



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

**MOACIR MARDEGAN**

## **CONTROLE DE ORÇAMENTO PARA BICICLETARIA**

**Assis  
2010**



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

**MOACIR MARDEGAN**

## **CONTROLE DE ORÇAMENTO PARA BICICLETARIA**

Trabalho final do curso de  
Processamento de dados, apresentado ao  
Instituto Municipal de Ensino Superior de  
Assis, IMESA, como requisito do Curso de  
graduação.

**Orientador:** Alex Sandro Romeo de Souza Poletto

**Orientado:** Moacir Mardegan

**Assis  
2010**

## FICHA CATALOGRÁFICA

MARDEGAN, Moacir

Controle de Orçamento para Bicletaria / Moacir Mardegan.  
Fundação Educacional do município de Assis - Assís, 2010 .  
45 pg.

Orientador: Alex Sandro Romeo de Souza Poletto .  
Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis

1.Tecnologia em Processamento de Dados. 2.Programação

CDD:001.61  
Biblioteca da Fema

# **CONTROLE DE ORÇAMENTO PARA BICICLETARIA**

**Moacir Mardegan**

Trabalho de Conclusão de Curso  
Apresentado ao Instituto Municipal  
De Ensino Superior de Assis como  
requerida do Curso de Graduação  
analisada pela seguinte comissão  
examinadora.

**Orientador:** Alex Sandro Romeo de Souza Poletto

**Assis  
2010**

## DEDICATÓRIA

Dedico este trabalho a minha querida esposa, Josiane, ao meu Filho Gabriel e a pequena Giovanna, que tiveram paciência e compreensão durante a elaboração deste presente trabalho.

## **AGRADECIMENTOS**

Aos meus professores que contribuíram para o meu crescimento, acadêmico e pessoal e vale aqui um agradecimento especial ao meu orientador Prof. Dr. Alex Poletto, pela paciência e as valiosas dicas dadas no decorrer do trabalho.

Aos amigos, Alexandre Latufe e Daniela Antonio Camolesi que participaram desse crescimento, durante esses três anos aqui na Fema, e também no meu estágio. Especialmente ao Ivan Camolese e Tiago Almeida Bungenstab (Roseta), amigos de trabalho, que me deram belas dicas técnicas.

Aos meus queridos pais, Celso e Adelaide, pela ajuda e grande apoio em varias ocasiões.

Ao meu querido irmão David pelas longas conversas em torno dessa difícil carreira. Ao meu grande irmão, Elói pela força de vontade e, lição de vida que demonstra, em todas as coisas que realiza. Realmente aprendo muito com ele.

## RESUMO

A finalidade do sistema será a automatização dos controles de orçamentos de produtos e serviços. O software será desenvolvido para Desktop, serão usados alguns Frameworks como o SWT, Hibernate e banco HSQLDB; Espera-se com tal software contribuir para um trabalho dinâmico, seguro e rápido para apresentar orçamentos aos clientes da bicicletaria.

**Palavras - chave:** Orçamento. Java. Frameworks. Hibernate. SWT. HSQLDB.

## **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 Desktop application, developed in Java, SWT, with HSQLDB database, using some frameworks like and hibernates. 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:** budgets.Java. Frameworks. Hibernate. SWT. HSQLDB.



## LISTA DE ILUSTRAÇÕES

Figura 1. Caso de Uso: Realizar Login.....	22
Figura 2. Caso de Uso: Manter Fornecedor .....	23
Figura 3. Caso de Uso: Manter Clientes .....	24
Figura 4. Caso de Uso: Manter Produtos .....	25
Figura 5. Caso de Uso: Manter Usuário .....	26
Figura 6. Caso de Uso: Efetuar Orçamento .....	27
Figura 7. Diagrama Geral de Caso de Uso .....	28
Figura 8. Diagrama de Classes de Orçamentos.....	29
Figura 9. Diagrama de Atividades: Orçamento.....	30
Figura 10. Diagrama de Sequência: Efetuar Login.....	31
Figura 11. Diagrama de Sequência: Manter Cliente.....	32
Figura 12. Diagrama de Sequência: Manter Peças.....	33
Figura 13. Diagrama de Sequência: Manter Usuários.....	34
Figura 14. Diagrama de Sequência: Movimentação de Orçamentos .....	35
Figura 15. Diagrama ER Venda e Orçamento.....	36
Figura 16. Diagrama WBS.....	37
Figura 17. Sequenciamento das Atividades Definidas .....	38
Figura 18. Cronograma das Atividades Definidas .....	40

## LISTA DE TABELAS

Tabela 1 - Orçamento do Projeto .....	42
---------------------------------------	----

## **LISTA DE ABREVIATURAS E SIGLAS**

UML	Unified Modeling Language
MVC	Model View Controller
IDE	Integrated Development Environment
JUDE	Java and UML Developers' Environment
WBS	Work Breakdown Structure
UC	Use Case

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
1.1	OBJETIVO	14
1.2	JUSTIFICATIVA E MOTIVAÇÃO	14
1.3	PÚBLICO ALVO	15
1.4	ESTRUTURA DO TRABALHO	15
<b>2</b>	<b>METODOLOGIA DA ANÁLISE DE ORIENTAÇÃO A OBJETOS</b>	<b>15</b>
2.1	CLASSES	16
2.2	OBJETO	16
2.3	ABSTRAÇÃO	17
2.4	ATRIBUTOS	17
2.5	MÉTODOS	17
2.6	ENCAPSULAMENTO	17
2.7	CONSTRUTOR	17
2.8	HERANÇA	18
2.9	POLIMORFISMO	18
<b>3</b>	<b>AMBIENTE DE DESENVOLVIMENTO</b>	<b>18</b>
3.1	FERRAMENTAS UTILIZADAS	18
3.1.1	Java	18
3.1.2	Hibernate	19
3.1.3	Eclipse-Ganymede IDE (Integrated Development Environment)	19
3.1.4	Hsqldb	19
3.1.5	JasperReport	19
3.1.6	Ferramentas para Análise	19
<b>4</b>	<b>PROJETO E ANÁLISE DO SISTEMA</b>	<b>20</b>
4.1	FORMA DE LEVANTAMENTO DE REQUISITOS	20
4.2	Análise dos Requisitos	20
4.3	RESTRIÇÕES DO SISTEMA	20
4.4	ESPECIFICAÇÃO DOS REQUISITOS	20
4.5	ESCOPO	21
4.5.1	Lista de Eventos	21
4.6	ESPECIFICAÇÃO DOS CASOS DE USOS	22
4.6.1	Efetuar Login	22
4.6.2	Manter Fornecedor	23
4.6.3	Manter Clientes	24

4.6.4 Manter Peças.....	25
4.6.5 Manter Usuário .....	26
4.7 EFETUAR ORÇAMENTO .....	27
4.8 DIAGRAMA GERAL DE CASO DE USO BICICLETARIA .....	28
4.9 DIAGRAMA DE CLASSE DE ORÇAMENTO.....	29
4.10 DIAGRAMA DE ATIVIDADES ORÇAMENTO .....	30
4.11 DIAGRAMA DE SEQUÊNCIA EFETUAR LOGIN .....	31
4.12 DIAGRAMA DE SEQUENCIA MANTER CLIENTES .....	32
4.13 DIAGRAMA DE SEQUÊNCIA MANTER PEÇAS.....	33
4.14 DIAGRAMA DE SEQUÊNCIA MANTER USUÁRIO .....	34
4.15 DIAGRAMA DE SEQUÊNCIA MOVIMENTAÇÃO DE ORÇAMENTO .....	35
4.16 DIAGRAMA ER DE ORÇAMENTOS .....	36
4.17 PLANEJAMENTO DO PROJETO.....	37
4.17.1 WBS – Work Breakdown Structure .....	37
4.17.1.1 Define o agrupamento e disposição dos elementos do projeto. ....	37
4.18 SEQUÊNCIA DAS ATIVIDADES DEFINIDAS .....	38
4.19 DEFINIÇÃO DAS ATIVIDADES.....	39
4.19.1 Levantamento dos requisitos;.....	39
TABELA DAS ATIVIDADES DEFINIDAS .....	40
<b>5 RECURSOS NECESSÁRIOS PARA O PROJETO .....</b>	<b>40</b>
5.1.1 Resumo da Estimativa de Custos do Orçamento do Projeto .....	42
5.2 Algumas Telas do Sistema. ....	43
5.2.1 Menu Principal .....	43
5.2.2 Cadastro de Produto .....	43
5.2.3 Mensagem de Cadastro de Produto .....	45
5.2.4 Pesquisa de Produto.....	46
5.2.5 Abertura de Orçamento .....	49
5.2.6 Banco de Dados HSQLDB.....	50

O sistema será desenvolvido para a empresa do ramo de bicicletas, tal estabelecimento leva o nome de Márcio Bike Ltda. O motor desse projeto foi o fato de que em doze anos (12) nesse mercado, o proprietário nunca se preocupou em instalar um sistema, para facilitar os trabalhos de sua oficina de bicicletas.

Hoje todo o seu trabalho, continua sendo com o auxílio de cadernetinhas; E o grande problema com elas é que se de alguma forma os dados da cadernetinha for depreciado, o dono do estabelecimento, amargará com o prejuízo.

No momento o proprietário da bicicletaria tem interesse em um módulo de sistema que efetue orçamentos, pois o mesmo hoje é feito de maneira bastante precária, e os dados acabam por se perder de alguma maneira, e acaba ficando bastante desgastante para o dia de trabalho.

## 1.1 OBJETIVO

A palavra que define o objetivo final desse trabalho é agilidade no andamento do trabalho a ser realizado pelo usuário do sistema.

Ao introduzir este sistema automatizado de Orçamentos, será criado um banco de dados (BD) de clientes, o que já seria de grande valia para as atividades desenvolvidas pela empresa. Assim sendo o sistema poderá melhorar o acompanhamento de todas as solicitações de um determinado orçamento. Este poderá ser um orçamento de peças, serviços, ou ainda um modelo mais elaborado de uma determinada bicicleta. Devido aos tipos de atividades relacionados a essa empresa, a parte cadastral do sistema será detalhada, haja vista que é de fundamental importância ter em mãos um número pré-definido de itens cadastrais de dados dos clientes e das peças e serviços também; Ao informatizar esta empresa, o grau de qualidade dos orçamentos, será elevado, o que por sua vez, tornará a empresa mais competitiva e mais próxima de seus clientes.

## 1.2 JUSTIFICATIVA E MOTIVAÇÃO

Hoje em dia as pessoas estão sempre com pressa e, não ficam nem um pouco felizes quando um simples orçamento começa a demorar, para ser entregue pelo funcionário da empresa, Com o sistema em funcionamento ficará muito mais fácil,

tirar qualquer orçamento que um cliente venha a precisar, este pode ser a cotação de materiais e serviços, ou mesmo a montagem de uma bicicleta inteira.

### 1.3 PÚBLICO ALVO

Este sistema poderá ser utilizado por todas as empresas que ainda não adotaram a informatização nos seus negócios, e operam com um grande volume de dados e que gastam muito tempo para elaborar, consultar ou localizar orçamentos; Mais precisamente, com o sistema fica mais fácil localizar Valor de peças, Serviços e valor de diversos modelos de bicicletas.

### 1.4 ESTRUTURA DO TRABALHO

O presente documento foi idealizado contendo cinco capítulos, sendo que o capítulo um mostra o assunto a serem tratados.

Os objetivos do sistema, assim como justificativa e motivação para desenvolver o sistema e, qual será o público alvo do sistema.

O capítulo dois trata dos conceitos da análise orientada a objetos, mostrando as ferramentas e tecnologias que serão usadas.

O capítulo três trata do levantamento de requisitos, e dos Diagramas: use case, Classe, Atividade e Seqüência. Por fim o planejamento do projeto, que contém a definição e o cronograma das atividades definidas, a estimativa de custos, e orçamento do Projeto.

O capítulo quatro aborda como o sistema, será codificado, e como o projeto foi estruturado.

