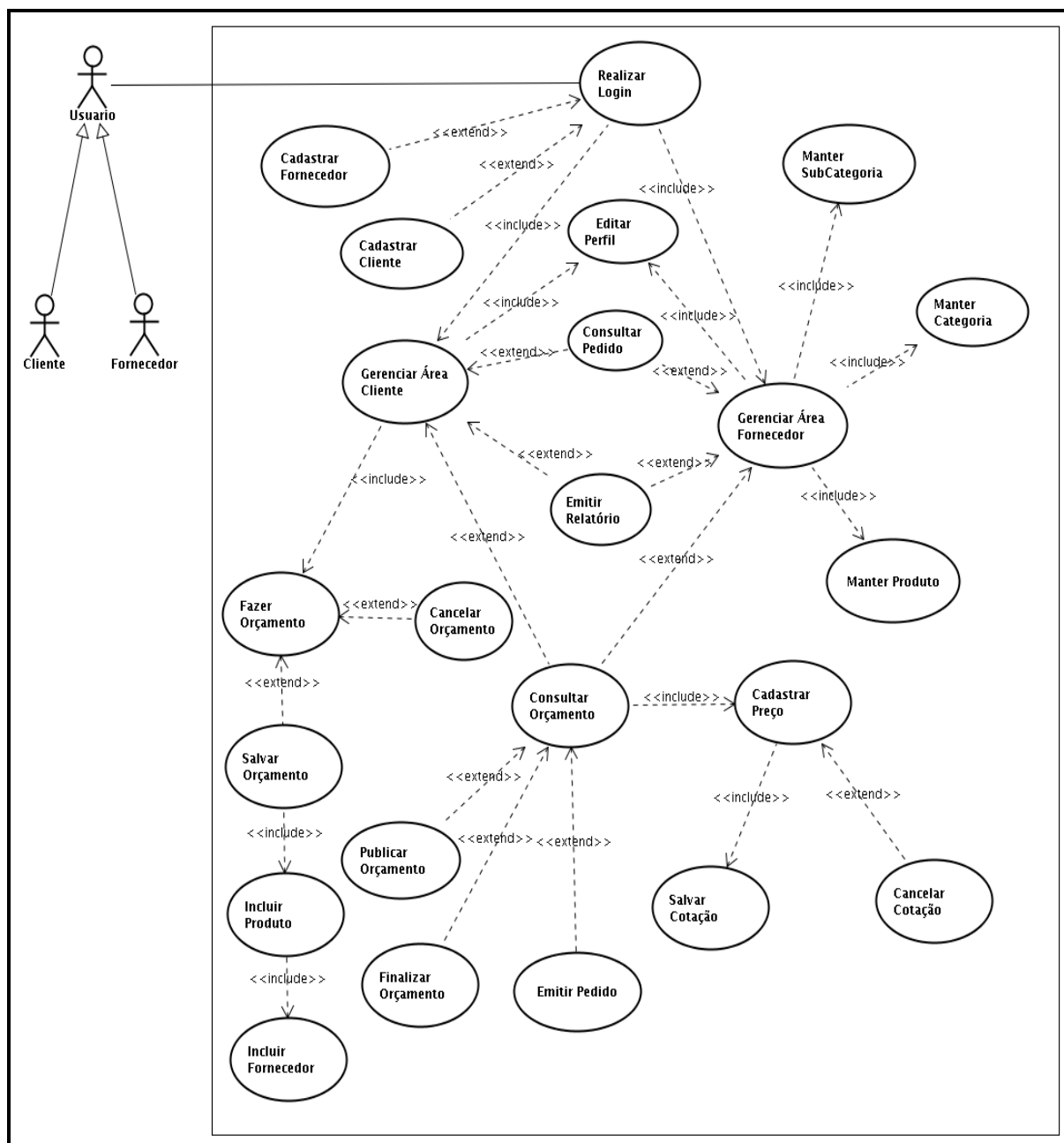


Figura 16 - Diagrama de Caso de Uso Global

### 3.2.3 Diagrama de Classe

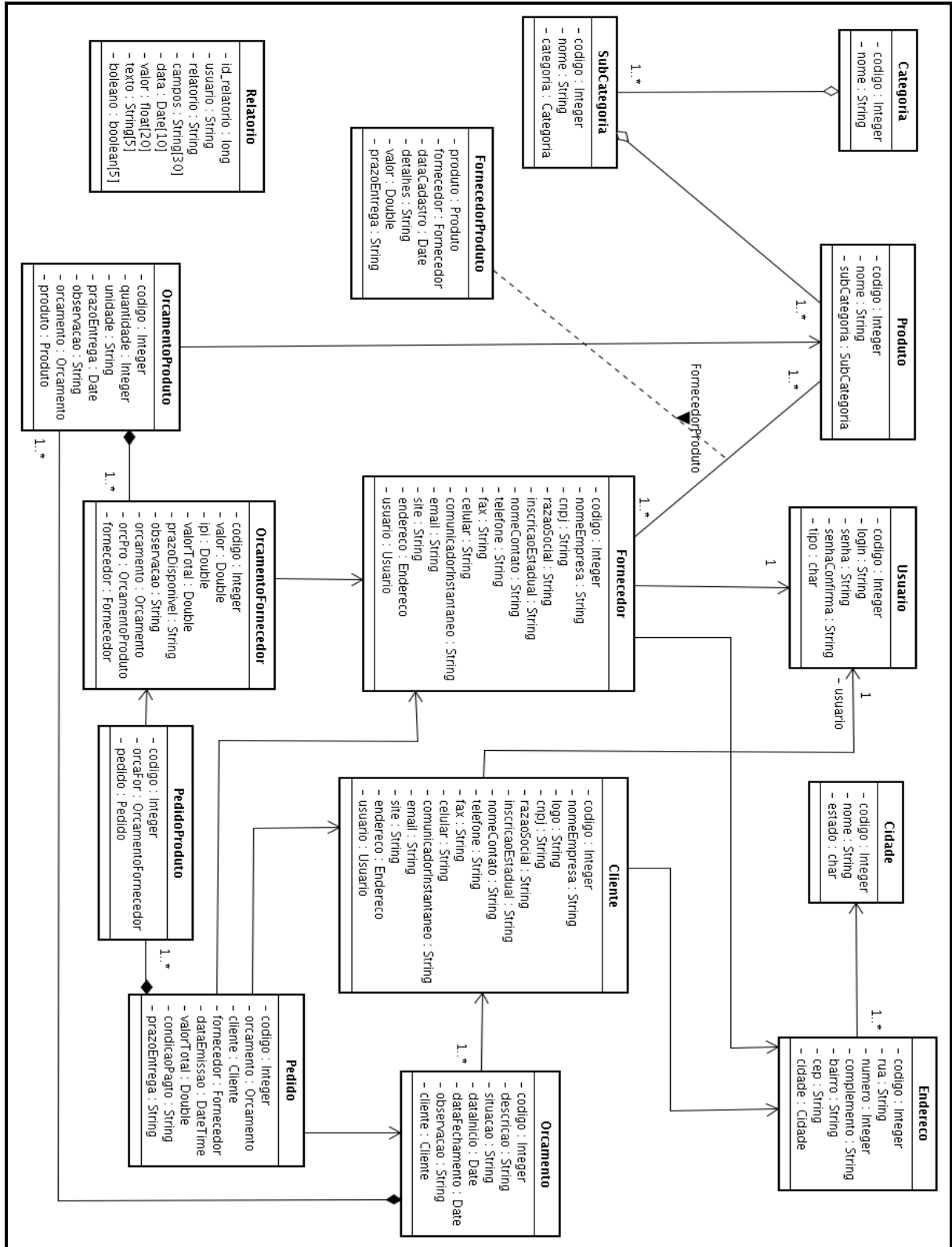
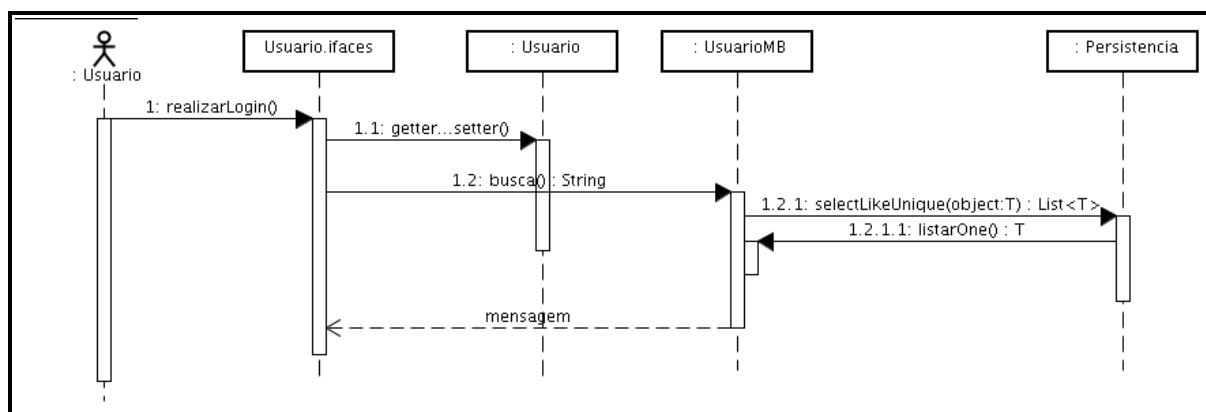


Figura 17 - Diagrama de Classes de Negócio

### 3.2.4 Diagrama de Sequência

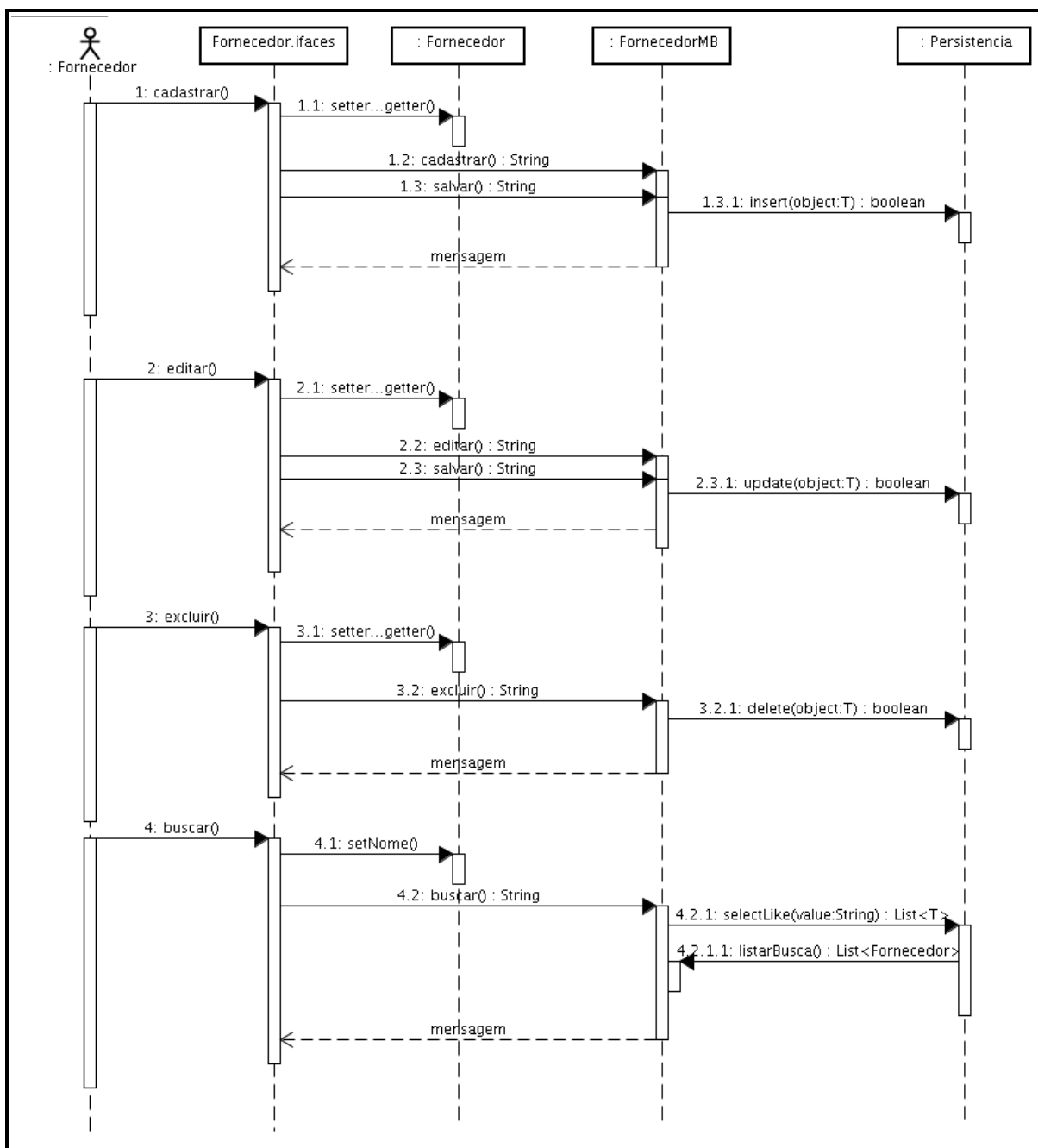
#### 3.2.4.1 Diagrama de Sequência: Realizar Login



