



**UNIVERSIDADE FEDERAL DE OURO PRETO**  
**ESCOLA DE MINAS**  
**DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO,**  
**ADMINISTRAÇÃO E ECONOMIA - DEPRO**



**MODELAGEM DE SOFTWARE: UM ESTUDO DE CASO**  
**EM UMA EMPRESA DE DISTRIBUIÇÃO DE FRUTAS**

**Marcelo Felisberto Rodrigues**

**Ouro Preto – MG**

**2013**

**Marcelo Felisberto Rodrigues**

**Marcelofr8@gmail.com**

# **MODELAGEM DE SOFTWARE: UM ESTUDO DE CASO EM UMA EMPRESA DE DISTRIBUIÇÃO DE FRUTAS**

Monografia submetida à apreciação da banca examinadora de graduação em Engenharia de Produção da Universidade Federal de Ouro Preto, como parte dos requisitos necessários para a obtenção de grau de graduado em Engenharia de Produção.

**Orientador:** Profº André Luís Silva

**Ouro Preto – MG**

**2013**

## RESUMO

As novas tecnologias mudaram a forma com que as organizações lidam com seus sistemas de informação, levando ao aumento da complexidade e novos desafios no desenvolvimento dos softwares. O objetivo deste trabalho é definir os requisitos e modelar um sistema informatizado destinado à gestão de uma micro empresa de distribuição de frutas. Neste intuito será apresentada uma revisão bibliográfica do assunto, a fim de se obter um embasamento teórico necessário. Após, foi definida a metodologia do trabalho, um estudo de caso, que foi realizado na empresa Sabor Brasil, uma micro empresa do ramo escolhido. Desta forma, o estudo de caso foi desenvolvido seguindo as etapas propostas pela Engenharia de Software, descrevendo primeiramente a análise dos requisitos necessários para a modelagem de um Software que atenda às necessidades da empresa estudada. Em seguida foram definidos os diagramas, que seriam desenvolvidos. E através das definições da UML, foram descritos os diagramas propostos com o auxílio do Software StarUML.

**Palavras-chaves:** Engenharia de Software. Análise de Requisitos. Unified Modeling Language. Modelagem.

## **ABSTRACT**

New technologies have changed the way organizations deal with their information systems, leading to increased complexity and new challenges in the development of software. The objective of this work is to define the requirements and modeling a computerized system for the management of a micro fruit distribution company. In this view, was presented a literature review of the subject in order to obtain a theoretical foundation required. After it was defined the work methodology, a case study, which was conducted at the company Sabor Brazil, a micro company. Thus the case study was developed following the steps proposed by the Software Engineering, describing the first analysis of the requirements for modeling a software that meets the needs of the company studied. Then we defined the diagrams, which would be developed. And through the definitions of UML diagrams were developed with the assistance of the proposed StarUML software and Microsoft Office Access.

**Key-words:** Software Engineering. Requirements Analysis. Unified Modeling Language. Modeling.

## LISTA DE FIGURAS

|                                                    |    |
|----------------------------------------------------|----|
| Figura 1: Pirâmide de propagação de erro .....     | 22 |
| Figura 2: Como Realmente funciona um projeto ..... | 24 |
| Figura 3: StarUML .....                            | 26 |
| Figura 4: Enterprise Architect .....               | 27 |
| Figura 5: ArgoUML .....                            | 27 |
| Figura 6: Violet UML .....                         | 28 |
| Figura 8: Diagrama de classe .....                 | 31 |
| Figura 7: Classe .....                             | 31 |
| Figura 9: Diagrama de objetos .....                | 32 |
| Figura 10: Diagrama de caso de uso .....           | 33 |
| Figura 11: Diagrama de atividades .....            | 34 |
| Figura 12: Diagrama de Caso de Uso 2 .....         | 40 |
| Figura 13: Diagrama de classes 2 .....             | 42 |
| Figura 14: Diagrama de atividade cliente .....     | 44 |
| Figura 15: Diagrama de atividade fornecedor .....  | 45 |
| Figura 16: Diagrama de atividade compras .....     | 46 |
| Figura 17: Diagrama de atividade vendas .....      | 47 |
| Figura 18: Diagrama de atividade recebimento ..... | 48 |
| Figura 19: Diagrama de atividade pagamento .....   | 49 |
| Figura 20: Diagrama de atividade produto .....     | 50 |
| Figura 21: Digrama de objeto 2 .....               | 51 |
| Figura 22: Implantação fornecedor .....            | 52 |
| Figura 23: Relatório fornecedor .....              | 53 |
| Figura 24: Implantação cliente .....               | 53 |
| Figura 25: Relatório cliente .....                 | 53 |
| Figura 26: Implantação compra .....                | 54 |
| Figura 27: Relatório compra .....                  | 55 |
| Figura 28: Implantação venda .....                 | 55 |
| Figura 29: Relatório venda .....                   | 56 |
| Figura 30: Implantação produtos .....              | 56 |

## **LISTA DE TABELAS**

|                                                       |    |
|-------------------------------------------------------|----|
| Tabela 1: Definição de Tecnologia de Informação. .... | 16 |
| Tabela 2: Definição de Sistema de Informação. ....    | 16 |
| Tabela 3: Definição de Software.....                  | 19 |
| Tabela 4: Definição de Engenharia de Software.....    | 20 |
| Tabela 5: Softwares de UML .....                      | 26 |

## Sumário

2

|          |                                                       |    |
|----------|-------------------------------------------------------|----|
| 1.       | Introdução.....                                       | 10 |
| 1.1.     | Formulação do Problema .....                          | 11 |
| 1.2.     | Objetivo.....                                         | 13 |
| 1.2.1.   | Objetivo geral .....                                  | 13 |
| 1.2.2.   | Objetivos específicos.....                            | 13 |
| 1.3.     | Justificativa do Trabalho.....                        | 13 |
| 2.       | Fundamentação Teórica .....                           | 15 |
| 2.4.     | Tecnologia de Informação e Sistema de Informação..... | 15 |
| 2.5.     | Engenharia de Software .....                          | 18 |
| 2.6.     | Engenharia de Requisitos.....                         | 22 |
| 2.7.     | Unified Modeling Language (UML) .....                 | 23 |
| 2.7.1.   | Definição .....                                       | 23 |
| 2.7.2.   | Histórico .....                                       | 24 |
| 2.7.3.   | Análise de Software de UML .....                      | 25 |
| 2.7.4.   | Itens, Relacionamentos e Diagramas.....               | 28 |
| 2.7.4.1. | Diagrama de classes .....                             | 30 |
| 2.7.4.2. | Diagrama de objetos .....                             | 32 |
| 2.7.4.3. | Diagrama de caso de uso .....                         | 32 |
| 2.7.4.4. | Diagrama de atividades .....                          | 33 |
| 2.7.4.5. | Diagrama de implantação .....                         | 34 |
| 3.       | Metodologia .....                                     | 35 |
| 3.4.     | Conceitos de Pesquisa.....                            | 35 |
| 3.5.     | Abordagem.....                                        | 35 |
| 3.6.     | Método de Pesquisa .....                              | 36 |
| 4.       | Estudo de caso .....                                  | 37 |

|          |                                |    |
|----------|--------------------------------|----|
| 4.4.     | Análise de requisitos .....    | 37 |
| 4.4.1.   | Requisitos funcionais.....     | 37 |
| 4.4.2.   | Requisitos não funcionais..... | 39 |
| 4.5.     | Diagramas .....                | 40 |
| 4.5.1.   | Diagrama de Caso de Uso .....  | 40 |
| 4.5.2.   | Diagrama de Classes.....       | 42 |
| 4.5.3.   | Diagrama de Atividade.....     | 43 |
| 4.5.4.   | Diagrama de Objetos .....      | 50 |
| 4.5.5.   | Diagrama de Implantação.....   | 51 |
| 4.5.5.1. | Fornecedor .....               | 52 |
| 4.5.5.2. | Cliente.....                   | 53 |
| 4.5.5.3. | Compra .....                   | 54 |
| 4.5.5.4. | Venda.....                     | 55 |
| 4.5.5.5. | Produto .....                  | 56 |
| 5.       | Discussão dos resultados.....  | 57 |
| 6.       | Conclusão .....                | 57 |
| 7.       | Bibliografia.....              | 59 |



## 1. Introdução

Segundo Laurindo (2008, p. 13) “os computadores eletrônicos surgiram logo após o final da segunda guerra mundial, quando se havia a necessidade de máquinas que pudessem fazer, os cada vez mais complexos cálculos balísticos”. Os computadores pessoais vieram apenas no final da década de 70 e foi a partir desta época que vieram novas alternativas e aplicações para o uso da informática.

Com a vinda dos computadores pessoais a importância destes e de suas ferramentas foram crescendo cada vez mais. Para Sommerville (2007) praticamente todos os países, dependem de sistemas complexos baseados em computadores. Infra-estruturas e serviços nacionais contam com sistemas baseados em computadores, e a maioria dos produtos eletrônicos inclui um computador e um software de controle. A importância dada aos computadores e seus recursos pode ser vista também em Rezende e Abreu (2010), que segundo estes autores, a tecnologia e os avanços em suas capacidades oferecem um maior número de alternativas de decisão aos negócios e às organizações de todos os portes.

Um dos principais benefícios trazidos pelos complexos sistemas baseados em computadores foi a capacidade disponibilizada por estes sistemas em armazenar, processar e manipular informações. A importância da informação é narrada em Rezende e Abreu (2010) que considera que a informação e o conhecimento serão os diferenciais das empresas que pretendem destacar-se no mercado. Segundo Laurindo (2008), informação seria o conjunto de dados devidamente tratados de forma a terem um significado, bem como organizados e classificados para alguma finalidade sendo úteis para as pessoas em processos, como a tomada de decisões. Já conhecimento seria o conjunto de ferramentas conceituais e categorias usadas pelas pessoas para criarem, colecionarem, armazenarem ou compartilharem informação.