Concluindo, com o capítulo cinco descreveremos as considerações finais, resultados do trabalho e os objetivos que serão alcançados.

## **2 METODOLOGIA DA ANÁLISE DE ORIENTAÇÃO A OBJETOS**

A orientação a objetos não é um conceito novo, já que era utilizado em 1967 pela linguagem Simuda 67, que foi criada por Johan Dahl e Kristen Nygaard, de quando o conceito de orientação a objetos foi criado, somente agora, seu uso vem ganhando terreno; Dentre as linguagens que utilizam o conceito de orientação a objetos estão

algumas tais como Java, C#, Ruby e por aí vai; Mas a linguagem que ficou mais famosa, foi sem dúvida o Java, porque além de ser orientada a objetos, ela é também multiplataforma, podendo funcionar em sistemas operacionais distintos, tais como Linux, Windows, Mac e mais algumas, prosseguindo com o conceito de análise orientada a objetos, eis que surgiu a Linguagem de Modelagem Unificada (UML), que utiliza elementos gráficos, que facilitam o entendimento e as iterações do projeto.

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

## 2.1 CLASSES

Classe é um protótipo de como deve ser um objeto (VARIÁVEL), os objetos são do tipo da classe; Analogicamente seria como a planta de uma casa. Pois com a planta da casa, podemos saber onde está cada coisa projetada, porém nada poderá ser usado, nada é real de verdade. Tomamos a descarga que estará projetada na planta e vamos tentar apertar a descarga no desenho esboçado, a fim de obter a sua funcionalidade real, que seria a descarga em si, logicamente a descarga da planta não funcionará. Para usarmos a descarga devemos construir a descarga e o seu sistema completo, para usufruirmos realmente da descarga. Então a planta da casa seria a classe, e a descarga da casa depois de pronta seria o objeto da classe banheiro.

## 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, podem ser criados 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. Relembrando o exemplo anterior, o objeto materializa a classe, faz com que a classe tome forma.



## 2.3 ABSTRAÇÃO

É uma maneira de focar nas partes principais do negócio, deixando de lado características que não são tão relevantes.

## 2.4 ATRIBUTOS

Comparando com a programação por procedimentos os atributos seriam as variáveis. Os atributos são as variáveis que vão guardar algum tipo de informação, porém ela reserva as características principais da classe, como os tipos de dados.

## 2.5 MÉTODOS

É o comportamento que tem a classes. Ou seja, são as funcionalidades, as operações que serão efetuadas pela classe, é o trabalho que ela realiza, alguns casos seriam as inserções de dados em banco de dados, fechamento de um módulo, alteração de um cadastro, o envio de mensagens de um módulo para outro e tudo que for necessário de comunicação e chamada de determinado modulo.

## 2.6 ENCAPSULAMENTO

O modo como um determinado atributo ou método de uma classe será acessado 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. Com o encapsulamento usamos um determinado objeto de outra classe, sem se importar como ele funciona, é como se fosse uma calculadora, onde não usamos suas funcionalidades sem saber como ela realmente funciona internamente.

## 2.7 CONSTRUTOR

Quando um objeto é instanciado, o construtor da classe é invocado para determinação do estado inicial do objeto. Quando nenhum construtor for definido no corpo da classe, o objeto é instanciado com seus atributos vazios. Caso um

construtor não seja definido no corpo da classe, o construtor da classe Object é invocado por default, pois todas as classes em java são derivadas de Object.

## 2.8 HERANÇA

É a parte que trata sobre a reutilização do código, atributos e métodos de outras classes. 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.

## 2.9 POLIMORFISMO

Polimorfismo é o princípio pelo qual duas ou mais classes derivadas de uma mesma superclasse podem invocar métodos que têm a mesma identificação (assinatura) mas comportamentos distintos, especializados para cada classe derivada, usando para tanto uma referência a um objeto do tipo da superclasse. A decisão sobre qual o método que deve ser selecionado, de acordo com o tipo da classe derivada, é tomada em tempo de execução, através do mecanismo de ligação tardia. No caso de polimorfismo, é necessário que os métodos tenham exatamente a mesma identificação, sendo utilizado o mecanismo de redefinição de métodos.

# 3 AMBIENTE DE DESENVOLVIMENTO

## 3.1 FERRAMENTAS UTILIZADAS

### 3.1.1 Java

Será usada no desenvolvimento do sistema a linguagem Java, tal linguagem foi introduzida no mercado pela Sun Microsystem. A linguagem é multiplataforma, ou seja, roda em Windows, Linux, Mac, Solaris e muito mais; Prosseguindo no raciocínio, Java também roda em celulares, geladeiras, relógios, satélites, dispositivos industriais e, é por esse motivo Java é tão cobiçada pelos aspirantes a programadores.

### **3.1.2 Hibernate**

Hibernate é um Framework de persistência escrito em Java que se posiciona entre o banco de dados e a aplicação Java, ele faz a comunicação entre a aplicação Java que é orientada a objetos e o banco de dados que é relacional.

### **3.1.3 Eclipse-Ganymede IDE (Integrated Development Environment)**

É um ambiente integrado para desenvolvimento de software, escrito em Java, é usado por muitos desenvolvedores que utilizam linguagem Java para desenvolver.

Esta ferramenta IDE podem ser adquiridas gratuitamente e auxiliar os programadores a escrever, debugar, compilar e executar o programa.

### **3.1.4 Hsqldb**

O Hsqldb é um banco de dados desenvolvido inteiramente em Java ele é extremamente pequeno e poderoso. Seu “jar” possui um tamanho de 500 kb; O Hsqldb funciona assim como o Java em diversas plataformas, e nele pode-se manipular as instruções de SQL, de relacional, como na maioria dos grandes bancos de dados.

### **3.1.5 JasperReport**

Ferramenta utilizada para a elaboração de relatórios do sistema, ou seja, é uma ferramenta extremamente importante, pois são os relatórios, quem apresentarão o que foi feito no sistema, é quando o usuário começa a ter as respostas do que ele necessita.

### **3.1.6 Ferramentas para Análise**

O JUDE é uma ferramenta desenvolvida em Java, que pode ser usada em qualquer sistema operacional, e 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. Após ter elaborada a modelagem, o JUDE possibilita que os diagramas possam ser exportados para tipos de arquivo de Imagem.

## 4 PROJETO E ANÁLISE DO SISTEMA

### 4.1 FORMA DE LEVANTAMENTO DE REQUISITOS

É extremamente importante a análise do sistema a ser desenvolvido, pois neste momento vamos entender como funciona o ramo do cliente, para que possamos resolver os problemas apresentados de forma simples e eficiente.

O primeiro passo será a numeração dos problemas enfrentados pela bicicletaria para efetuar o orçamento; Tal numeração será feita entrevistando, diretamente o proprietário do estabelecimento, descobrindo assim se existe a possibilidade de melhorar a maneira como o proprietário efetua seus orçamentos.

### 4.2 Análise dos Requisitos

- Cadastrar clientes;
- Cadastrar usuários;
- Cadastrar produtos;
- Controlar serviços;
- Controlar os orçamentos;
- Emitir relatórios dos orçamentos, produtos, usuários e serviços.

### 4.3 RESTRIÇÕES DO SISTEMA

O sistema será Desktop, e será necessário o seguinte recurso : Computador com processador Intel Pentium dual core 2.20 GHz, 2 Gb de memória RAM, HD de 320 GB, e sistema operacional de 32 bits.

### 4.4 ESPECIFICAÇÃO DOS REQUISITOS

O sistema a ser desenvolvido irá controlar e gerenciar os orçamentos da empresa, possibilitando que os clientes recebam seus orçamentos de maneira muito rápida.

## 4.5 ESCOPO

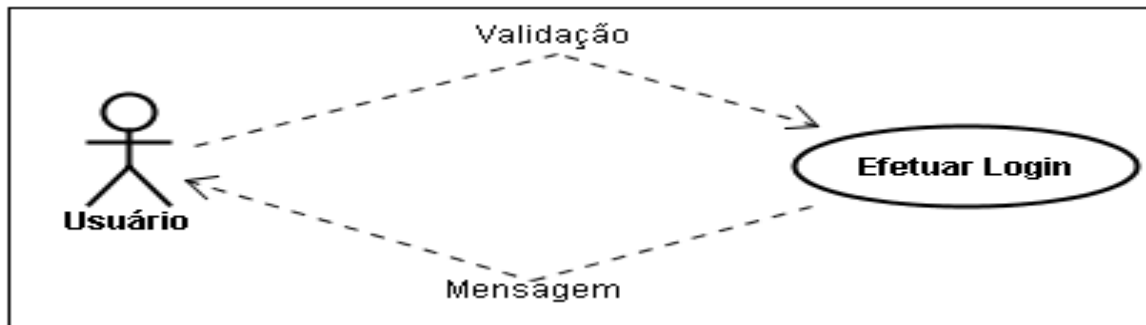
Os clientes, fornecedores, Itens (peças e serviços) novos usuários, formas de pagamento, devem ser cadastradas pelo Usuário (administrador). Ao final de um orçamento, o cliente tem a possibilidade de gerar um pedido. Diversos relatórios serão disponibilizados para o usuário do sistema. Por questão de segurança o sistema contará com um recurso de autenticação de usuários.

### 4.5.1 Lista de Eventos

- Manter Usuário
- Manter Cliente
- Manter Produto
- Manter Serviço
- Movimentar Orçamento
- Emitir Relatório de Produtos
- Emitir Relatório de Serviços
- Emitir Relatório de Orçamentos por Cliente

## 4.6 ESPECIFICAÇÃO DOS CASOS DE USOS

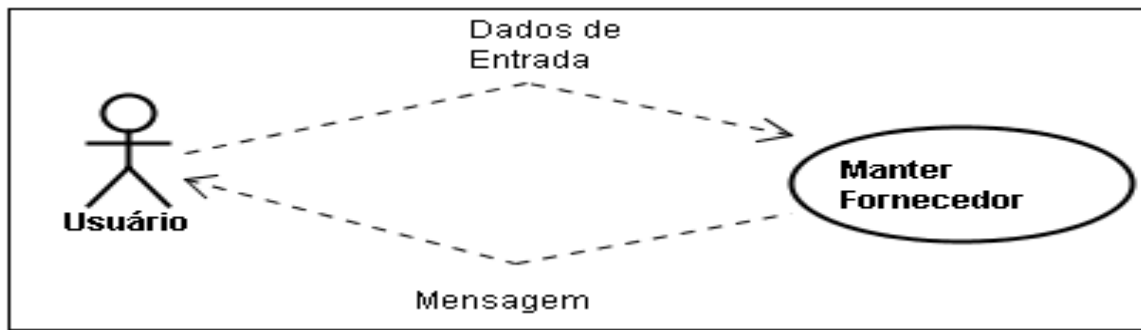
### 4.6.1 Efetuar Login



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

Nome do use case	Realizar login
Cenário Principal	O usuário fornece seu nome de usuário e senha. Depois de passar pela autenticação a área de controle e gerenciamento do sistema fica disponível para o usuário.
Cenário alternativo	Se for um novo usuário é necessário fazer o cadastro, e em seguida logar no sistema. Se o usuário ou senha estiver errado, será apresentada uma mensagem de erro de login. Após efetuar o cadastro o sistema fica disponível para o usuário.
Fluxos de Exceção	Se o usuário entrar com os dados incorretos, o sistema não abre e é necessário realizar o processo de autenticação novamente.