**Figura 18 - Diagrama de Sequência: Realizar Login**

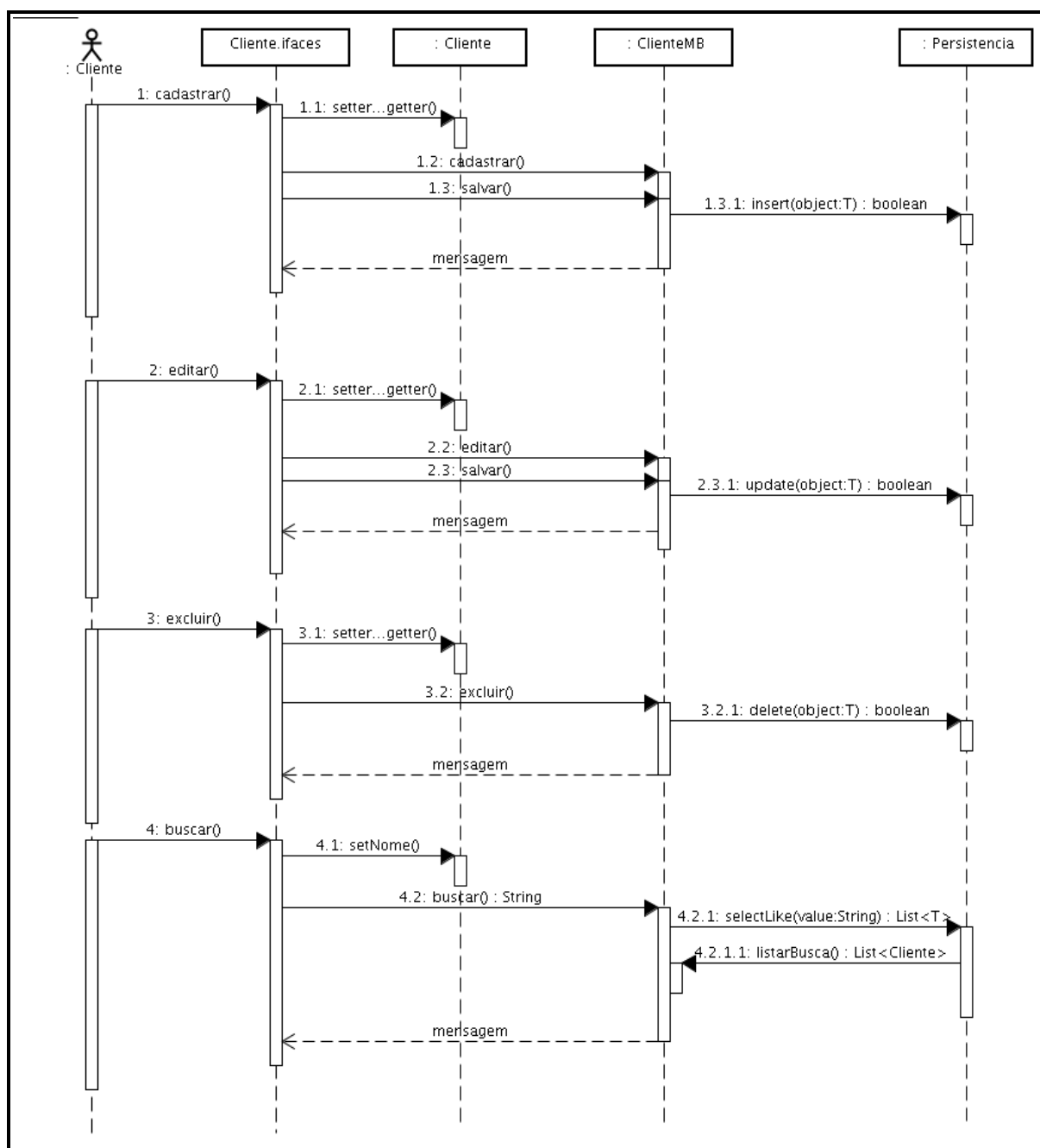


### 3.2.4.2 Diagrama de Sequência: Manter Fornecedor



**Figura 19 - Diagrama de Sequência: Manter Fornecedor**

### 3.2.4.3 Diagrama de Sequência: Manter Cliente



**Figura 20 - Diagrama de Sequência: Manter Cliente**

## 3.2.4.4 Diagrama de Sequência: Manter Usuário

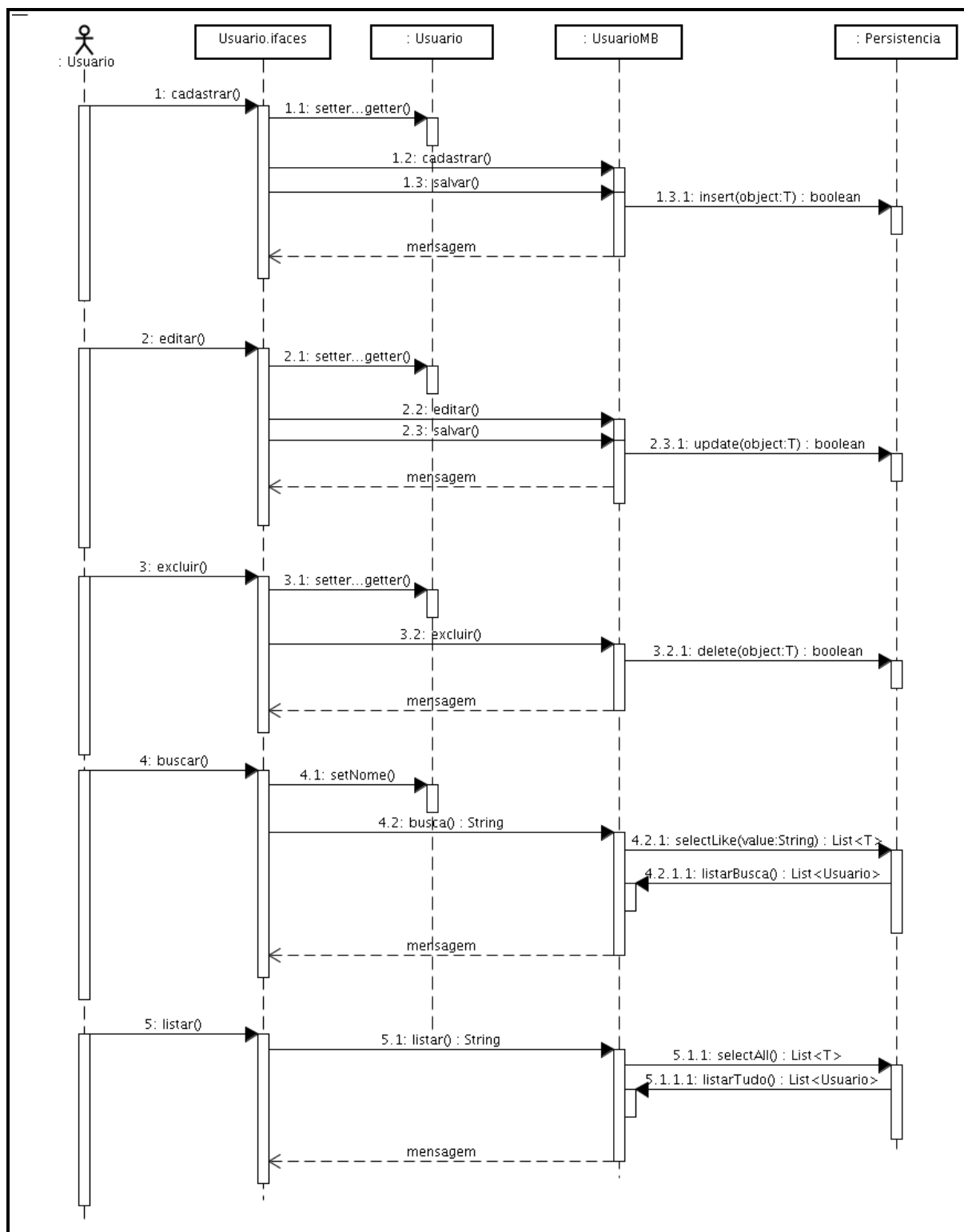


Figura 21 - Diagrama de Sequência: Manter Usuário

## 3.2.4.5 Diagrama de Sequência: Manter Produto

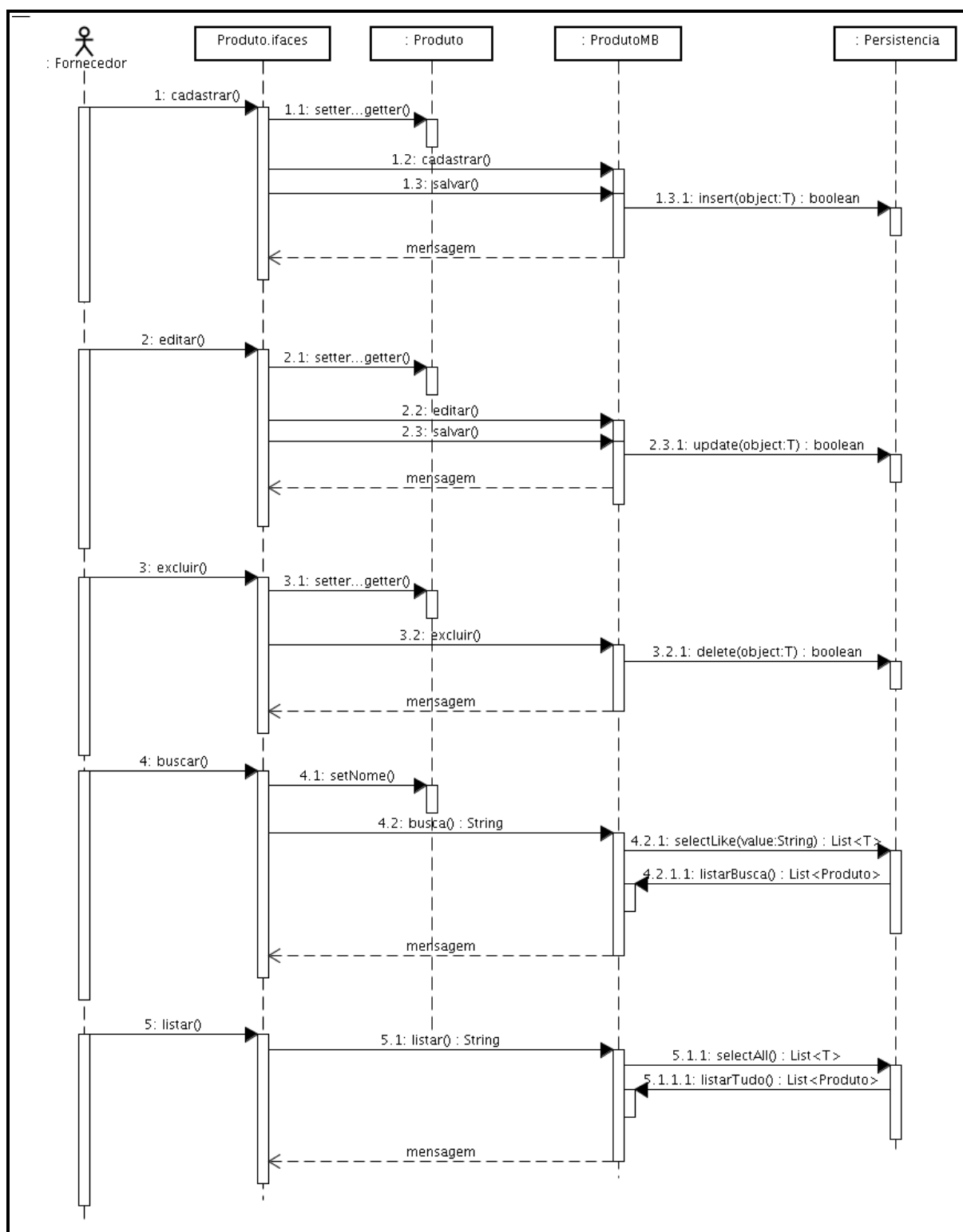
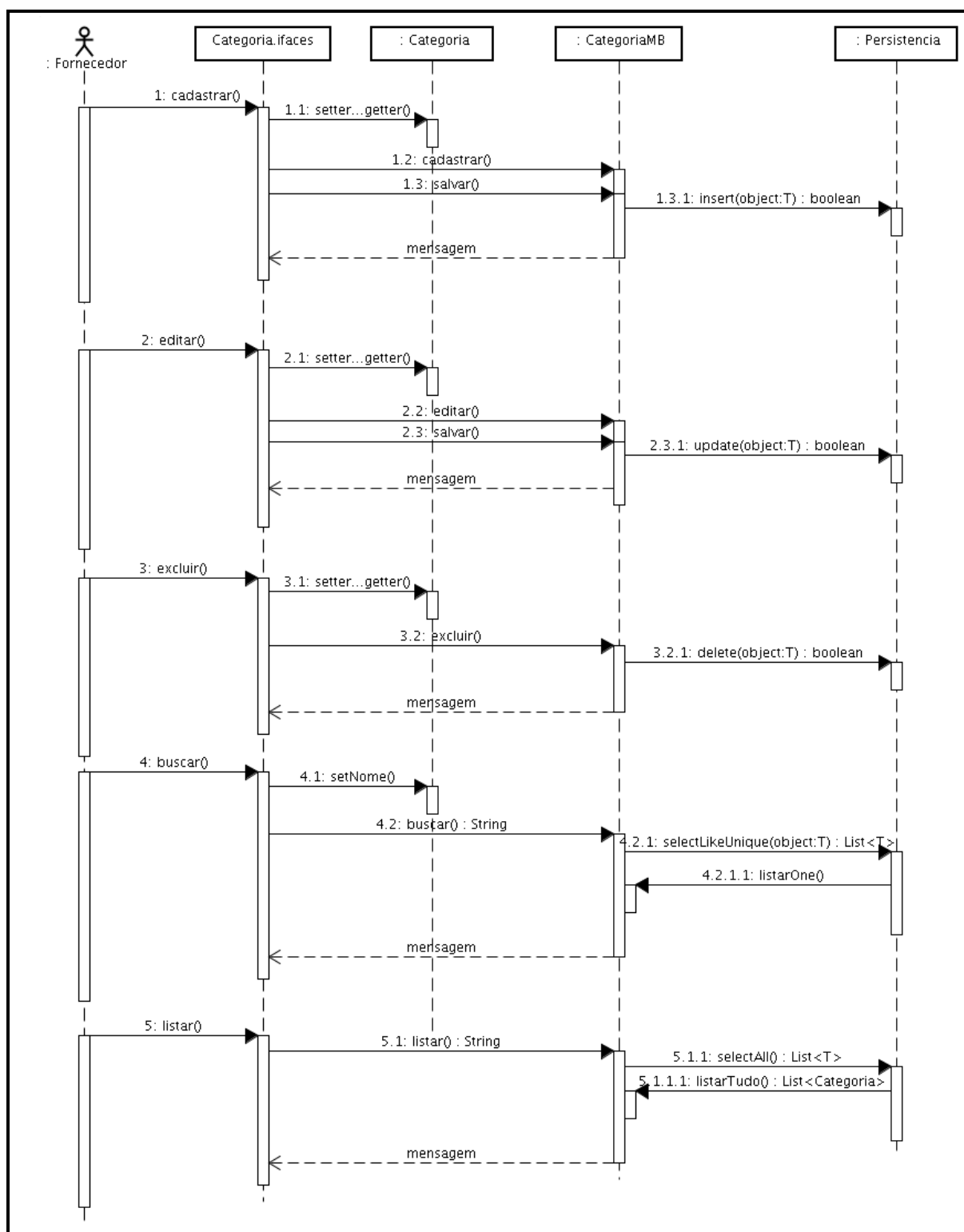