As empresas possuem diversas opções de tecnologia para auxiliá-las na gestão de informação e muitas destas se enquadram nas Tecnologias de Informação e nos Sistemas de Informação. Como afirma Tonsig (2008, p. 32) “a obtenção de informações que sejam relevantes para um negocio dá-se através de algum Sistema de Informações que tenha sido eficientemente planejado.”. Os autores Laudon e Laudon (2007) também afirmam que as empresas estão sempre tentando melhorar a eficiência de suas operações a fim de conseguir maior lucratividade, sendo que as Tecnologias e os Sistemas de Informação estão entre as mais importantes para atingir altos níveis de eficiência e produtividade nas operações.

Além destes pontos, Turban, Jr. e Potter, (2007) defendem a ideia de que as organizações não podem funcionar sem a Tecnologia de Informação.

Dessa forma, é possível perceber a importância das Tecnologias de Informação e dos Sistemas de Informação no gerenciamento das informações.

Uma Tecnologia de Informação muito utilizada pelas empresas é o Software como mostra Laudon e Laudon (2007, p. 9) “por Tecnologia da Informação, entenda-se todo Software e todo hardware de que uma empresa necessita para atingir seus objetivos organizacionais.”.

Para Pressman (2010, tradução nossa) os Softwares utilizados em empresas têm como finalidade facilitar e resolver operações de negócios ou ajudar em tomadas de decisões. Além de aplicações em processamento de dados, os Softwares são utilizados no controle de funções de mercado em tempo real.

Diversos autores como Pressman (2010) e Tonsig (2008) defendem a ideia de que os Softwares devem ser criados através de processos de engenharia. Para isso utiliza-se a Engenharia de Software.

Segundo Sommerville (2007, p. 3) “a Engenharia de Software é um ramo da engenharia cujo foco é o desenvolvimento dentro de custos adequados de sistemas de Software de alta qualidade.”

Porém, mesmo com a ajuda de recursos, para Tonsig (2008) ainda é muito frequente empresas desenvolvendo e entregando Softwares de má qualidade de funcionamento e mal estruturados para clientes, além de entregarem o produto em atraso.

Tendo em vista a importância da modelagem de Softwares para uma empresa, será desenvolvido neste trabalho um modelo de Software para a utilização em uma microempresa, com o intuito de ajudá-la a armazenar, processar e manipular suas informações e gerar conhecimentos.

### **1.1. Formulação do Problema**

Foia( 2006, p. 1) demonstra o quanto a Tecnologia de Informação era de difícil acesso e entendimento para as empresas em um passado próximo “em um passado próximo, a Tecnologia de Informação era tratada como algo quase esotérico com sua própria terminologia, seus problemas eram de difícil entendimento para os não iniciados, seus cronogramas, fictícios e seus orçamentos, sempre megalomaniacos.”. Os custos eram muito altos e existiam poucas pessoas que eram capazes de um bom manuseio de seus recursos.

Segundo Rezende e Abreu (2010, p. 43), “o mundo esta na era da informatização em que o conhecimento fará a diferença.” e Foina (2006) completa dizendo que a informação com qualidade e no tempo certo é um dos produtos mais valiosos para a gestão de uma empresa, e deve ser tratada com profundidade por sua alta direção, tendo esta a função entender e alinhar as demandas atuais de tecnologia, o que demonstra a importância atingida pelas Tecnologias de Informação atualmente e o valor que a elas deve ser dado pelas empresas.

Os Sistemas de Informação são ferramentas utilizadas na transformação de informação em conhecimento, que segundo Rezende (1999) apud REZENDE (2010, p. 38) “todo sistema, usando ou não recursos de Tecnologia da Informação, que manipula e gera informação pode ser genericamente considerado Sistema de Informação”. Os Sistemas de Informação vieram para auxiliar os gestores através do gerenciamento das informações, podendo ou não estar atrelado aos recursos tecnológicos.

O Software é uma Tecnologia da Informação muito utilizada nos Sistemas de Informação pelos gestores e que segundo Rezende e Abreu (2010) são um conjunto de comandos, instruções ou ordens elaboradas pelo cliente para o computador cumprir, visando resolver problemas e desenvolver atividades ou tarefas específicas. Para se ter uma idéia da importância dos Softwares, segundo Laudon e Laudon (2007, p. 5) “em 2006, as empresas norte-americanas investirão 1,8 trilhão de dólares em hardware e Software para Sistemas de Informação”.

Algumas empresas acreditam que apenas se informatizar é o melhor caminho para se obter resultados. Como demonstra Rezende e Abreu (2010, p.33) “muitas empresas ainda acreditam que o simples ato de informatizar a empresa, espalhando computadores e impressoras pelas unidades departamentais, ligando-os em rede e instalando sistemas aplicativos, possa organizar as mesmas”. Porém apenas informatizar não é suficiente, é preciso que os usuários saibam utilizar os recursos da tecnologia aplicada e que ela funcione sempre em conformidade com as necessidades estipuladas.

Para Magela (2006) a teoria e prática na construção de Software têm evoluído substancialmente ao longo dos anos. Tal evolução fez com que profissionais percebessem a falta de habilidade existente para o desenvolvimento de Softwares mais complexos e que exijam uma maior funcionalidade, iniciando-se então um movimento de transição gradativa no papel de construção de um Software, da Ciência da Computação para a Engenharia de Software.

Para Sommerville (2007), a grande diferença entre Engenharia de Software e Ciência da Computação é que a segunda está relacionada com teorias e fundamentos, enquanto a primeira com prática de desenvolvimento e entrega de software útil.

Os Softwares auxiliam gestores em diversos setores. Um deles é as micro e pequenas empresas. Segundo a lei complementar 123/2006 e o Serviço Brasileiro de Apoio às Micro e Pequenas Empresas Sebrae (2012), microempresas são aquelas que possuem um faturamento bruto anual de até R\$360 mil reais. As microempresas possuem um papel fundamental no desenvolvimento do Brasil, sendo que, segundo o Instituto Brasileiro de Geografia e Estatística IBGE (2001), em 2011, quase 98% das empresas existentes no Brasil eram micro e pequenas empresas, sendo responsáveis por cerca de 60% de empregos constituintes no país, mas que representam apenas cerca de 20% do Produto Interno Bruto brasileiro.

O estudo aqui apresentado nesta monografia será realizado em uma microempresa de distribuição de frutas situada em Ipatinga/MG, que está no mercado desde 2006. A distribuidora atende hipermercados, supermercados, mercearias hortifrutis em geral e também realiza vendas no CEASA de Caratinga e no mercado do produtor de Ipatinga.

Frente aos itens levantados, a questão problema a ser trabalhada neste projeto foi:

## **QUAIS REQUISITOS DEVEM SER UTILIZADOS NA MODELAGEM DE UM SISTEMA INFORMATIZADO DESTINADO A GESTÃO DE UMA DISTRIBUIDORA DE FRUTAS?**

### **1.2. Objetivo**

#### **1.2.1. Objetivo geral**

Definir, modelar e descrever um sistema informatizado destinado à gestão de uma distribuidora de frutas (Micro empresa).

#### **1.2.2. Objetivos específicos**

Para atingir o objetivo geral será preciso estabelecer objetivos auxiliares, sendo eles:

- Definir as atividades que serão geridas pelo Software a ser feito;
- Definir os requisitos a serem atendidos pelo Software;
- Modelar os diagramas que o Software irá possuir;
- Descrever cada diagrama.

### **1.3. Justificativa do Trabalho**

A Engenharia de Produção envolve diversas áreas e subáreas, dentre elas a Pesquisa Operacional, que em parte é definida pela Associação Brasileira de Engenharia de Produção

(ABEPRO) como “aplica conceitos e métodos de outras disciplinas científicas na concepção, no planejamento ou na operação de sistemas para atingir seus objetivos.” (ABEPRO, 2008). E a Engenharia Organizacional, que é definida como:

“conjunto de conhecimentos relacionados à gestão das organizações, englobando em seus tópicos o planejamento estratégico e operacional, as estratégias de produção, a gestão empreendedora, a propriedade intelectual, a avaliação de desempenho organizacional, os sistemas de informação e sua gestão e os arranjos produtivos.” ABEPRO (2008)

O desenvolvimento deste trabalho se dará dentro destas áreas, onde os Softwares constituem um papel importante nas empresas, no intuito de se obter um melhor resultado e controle das informações e dados processados.

A Tecnologia da Informação é uma ferramenta utilizada por gestores de empresas com diversas finalidades, como citado em Rezende (1999) apud REZENDE e ABREU, (2010, p. 179) “as empresas têm como opção a utilização de diversas tecnologias modernas, para facilitar o processo de tomada de decisão dos gestores, visando atender a sua complexidade, seu crescimento, sua modernidade, sua perenidade, sua rentabilidade e sua competitividade”.

Porém, muitas empresas desconhecem os benefícios trazidos pelas Tecnologias de Informações ou então possuem dificuldades em sua utilização. Como propõe Rezende e Abreu (2010, p 53) “o desconhecimento elementar da Tecnologia da Informação e de seus recursos tem causado muitos problemas e dificuldades dentro das empresas, principalmente para atividades ligadas a Planejamento Estratégico, Sistemas de Informação e Gestão de Tecnologia da Informação.”. O projeto ajudará o usuário a possuir um melhor controle sobre suas informações, fornecendo um modelo de Software capaz armazenar e fornecer dados que trarão uma melhor qualidade no gerenciamento da empresa, além de uma economia de tempo, gerada pela precisão e velocidade do fornecimento das informações por parte do sistema.

Os Sistemas de Informação e Tecnologias de Informação são de fundamental utilização na gestão de diversos fatores fundamentais para uma empresa como sugere ABEPRO( 2008), sendo eles:

- Gestão Estratégica e Organizacional;
- Gestão da Informação;
- Gestão da Tecnologia;
- Gestão do Conhecimento.

Segundo os autores Moraes, Terence e Filho (2004), as pequenas empresas carecem de tecnologias que lhes dêem suporte e melhorem a qualidade tanto dos seus processos decisórios estratégicos quanto da gestão estratégica de seus negócios. O projeto desenvolvido terá como objetivo suprir esta carência de tecnologia, no controle de compra e venda de produtos, de uma micro empresa, oferecendo ao usuário uma melhor qualidade no controle de suas informações, facilitando assim a gestão estratégica de seus negócios.