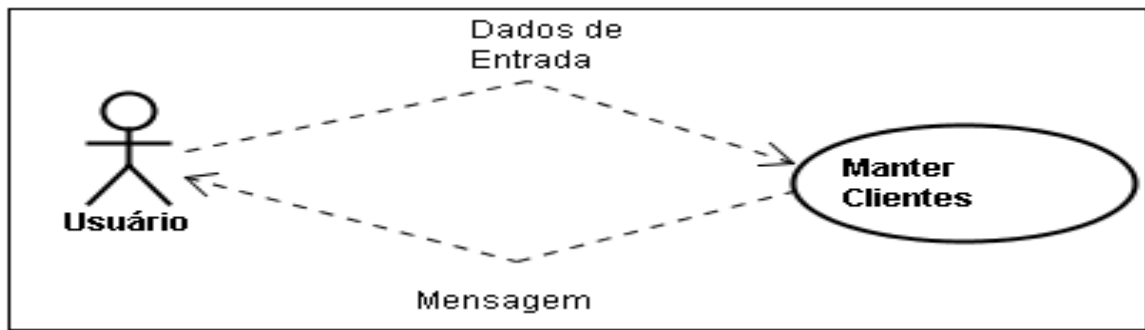
#### 4.6.2 Manter Fornecedor



**Figura 2. Caso de Uso: Manter Fornecedor**

Nome do use case	Manter Fornecedor
Cenário Principal	O usuário efetua uma pesquisa de nome, CNPJ ou IE do fornecedor no sistema. Com isso o usuário pode comprar de um determinado Fornecedor.
Cenário alternativo	Se o fornecedor, não for encontrado no sistema, significa que o fornecedor ainda não está cadastrado no sistema, então bastará cadastra-lo.
Fluxos de Exceção	Se o usuário efetuar uma pesquisa por nome de fornecedor, e encontrar dois nomes idênticos no sistema, então o sistema pedira para que seja feita uma pesquisa por CNPJ ou IE.

### 4.6.3 Manter Clientes

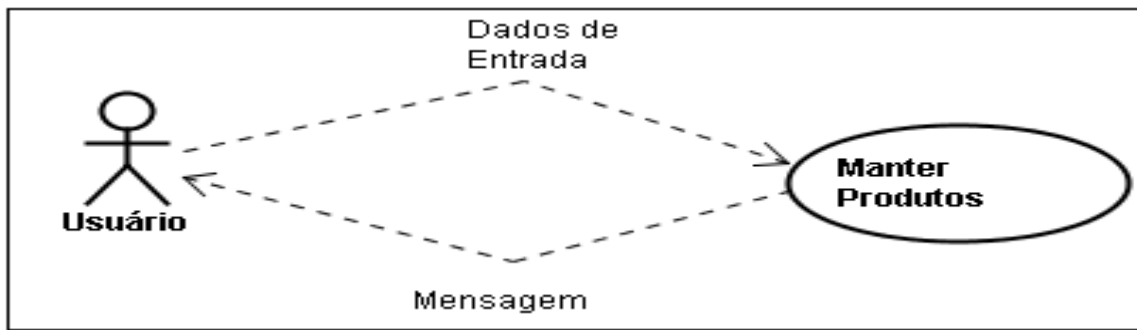


**Figura 3. Caso de Uso: Manter Clientes**

Nome do use case	Manter Clientes
Cenário Principal	O usuário efetua uma pesquisa de nome, RG ou CPF do cliente no sistema. Com isso o usuário pode cadastrar ou vender para um cliente.
Cenário alternativo	Se o Cliente for pesquisado por, nome, RG ou CPF, e não for encontrado, então o usuário poderá efetuar o cadastro do Cliente no sistema.
Fluxos de Exceção	Se o usuário efetuar uma pesquisa por nome de cliente, e encontrar dois nomes idênticos no sistema, então o sistema pedira para que seja feita uma pesquisa por RG ou CPF.



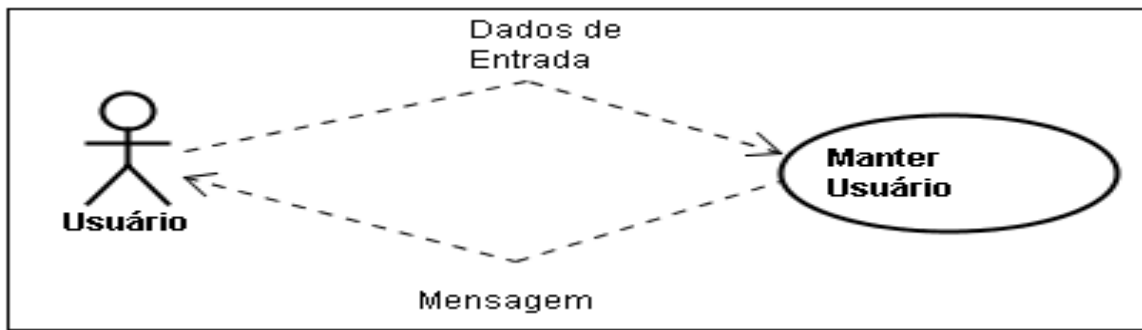
#### 4.6.4 Manter Peças



**Figura 4. Caso de Uso: Manter Produtos**

Nome do use case	Manter Peças
Cenário Principal	O usuário efetua uma pesquisa de peças por nome, tipo valor, e fornecedor.
Cenário alternativo	Se o nome da peça não for encontrada então deverá ser feito o cadastro da mesma, e incluir o fornecedor da peça.
Fluxos de Exceção	Caso o Fornecedor, ainda não exista o mesmo deverá ser cadastrado também.

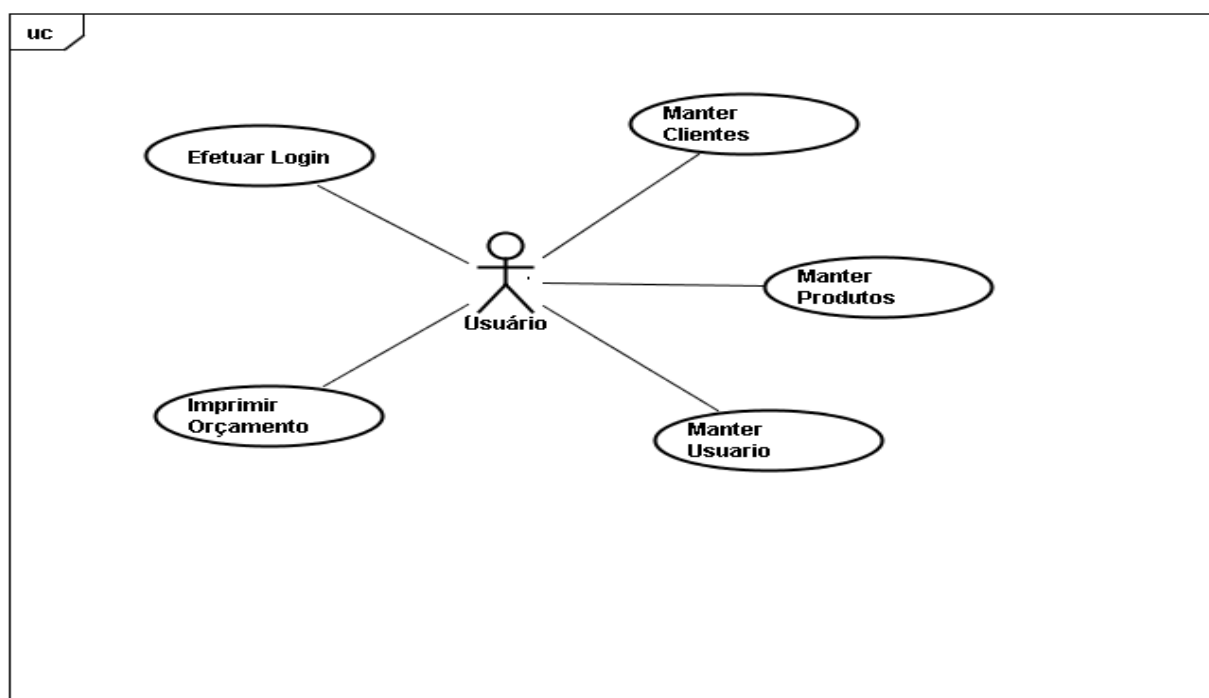
#### 4.6.5 Manter Usuário



**Figura 5. Caso de Uso: Manter Usuário**

Nome do use case	Manter Usuários
Cenário Principal	O usuário máster efetua manutenção de nome e endereço de usuários, com privilégios menores.
Cenário alternativo	Se o nome não for encontrado, o Usuário máster poderá efetuar o cadastrar, se for preciso.
Fluxos de Exceção	Se o Cliente com privilegio menor estiver sendo desligado da empresa, o mesmo deverá ser inativado.

## 4.7 EFETUAR ORÇAMENTO



**Figura 6. Caso de Uso: Efetuar Orçamento**

Nome do use case	Efetuar Orçamento
Cenário Principal	O usuário abre um pedido de Orçamento, nele deverá haver produtos, bicicletas completas ou apenas serviços. O valor de cada item do Orçamento deveser descrito de forma clara .
Cenário alternativo	Se um serviço for solicitado e o mesmo ainda não estiver no sistema, o mesmo poderá ser incluso no sistema, juntamente com o seu valor.
Fluxos de Exceção	O orçamento poderá ser Modificado mesmo depois de salvo.