Figura 22 - Diagrama de Sequência: Manter Produto

### 3.2.4.6 Diagrama de Sequência: Manter Categoria



**Figura 23 - Diagrama de Sequência: Manter Categoria**

## 3.2.4.7 Diagrama de Sequência: Manter SubCategoria

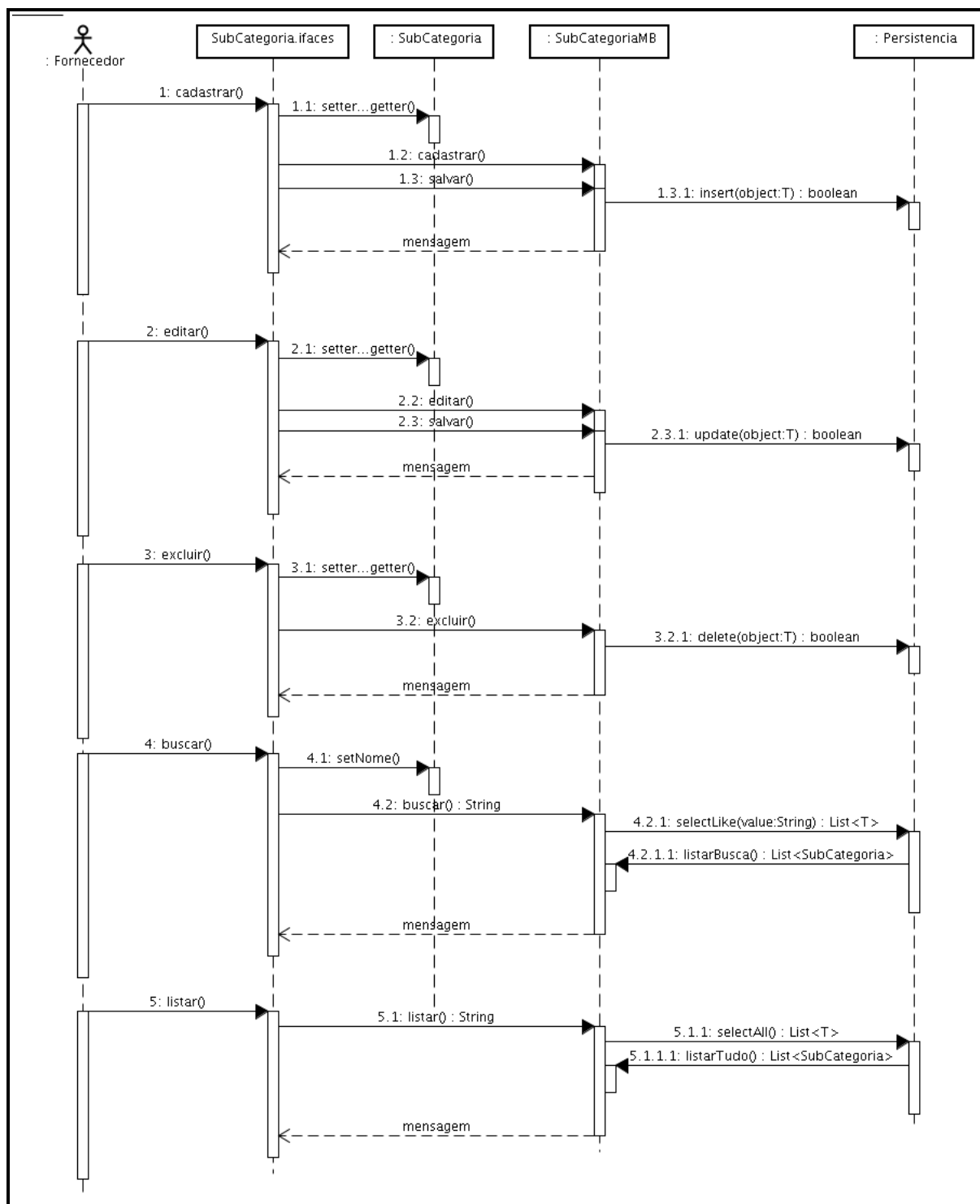


Figura 24 - Diagrama de Sequência: Manter SubCategoria

## 3.2.4.8 Diagrama de Sequência: Movimentação Orçamentos

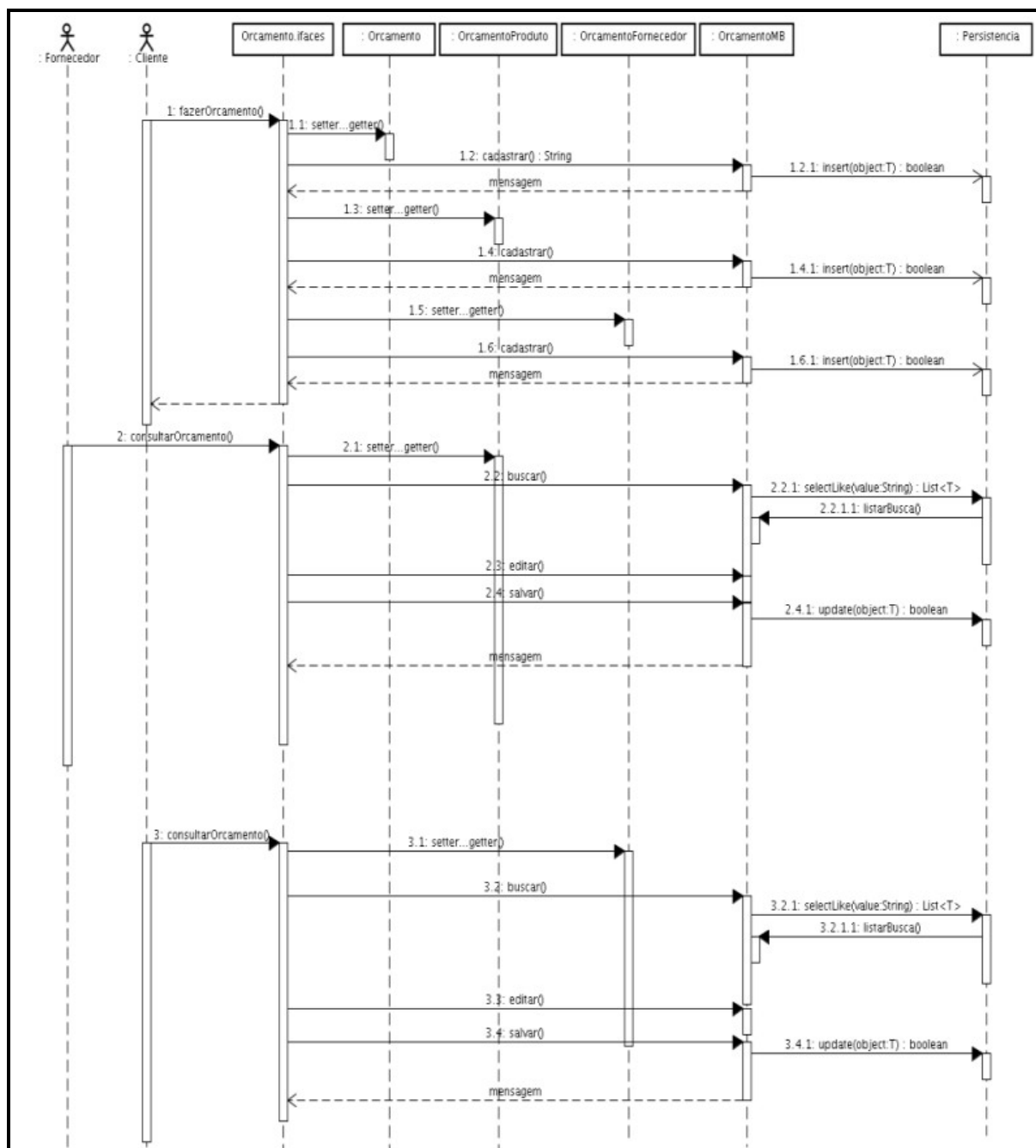
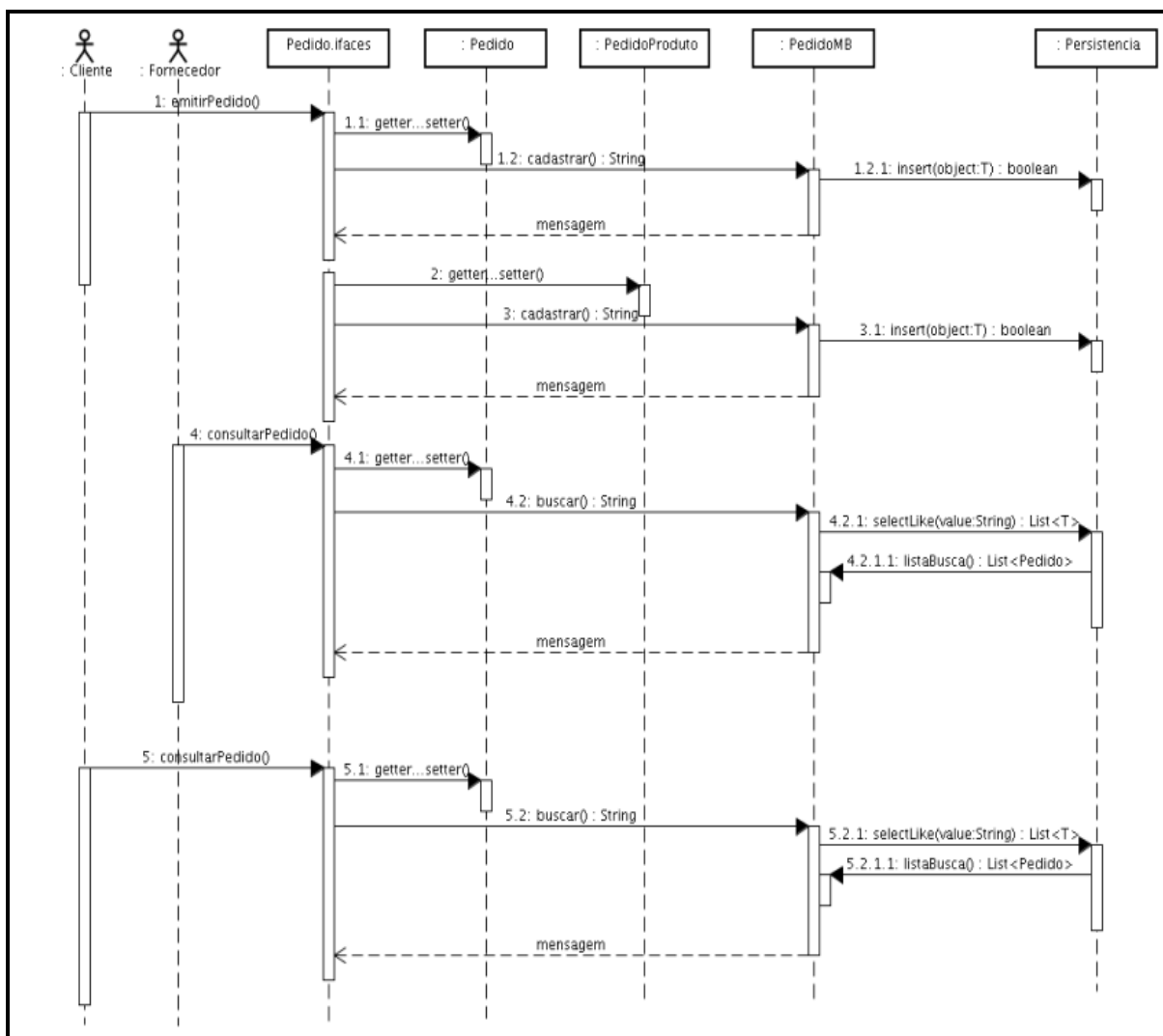