A escolha pela empresa Sabor Brasil se deu pela facilidade encontrada para a obtenção dos dados necessários, tendo em vista a realização de um trabalho de tamanha amplitude e necessidade de recolhimento de dados a fim de se obter um melhor resultado. Outro fator seria a dificuldade relatada pelo proprietário em gerenciar suas informações e a possibilidade de melhoria oferecida pelo estudo de caso.

A linguagem UML foi escolhida por ser a linguagem padrão para a elaboração da estrutura de um projeto de Software. Por ser uma linguagem, a UML indica como criar e ler modelos bem formulados, que permitem a compreensão de um sistema. Segundo os autores Booch, Rumbaugh e Jacobson (2005), a UML é uma linguagem destinada a visualizar, especificar, construir e documentar os artefatos de um sistema complexo de Software.

Alem disto, como afirmam Booch, Rumbaugh e Jacobson (2005) a UML, apesar de sua expressividade, não é de difícil entendimento e manuseio.

Os diagramas aqui apresentados foram desenvolvidos utilizando o Software StarUML, uma plataforma que utiliza a linguagem UML 2.0 em sua notação. Tal programa foi escolhido por apresentar a possibilidade de criação de todos os diagramas necessários neste projeto, além de ser um Software de fácil acesso e possuir uma versão gratuita.

## **2. Fundamentação Teórica**

### **2.4. Tecnologia de Informação e Sistema de Informação**

Primeiramente é importante diferenciar Tecnologia da Informação e Sistema de Informação, pois não é algo claro quais são suas diferenças e semelhanças.

Pode-se definir a primeira como:

|                           |                                                                                                                                                                                           |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Foina (2006, p. 17)       | “A Tecnologia de Informação é um conjunto de métodos e ferramentas, mecanizadas ou não, que se propõe a garantir a qualidade e pontualidade das informações dentro da malha empresarial.” |
| Rezende e Abreu (2010, p. | “Pode-se conceituar a Tecnologia de Informação como recursos tecnológicos e computacionais para geração e uso da informação.”                                                             |

|                                         |                                                                                                                                                                                                                                        |
|-----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 54)                                     |                                                                                                                                                                                                                                        |
| Cruz (1998) apud REZENDE, (2010, p. 54) | “Tecnologia da Informação pode ser todo e qualquer dispositivo que tenha capacidade para tratar dados e ou informações, tanto de forma sistêmica como esporádica, quer esteja aplicada ao produto, quer esteja aplicada no progresso.” |

**Tabela 1: Definição de Tecnologia de Informação, fonte: Pesquisa bibliográfica.**

Atraves das citações acima, entende-se por Tecnologia de Informação um conjunto de ferramentas que tenha a capacidade de tratar e processar dados e informações. Em uma era onde a informação é fundamental para o gerenciamento das empresas, as Tecnologias de Informação assumem um papel destacado no gerenciamento de tais informações.

E para os seguintes autores, Sistema de Informação pode ser definido tal como apresentado na tabela 2:

|                                                          |                                                                                                                                                                                                                                                                                |
|----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Para Zwass (1998) apud LAURINDO, (2008, p. 27)           | Define Sistema de Informação como um conjunto organizado de componentes para coletar, transmitir, armazenar e processas dados de modo a fornecer informação voltada para a ação.                                                                                               |
| Para Laudon e Laudon (2003) apud LAURINDO, (2008, p. 27) | Um Sistema de Informação seria um conjunto de componentes inter-relacionados para coletar, recuperar, processar, armazenar e distribuir informações com a finalidade de facilitar o planejamento, o controle a coordenação, a análise e a tomada de decisões nas organizações. |
| Xavier e Gomes (2000) apud TONSIG (2008, p. 40)          | Um Sistema de Informação apresenta-se como um agente que auxilia a dinâmica de mudanças organizacionais, que, juntamente com outras tecnologias e processos, pode ser fator de vantagem competitiva para os negócios da empresa                                                |

**Tabela 2: Definição de Sistema de Informação, fonte: Pesquisa bibliográfica.**

Um Sistema de Informação se caracteriza por ser um conjunto de componentes, que possuem como principal finalidade coletar, armazenar e disribuir dados e informações, com a finalidade de facilitar o planejamento e dar vantagem competitiva aos negócios da empresa.

Percebe-se uma grande semelhança entre as definições dos dois termos, como a finalidade de se processar informações e dados. Porém como demonstram autores como Alter (1992) apud LAURINDO (2008, pag. 26) há uma distinção entre Tecnologia de Informação e Sistemas de Informação. Segundo este a primeira restringe apenas aos aspectos técnicos,

enquanto que à segunda corresponderia às questões relevantes ao fluxo do trabalho, pessoas e informações envolvidas.

Segundo Bio (2008) a informática por si só não é capaz de garantir qualidades aos Sistemas de Informação. De forma análoga, a ausência de recursos tecnológicos que suportem os Sistemas de Informação pode representar diretamente perda de eficiência e desempenho. Tal afirmação demonstra a importância das Tecnologias de Informação e dos Sistemas de Informação em trabalharem em conjunto, para assim um melhor aproveitamento dos recursos.

Os Sistemas de Informação podem ser classificados de diversas maneiras segundo diferentes autores.

Alter (1992) apud LAURINDO (2008, p. 30) propôs a seguinte classificação:

- Sistemas Transacionais;
- Sistemas de Informação Gerenciais;
- Sistemas de Apoio à Decisão;
- Sistemas de Informação Executivos;
- Sistemas Especialistas;
- Sistemas de Automação de Escritórios.

Já Turban, MClean e Wetherbe (2002) apud LAURINDO (2008, p. 31) classificam os Sistemas de Informação da seguinte maneira:

- Sistemas de Processamento de Transação;
- Sistemas de Informação Gerenciais;
- Sistemas de Apoio a Decisões;
- Sistemas de Automação de Escritórios;
- Sistema de Informação Empresarial;
- Sistemas de Suporte Inteligente;
- Sistemas de Gestão do conhecimento;
- Sistemas de apoio a grupos.

Bio (2008) classifica os Sistemas de Informação em apenas dois grupos:

- Sistemas de apoio às operações;
- Sistemas de apoio à gestão.

Para K.C Laudon e J.P. Laudon (2007):



- Sistemas sob a perspectiva funcional
  - Sistemas de venda e marketing;
  - Sistemas de manufatura e produção;
  - Sistemas financeiros e contábeis;
  - Sistemas de recursos humanos;
- Sistemas sob a perspectiva de grupos usuários
  - Sistemas de processamento de transações;
  - Sistemas de informações gerenciais e sistemas de apoio à decisão;
  - Sistemas de apoio ao executivo.

As diferentes classificações apresentadas pelos autores demonstram uma falta de padronização em relação a classificação dos Sistemas de Informação, dificultando assim um melhor entendimento por parte dos usuários em relação à funcionalidade do sistema. Bio (2008) afirma que a própria natureza interdependente dos subsistemas não recomenda a tendência de classificá-los rigidamente.

Tal falta de padronização é decorrente de o Sistema de Informação ser um subsistema do “sistema empresa” como afirma Bio (2008). Pode-se observar pelas classificações dos autores, que as duas mais antigas têm uma visão mais global em suas divisões, sem um foco maior nos setores da empresa. Uma classificação mais recente, como a apresentada por Laudon e Laudon (2007), demonstra uma maior preocupação com os setores sob uma perspectiva da empresa, sendo os Sistemas de Informação classificados e divididos em seus segmentos.

## **2.5. Engenharia de Software**

Nas empresas, a Tecnologia da Informação pode ser dividida em componentes, para que possam ser utilizados, segundo Rezende e Abreu (2010) tais componentes são:

- Hardware e seus dispositivos;
- Software e seus recursos;
- Sistemas de telecomunicação;
- Gestão de dados e informações.

Sendo que neste texto será desenvolvido apenas um estudo sobre os softwares e seus recursos.

Na tabela 3 abaixo se pode ver algumas definições de Software:

|                               |                                                                                                                                                         |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| Turban, Jr. e Potter (2007)   | Software é um programa ou conjunto de programas que permite que o hardware processe os dados                                                            |
| Laudon e Laudon (2007, p. 12) | “O Software consiste em instruções detalhadas e pré programadas que controlam e coordenam os componentes do Hardware de um Sistema de Informação”       |
| Pfleeger (2004, p.1)          | “O conceito mais geral de Software compreende todo o conjunto de programas, procedimentos, dados e documentação associados a um sistema de computador.” |

**Tabela 3: Definição de Software, fonte: Pesquisa bibliográfica.**

O conceito de Software é percebido como um conjunto de programas e procedimentos, que processam dados gerando informações ao hardware. A importância e complexidade dos Softwares foram aumentando ao longo dos anos,

“As primeiras aplicações de Software dos computadores em empresas foram desenvolvidas no início da década de 1950. O Software era menos importante (e menos caro) nos sistemas de computação da época. Hoje, o Software representa uma porcentagem muito maior do custo dos sistemas de computação modernos. O preço dos Software aumentaram assustadoramente.” (TURBAN, JR. E POTTER, 2007, P. 318)

A partir desta modernização e aumento dos custos de um Software veio a necessidade de uma nova maneira de se desenvolver um Software.

Segundo Sommerville (2007, p. 3) “o conceito de Engenharia de Software foi inicialmente proposto em 1968, em uma conferencia organizada para discutir o que foi então chamado de “crise de software”. A engenharia surgiu como uma ferramenta auxiliar na construção de um Software, valorizando e tornando-os mais eficientes, como cita Tonsig (2008, p. 65) “objetos que são frutos do trabalho de engenharia, constituem-se soluções economicamente viáveis para problemas práticos, obtidas do conhecimento científico sobre um determinado domínio tecnológico, que podem partilhar soluções anteriores.”