## 4.8 DIAGRAMA GERAL DE CASO DE USO BICICLETARIA

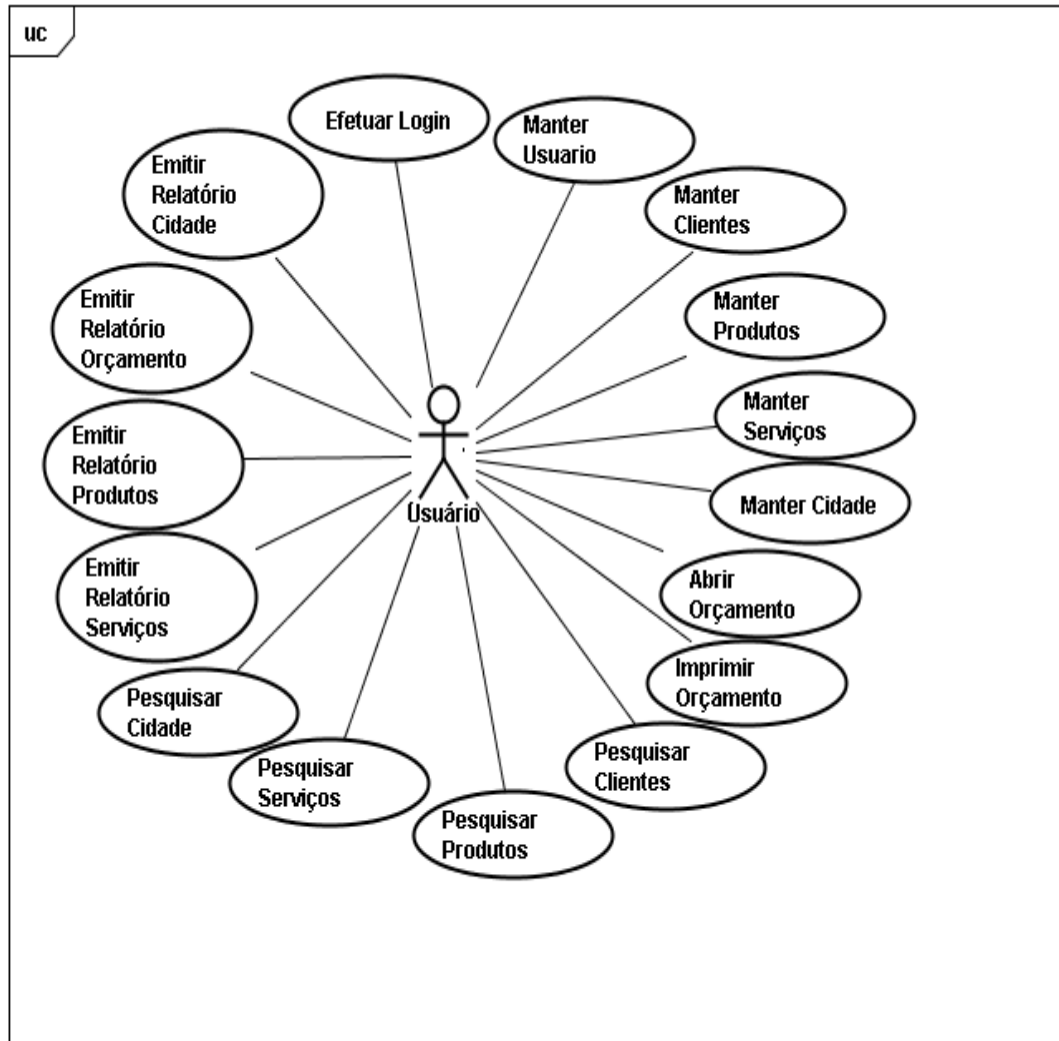


Figura 7. Diagrama Geral de Caso de Uso

## 4.9 DIAGRAMA DE CLASSE DE ORÇAMENTO

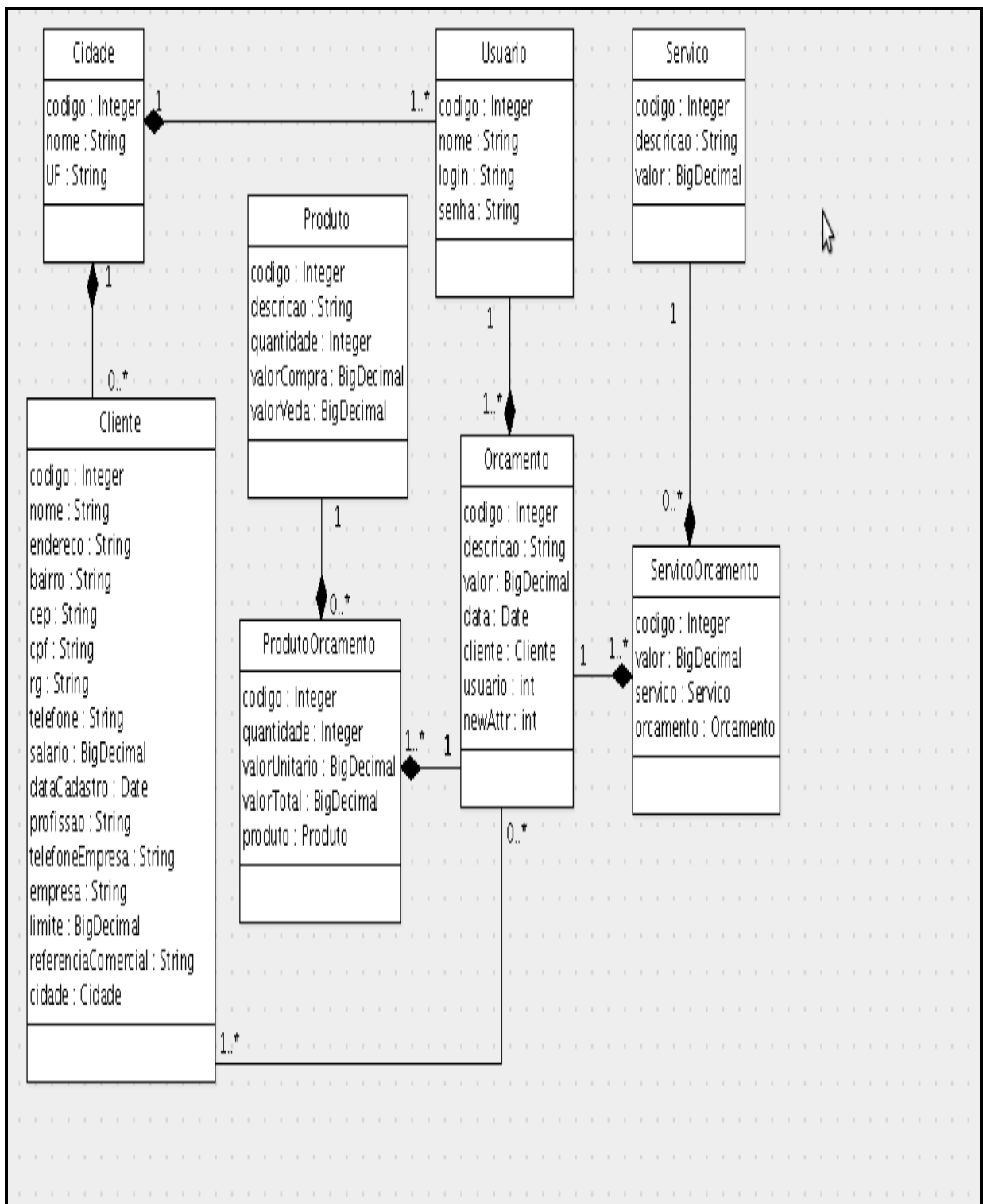


Figura 8. Diagrama de Classes de Orçamentos

## 4.10 DIAGRAMA DE ATIVIDADES ORÇAMENTO

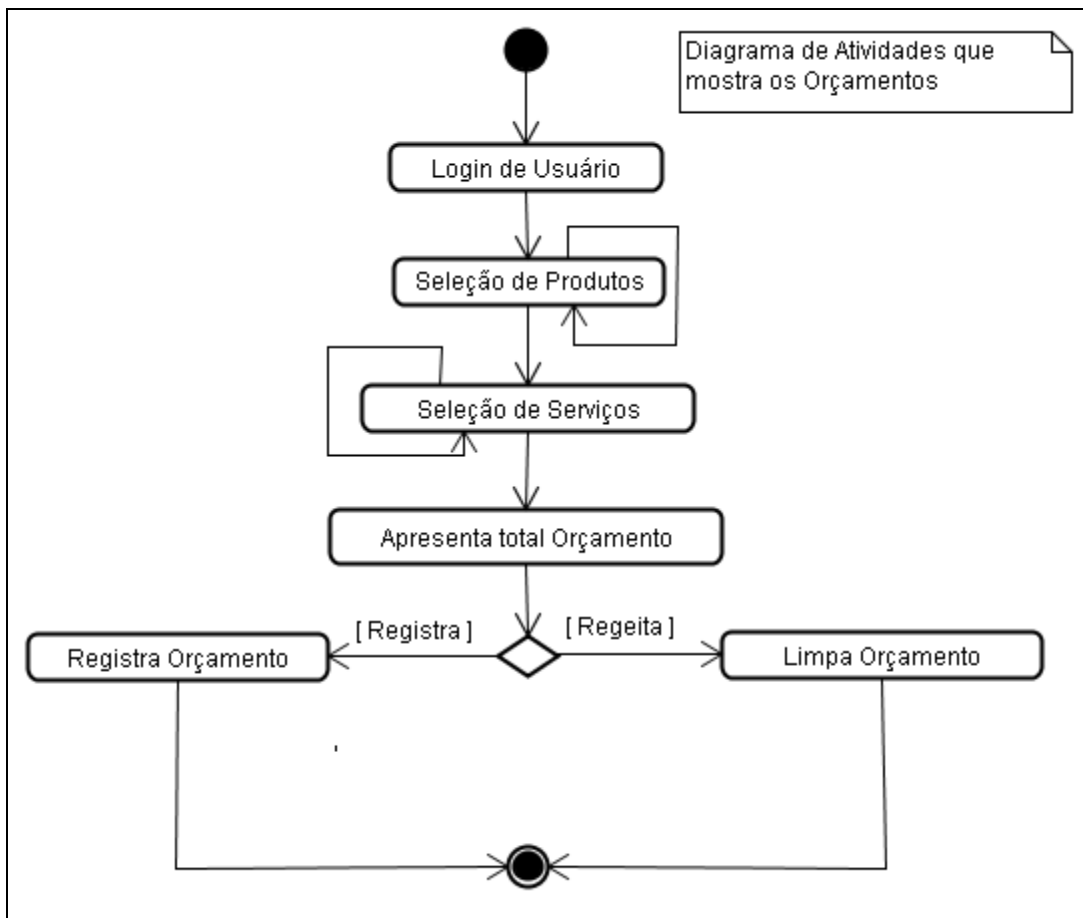


Figura 9. Diagrama de Atividades: Orçamento

## 4.11 DIAGRAMA DE SEQUÊNCIA EFETUAR LOGIN

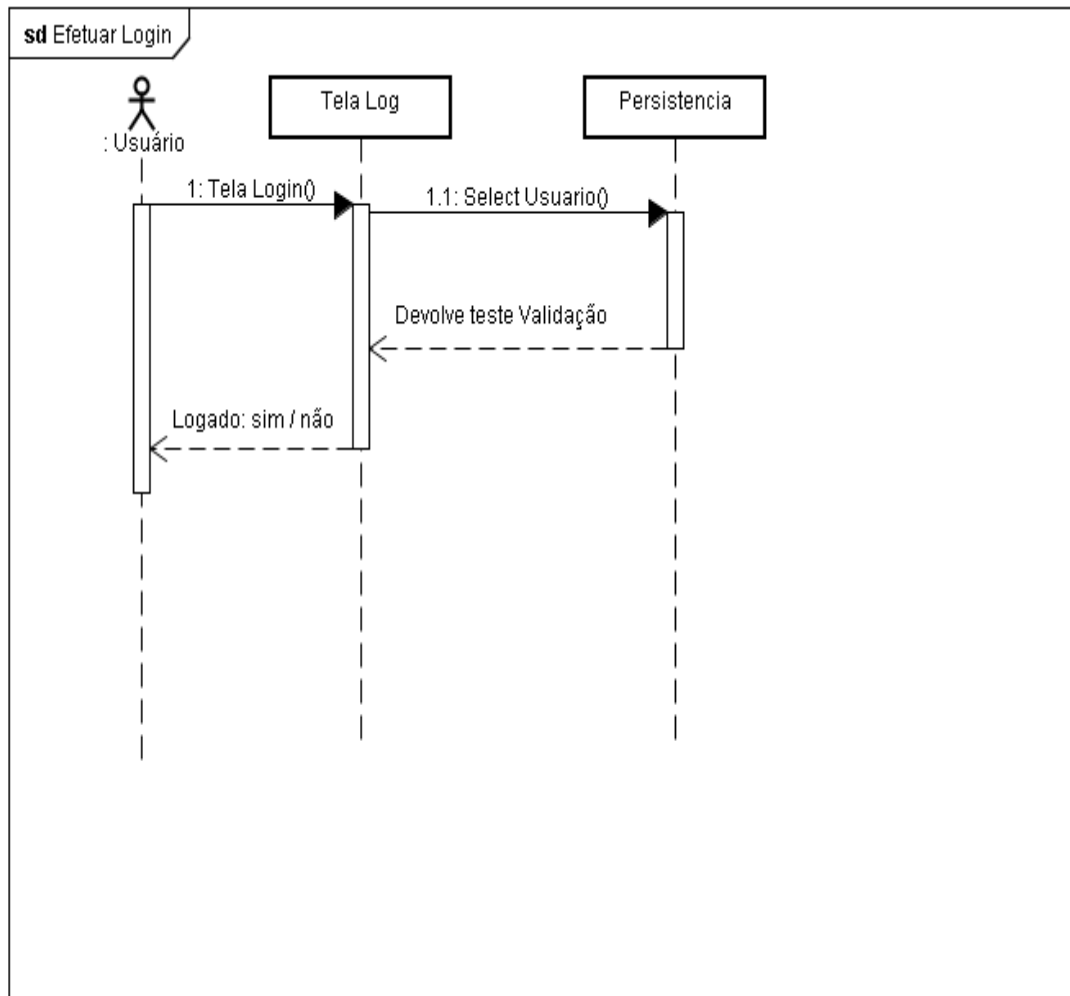
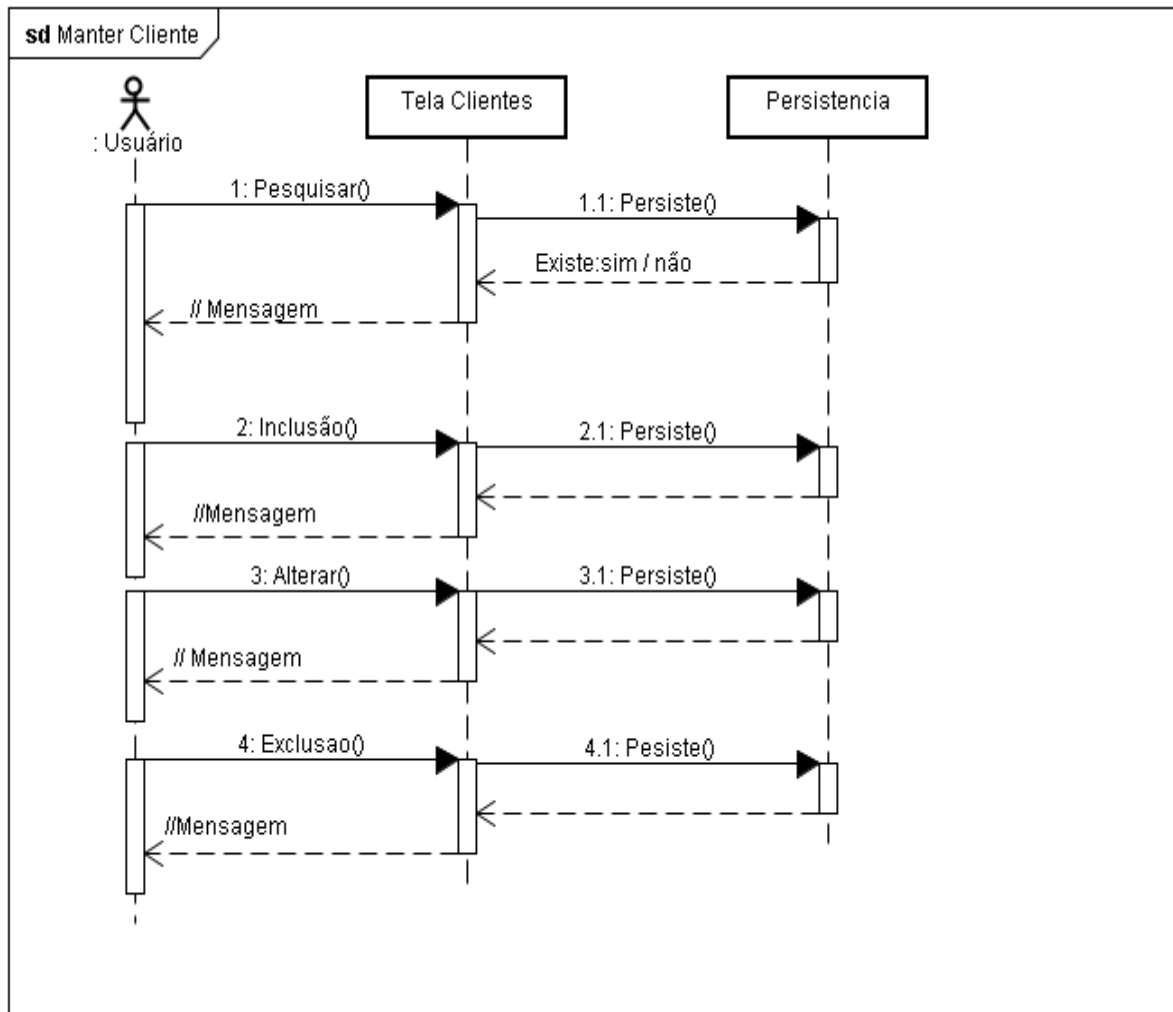