Figura 25 - Diagrama de Sequência: Movimentação Orçamentos

### 3.2.4.9 Diagrama de Sequência: Emitir Pedido de Compra

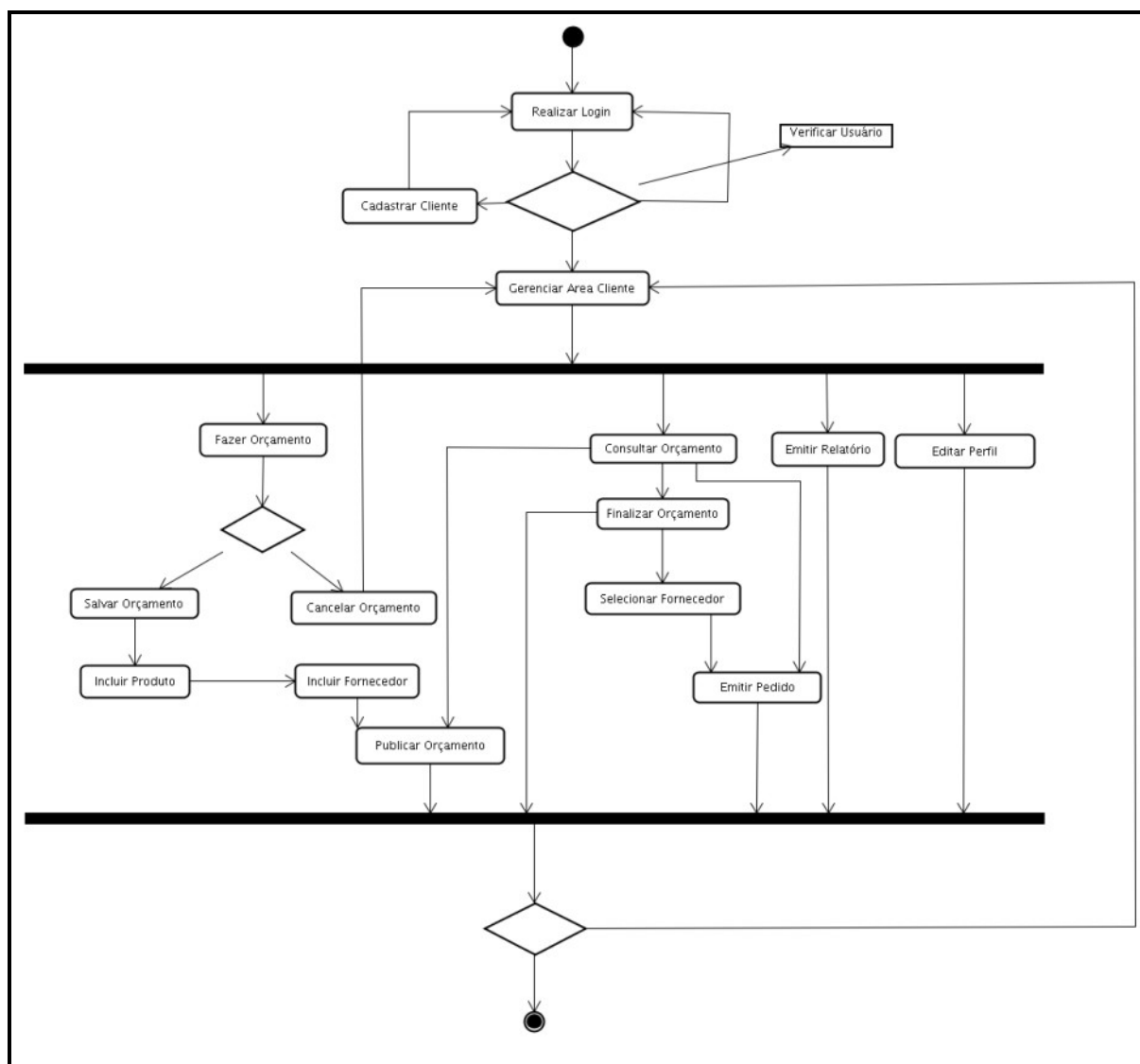


**Figura 26 - Diagrama de Sequência: Emitir Pedido de Compra**



### 3.2.5 Diagrama de Atividades

#### 3.2.5.1 Diagrama de Atividade Cliente



**Figura 27 - Diagrama de Atividade: Cliente**

## 3.2.5.2 Diagrama de Atividade Fornecedor

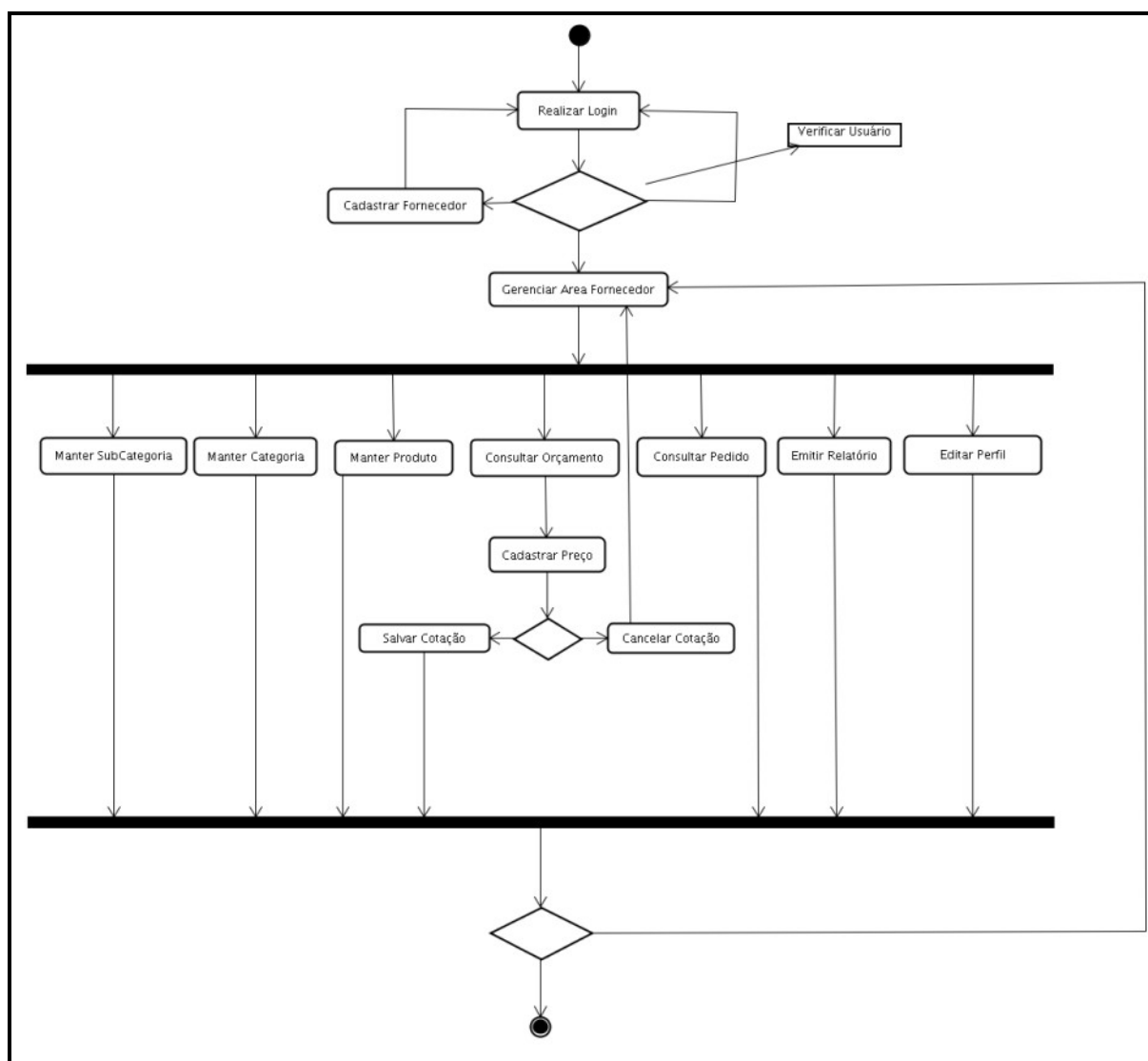


Figura 28 - Diagrama de Atividade: Fornecedor

### 3.3 PLANEJAMENTO DO PROJETO

#### 3.3.1 WBS – Work Breakdown Structure

WBS é um agrupamento de elementos do projeto que organiza e define total abrangência do projeto.

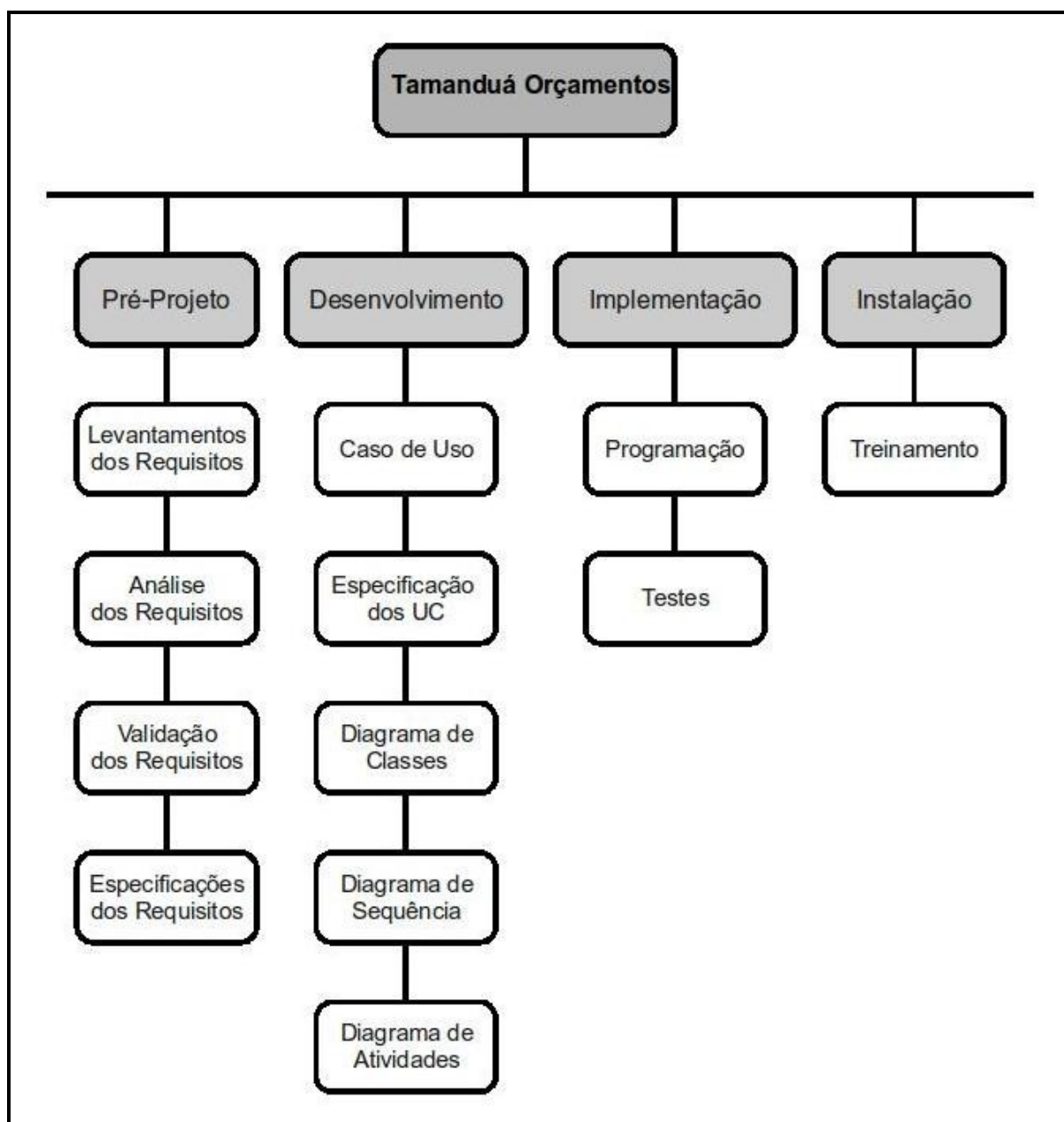


Figura 29 - Diagrama WBS

### **3.3.2 Definição das Atividades**

- Levantamento dos requisitos;
- Análise do requisitos;
- Validação dos requisitos;
- Especificação dos requisitos;
- Caso de Uso;
- Especificação dos casos de uso;
- Diagrama de classes;
- Diagrama de sequência;
- Diagrama de atividades;
- Programação;
- Testes;
- Treinamento.

### 3.3.3 Sequenciamento das Atividades Definidas

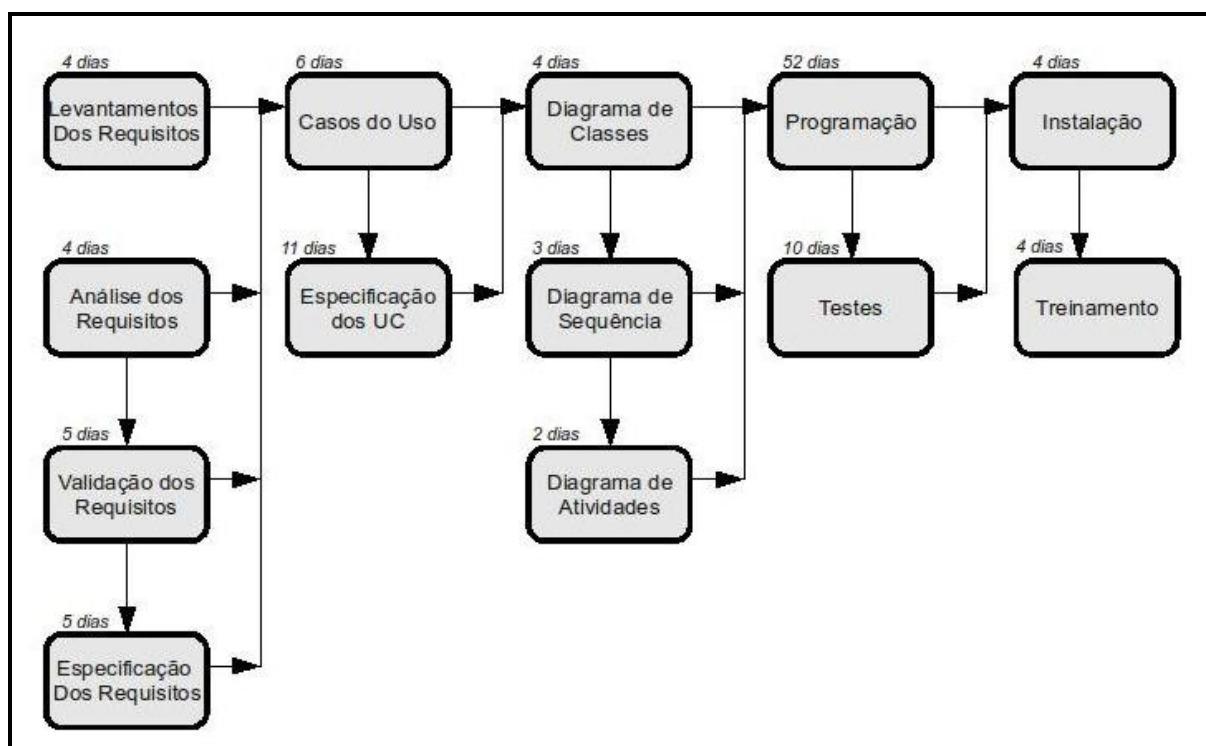
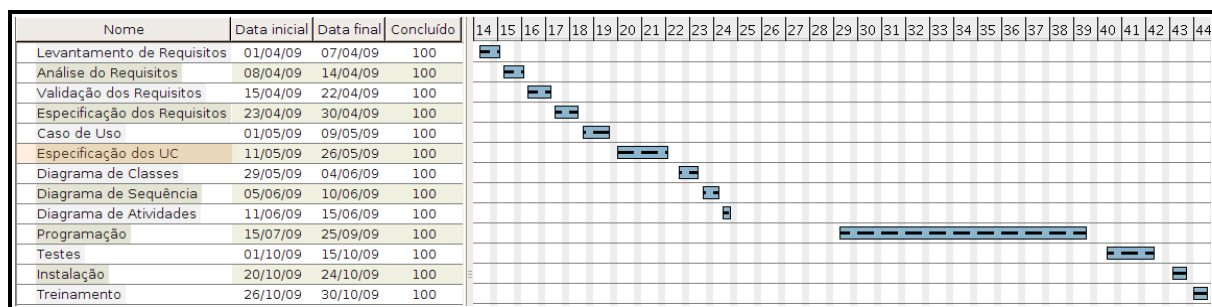


Figura 30 - Sequenciamento das Atividades Definidas

### 3.3.4 Cronograma das Atividades Definidas



**Figura 31 - Cronograma das Atividades Definidas**

### 3.3.5 Recursos Necessários para o Desenvolvimento do Projeto

1 (um) Analista de Sistema;

1 (um) Programador Java;

1 (um) Notebook;

1 (uma) Impressora;

Sistema Operacional Linux – Distribuição Ubuntu;

IDE para o desenvolvimento: *Netbeans*;

IDE para modelagem de dados: *JUDE*;

Linguagem para o desenvolvimento: *Java*;

Servidor de Aplicação: *GlassFish*;

Bando de dados: *MySQL*;

Aplicativos: *OpenOffice* e *GanttProject*.