A Engenharia de Software pode ser definida da seguinte maneira:

|                          |                                                                                                                                                                                            |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Sommerville (2007, p. 5) | “Engenharia de Software é uma disciplina de engenharia relacionada a todos os aspectos de produção de software, desde os estágios iniciais de especificação do sistema até sua manutenção, |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                                 |                                                                                                                                                                                                                                                    |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                 | depois que este entrar em operação.”                                                                                                                                                                                                               |
| Pressman (2010, tradução nossa) | Engenharia de Software é a utilização dos recursos e princípios da engenharia em função de obter um Software confiável e que funcione eficientemente em uma máquina real.                                                                          |
| Pfleeger (2004)                 | A Engenharia de Software é responsável por agrupar os conhecimentos teóricos e funções computacionais da Ciência da Computação, juntamente com o problema proposto pelo cliente a fim de se obter ferramentas e técnicas para resolver o problema. |

**Tabela 4: Definição de Engenharia de Software, fonte: Pesquisa bibliográfica.**

As definições apresentadas demonstram a clareza do significado de Engenharia de Software, pois todas seguem a mesma linha da importância da utilização dos recursos da engenharia no propósito de desenvolvimento de um Software.

O Software sendo desenvolvido através de processos de engenharia tornou-se um produto mais confiável e eficiente, como afirma Tonsig (2008, p. 65) “o software como fruto de engenharia é o grande objetivo que se busca, a fim de conseguir obter-se um produto confiável, funcionando de uma maneira eficiente e economicamente viável.”

O processo de Engenharia de Software pode contemplar diferentes etapas como demonstra Pfleeger (2004), que distribui o desenvolvimento de um Software nas seguintes atividades:

- Análise e definição dos requisitos;
- Projeto do sistema;
- Projeto do programa;
- Escrever os programas;
- Testes das unidades;
- Testes de integração;
- Testes do sistema;
- Entrega do sistema;
- Manutenção.

Já para Magela (2006) o desenvolvimento de um Software se dá pelas seguintes etapas:

- Análise de requisitos;
- Análise de negócio;

- Análise de sistema;
- Projeto de sistema;
- Implementação;
- Teste de sistema;
- Implantação e distribuição de Software;
- Avaliação, qualidade e melhoria.

E para Sommerville (2007) o processo de criação de um Software orientado a objetos é constituído pelos seguintes passos:

- Compreender e definir o contexto e os modos de uso do sistema;
- Projetar a arquitetura do sistema;
- Identificar os objetivos principais do sistema;
- Desenvolver modelos de projeto;
- Especificar interfaces de objetos.

Apesar das diferenças de etapas na construção de um Software apresentadas pelos autores, percebe-se que a linha seguida é semelhante, sendo a etapa inicial caracterizada pela realização de análises, tanto de requisitos quanto de sistemas, onde serão definidas as necessidades que o projeto deverá atender, tanto em caráter operacional quanto em funcional. Nesta etapa é fundamental também a definição dos objetivos principais do sistema, como demonstra Sommerville (2007).

Em seguida, segundo Pfleeger (2004), deverão ser desenvolvidos os modelos do projeto, podendo assim se ter uma melhor visão daquilo que será produzido e se atenderá as necessidades propostas pelo futuro usuário. Tendo os modelos desenvolvidos, o projeto do sistema e programa deverão ser realizados, sendo aqui desenvolvido o produto.

A penúltima etapa, ainda segundo Pfleeger (2004), seriam os testes a serem realizados, etapa também de suma importância para a consolidação do projeto e eliminação dos erros. Finalizando, com a implantação do Software e a realização de manutenções, como afirmar Magela (2006).

Neste projeto serão seguidas as etapas de análise e definição dos requisitos e desenvolvimento dos modelos do projeto.

## 2.6. Engenharia de Requisitos

A Engenharia de Requisitos é uma etapa inicial do processo de Engenharia de Software. Para Magela (2006, p. 2) “requisito é o conjunto de sentenças condicionadas pelos processos e pela política de negócio da empresa que visam a definir as funcionalidades que devem estar presentes em um Software.” E segundo Sommerville (2007, p. 79), “os requisitos de um sistema são descrições dos serviços fornecidos pelo sistema e as suas restrições operacionais.”, o que demonstra que os requisitos têm a função de definir o que o Software deverá fazer.

Magela (2006) demonstra a importância da análise dos requisitos na figura 1 – pirâmide de propagação de erro, apresentada abaixo, onde os números são considerados valores monetários.



Figura 1: Pirâmide de propagação de erro, fonte: Magela (2006, p.13)

Segundo Magela (2006) a pirâmide demonstra que uma omissão ou especificação errada de um requisito pode custar posteriormente 20 vezes mais, se descoberto na fase de implementação do Software, ou então 200 vezes mais na fase de manutenção. O que demonstra a importância de se ter uma análise dos requisitos realizada de maneira correta.

Autores possuem diferentes opiniões em relação aos requisitos necessários pelo sistema. Para Pfleeger (2004), os requisitos são de dois tipos, funcionais e não funcionais. Segundo Magela (2006) são necessários os requisitos de usuário, do Software e de ambiente. Já na

visão de Sommerville (2007), podem-se classificar os requisitos como de usuário ou de sistema.

Segundo Sommerville (2007), os requisitos de sistema definem as funções, serviços e as restrições operacionais do sistema. Ele deve definir exatamente o que será implementado.

E os requisitos de usuário segundo Magela (2006) são os requisitos definidos diretamente pelo usuário em linguagem natural. Representam o que o sistema deverá ser capaz de executar sob o ponto de vista do usuário. O requisito do usuário deve focar nos recursos a serem fornecidos, utilizando a linguagem natural, sem termos técnicos de Software.

Sommerville (2007) divide os requisitos de sistema de Software em três, funcionais, não funcionais e de domínio. Os primeiros, segundo Magela (2006) são os que definem o que o Software deverá fornecer e seu comportamento durante a interação com o ambiente externo ao Software que está sendo produzido. Os segundos, para Sommerville (2007, p. 80) “são restrições sobre os serviços ou as funções oferecidas pelo sistema. Eles incluem restrições de tempo, restrições sobre o processo de desenvolvimento e padrões.” E os requisitos de domínio são descritos por Sommerville (2007, p. 81) “são requisitos provenientes do domínio da aplicação do sistema e que refletem as características e as restrições desse domínio. Podem ser requisitos funcionais ou não funcionais.”

## **2.7. Unified Modeling Language (UML)**

### **2.7.1. Definição**

Primeiramente é importante salientar a importância da modelagem. Segundo os autores Booch, Rumbaugh e Jacobson (2005, p. 3), “uma empresa de Software bem sucedida é aquela que fornece Software de qualidade e capaz de atender às necessidades dos respectivos usuários.” Estes autores defendem que a modelagem tem papel fundamental nas atividades que levam a implantação de um bom Software.

Ainda segundo Booch, Rumbaugh e Jacobson (2005, p. 6) “a modelagem é uma técnica de engenharia aprovada e bem aceita. Construímos modelos de arquitetura de casas e de grandes prédios para auxiliar seus usuários a visualizar qual será o produto final.” Podendo-se dizer que a modelagem é uma simplificação da realidade e que a construção de modelos se dá no intuito de compreender melhor o sistema a ser desenvolvido.

A figura 2 - Como realmente funciona um projeto. A seguir demonstra a dificuldade encontrada pelas empresas em entregar o produto desejado pelo cliente. Facilitar este processo é uma das funções da modelagem.

### Como realmente funciona um projeto

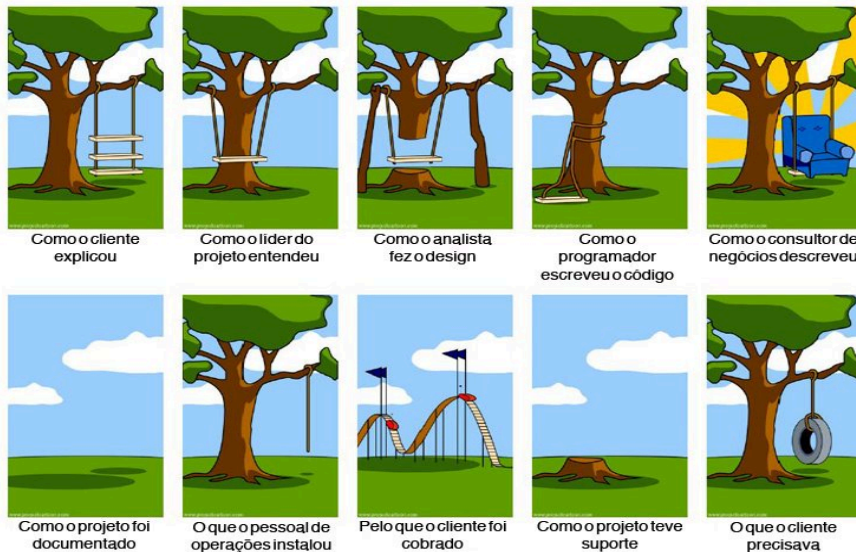


Figura 2: Como Realmente funciona um projeto, fonte: Lima (2011, p. 173)

O autor Lima (2011, p. 27) defende que “na Tecnologia da Informação, a construção de modelos exige uma linguagem de modelagem que inclua elementos visuais para expressar conceitos e uma notação simples, mas poderosa para esses elementos”.

Para Booch, Rumbaugh e Jacobson (2005) a UML é uma linguagem padrão para a elaboração da estrutura de projetos de Softwares, sendo somente uma parte de um método para desenvolvimento de Software. Já para Melo (2010), UML é uma linguagem para especificação, visualização, construção e documentação de artefatos de sistemas de Software. As afirmações dos dois autores demonstram que a UML é a linguagem padrão utilizada para a modelagem de projetos de Softwares.

#### 2.7.2. Histórico

Com o crescimento da utilização dos Softwares e as melhores oportunidades oferecidas pelos hardwares, segundo Lima (2011, p. 28) “entre 1970 e 1980, muitos metodologistas desenvolveram linguagens (ou métodos) para modelar e especificar sistemas orientados a objetos, alguns influenciados por técnicas convencionais, como, por exemplo, o Modelo Entidade-Relacionamento.” O problema se deu pela quantidade de linguagens desenvolvidas e a falta de padronização apresentada por estas. Segundo os autores Lima (2011) e Melo (2010) existiam mais de cinquenta linguagens diferentes em meados da década de 90, período este que ficou conhecido como “guerra dos métodos”.

Ainda segundo os autores Lima (2011) e Melo (2010) as principais linguagens apresentadas naquela época eram: Booch (de Grady Booch), OMT-2 (de Jim Rumbaugh) e OOSE (de Ivar Jacobson).