Figura 10. Diagrama de Sequência: Efetuar Login

## 4.12 DIAGRAMA DE SEQUENCIA MANTER CLIENTES



**Figura 11. Diagrama de Sequência: Manter Cliente**



## 4.13 DIAGRAMA DE SEQUÊNCIA MANTER PEÇAS

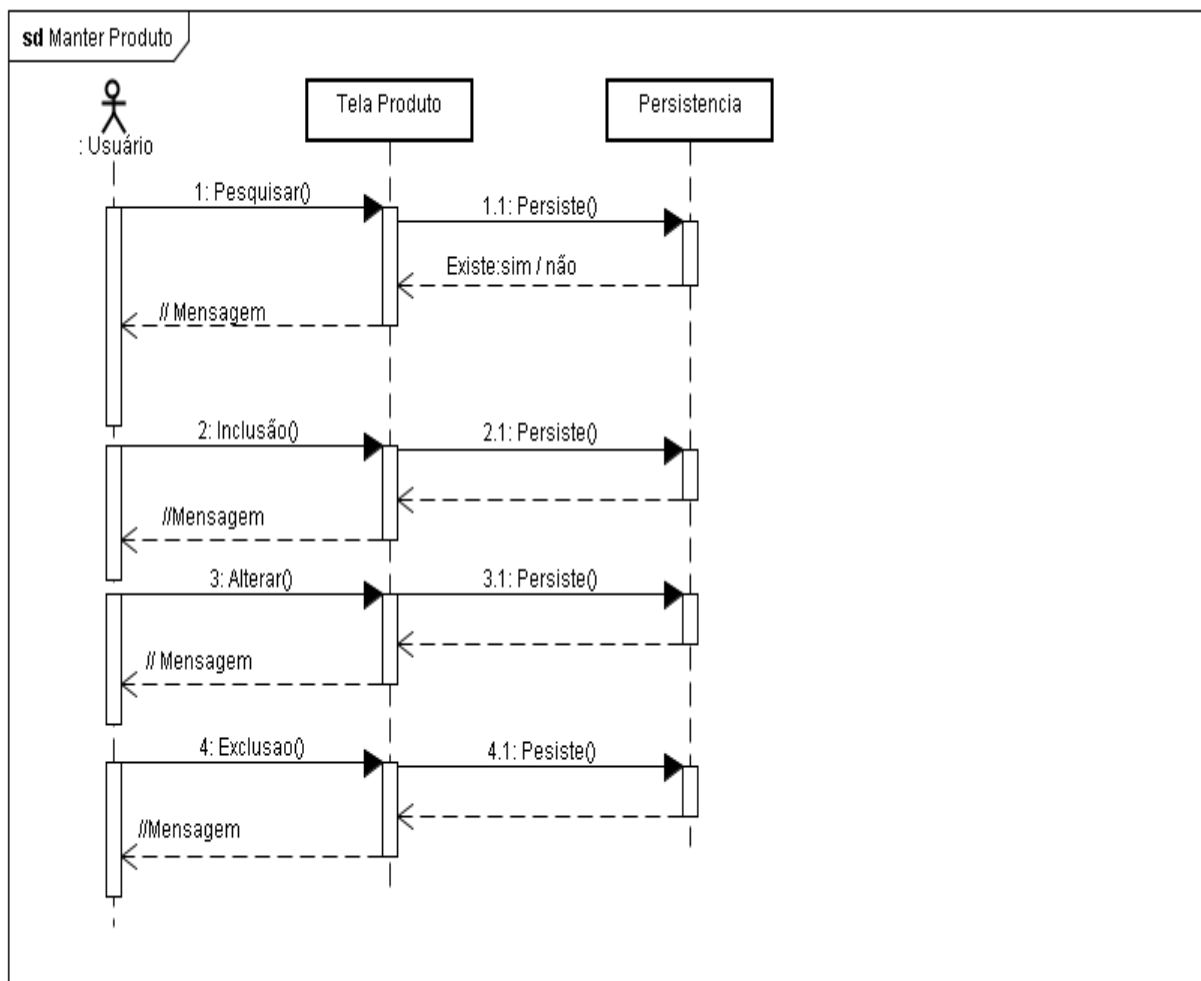


Figura 12. Diagrama de Sequência: Manter Peças

## 4.14 DIAGRAMA DE SEQUÊNCIA MANTER USUÁRIO

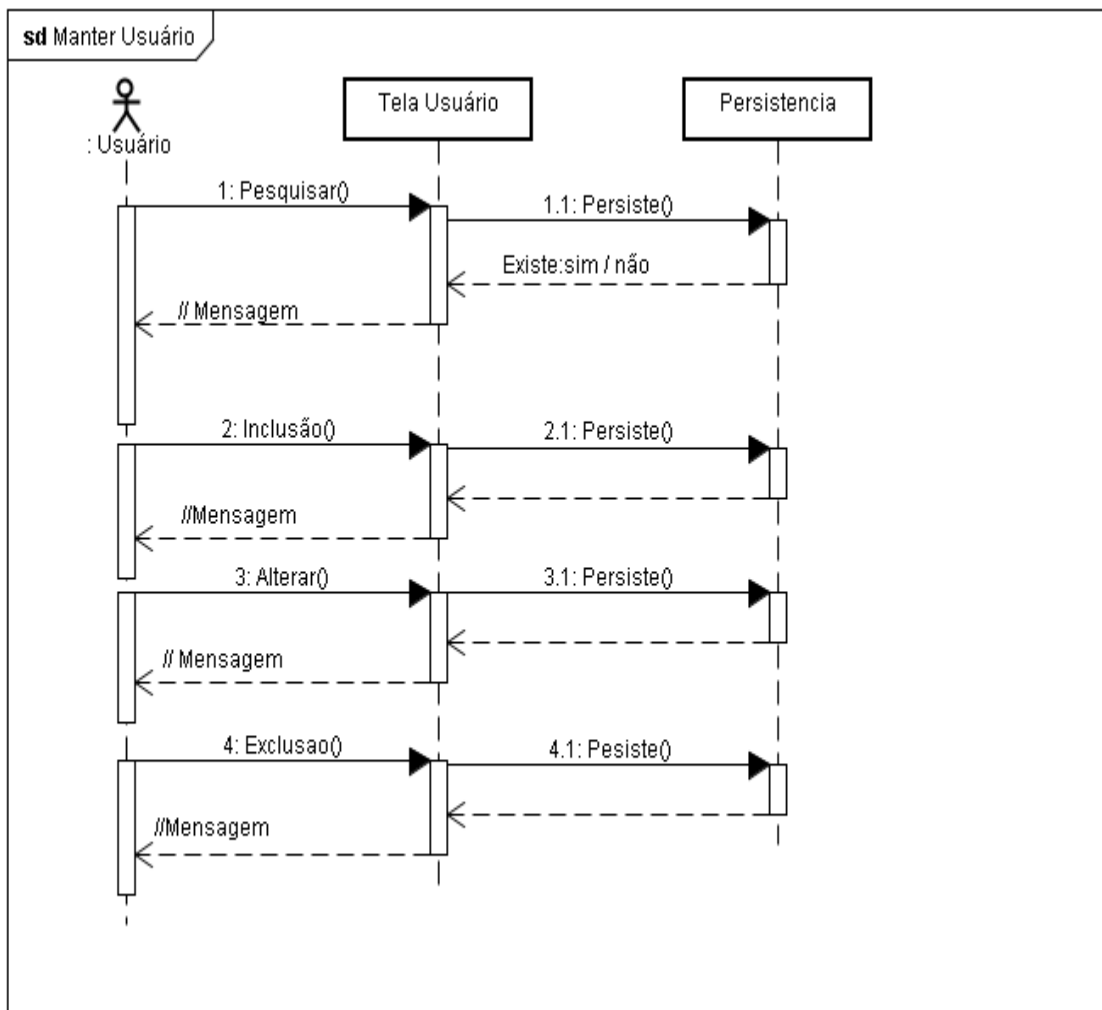
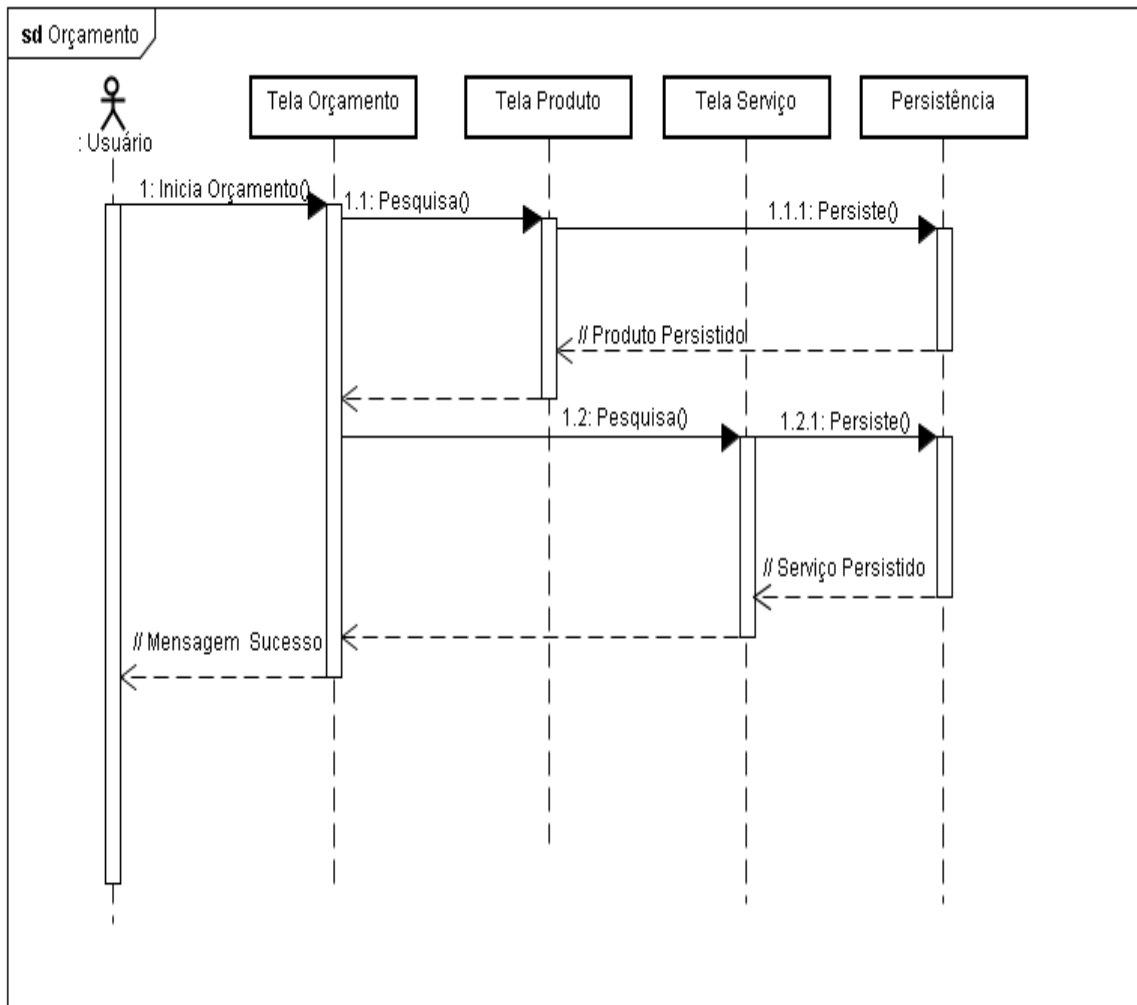