### 3.3.6 Estimativa de Custos

#### **Analista de Sistema**

Quantidade dias: 44

Custo diário: R\$ 80,00 (Oitenta reais)

Custo total: (Total de dias \* Custo diário) = R\$ 3.520,00 (Três mil quinhentos e vinte reais)

#### **Programador**

Quantidade dias: 66

Custo diário: R\$ 65,00 (Sessenta e cinco reais)

Custo total: (Total de dias \* Custo diário) = R\$ 4.290,00 (Quatro mil duzentos e noventa reais)

#### **NoteBook**

Valor Unitário: R\$ 2.200,00 (Dois mil e duzentos reais)

Dias de uso: 110 dias

Depreciação: (R\$ 2.200,00 / 24 meses) = R\$ 91,67 (Noventa um reais e sessenta e sete centavos) por mês

Custo dia: (R\$ 91,67 / 30 dias) = R\$ 3,05 (Três reais e cinco centavos) por dia

Custo do Notebook: (Total de dias \* Custo dia) = R\$ 335,00 (Trezentos e trinta e cinco reais)

#### **Impressora**

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

Dias de uso: 44 dias

Depreciação: (R\$ 300,00 / 24 meses) = R\$ 12,50 (Doze reais e cinquenta centavos) por mês

Custo dia: (R\$ 12,50 / 30 dias) = R\$ 0,41 (Quarenta e um centavos) por dia

Custo da Impressora: (Total de dias \* Custo dia) = R\$ 18,04 (Dezoito reais e quatro centavos)

### **Sistema Operacional Linux – Distribuição Ubuntu**

Custo: R\$ 0,00 (*Freeware*)

### **IDE para o desenvolvimento - Netbeans**

Custo: R\$ 0,00 (*Freeware*)

### **IDE para a modelagem de dados - JUDE**

Custo: R\$ 0,00 (*Freeware*)

### **Linguagem para o desenvolvimento: Java;**

Custo: R\$ 0,00 (*Freeware*)

### **Servidor de Aplicação: GlassFish;**

Custo: R\$ 0,00 (*Freeware*)

### **Bando de dados: MySQL;**

Custo: R\$ 0,00 (*Freeware*)

### **Aplicativos: OpenOffice e GanttProject.**

Custo: R\$ 0,00 (*Freeware*)



### 3.3.7 Orçamento do Projeto

Recursos	Valor
Analista de Sistema	R\$ 3.520,00
Programador Java	R\$ 4.290,00
NoteBook	R\$ 355,00
Impressora	R\$ 18,04
Sistema Operacional Linux – Distribuição Ubuntu;	R\$ 0,00
IDE para o desenvolvimento: <i>Netbeans</i> ;	R\$ 0,00
IDE para modelagem de dados: JUDE;	R\$ 0,00
Linguagem para o desenvolvimento: Java;	R\$ 0,00
Servidor de Aplicação: <i>GlassFish</i> ;	R\$ 0,00
Bando de dados: <i>MySQL</i> ;	R\$ 0,00
Aplicativos: <i>OpenOffice</i> e <i>GanttProject</i> .	R\$ 0,00
<b>Custo Total do Projeto:</b>	<b>R\$ 8.183,04</b>

**Tabela 1 - Orçamento do Projeto**

## 4 IMPLEMENTAÇÃO

O sistema foi desenvolvido usando o conceito de MVC, que separa uma aplicação em três camadas distintas, sendo: *Model*, *View* e *Controller*.

O *Model* (Modelo) é o objeto que representa os dados do programa. Gerencia esses dados e controlam todas suas informações (GONÇALVES, 2008).

A *View* (Apresentação) é a que opera a apresentação visual dos dados representados pelo *Model*, está relacionada a exibir os dados ou informações da aplicação (GONÇALVES, 2008).

O *Controller* (Controlador) é responsável por coordenar os dois anteriores, exibindo a apresentação correta ou executando algum trabalho que o sistema precisa finalizar (GONÇALVES, 2008).

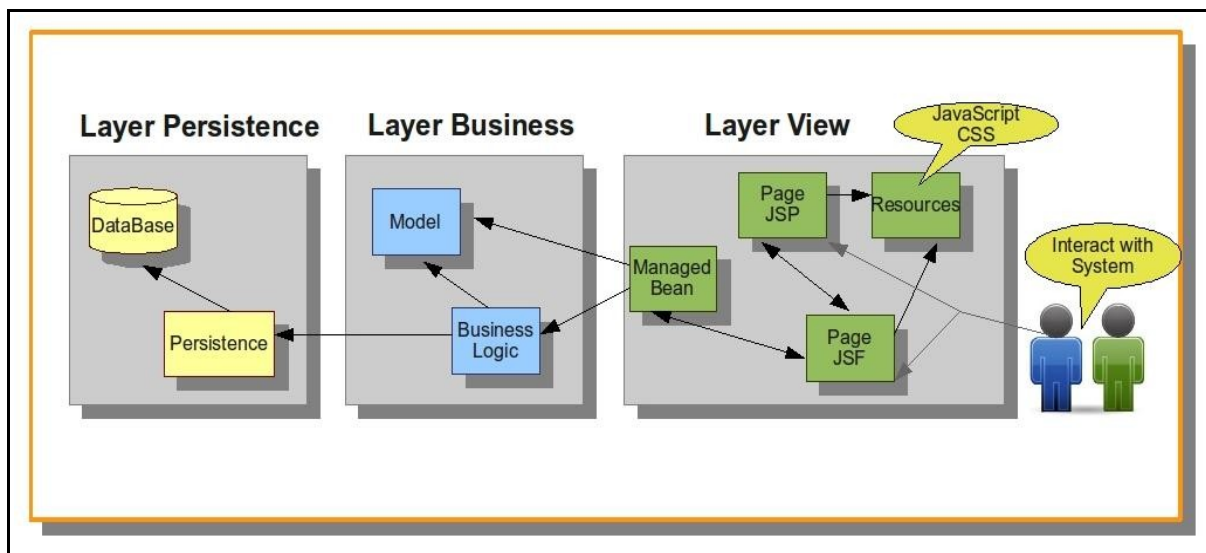
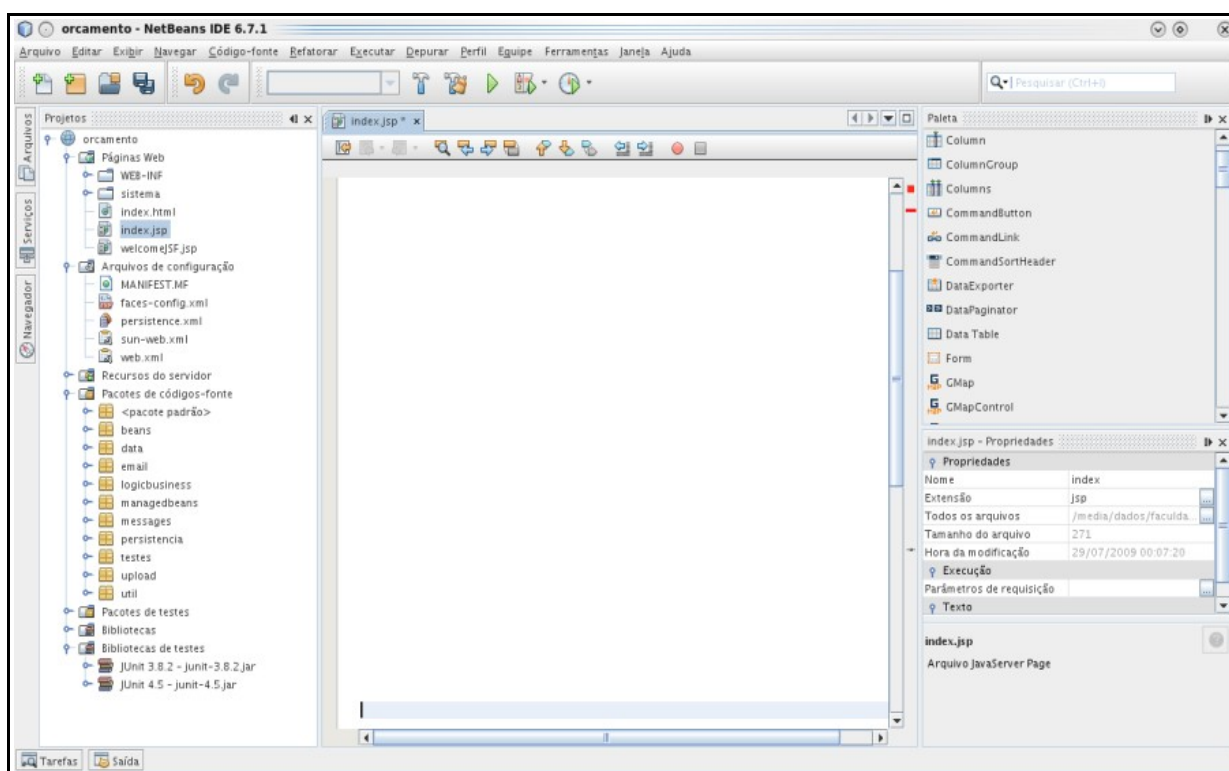


Figura 32 - Arquitetura da Aplicação

O sistema foi desenvolvido utilizando o *Netbeans*, o tipo do projeto escolhido foi uma aplicação para web, sendo assim, o *Netbeans* separa as partes que são referentes as páginas web, os pacotes de códigos de fonte, arquivos de configuração e bibliotecas.



**Figura 33 - Estrutura de Aplicações Web no Netbeans**

A primeira parte da aplicação é onde está localizada as páginas web, neste diretório contém todos os arquivos de páginas com a extensão *.jsp*, e todos eles estão separados por diretório de acordo com as funcionalidades que o sistema possui. Nesta parte, também possuem, o diretório das imagens que foram usadas no sistema, o arquivo de CSS padrão, CSS para impressão e o arquivo de Javascript.

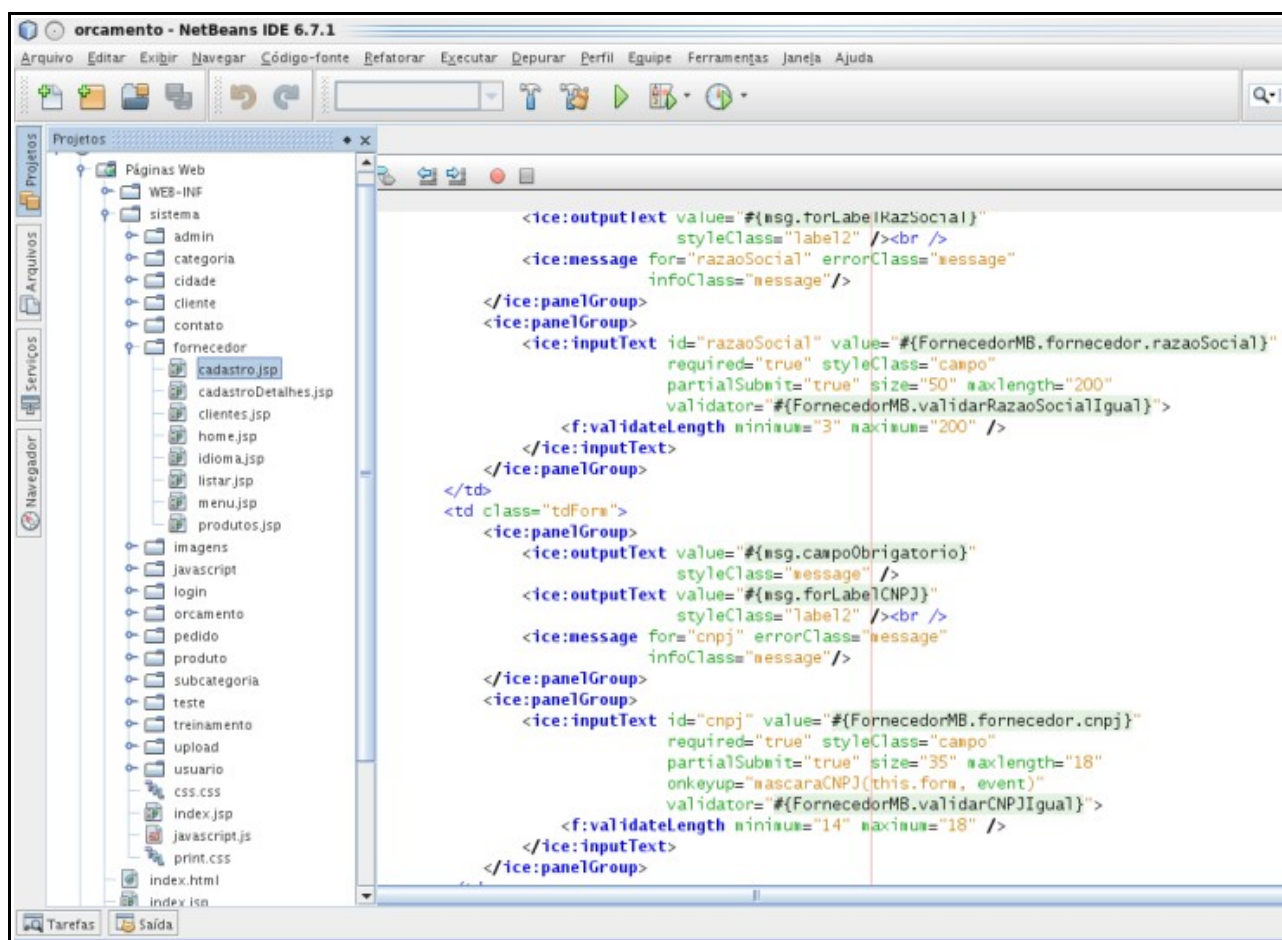


Figura 34 - Páginas Web no Netbeans

A segunda parte da aplicação é onde está localizados os arquivos de configuração, neste diretório contém alguns arquivos, como por exemplo:

O arquivo *faces-config.xml* é usado pelo JSF para configurar os *managed beans*, regras de navegação, *locale*, *resource-bundle*, etc.

O arquivo *persistence.xml* é usado para criar uma unidade de persistência, neste arquivo deve-se selecionar qual *data-source* será utilizado.

O arquivo *web.xml* contém diversas configurações da aplicação, como: configurar o tempo de sessão, mapeamento de servlet e filtro, etc.