Lima (2011) também cita que em outubro de 1995 os três autores se juntaram e unificaram suas linguagens em uma só, criando então o Método Unificado na versão 0.8 e em 1996, o Método Unificado passou a se chamar UML (Unified Modeling Language) e foi lançado na versão UML 0.91.

A autora Melo (2010) afirma que já em 1996, a UML era vista pelas organizações como ótima estratégia para seus negócios.

Em janeiro de 1997 foi lançada uma nova versão 1.0 da UML e que segundo Costa (2001) apenas em novembro do mesmo ano, ela foi adotada como padrão pelo Object Management Group (OMG), uma organização internacional, fundada em 1989, que promove a teoria e prática da tecnologia orientada a objeto em desenvolvimento de sistemas. Em seguida, segundo Melo (2010), diversas versões foram desenvolvidas. Já em julho de 1998 foi apresentada a versão 1.2 da UML, no mesmo ano a UML 1.3. Em maio de 2001, a UML 1.3 foi aprovada, sendo substituída no mesmo ano pela UML 1.4. Segundo afirma Melo (2010) até então eram utilizados os seguintes diagramas: de atividades, de caso de uso, de classes, de colaboração, de componentes, de gráfico de estado, de implantação, de objetos e de sequência.

Em 2005, segundo Guedes (2011) foi lançada a versão UML 2.0, que trouxe grandes inovações na linguagem. Primeiramente, segundo Melo (2010), a nova versão trouxe um forte conceito em herança de modelo além da introdução de 4 novos diagramas: o temporal, de visão geral, de estrutura composta e de pacotes.

E já em fevereiro de 2009, era utilizada a UML 2.2 segundo Melo (2010), versão que introduziu o diagrama de perfil, e que depois foi substituída pela UML 2.3 ainda no mesmo ano. A última versão disponível, segundo OMG (2013) é o UML 2.5, lançado em outubro de 2012.

### 2.7.3. Análise de Software de UML

Por se tratar apenas de uma linguagem a UML precisa de Softwares que sirvam como plataformas para o desenvolvimento dos diagramas. No mercado, existem diversos programas que funcionam desta maneira, e cabe ao desenvolvedor escolher aquele que melhor se adapta a suas necessidades e atenda a seus requisitos.

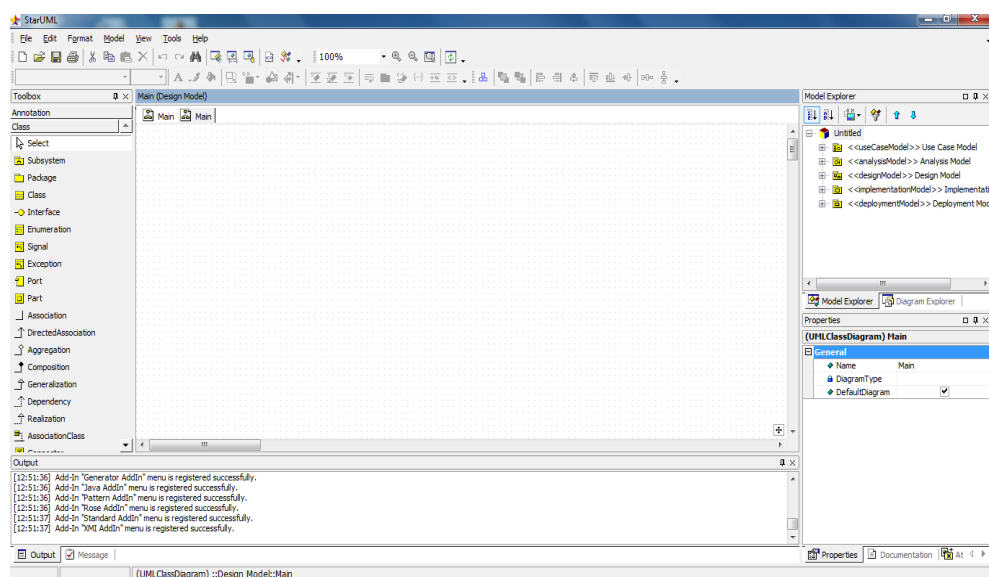


Alguns destes Softwares serão apresentados na tabela 05 a seguir, mostrando as suas especificações e sistemas necessários:

| Nome                    | Versão da UML utilizada | Espaço em disco rígido necessário | Sistema operacional                        | Língua | Preço                 |
|-------------------------|-------------------------|-----------------------------------|--------------------------------------------|--------|-----------------------|
| StarUML<br>5.0.2.1570   | 2.0                     | 150 MB                            | Windows XP ou melhor, Linux ou Mac         | Inglês | Gratuito              |
| Enterprise Architect 10 | 2.4                     | 175 MB                            | Windows 2000 ou melhor, Linux ou Intel Mac | Inglês | US \$135-<br>US \$849 |
| ArgoUML<br>0.21.1       | 1.4                     | 17 MB                             | Windows XP ou melhor                       | Inglês | Gratuito              |
| Violet UML<br>0.21.1    | 2.0                     | 1,7 MB                            | Windows XP ou melhor                       | Inglês | Gratuito              |

**Tabela 5: Softwares de UML, fonte: Pesquisa bibliográfica**

O StarUML é um Software gratuito e disponível na web. Utiliza a versão 2.0 da UML e possui um manual de instruções que auxilia o usuário quando necessário. É possível desenvolver oito tipos diferentes de diagramas através deste sistema, que possui uma interface dinâmica como mostrado na figura 3 - StarUML, que se segue.



**Figura 3: StarUML, fonte: Autoria própria**

O Enterprise Architect é um Software mais completo, que possui uma versão gratuita por trinta dias. Porém, após esta data, é necessária a compra de uma licença, que varia entre

\$135 e \$849 dólares americanos. Utiliza a versão 2.4 da UML. É capaz de desenvolver todos os quatorze diagramas e possui uma plataforma mais completa como demonstra a figura 4 - Enterprise Architect, abaixo.

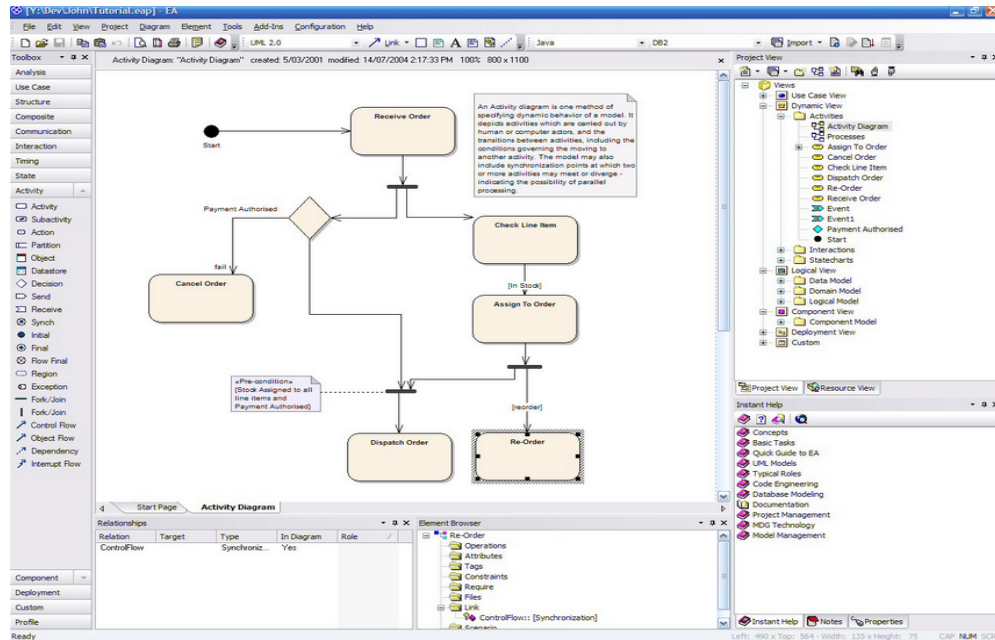


Figura 4: Enterprise Architect, fonte: Autoria própria

O Software ArgoUML é um programa mais simples, de fácil acessibilidade e gratuito. É implementado em Java e utiliza uma versão mais antiga da UML, a 1.4. Possui manual de instruções e da suporte para se desenvolver sete diagramas diferentes. A interface do Software é apresentada na figura 5 – ArgoUML, a seguir.