Figura 13. Diagrama de Sequência: Manter Usuários

## 4.15 DIAGRAMA DE SEQUÊNCIA MOVIMENTAÇÃO DE ORÇAMENTO



**Figura 14. Diagrama de Sequência: Movimentação de Orçamentos**

## 4.16 DIAGRAMA ER DE ORÇAMENTOS

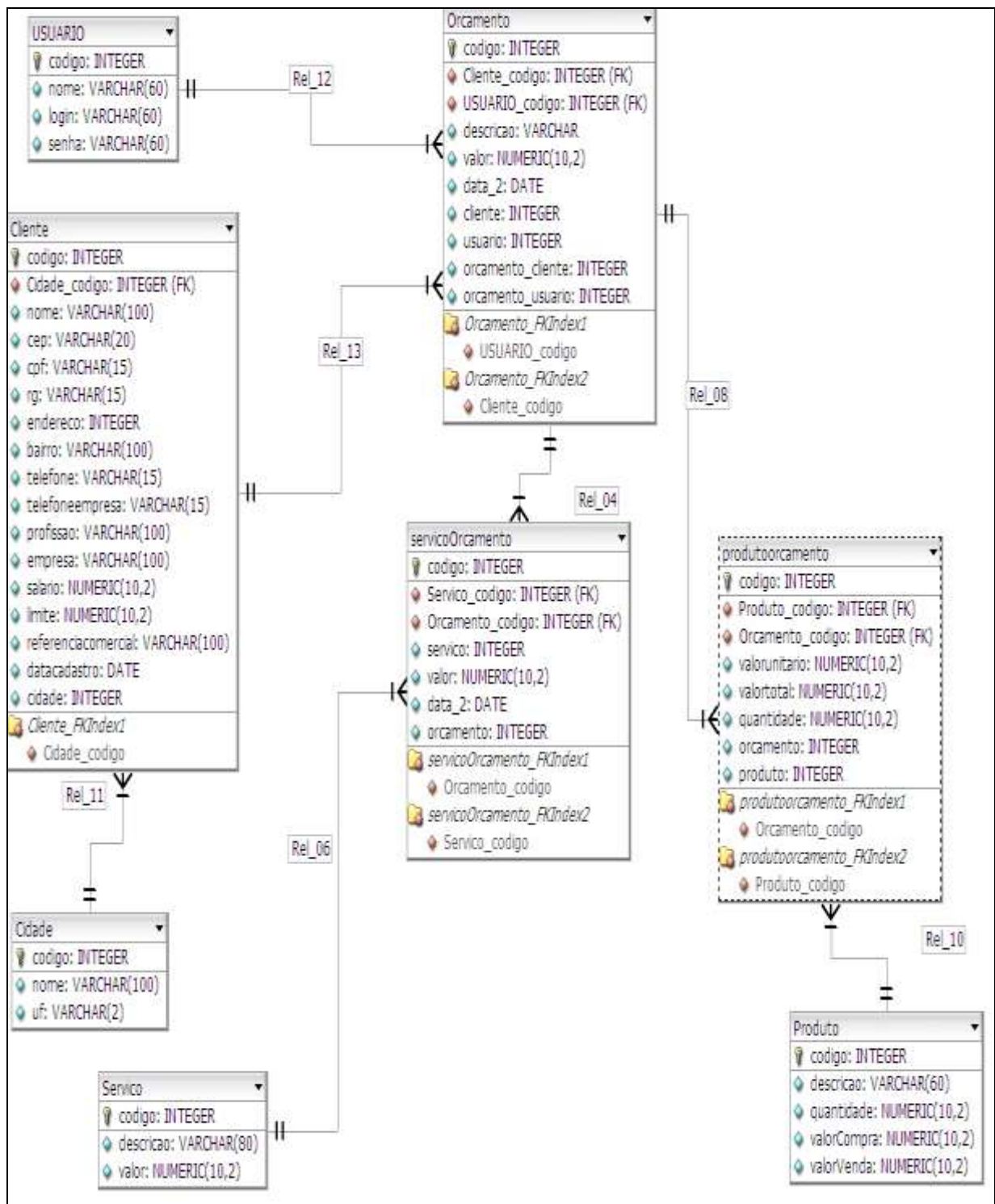


Figura 15. Diagrama ER Venda e Orçamento

## 4.17 PLANEJAMENTO DO PROJETO

### 4.17.1 WBS – Work Breakdown Structure

4.17.1.1 Define o agrupamento e disposição dos elementos do projeto.

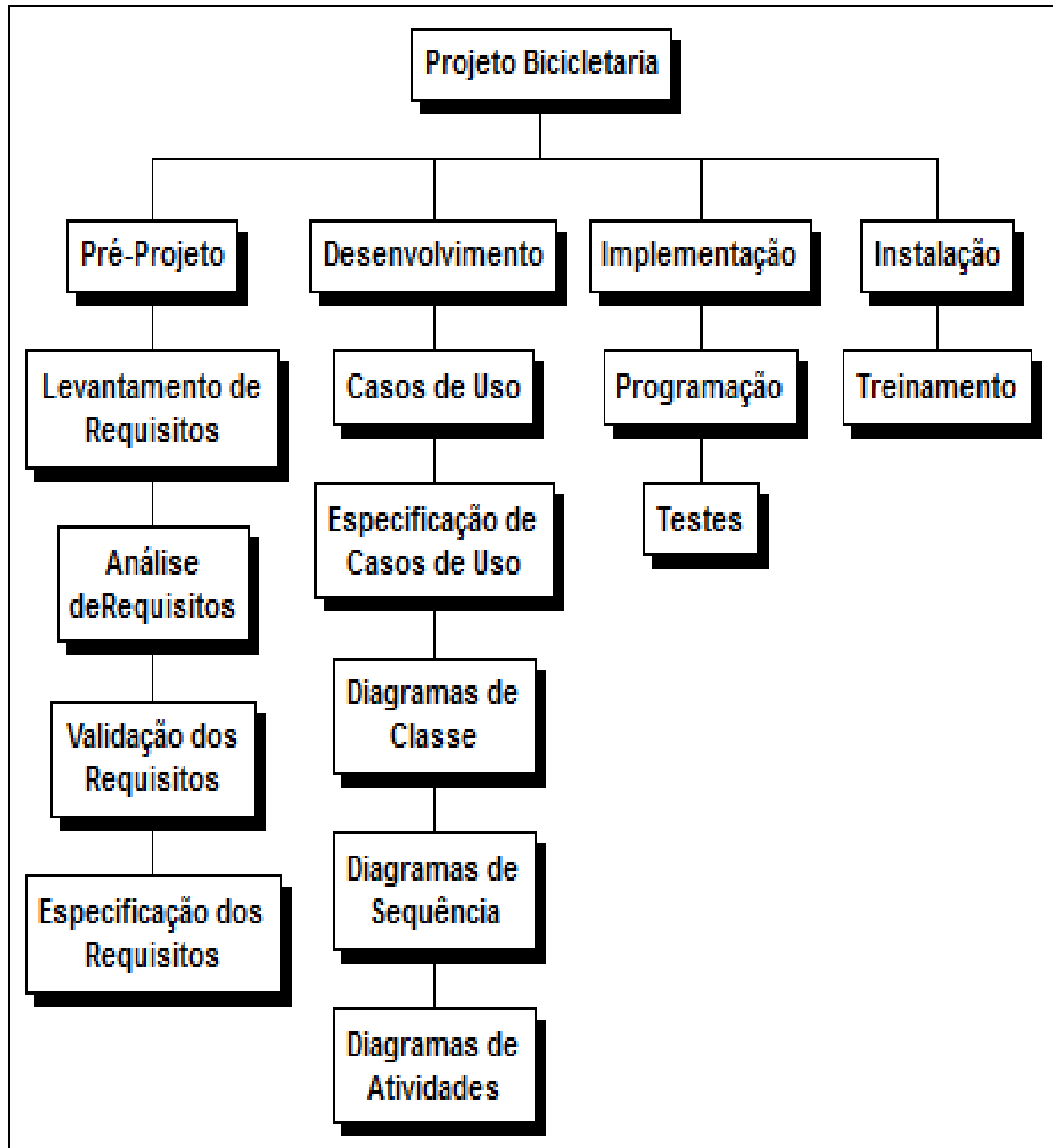


Figura 16. Diagrama WBS

## 4.18 SEQUÊNCIA DAS ATIVIDADES DEFINIDAS

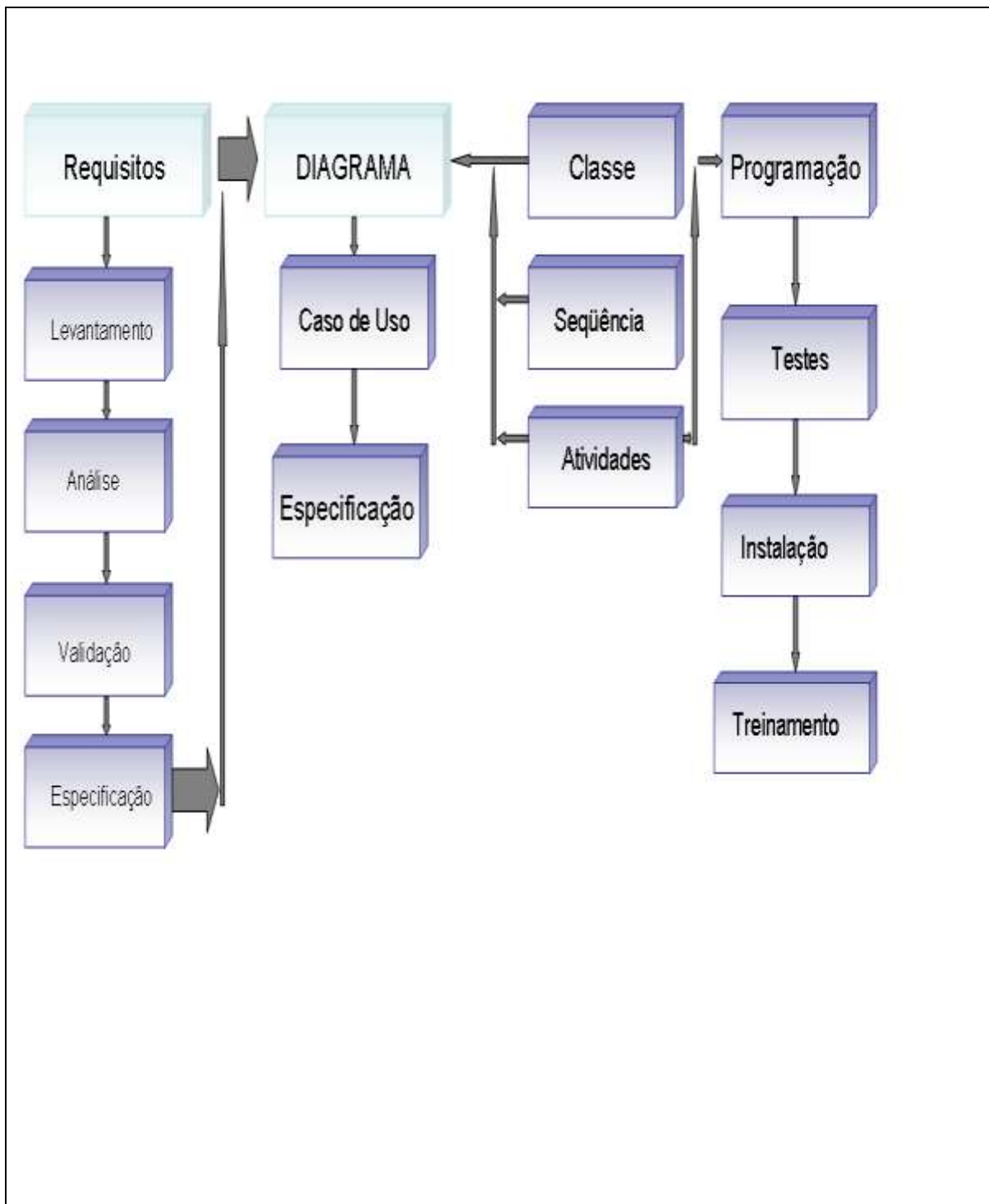


Figura 17. Sequenciamento das Atividades Definidas

## 4.19 DEFINIÇÃO DAS ATIVIDADES

### 4.19.1 Levantamento dos requisitos;

- Análise dos 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;
- Instalação;
- Treinamento;

## TABELA DAS ATIVIDADES DEFINIDAS

	Nome	Data Inicial	Data Final	Dias	Mês
REQUISITOS	Levantamento	01/03/2010	05/03/2010	5	MARÇO
	Análise	08/03/2010	12/03/2010	5	
	Validação	15/03/2010	19/03/2010	5	
	Especificação	22/03/2010	26/03/2010	5	
DIAGRAMAS	Casos de uso	29/03/2010	02/04/2010	5	ABRIL
	Classes	05/04/2010	16/04/2010	12	
	Seqüência	17/04/2010	23/04/2010	7	
	Atividades	24/04/2010	30/04/2010	7	
DESENVOLVIMENTO	Programação	03/05/2010	15/11/2010	192	MAIO JUNHO JULHO AGOSTO SETEMBRO
	Testes	16/11/2010	20/11/2010	5	
	Instalação	21/11/2010	21/11/2010	1	
	Treinamento	22/11/2010	26/11/2010	5	

Figura 18. Cronograma das Atividades Definidas

## 5 RECURSOS NECESSÁRIOS PARA O DESENVOLVIMENTO DO PROJETO

- 01 (um) Analista de Sistema;
- 01 (um) Programador Java;
- 01 (um) Notebook;
- 01 (uma) Impressor;
- Sistema Operacional Windows Sete Starter;
- IDE para o desenvolvimento: Eclipse;
- IDE para modelagem de dados: JUDE;
- Linguagem para o desenvolvimento: Java;
- Bando de dados: Hsqldb;
- Aplicativos: Pacote Office;



## ESTIMATIVA DE CUSTOS

### Analista de Sistema

- Valor Hora R\$ 15,00
- Horas Trabalhadas Dia: 8
- Custo diário: R\$ 120,00 (Cento e vinte reais)
- Quantidade dias: 46
- Custo total: (Dias \* Custo diário) = R\$ 5.520,00 (Cinco mil quinhentos e vinte reais)

### Programador

- Valor Hora R\$ 10,00
- Horas Trabalhadas Dia: 8
- Custo diário: R\$ 80,00 (Oitenta reais)
- Quantidade dias: 211
- Total:(Dias\*Custo diário) = R\$16.240,00 (Dezesseis mil duzentos e quarenta reais)
- Notebook Valor Unitário: R\$ 2.200,00 (Dois mil e duzentos reais)
- Total de dias de uso do Notebook: 257 (Oito meses e meio)
- Depreciação do Notebook: ((R\$ 2.200,00 / 36 meses) = R\$ 61,11) por mês
- Depreciação do notebook no projeto (R\$ 61,11 \* 8,5 meses) = R\$ 519,44
- 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).
- Custo dia: (R\$ 12,50 / 30 dias) = R\$ 0,41 (Quarenta e um centavos/dia) .
- Valor Impressora: (Dias\*Custo dia) = R\$18,04(Dezoito Reais e quatro centavos).
- IDE para o desenvolvimento – Eclipse Custo: R\$ 0,00.
- IDE para a modelagem de dados – JUDE Custo: R\$ 0,00.
- Linguagem para o desenvolvimento: Java Custo: R\$ 0,00.

Custo Total do Projeto:R\$ 21.760,00 (Vinte e um Mil setecentos e sessenta Reais)

**5.1.1 Resumo da Estimativa de Custos do Orçamento do Projeto**

Recurso Necessário	Valor
Analista de Sistemas	R\$ 5.520,00
Programador Java	R\$ 16.240,00
Notebook	R\$ 519,44
Impressora	R\$ 18,04
Eclipse IDE	R\$ 0,00
Jude	R\$ 0,00
Linguagem Java	R\$ 0,00
Banco Hsqldb	R\$ 0,00
Valor Total do Projeto:	R\$ 21.760,00