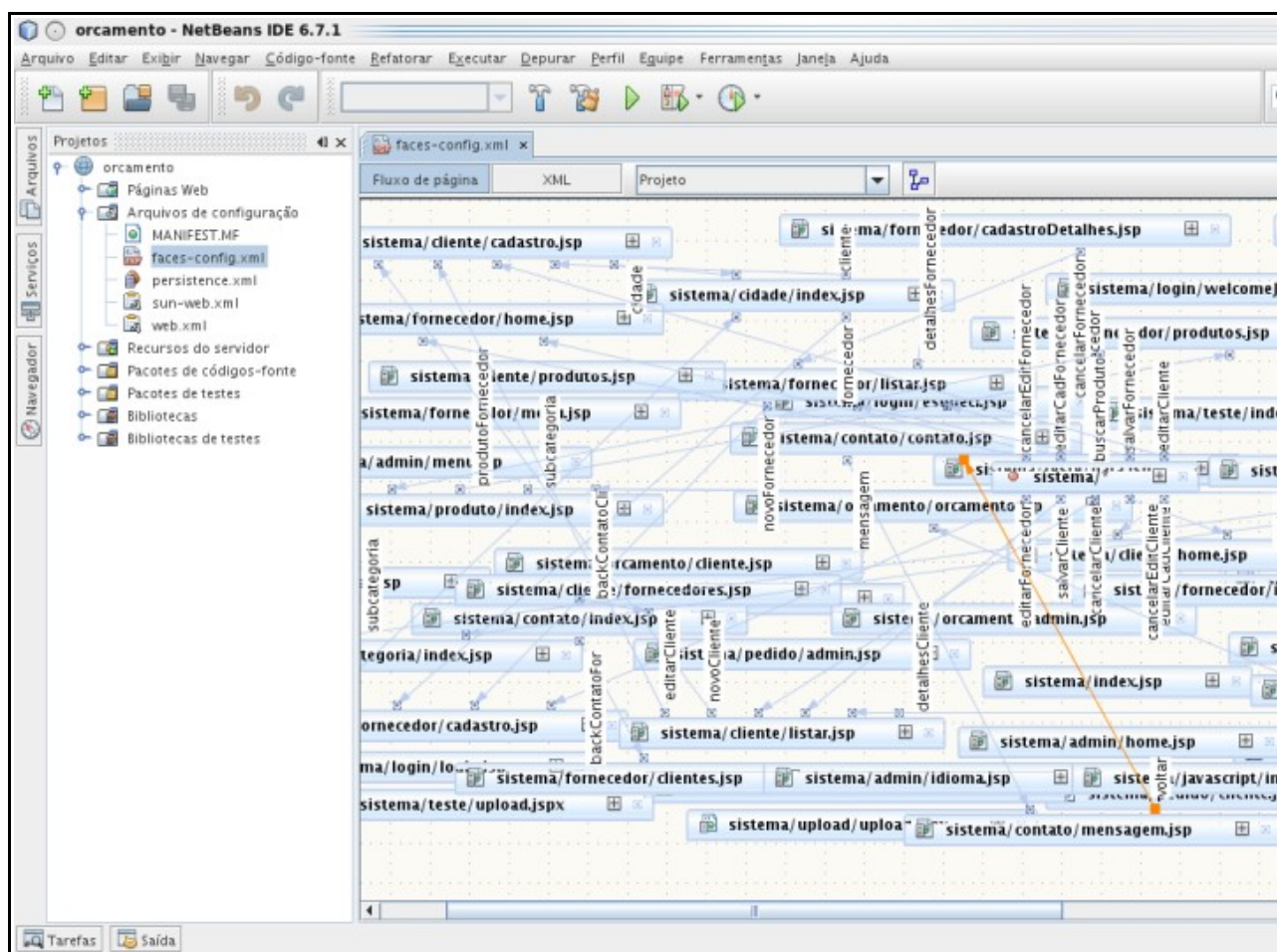


Figura 35 - Arquivos de Configuração Web no Netbeans

A última parte da aplicação é a principal, é onde estão localizados os códigos fontes do sistema, todos eles separados por pacote, como por exemplo, as classes de modelos estão dentro de um pacote chamado *beans*, a camada de persistência está dentro do pacote *persistencia*, as regras de negócios estão localizadas dentro do pacote *logicbusiness*, as classes controladoras estão dentro do pacote *managedbeans*, as mensagens de internacionalização usada pelo sistema ficam no pacote *messages*, enfim, esse são os principais pacotes e classes que se constam na aplicação.

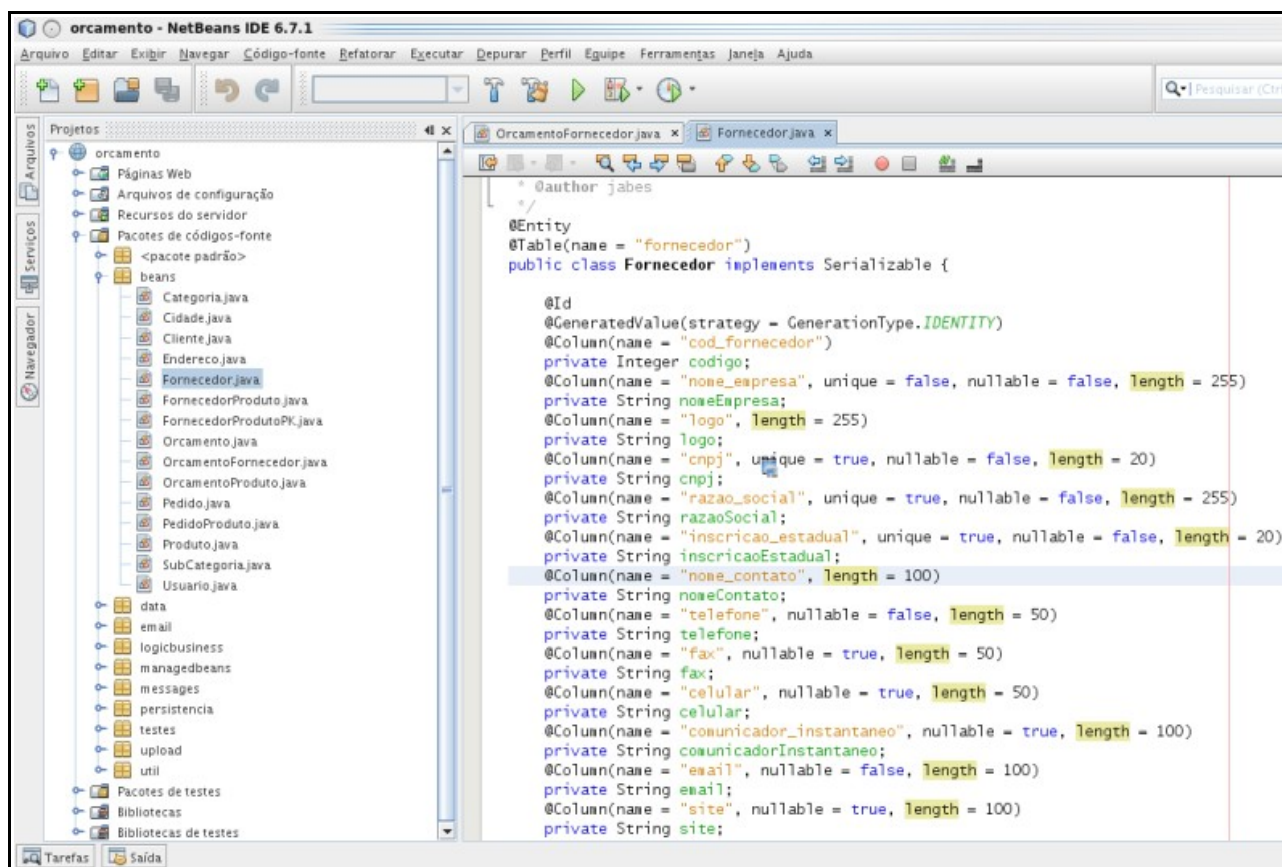


Figura 36 - Código Fonte com os Pacotes no Netbeans



O objetivo do sistema desenvolvido é permitir o controle eletrônico de orçamentos, por ser uma aplicação web, possui um recurso de autentificação de usuário, ou seja, um login para gerenciar as três áreas de gerenciamento, sendo: usuário administrador, cliente e fornecedor.

O administrador tem o privilégio de controlar a maioria das funcionalidades do sistema, como por exemplo: gerenciar as categorias, subcategorias, produtos, orçamentos, pedidos, e também cadastrar os usuários, que são do tipo cliente e fornecedor.

O fornecedor possui a permissão para editar seu perfil, escolher qual será o idioma padrão do sistema, consultar os clientes que estão cadastrados no sistema, gerenciar as categorias, subcategorias, consultar os seus orçamentos, se os orçamentos estiverem na situação de publicado, o fornecedor poderá incluir os valores nos mesmos, também pode consultar os pedidos de compra que lhe foram solicitados, e por fim a funcionalidade mais importante de sua área, que é gerenciar os produtos que os clientes irão orçar.

O cliente é o usuário que mais faz as movimentações do sistema, ele cria os orçamentos, incluindo quantos produtos e fornecedores forem necessários, após finalizar um orçamento, o cliente poderá emitir um ou mais pedido de compra para um ou mais fornecedores de cada vez, também ele possui outras funcionalidades, como: editar seu perfil, escolher qual será o idioma padrão do sistema, poderá realizar consultas dos produtos e fornecedores que estão cadastrados no sistema, etc.

As figuras que mostram as principais funcionalidades que o sistema possuem, estão no final deste documento, em anexo.

## 5 CONCLUSÃO

O trabalho proposto apresentou-se através de um software desenvolvido para uso na Internet ou intranet, que permite que qualquer organização, independente do tamanho ou seguimento, possa utilizá-lo eficientemente. Que nos processos de cotações e orçamentos estejam totalmente sistematizados, realizem suas atividades com maior eficácia, proporcionando a organização, maior agilidade e mais enfoque em obter mais presença no mercado e garantindo mais competitividade.

Com o uso do software, observou-se que os orçamentos são gerenciados e controlados de maneira coesa, agregando mais qualidade no serviço, obtendo mais agilidade e segurança para os clientes e os fornecedores. Através destas características importantes, a organização evita a entropia.

Tratando-se de um sistema desenvolvido para web, percebeu-se uma grande vantagem no quesito mobilidade, tal que, a aplicação poderá oferecer todas as informações sobre os históricos de orçamentos e pedidos, assim como o acompanhamento em trânsito de pessoas, que facilita a integração de outros gerentes da organização.

Portanto, o sistema responde de forma rápida o gerenciamento dos orçamentos da organização. Compartilha qualquer informação de forma responsável e segura. Trata com imparcialidade qualquer preço praticado pelos fornecedores que estejam participando daquela cotação. E finaliza o pedido com os fornecedores, enviando rapidamente os pedidos aos fornecedores correspondentes, contendo os produtos e as quantidades solicitadas.

Para desenvolvimentos futuros, pretende-se implantar o uso de EJB, que é um dos principais componentes da plataforma JEE, através do EJB, o sistema terá melhor desempenho para atender inúmeros usuários que acessam simultaneamente o sistema.



## REFERÊNCIAS BIBLIOGRÁFICAS

BEZERRA, Eduardo. **Princípios de Análise e Projeto de Sistemas com UML**. Rio de Janeiro: Editora Campus, 2002.

GEARY, David; HORSTMANN, Cay. **Core JavaServer Faces Fundamentos**. 2. ed. Tradução de Patrícia Azeredo e Fausto Magalhães, Rio de Janeiro: Editora Alta Books, 2007.

GONÇALVES, Edson. **Desenvolvendo Aplicações Web com JSP, SERVLETS, JAVASERVER FACES, HIBERNATE, EJB 3 PERSISTENCE E AJAX**. Rio de Janeiro: Editora Ciência Moderna Ltda, 2007.

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

JÚNIOR, Peter Jandl. **Introdução ao Java**. 1999. 196 p. Núcleo de Educação a Distância – Universidade São Francisco, São Francisco, 1999.

OLIVEIRA, Alcione Paiva; MACIEL, Vinícius Valente. **Java na Prática**. 2002. 432 p. Departamento de Informática – Universidade Federal de Viçosa, Minas Gerais, 2002.

STADZISZ, Paulo César. **Projeto de Software usando a UML**. 2002. 69 p. Departamento Acadêmico de Informática – Centro Federal de Educação Tecnológica do Paraná, Paraná, 2002.

SILVA, Simone Cardoso. **Vantagens e Desvantagens de se usar o Framework Hibernate**. 2009. Ciência da Computação – Fundação Educacional do Município de Assis, Assis, São Paulo, 2009.

## REFERÊNCIAS ELETRÔNICAS

**As Principais Características do MySQL.** Disponível em: <<http://dev.mysql.com/doc/refman/4.1/pt/features.html>>. Acessado em: 18 out, 2009.

**Cascading Style Sheets - Home Page.** Disponível em: <<http://www.w3.org/Style/CSS/>>. Acessado em: 11 abr, 2009.

**Diagrama de Classes.** Disponível em: <<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/diagramas/classes/classes1.htm>>. Acessado em: 14 abr, 2009.

**Entity Beans e Chaves Primárias Simples e Composta.** Disponível em: <<http://rfiume.blogspot.com/2007/04/entity-beans-e-chaves-primrias-simples.html>>. Acessado em: 18 jul, 2009.

FURUTANI, Roberto. **Como criar relatórios Java para Web com JasperReports e iReport.** Disponível em <[http://www.furutani.eti.br/tutoriais/Mini-Tutorial\\_Relatorios\\_Java\\_JasperReports\\_e\\_iReport.pdf](http://www.furutani.eti.br/tutoriais/Mini-Tutorial_Relatorios_Java_JasperReports_e_iReport.pdf)>. Acesso 02 abr. 2009.

MENDONÇA, Alexandre; PETROSKI, Bruno; SANTIN, Dyego; CARVALHO, Felipe.  
**Análise e Projeto de Sistemas Computadorizados.** Disponível em:  
<<http://www.inf.ufsc.br/~fgcarval/ine5319/index.html>>. Acessado em: 14 abr, 2009.

**Model - View - Controller.** Disponível em:  
<<http://java.sun.com/blueprints/patterns/MVC.html>>. Acessado em: 18 jul, 2009.

**NetBeans IDE - Connecting Developers.** Disponível em:  
<<http://www.netbeans.org/features/>>. Acessado em: 11 abr, 2009.

**Tutorial – ICEfaces.** Disponível em:  
<<http://www.icefaces.org/main/resources/tutorials.iface>>. Acessado em: 09 abr, 2009.

**Unified Modeling Language (UML), version 2.2.** Disponível em:  
<<http://www.omg.org/technology/documents/formal/uml.htm>>. Acessado em:  
07 jul, 2009.