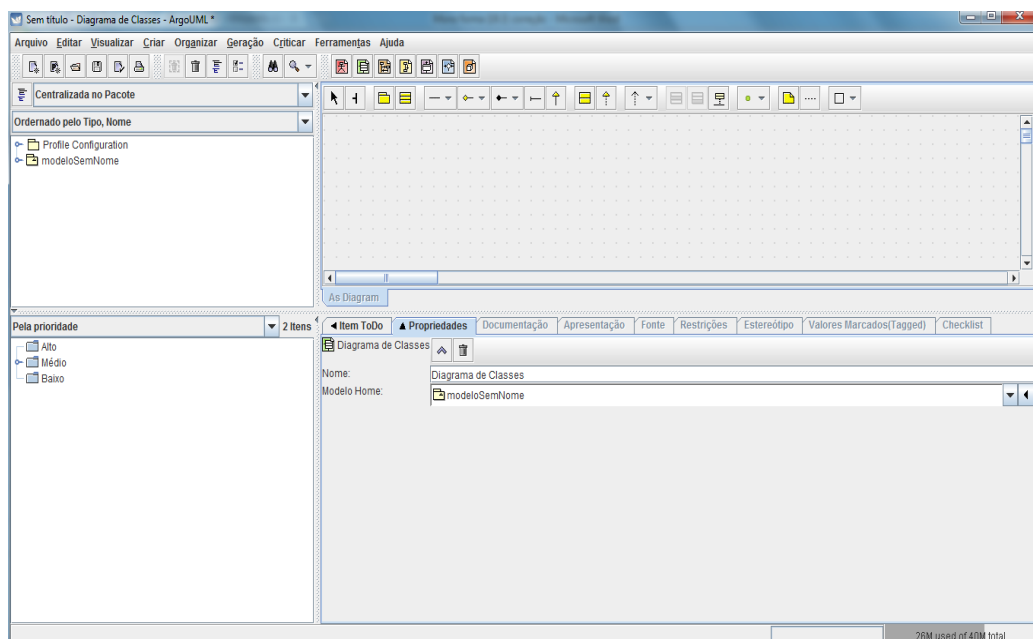
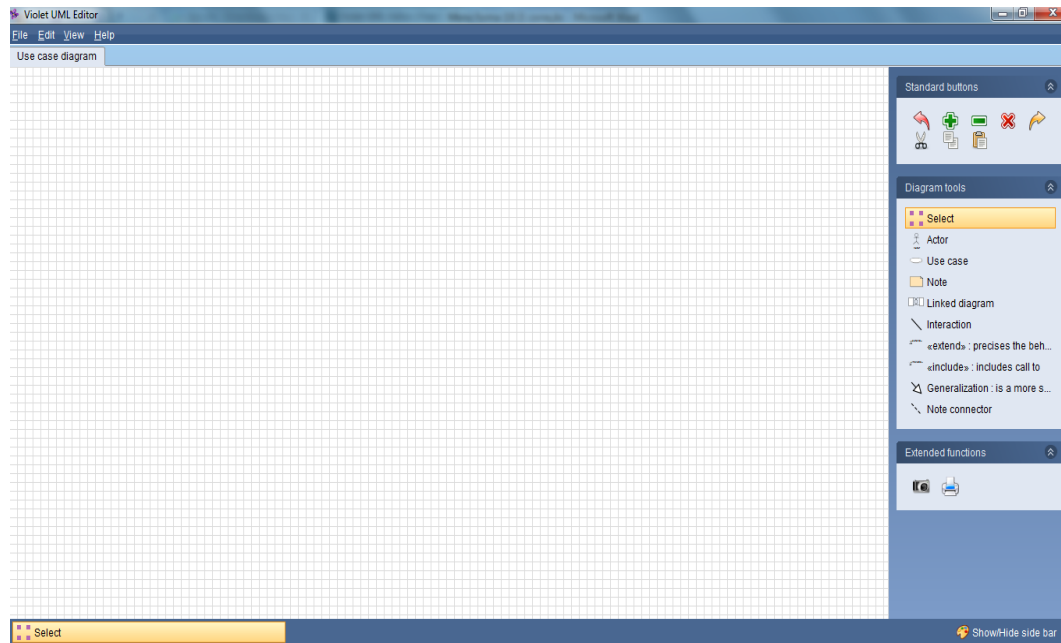


Figura 5: ArgoUML, fonte: Autoria própria

O Violet UML também é um Software simples, de fácil acesso e que trabalha em Java e por isto não precisa ser instalado para ser utilizado. O programa possui manual de instruções disponível em uma pagina da internet, dificultando assim, o seu manuseio quando não conectado a internet. O Software possui a capacidade de desenvolvimento de seis diagramas, além de possuir uma interface simples como apresentada na figura 6 – Violet UML, a seguir.



**Figura 6: Violet UML, fonte: Autoria própria**

#### 2.7.4. Itens, Relacionamentos e Diagramas

Segundo Booch, Rumbaugh e Jacobson (2005) a construção da UML é dividida em três blocos: itens, relacionamentos e diagramas.

Os itens na UML podem ser divididos em quatro tipos:

- Estruturais, que são os elementos físicos e conceituais do modelo, as partes mais estáticas;
- Os comportamentais representam o comportamento no tempo e no espaço. São as ações realizadas no modelo;
- Os itens de agrupamento são as partes organizacionais do modelo, sendo blocos em que o modelo pode se decompuser;
- Por último os itens anotacionais sendo a parte explicativa do modelo, através de comentários e observações sobre qualquer elemento do modelo.

Os relacionamentos divididos em quatro tipos:

- Dependência, que representa o relacionamento entre dois elementos da modelagem, sendo que a alteração em um elemento implicará em mudança no outro;

- Associação é a ligação entre classes, onde o relacionamento ocorre entre objetos destas classes;
- Generalização é um relacionamento de especialização, entre um elemento mais genérico e outro mais específico, sendo o primeiro conhecido como pais e o segundo como filhos;
- Realização é um relacionamento entre uma especificação e sua implementação.

E por último os diagramas da UML, que são em uma totalidade de 14 diagramas, sendo eles:

- Diagrama de classes, que tem como finalidade apresentar um conjunto de classes, interfaces e colaborações mostrando o relacionamento entre eles;
- Diagrama de objetos, que exhibe os objetos e seus relacionamentos no modelo;
- Diagrama de componentes, que mostra as dependências entre os componentes do modelo, apresentando suas interfaces;
- Diagrama de estruturas compostas, que mostra a composição de uma estrutura;
- Diagrama de caso de uso, que mostra um conjunto de casos de usos e seus atores e seus relacionamentos;
- Diagrama de seqüências, que mostra a ordenação temporal das mensagens;
- Diagrama de comunicação, que mostra os objetos, seus relacionamentos e o fluxo de mensagem entre eles;
- Diagrama de gráfico de estados, que mostra as ações ocorridas em resposta ao recebimento de eventos;
- Diagrama de atividades, que mostra as ações e atividades no fluxo em que são realizadas;
- Diagrama de implantação, que mostra a configuração em tempo de execução e seus componentes;
- Diagrama de pacote, que mostra a decomposição do modelo em unidades organizacionais e suas dependências;
- Diagrama de temporização, que mostra a mudança de estado dos objetos no tempo real;
- Diagrama de visão geral da interação, que mostra de uma forma geral o fluxo de controle dentro do sistema.

Recentemente, segundo Lima (2011) mais um diagrama foi incluído, o diagrama de perfil.

Segundo Booch, Rumbaugh e Jacobson (2005) os diagramas são meios utilizados para a visualização de blocos de construção básicos, como classes, interfaces, colaborações,

componentes, nós, dependências, generalizações e associações. O diagrama tem como finalidade ajudar na compreensão do sistema a ser desenvolvido e cada um oferece uma visão diferente acerca dos elementos que constituem esse sistema.

Por tais motivos, é muito importante a escolha do conjunto de diagramas que será utilizados na modelagem de um Software. Ainda segundo os autores Booch, Rumbaugh e Jacobson (2005) alguns diagramas contribuem para a visualização de partes estáticas, sendo conhecidos como diagramas estruturais:

- Diagrama de classes;
- Diagrama de objetos;
- Diagrama de componentes;
- Diagrama de estruturas compostas;
- Diagrama de implantação.

E outros diagramas contribuem para visualizar as partes dinâmicas de um sistema, sendo conhecidos como diagramas comportamentais:

- Diagrama de caso de uso;
- Diagrama de seqüências;
- Diagrama de comunicação;
- Diagrama de gráfico de estados;
- Diagrama de atividades.

Para a modelagem deste sistema foram selecionados os seguintes diagramas: diagrama de classes, diagrama de objetos, diagrama de caso de uso, diagrama de atividades e diagrama de implantação. A seguir tais diagramas serão mais bem descritos, a fim de se obter um melhor entendimento de seus funcionamentos demonstrando o porquê, de sua escolha.

#### 2.7.4.1. Diagrama de classes

O diagrama de classes é fundamental na modelagem de um Software, como demonstra Melo (2010), afirmando que ele é a estrela principal de um sistema orientado a objetos. Tal diagrama tem como principal finalidade detalhar as classes pertencentes ao modelo e identificar seus relacionamentos.

Segundo Booch, Rumbaugh e Jacobson (2005, p. 51) “classe é uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica.” Cada classe possui seus atributos e operações. Os atributos representam alguma propriedade da classe na qual ele está inserido, uma classe não necessariamente precisa ter

atributos. Já uma operação é uma atividade que pode ser realizada por um objeto da classe na qual esta operação está inserida.

Como afirmam os autores Booch, Rumbaugh e Jacobson (2005) e Melo (2010), a maioria das classes não trabalham sozinhas, elas precisam se relacionar, colaborando uma com a outra. Os primeiros autores defendem a ideia de que existem três tipos de relacionamentos na modelagem orientada a objetos: dependências, generalizações e associações. A segunda autora inclui ainda mais um relacionamento nesta lista, o de realização.

Um exemplo de uma classe pode ser visto na Figura 7- Classe, a seguir:

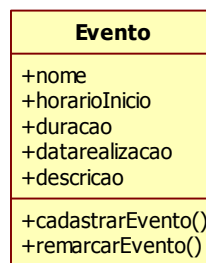


Figura 7: Classe, fonte: Melo (2010, p.105)

Um exemplo de um diagrama de classe pode ser visto na figura 8 – Diagrama de classe a seguir:

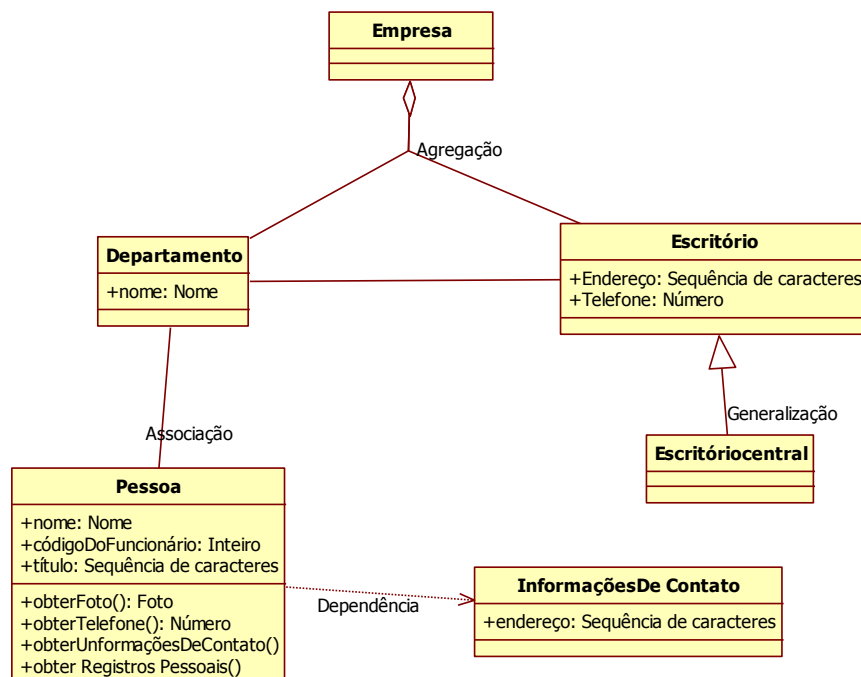


Figura 8: Diagrama de classe, fonte: Adaptado de Booch, Rumbaugh e Jacobson (2005)

#### 2.7.4.2. Diagrama de objetos

Os autores Booch, Rumbaugh e Jacobson (2005) e Melo (2010) afirmam que o diagrama de objetos representa uma instância de itens contidos em diagramas de classes, fazendo uma visão estática do projeto, modelando o sistema em determinado momento do processo do sistema.