**Tabela 1 - Resumo da Estimativa de Custos**

## IMPLEMENTAÇÃO

O sistema será desenvolvido usando a linguagem Java.

Os pacotes serão Divididos em: Telas, Ação das telas e, Regras de Negócio.

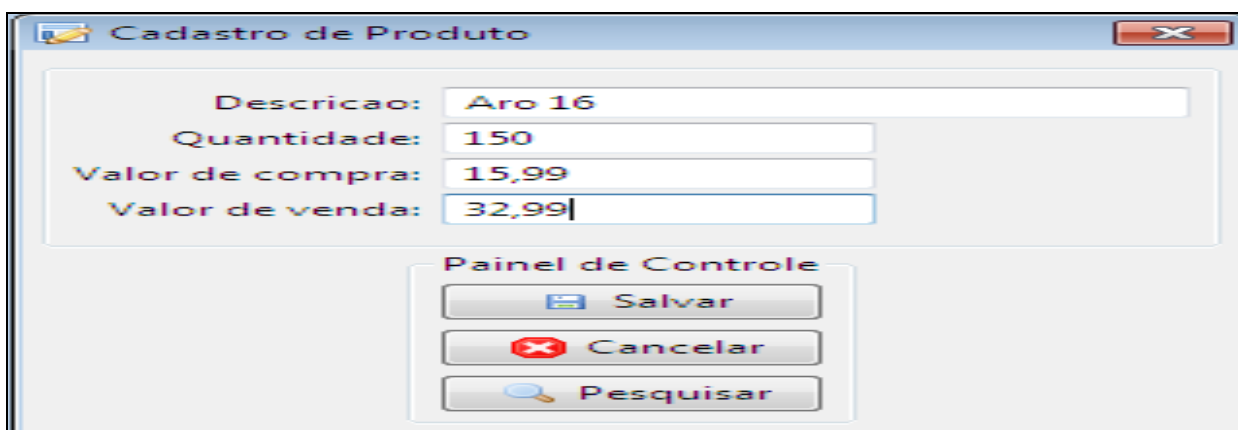
### 5.2 Algumas Telas do Sistema.

#### 5.2.1 Menu Principal



Figura - 18 Menu Principal

#### 5.2.2 Cadastro de Produto



Descrição:	<input type="text" value="Aro 16"/>
Quantidade:	<input type="text" value="150"/>
Valor de compra:	<input type="text" value="15,99"/>
Valor de venda:	<input type="text" value="32,99"/>

Painel de Controle

Figura – 19 Cadastro de Produtos

## Código necessário para a montagem da tela

```

public class CadastrarProduto {

    protected Shell sShell = null;
    protected Shell pai = null;
    protected Group groupBuscar = null;
    protected Button buttonSalvar = null;
    protected Button buttonCancelar = null;
    protected Button buttonBuscar = null;
    protected Group group = null;
    protected Label labelDescricao = null;
    protected Text textDescricao = null;
    protected Label labelQuantidade = null;
    protected Text textQuantidade = null;
    protected Label labelValorCompra = null;
    protected Text textValorCompra = null;
    protected Label labelvalorVenda = null;
    protected Text textValorVenda = null;

    public CadastrarProduto(Shell pai) {
        this.pai = pai;
        createSShell();
    }

    private void createSShell() {
        if (pai == null) {
            sShell = new Shell(SWT.DIALOG_TRIM);
        } else {
            sShell = new Shell(pai, SWT.DIALOG_TRIM | SWT.APPLICATION_MODAL);
        }
        GridLayout gridLayout = new GridLayout();
        gridLayout.numColumns = 1;
        gridLayout.marginWidth = 5;
        gridLayout.marginHeight = 5;
        gridLayout.verticalSpacing = 3;
        sShell.setText("Cadastrc de Produto");
        sShell.setImage(new Image(Display.getCurrent(), getClass().getResource
        sShell.setLayout(gridLayout);
        sShell.setSize(new Point(320, 286));
        createGroup();
        createGroupBascar();
    }
}

```

```

private void createGroupBuscar() {
    GridData gridData11 = new GridData();
    gridData11.heightHint = -1;
    gridData11.widthHint = 100;
    GridData gridData2 = new GridData();
    gridData2.heightHint = -1;
    gridData2.widthHint = 100;
    GridData gridData1 = new GridData();
    gridData1.heightHint = -1;
    gridData1.widthHint = 100;
    GridData gridData10 = new GridData();
    gridData10.horizontalAlignment = GridData.CENTER;
    gridData10.grabExcessVerticalSpace = false;
    gridData10.verticalAlignment = GridData.CENTER;
    groupBuscar = new Group(sShell, SWT.NONE);
    groupBuscar.setLayout(new GridLayout());
    groupBuscar.setText("Painel de Controle");
    groupBuscar.setLayoutData(gridData10);
    buttonSalvar = new Button(groupBuscar, SWT.NONE);
    buttonSalvar.setText("&Salvar");
    buttonSalvar.setLayoutData(gridData1);
    buttonSalvar.setImage(new Image(Display.getCurrent())
    buttonCancelar = new Button(groupBuscar, SWT.NONE);
    buttonCancelar.setText("&Cancelar");
    buttonCancelar.setLayoutData(gridData2);
    buttonCancelar.setImage(new Image(Display.getCurrent())
    buttonBuscar = new Button(groupBuscar, SWT.NONE);
    buttonBuscar.setText("&Pesquisar");
    buttonBuscar.setLayoutData(gridData11);
    buttonBuscar.setImage(new Image(Display.getCurrent())
}