## ANEXO – INTERFACE DO SISTEMA

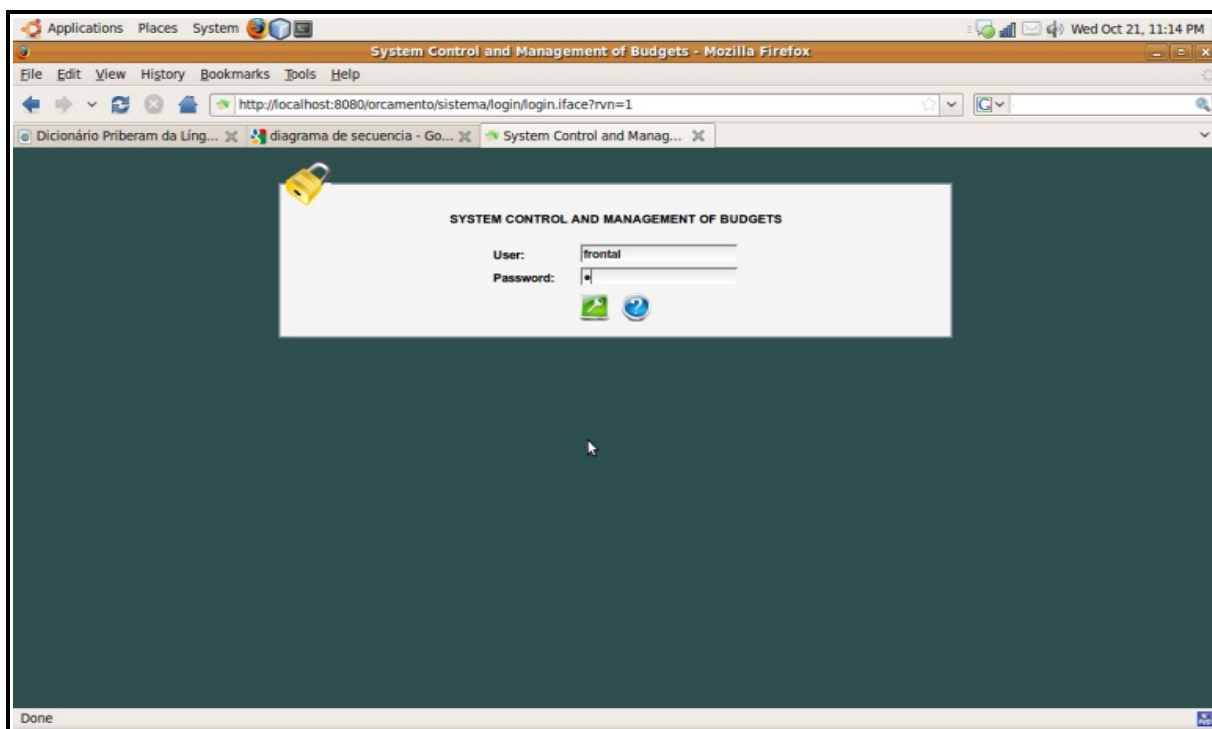


Figura 37 - Tela de Login

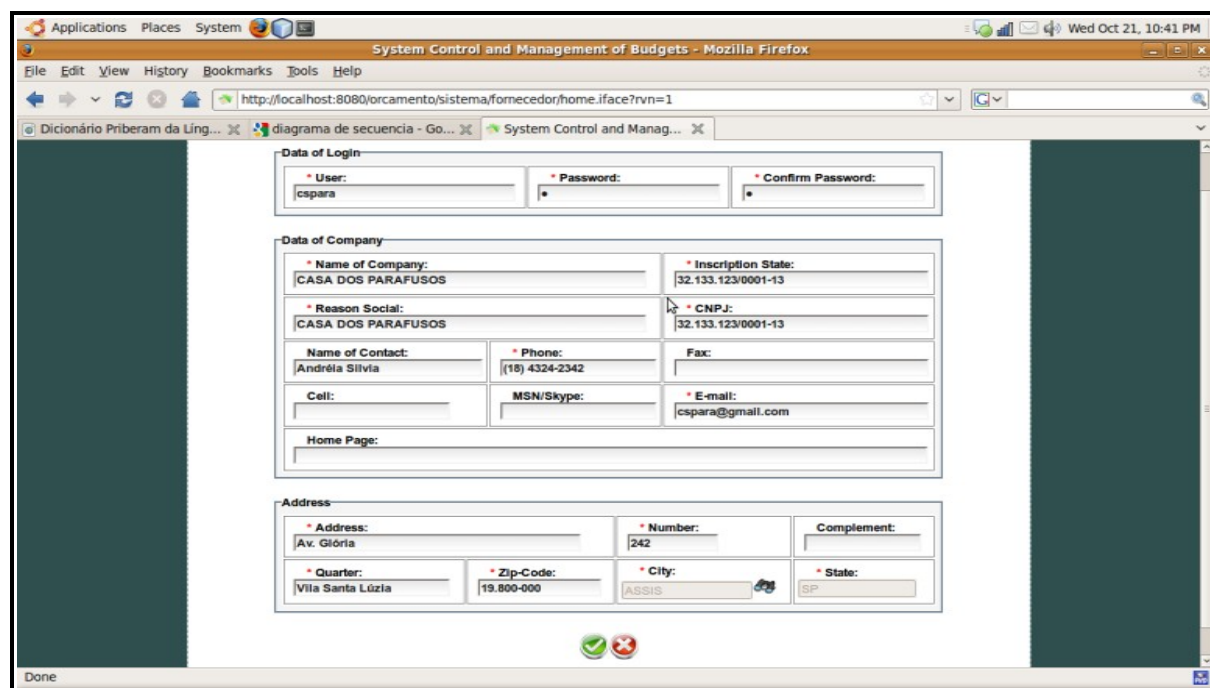


Figura 38 - Tela de Cadastro de Fornecedores

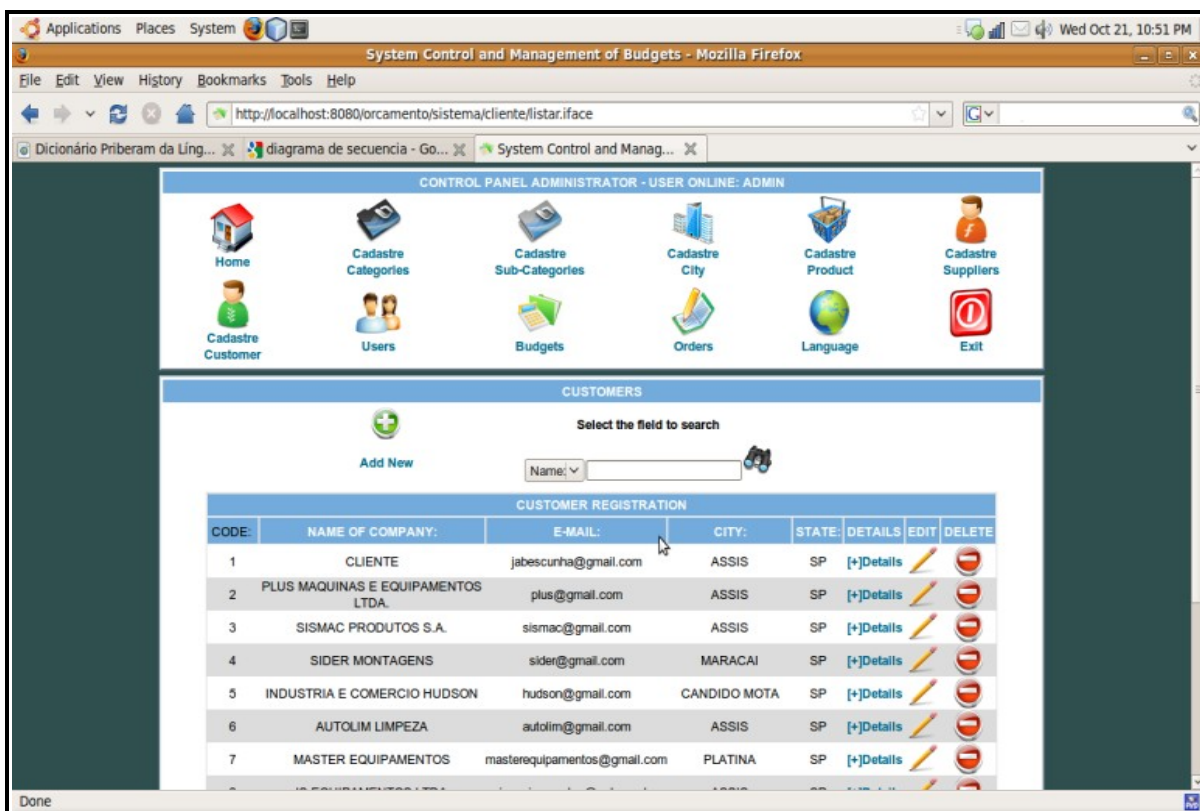


Figura 39 - Tela de Clientes Cadastrados

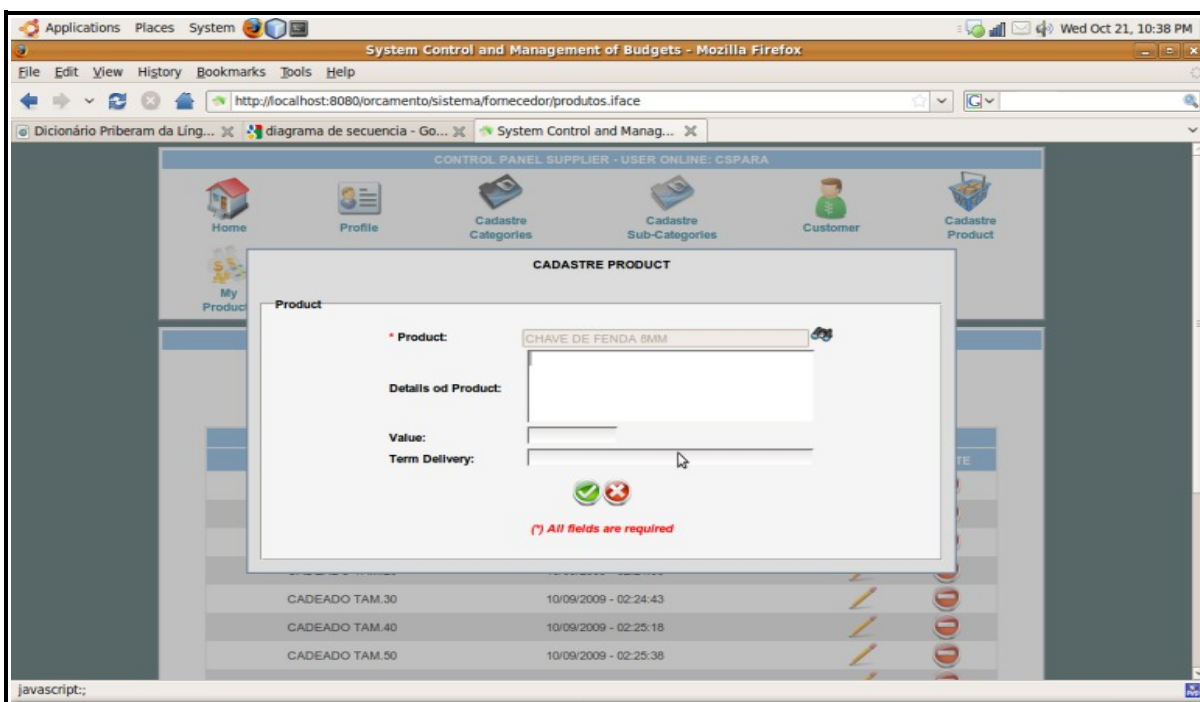


Figura 40 - Tela de Cadastro de Produtos

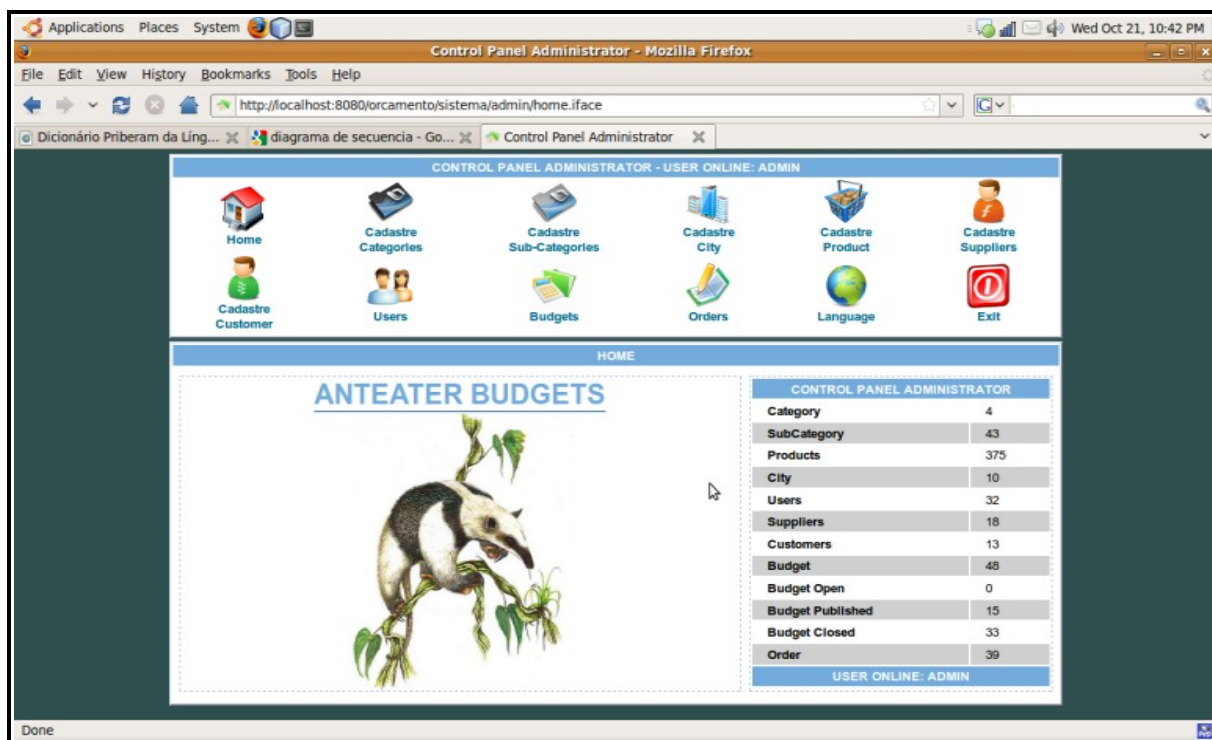


Figura 41 - Tela Inicial do Administrador

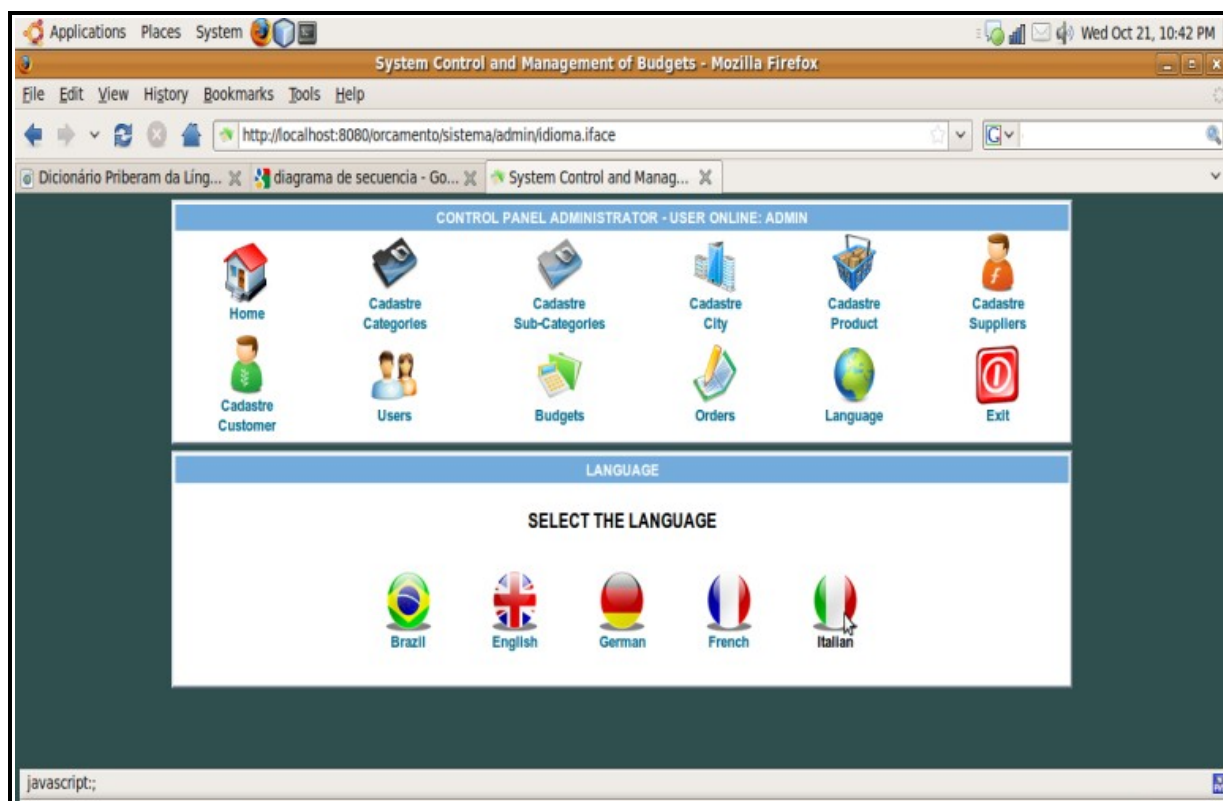


Figura 42 - Tela para Escolha de Idiomas

Budget ID:49 - Desc.: Clarineta - Situation: FECHADO

Hide || Emit Order || Home || Exit

\* Code: 49

\* Description: Clarineta

\* Date Begin: 14/10/2009

\* Date Close: 15/10/2009

October Open popup calendar

PRODUCTS REGISTRATION

NAME:	UNIT	QUANTITY	OBSERVATION:
AHA 17 CH SIB NIQ YCL 250	un.	15	

1 Product(s) found(s), display 1 Product(s), From: 1 To: 1. Page 1 / 1.

Suppliers

Display

Figura 43 - Tela Orçamento I

Budget ID:49 - Desc.: Clarineta - Situation: FECHADO

Display || Emit Order || Home || Exit

Products

Suppliers