Para Booch, Rumbaugh e Jacobson (2005), a definição de diagrama de objetos é “diagrama que mostra um conjunto de objetos e seus relacionamentos em um ponto do tempo. Graficamente, o diagrama de objetos é uma coleção de vértices e arcos”. Já para Melo (2010) “um diagrama de objetos consiste numa instância do diagrama de classes, no qual para cada classe tem-se um objeto (sua instância) em um determinado ponto do tempo.

Tendo em vista a Figura 8 – Diagrama de Classe, pode ser visto a seguir, na Figura 9 – Diagrama de objetos, um exemplo de tal diagrama, representando uma instância de itens contidos em um diagrama de classe.

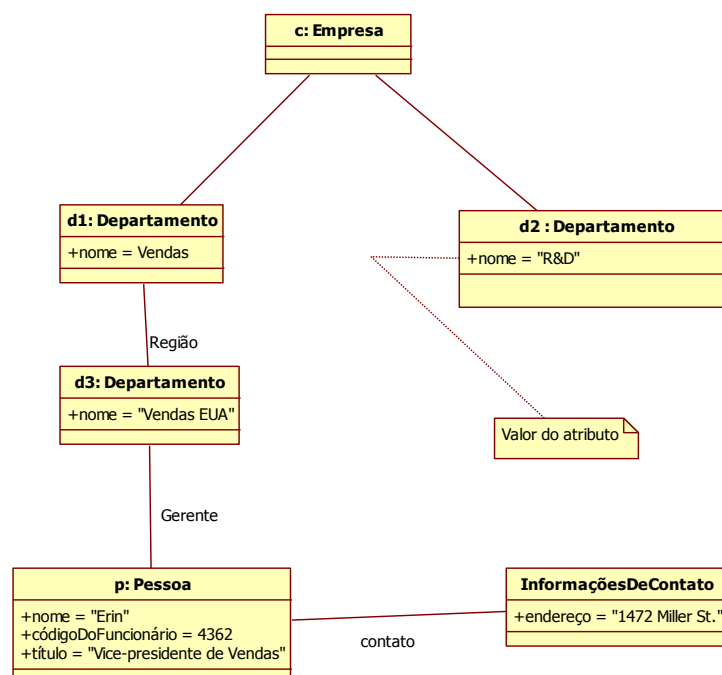


Figura 9: Diagrama de objetos, fonte: Adaptado de Booch, Rumbaugh e Jacobson (2005)

#### 2.7.4.3. Diagrama de caso de uso

Primeiramente é importante saber a definição de casos de uso, que segundo Melo (2010, p. 57) “um caso de uso descreve uma sequência de ações que representam um cenário principal (perfeito) e cenários alternativos, com o objetivo de demonstrar o comportamento de um sistema (ou parte dele), através de interações com autores.” E ainda segundo Booch, Rumbaugh e Jacobson (2005, p. 231) “um caso de uso é uma descrição de um conjunto de

seqüências de ações, inclusive variantes, que um sistema executa para produzir um resultado de valor observável por um ator.”

Para os autores Lima (2011) e Booch, Rumbaugh e Jacobson (2005) o diagrama de casos de uso tem um papel central na modelagem do comportamento de um sistema, representando toda a funcionalidade de um sistema, evidenciando interações externas com outras entidades.

Os autores Booch, Rumbaugh e Jacobson (2005) definem “um diagrama de caso de uso como um diagrama que mostra um conjunto de casos de uso e atores e seus relacionamentos.”

Um exemplo de um diagrama de caso de uso pode ser visto na Figura 10 – Diagrama de caso de uso, a seguir:

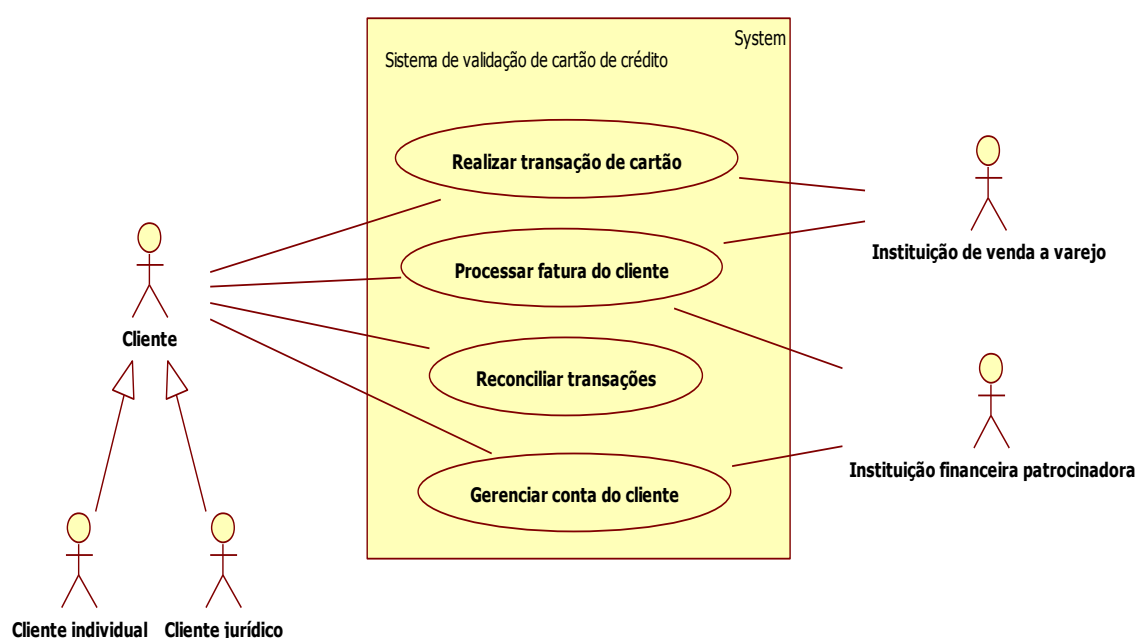


Figura 10: Diagrama de caso de uso, fonte: Booch, Rumbaugh e Jacobson (2005, p. 246)

#### 2.7.4.4. Diagrama de atividades

Para Lima (2011, p. 260) “o diagrama de atividades permite modelar o comportamento do sistema, denotando os caminhos lógicos que um processo pode seguir. Ele é um dos diagramas que compõem a visão dinâmica da UML.”

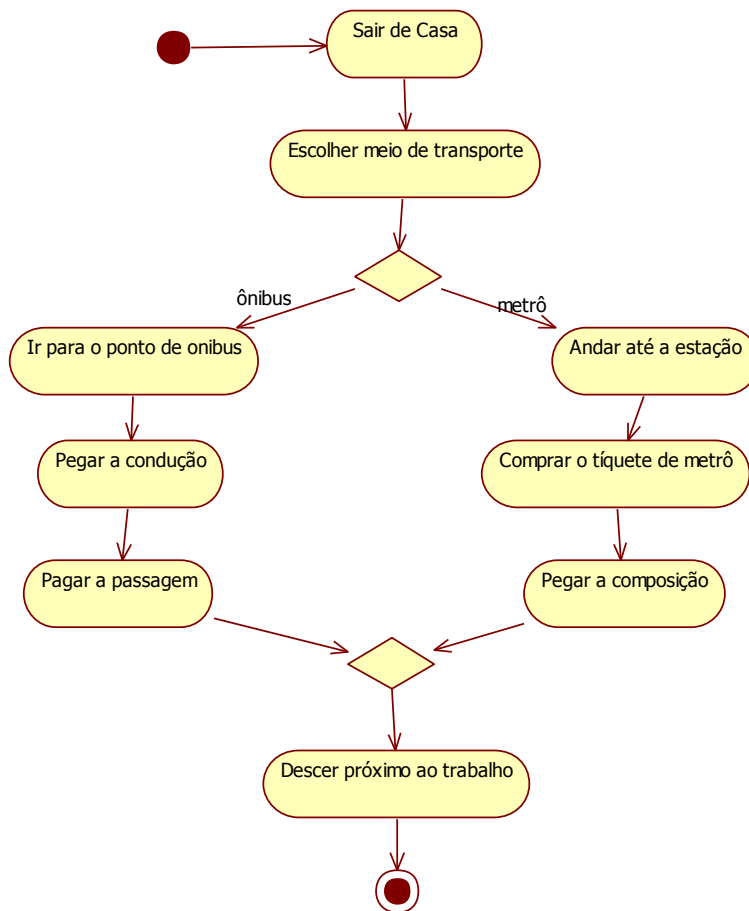
O diagrama de atividades é um modelo gráfico que analisa a estrutura e as etapas seqüenciais em um processo computacional, dando ênfase ao fluxo de controle de uma etapa para outra.

Autores definem um diagrama de atividade da seguinte maneira:



“um diagrama de atividades mostra o fluxo de uma atividade para outra. Uma atividade é uma execução em andamento não-atômica em uma máquina de estados. As atividades efetivamente resultam em alguma ação, formada pelas computações executáveis atômicas que resultam em uma mudança de estado do sistema ou o retorno de um valor. As ações abrangem a chamada a outras operações, enviando um sinal, criando ou destruindo um objeto ou alguma computação pura, como o cálculo de uma expressão.” Booch, Rumbaugh e Jacobson (2005, p. 270)

Um exemplo de diagrama de atividades pode ser observado na Figura 11 – Diagrama de atividades, a seguir:



**Figura 11: Diagrama de atividades, fonte: Melo (2010, p. 200)**

#### 2.7.4.5. Diagrama de implantação

O diagrama de Implantação segundo Booch, Rumbaugh e Jacobson (2005) e Melo (2010) mostra a estrutura de nós nos quais os componentes e artefatos são implantados. Ele é empregado para a modelagem da visão estática da implantação de um sistema. E permite definir a arquitetura de execução dos sistemas que representam a designação de artefatos de Software.