```

### 5.2.3 Mensagem de Cadastro de Produto

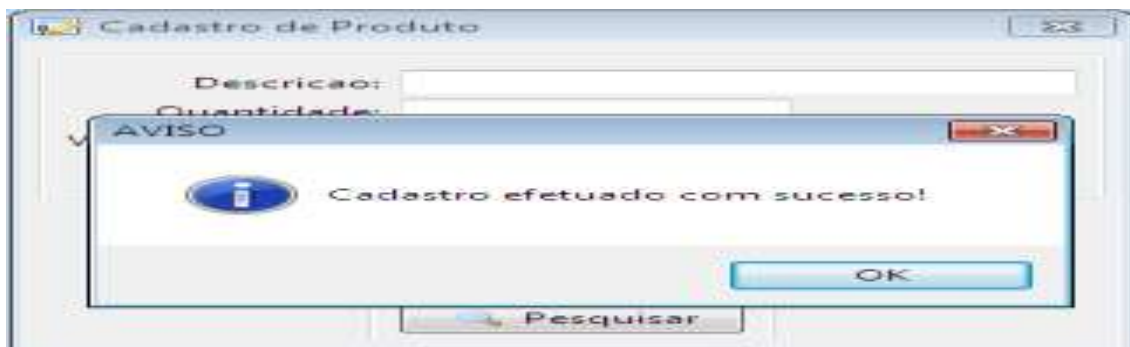


Figura 20 – Mensagem de Cadastro de Produtos

#### Código do Método Mensagem

```

public class Mensagem {

    public static void mensagemSucesso(Shell shell) {
        MessageBox box = new MessageBox(shell, SWT.ICON_INFORMATION | SWT.OK);
        box.setText("AVISO");
        box.setMessage("Cadastro efetuado com sucesso!");
        box.open();
    }
}

```

## 5.2.4 Pesquisa de Produto

Interface de pesquisa de produtos. O campo de busca está configurado para 'Descrição'. O resultado encontrado é:

Cód...	Descrição	Quan...	Valor Compra	Valor Venda
1	Aro 16	150,00	R\$ 15,99	R\$ 32,99

Figura 21 – Pesquisa de Produtos

## Códigos para Pesquisar, Editar e Excluir

```

public class PesquisaProduto extends Pesquisa {

    private List<Produto> produtos = null;
    private List<String> campos = Arrays.asList("Codigo", "Descrição", "Quantidade", "Valor Compra", "Valor Venda");
    private List<String> atributos = Arrays.asList("codigo", "descricao", "quantidade", "valorCompra", "valorVenda");
    private Integer codigoSelecionado = null;
    private boolean pesquisa;

    public PesquisaProduto(Shell shell, boolean pesquisa) {
        super(shell);
        this.pesquisa = pesquisa;
        preencherCampos();
        criarColunasTabela();
        iniciarTela();
    }
    private void iniciarTela () {
        toolBarEventos.setEnabled(pesquisa);
        adicionarEventosButton();
        adicionarEventosToolBar();
        adicionarEventosTable();
        adicionarEventosText();
        SwtUtil.centralize(sShell);
        sShell.open();
        SwtUtil.janelaModal(sShell);
    }

    private void adicionarEventosToolBar () {
        toolBarPesquisa.getItem(0).addSelectionListener(new SelectionAdapter() {

            public void widgetSelected (SelectionEvent e) {
                pesquisar(false);
            }
        });
        toolBarEventos.getItem(0).addSelectionListener(new SelectionAdapter() {

            public void widgetSelected (SelectionEvent e) {
                editar(tableResultado.getSelectionIndex());
            }
        });
        toolBarEventos.getItem(1).addSelectionListener(new SelectionAdapter() {

            public void widgetSelected (SelectionEvent e) {
                excluir(tableResultado.getSelectionIndex());
            }
        });
    }

    private void adicionarEventosButton () {
        buttonSair.addSelectionListener(new SelectionAdapter() {

            public void widgetSelected (SelectionEvent e) {
                sShell.close();
            }
        });
        buttonImprimir.addSelectionListener(new SelectionAdapter() {
            public void widgetSelected (SelectionEvent e) {
                pesquisar(true);
            }
        });
    }
}

```

```

private void selecionar (int index) {
    try {
        if (index < 0) {
            Mensagem.mostrarMensagem(sShell, "Selecione um item válido");
            return;
        } else {
            if (pesquisa) {
                editar(index);
            } else {
                setCodigoSelecionado(Integer.parseInt(tableResultado.getItem(index).getText(0)));
                sShell.close();
            }
        }
    } catch (Exception e) {
        Mensagem.mostrarErro(sShell, e);
    }
}

private void preencherCampos () {
    for (String s : campos) {
        combo.add(s);
    }
    combo.select(1);
}

private void criarColunasTabela () {
    TableColumn column1 = new TableColumn(tableResultado, SWT.CENTER);
    column1.setText("Código");
    column1.setWidth(50);
}

private void pesquisar(boolean isReport) {
    try {
        String atributo = atributos.get(combo.getSelectionIndex());
        String argumento = textBusca.getText();
        HibernateUtil.begin();
        ProdutoDao dao = new ProdutoDao();
        produtos = new ArrayList<Produto>();
        if (atributo.equals("codigo")) {
            Produto produto = dao.carregarPorId(argumento);
            if (produto != null) {
                produtos.add(produto);
            }
        } else {
            produtos = dao.pesquisar(atributo, argumento);
        }
        if(isReport){
            ReportUtil.showReport("Report/RelatorioProduto.jasper", produtos);
        }
        tableResultado.removeAll();
        for (Produto p : produtos) {
            TableItem item = new TableItem(tableResultado, SWT.NONE);
            item.setText(0, p.getCodigo().toString());
            item.setText(1, p.getDescricao());
            item.setText(2, Converter.converterBigDecimalTexto(p.getQuantidade()));
            item.setText(3, Converter.converterValorTexto(p.getValorCompra()));
            item.setText(4, Converter.converterValorTexto(p.getValorVenda()));
        }
        HibernateUtil.commit();
    } catch (NumberFormatException e) {
        Mensagem.mostrarErro(sShell, "Pesquisa por codigo aceita somente numeros!");
    } catch (Exception e) {
    }
}

```



```

    }
}

private void editar (int index) {
    if (index < 0) {
        Mensagem.mostrarErro(sShell, "Selecione um item para editar!");
    } else {
        CadastrarProdutoAcao acac = new CadastrarProdutoAcao(sShell, Integer.parseInt(tableResultado.getItem(index).getText(0)));
        if (acac.isConcluido()) {
            pesquisar(false);
        }
    }
}

private void excluir (int index) {
    try {
        if (index < 0) {
            Mensagem.mostrarErro(sShell, "Selecione um item para excluir!");
        } else {
            HibernateUtil.begin();
            new ProdutoDao().excluir(tableResultado.getItem(index).getText(0));
            HibernateUtil.commit();
            pesquisar(false);
        }
    } catch (Exception e) {
        HibernateUtil.rollback();
        Mensagem.mostrarErro(sShell, e);
    }
}

public Integer getCodigoSelecionado () {

```

## 5.2.5 Abertura de Orçamento

The screenshot shows a software window titled "Abertura de orçamento". It features several input fields: "Descrição", "Cliente", "Usuário", "Data", and "Hora". Below these is a table with four columns: "Descrição", "Quantidade", "Valor Unitário", and "Valor total". At the bottom of the window, there are three buttons: "Incluir", "Editar", and "Excluir". To the right of these buttons is a field labeled "Total em produtos" with the value "R\$ 0,00". At the bottom right, there is a larger field labeled "Total orçamento" also showing "R\$ 0,00". Below this field are two buttons: "Gravar" (Save) and "Cancelar" (Cancel).

Figura 22 – Abertura de Orçamento

## 5.2.6 Banco de Dados HSQLDB

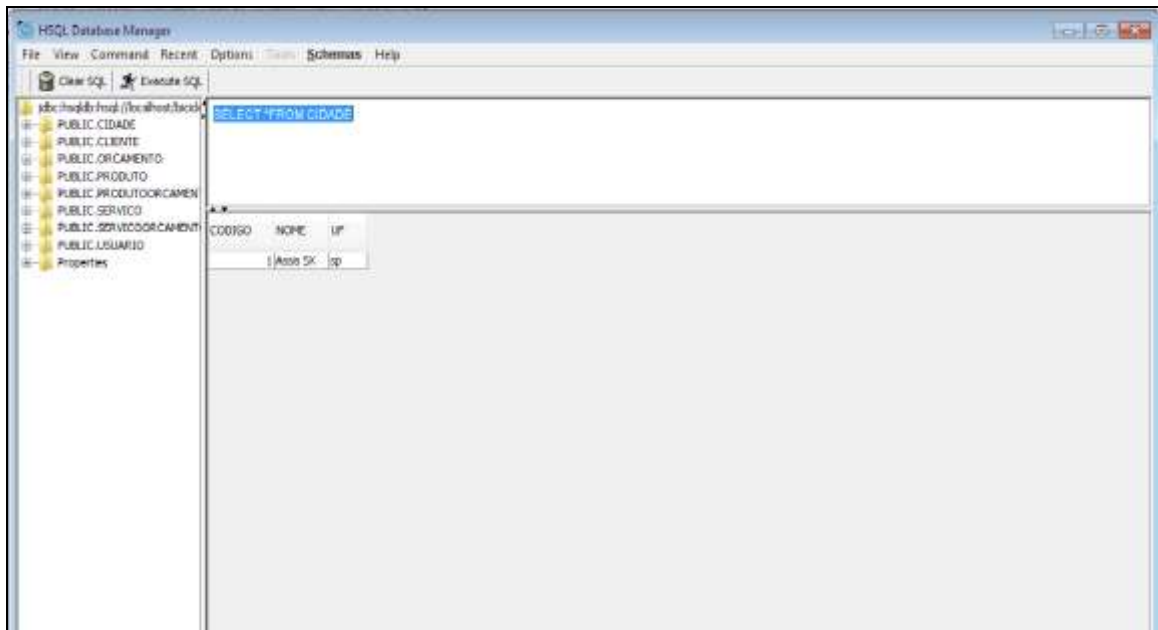


Figura 23 - Banco de Dados HSQLDB

## **6 CONCLUSÃO**

O documento presente mostra o caminho para o desenvolvimento de um software do tipo Desktop, e que o mesmo poderá atender a empresas que desejam manter um controle de orçamentos para os clientes.

Prevê também que as empresas detentoras do software, realizem suas atividades com maior eficácia, proporcionando, organização, maior agilidade e garantindo mais competitividade.

Com base em projeções com o uso do software, observou-se que os orçamentos serão gerenciados e controlados de maneira coesa, agregando mais qualidade no serviço, obtendo mais agilidade e segurança para os clientes. Através destas características importantes, a organização evita o caos. A aplicação poderá oferecer todas as informações sobre os históricos de orçamentos e os que estão em trânsito.

Futuramente será desenvolvido o módulo de cotação de preços vindo de diferentes fornecedores, e também o modulo de vendas e contas a pagar.

## REFERÊNCIAS

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

METSKER, Steven John. **Padrões de Projeto em Java**, 1ª ed. Werner Loeffler, Porto Alegre RS: Editora Bookman, 2004.

PRADO, Antonio Francisco do. **Modelagem Orientada a Objetos**, Universidade federal de São Carlos, Departamento de Computação.

PRADO, Antonio Francisco do. **Desenvolvimento de Software Orientado a Objetos**, Universidade federal de São Carlos, Departamento de Computação.

GONÇALVES, Edson. **Dominando Eclipse**, Rio de Janeiro: Editora Ciência Moderna, 2006.

SEVERO, Carlos Emílio Padilha. **NetBeans IDE**. Rio de Janeiro: Editora Brasport, 2005.

SIERRA, kathy. **Use a Cabeça Java**. Rio de Janeiro: 2ª ed. Editora Alta Books 2004.