SUPPLIER REGISTRATION

CODE	SUPPLIER	PRODUCT	UNIT	QUANTITY	VALUE UNIT	IPI (%)	VALUE TOTAL	TERM DELIVERY	OBS
630	SO SOM INSTRUMENTOS MÚSICAIS	CLARINETA YAMAHA 17 CH SIB NIQ YCL 250	un.	15	R\$ 312.50	5.0	R\$ 4,921.88	8 dias	nada
631	CORDIAL INSTRUMENTOS MÚSICAIS	CLARINETA YAMAHA 17 CH SIB NIQ YCL 250	un.	15	R\$ 342.00	0.0	R\$ 5,130.00		
632	PLAYMUSIC INSTRUMENTOS LTDA.	CLARINETA YAMAHA 17 CH SIB NIQ YCL 250	un.	15	R\$ 300.00	5.0	R\$ 4,725.00	imediate	nd.

3 Supplier(s) found(s), display 3 Supplier(s), From: 1 To: 3. Page 1 / 1.

VALUE TOTAL OF SUPPLIERS

SUPPLIER	VALUE TOTAL
SO SOM INSTRUMENTOS MÚSICAIS	R\$ 4,921.88
CORDIAL INSTRUMENTOS MÚSICAIS	R\$ 5,130.00
PLAYMUSIC INSTRUMENTOS LTDA.	R\$ 4,725.00

Figura 44 - Tela Orçamento II



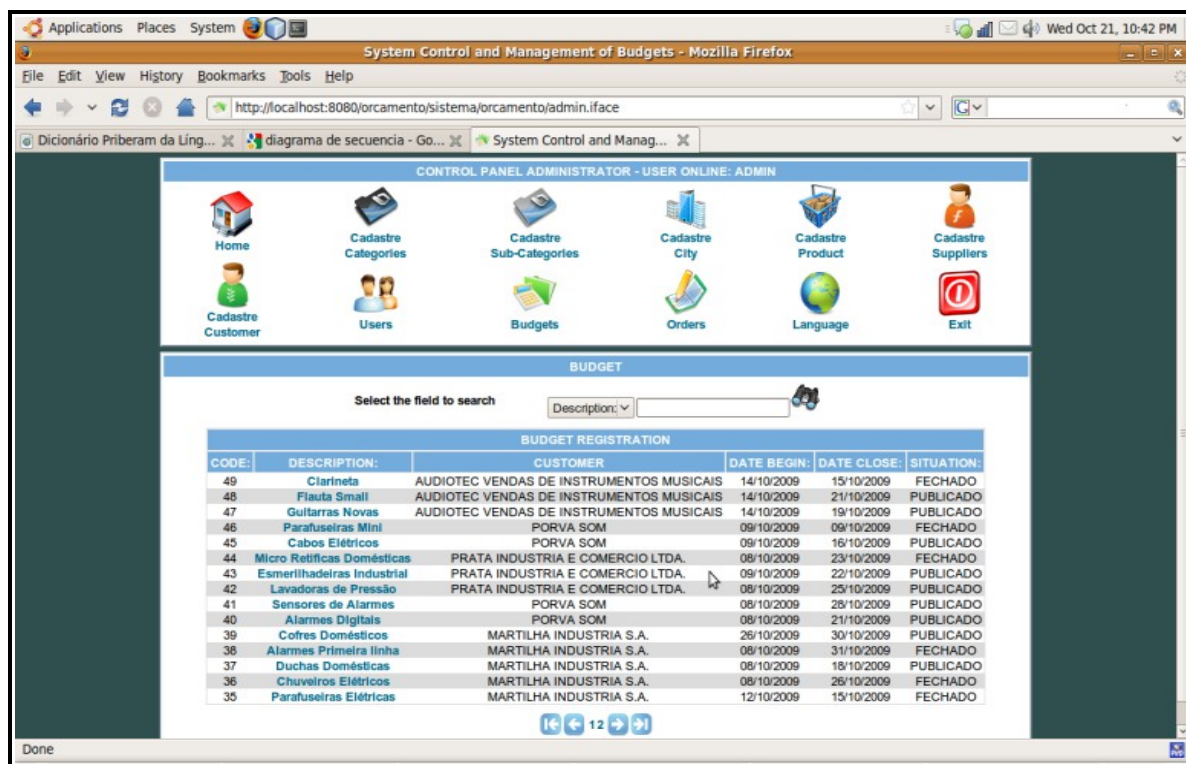


Figura 45 - Tela Orçamentos Cadastrados

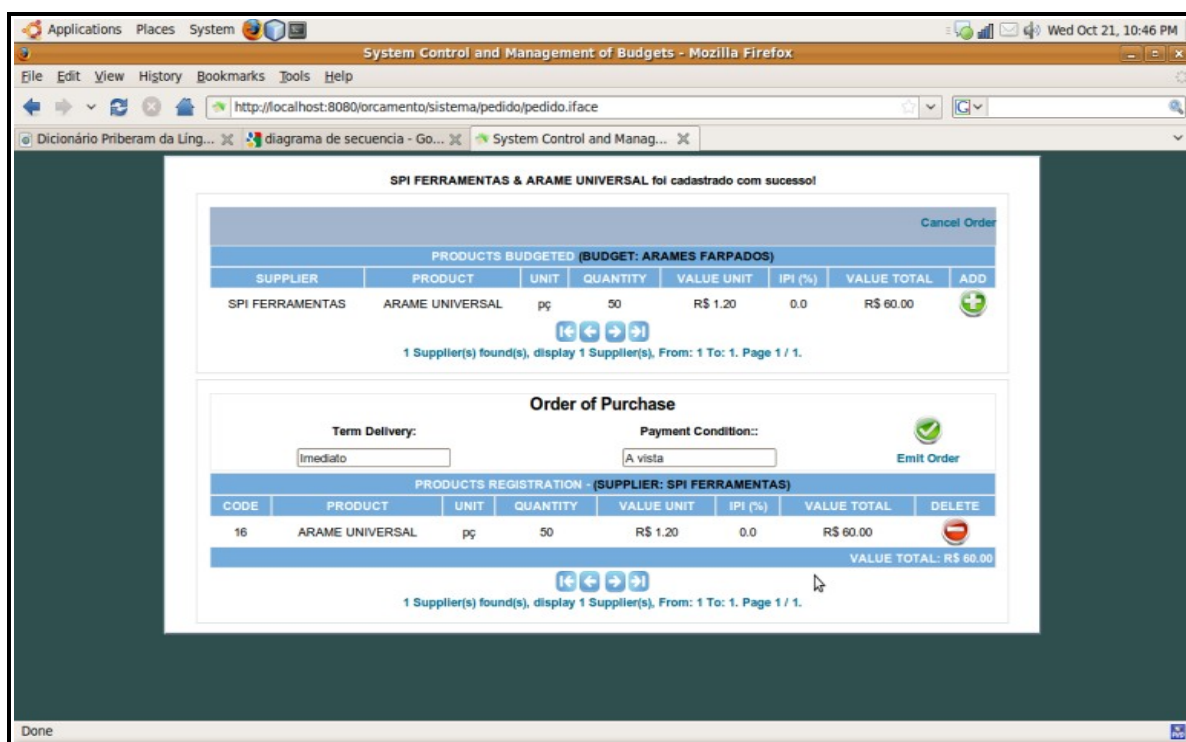


Figura 46 - Tela Emitir Pedido de Compra

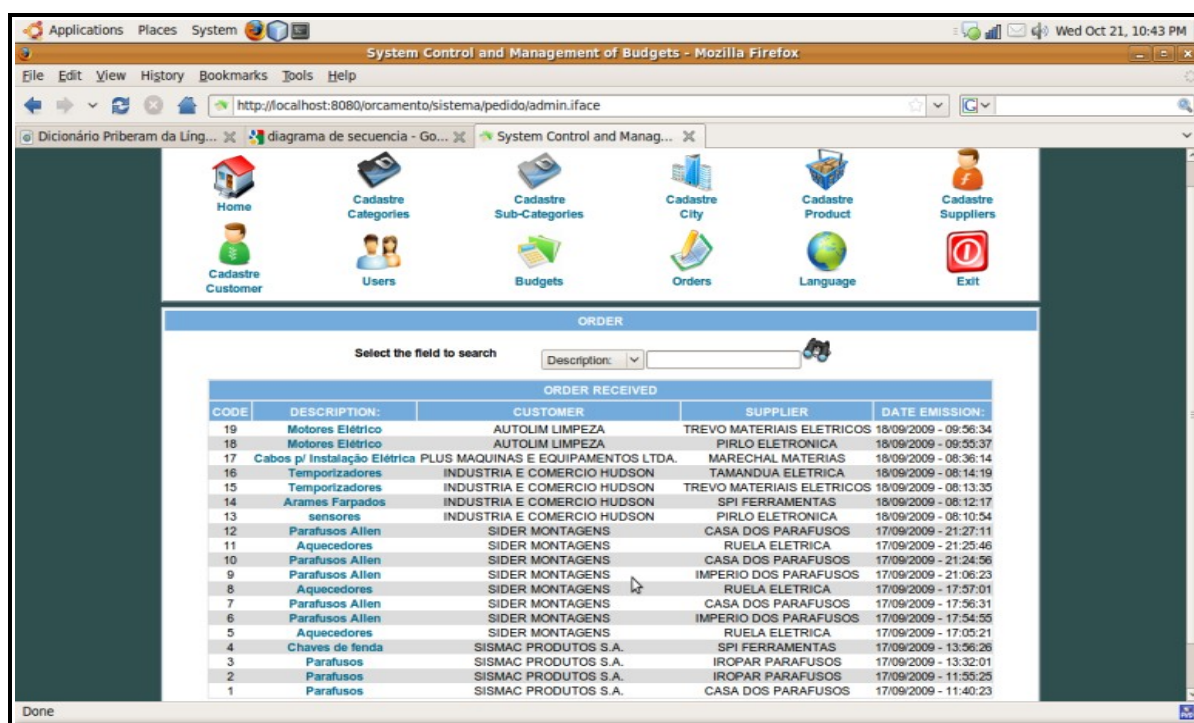


Figura 47 - Tela Pedidos Cadastrados

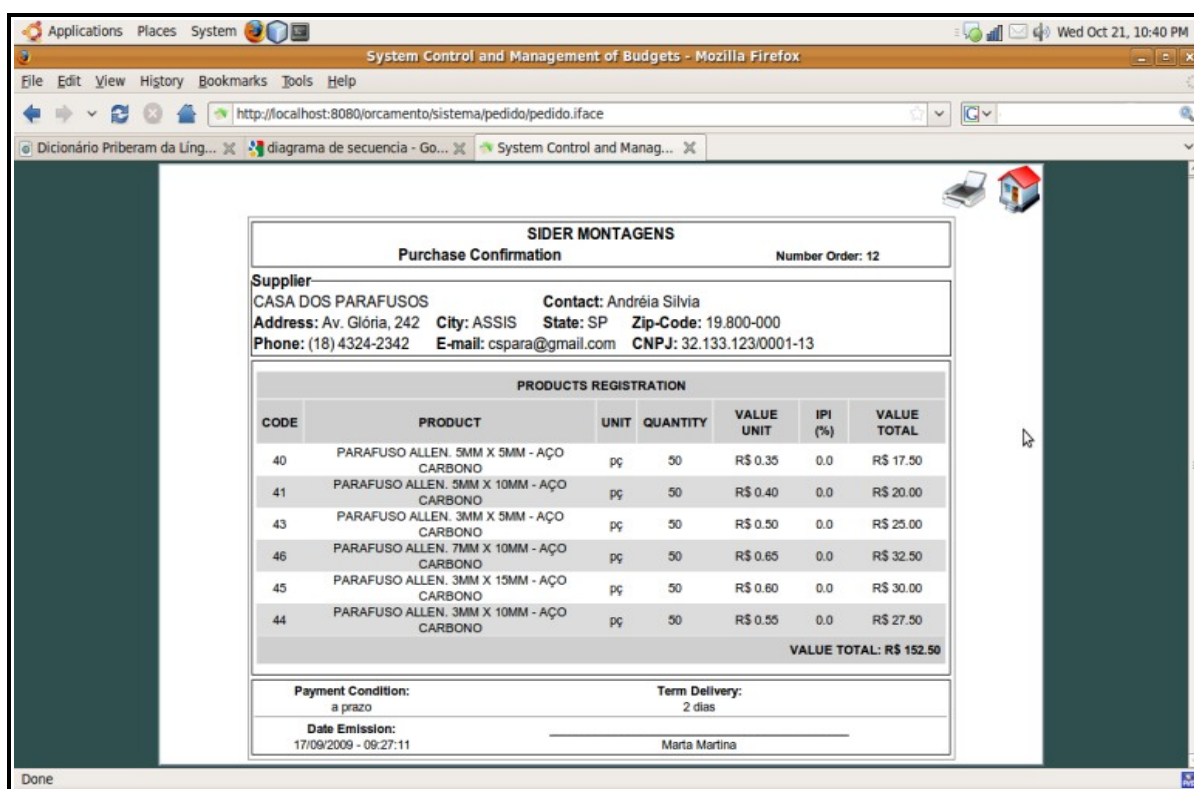


Figura 48 - Tela Pedido de Compra