Para Lima (2011, p. 322) “o diagrama de implantação especifica o conjunto de elementos usado para definir a arquitetura do sistema e que representa a distribuição dos artefatos da aplicação pelos nós da tipologia de hardware em que o sistema será executado.” Já para Booch, Rumbaugh e Jacobson (2005, p. 413) “o diagrama de implantação é um diagrama que mostra a configuração de nós de processamento em tempo de execução e os artefatos que neles existem.”

### **3. Metodologia**

Este capítulo tem como objetivo descrever a metodologia a ser utilizada no desenvolvimento do projeto final de graduação, visando à concretização dos objetivos propostos.

#### **3.4. Conceitos de Pesquisa**

Segundo Ribeiro e Zabadal (2010), entende-se por pesquisa, uma busca a geração de conhecimento, e Wazlawick (2008) define pesquisa científica, como sendo uma busca por produção de conhecimento novo.

Um projeto final de graduação consiste em uma pesquisa científica, que segundo Chinneck (1988) apud Wazlawick (2008), deve apresentar uma contribuição original de conhecimento.

Por tais motivos, este projeto se trata de uma pesquisa científica, que busca encontrar uma solução para problema encontrado através da metodologia proposta.

#### **3.5. Abordagem**

Na Engenharia de produção, segundo Ganga (2012) e Miguel (2012) pode-se classificar a abordagem do problema de pesquisa de maneira qualitativa e quantitativa. O ato de mensurar variáveis de pesquisa é a característica mais marcante da abordagem quantitativa (MIGUEL, 2012). E segundo Ganga (2012, p. 210) “o papel do pesquisador numa pesquisa qualitativa é obter informações do fenômeno segundo a visão dos indivíduos, bem como observar e coletar evidências que possibilitem interpretar o ambiente em que a problemática ocorre.”

O que não impede uma pesquisa de caráter qualitativo de possuir abordagens de quantificação de variáveis como afirma Miguel (2012, p. 52) “a característica distintiva, em contraste com a pesquisa quantitativa, é a ênfase na perspectiva do indivíduo que está sendo estudado.”

Considerando os critérios apresentados, este estudo pode ser classificado como de abordagem qualitativa, por apresentar uma ênfase na perspectiva do indivíduo que está sendo estudado e buscar interpretar as evidências do ambiente em que a problemática ocorre.

### **3.6. Método de Pesquisa**

Tendo definido o objetivo geral do projeto, é preciso definir então o método a ser utilizado, que segundo Wazlawick (2008, p. 40) “o método consiste na sequência de passos necessários para demonstrar que o objetivo proposto foi atingido, ou seja, se os passos definidos no método forem executados, os resultados obtidos deverão ser convincentes.”

E também segundo Ganga (2012) a escolha do método é de suma importância para a condução do projeto, pois assegura que o novo conhecimento gerado estará consonante com as exigências do cliente.

Segundo Ganga (2012) os seguintes métodos podem ser utilizados em uma pesquisa científica:

- Pesquisa bibliográfica;
- Desenvolvimento teórico conceitual;
- Pesquisas experimentais;
- Survey;
- Modelagem e simulação;
- Estudo de caso;
- Pesquisa-ação.

E segundo Miguel (2012):

- Survey;
- Estudo de caso;
- Modelagem;
- Simulação;
- Estudo de campo;
- Experimento;
- Teórico / conceitual.

No entanto, não será detalhada a diferença entre tais metodologias, pois elas não terão influencia no desenvolvimento deste projeto. Este trabalho realizará um estudo de caso.

Segundo Miguel (2012), “o estudo de caso é um trabalho de caráter empírico que investiga um dado fenômeno dentro de um contexto real contemporâneo por meio de análise aprofundada de um ou mais objetos de análise.” E para Ganga (2012) “o papel de um pesquisador, num estudo de caso, é obter informações do fenômeno segundo a visão dos indivíduos, bem como observar e coletar evidências que possibilitem interpretar o ambiente em que a problemática ocorre.”

## **4. Estudo de caso**

Este capítulo visa apresentar o estudo de caso da modelagem de um Software a ser implantado em uma distribuidora de frutas. Para isto será indicado os requisitos que serão necessários para o funcionamento deste sistema, e em seguida a apresentação dos diagramas demonstrando como deverá funcionar e apresentar o sistema a ser desenvolvido.

O estudo foi desenvolvido na micro empresa Sabor Brasil, que tem sede na cidade de Ipatinga/MG. A empresa atua no ramo de distribuição de frutas, trabalhando apenas com bananas. O trabalho contou com a colaboração do proprietário, que definiu quais seriam as necessidades a serem preenchidas pelo sistema.

A empresa possui oito funcionários e vende uma média de mil caixas de banana por semana, sendo comercializadas com mais de cem clientes. Os produtos são adquiridos de um grupo menor de fornecedores, atualmente a empresa trabalha com menos de dez fornecedores.

O Software a ser desenvolvido tem como finalidade ajudar no controle das compras e vendas de produtos da empresa, além de controlar os dados dos clientes e fornecedores da empresa.

### **4.4. Análise de requisitos**

Nesta seção serão apresentados os requisitos funcionais e não funcionais do Software a ser desenvolvido, com a finalidade de demonstrar as tarefas a serem desenvolvidas pelo sistema.

#### **4.4.1. Requisitos funcionais**

Somente o administrador será capaz de logar no sistema. Para isto, ele deve se cadastrar ao habilitar o sistema e seu login será sempre feito através de uma senha.

Na tela inicial do programa, o usuário terá a opção de ir a todas as demais operações do sistema (fornecedores, compra, venda, produtos, clientes,), além da opção de sair do sistema.

Na opção de compra, o usuário poderá cadastrar novas compras, além de atualizá-las, pesquisá-las e verificar relatórios. Cada compra deverá ter os seguintes atributos: fornecedor,

produto, quantidade, valor total, valor do quilo, forma de pagamento e data, assim que cadastrada deve ser salva e armazenada pelo sistema.

Ao se cadastrar uma nova compra, o programa deverá fornecer uma visualização de todos os fornecedores e produtos cadastrados, a fim de facilitar o preenchimento por parte do usuário.

As funções na parte de vendas serão semelhantes à de compras. Assim sendo, o usuário poderá cadastrar novas vendas, além de atualizá-las, pesquisá-las e verificar relatórios. Tendo cada venda os seguintes atributos: cliente, produto, quantidade, valor total, valor do quilo, forma de pagamento e data. Assim que cadastrada, deve ser salva e armazenada pelo sistema.

Ao se cadastrar uma venda deverão ser apresentados todos os clientes e produtos cadastrados.

Tanto na área de compras como de vendas, o programa deve apresentar uma lista com as últimas compras/ vendas realizadas, sendo elas classificadas primeiramente por data. Porém deverá haver um filtro para que o usuário possa pesquisar como desejar, dentre as opções fornecidas pela propriedade de cada área.

Cada compra ou venda terá o seu número de identificação, que não poderá ser repetido em outra compra e será fornecido pelo sistema, em ordem cronológica.

Os produtos também devem ser cadastrados pelo usuário, sendo necessários os seguintes dados: nome e métrica. Nesta seção, o usuário será capaz de fazer a verificação do estoque.

Na opção clientes, será possível cadastrar novos clientes, assim como alterar seus dados futuramente quando necessário. Haverá também a opção pesquisar para facilitar a busca por um determinado cliente.

Os cadastros dos clientes deverão ter os seguintes dados: responsável, endereço, telefone, celular, email, CNPJ/CPF, proprietário, empresa e nome fantasia. Cada cliente cadastrado receberá um número de identificação único, gerado em ordem cronológica.

Quanto aos fornecedores, o programa deverá cadastrar novos fornecedores, assim como alterar seus dados futuramente. Haverá também a opção pesquisar para facilitar a busca por um determinado fornecedor.

Os cadastros dos fornecedores deverão ter os seguintes dados: responsável, endereço, telefone, celular, email, CNPJ/CPF, proprietário e produtos. Cada fornecedor cadastrado receberá um número de identificação, gerado em ordem cronológica.

Na opção, fornecedor, o sistema exibirá um histórico com todas as compras realizadas com cada fornecedor. O mesmo acontecerá em cliente, onde o sistema também exibirá um histórico com todas as vendas do cliente.

Através das compras e vendas realizadas o sistema conseguirá realizar um controle de estoque, para que o proprietário esteja sempre informado sobre a quantidade disponível de produtos que ele possui. Ao se cadastrar ou atualizar uma nova compra ou venda o sistema deverá atualizar automaticamente o setor de estoque.

Novos clientes, fornecedores, produtos, compras e vendas que forem devidamente cadastrados serão inseridos e salvos no sistema, ganhando um número de identificação gerado em ordem cronológica para cada item, que terá o nome de id.

Ao se atualizar clientes, fornecedores, produtos, compras e vendas, o sistema deverá substituir o item atualizado, eliminando assim os dados antigos.

Não deverão ser aceitos o cadastramento e atualizações de itens com campos obrigatórios sem o devido preenchimento.

O sistema deverá ser capaz de armazenar até dois mil clientes, mil fornecedores, cinco mil compras, cinco mil vendas e 50 produtos.

#### 4.4.2. Requisitos não funcionais

O Software deve facilitar o gerenciamento do administrador em seu trabalho, para isto deve contar com um manuseio fácil e rápido, além de oferecer informações precisas.

O Software deverá ser desenvolvido na língua portuguesa brasileira, utilizando uma linguagem simples e de fácil entendimento.

A usabilidade do sistema deverá ser fácil, com o intuito de ser utilizada por pessoas de baixo conhecimento tecnológico. Os erros cometidos pelo usuário deverão ser sinalizados, além de ser oferecida a opção de ajuda, com um manual do usuário e a documentação referente ao programa.

O sistema deverá ser desenvolvido para o sistema operacional Windows, deverá trabalhar em modo off-line. Depois de instalado deverá funcionar sem a necessidade de utilização de um CD-ROM.

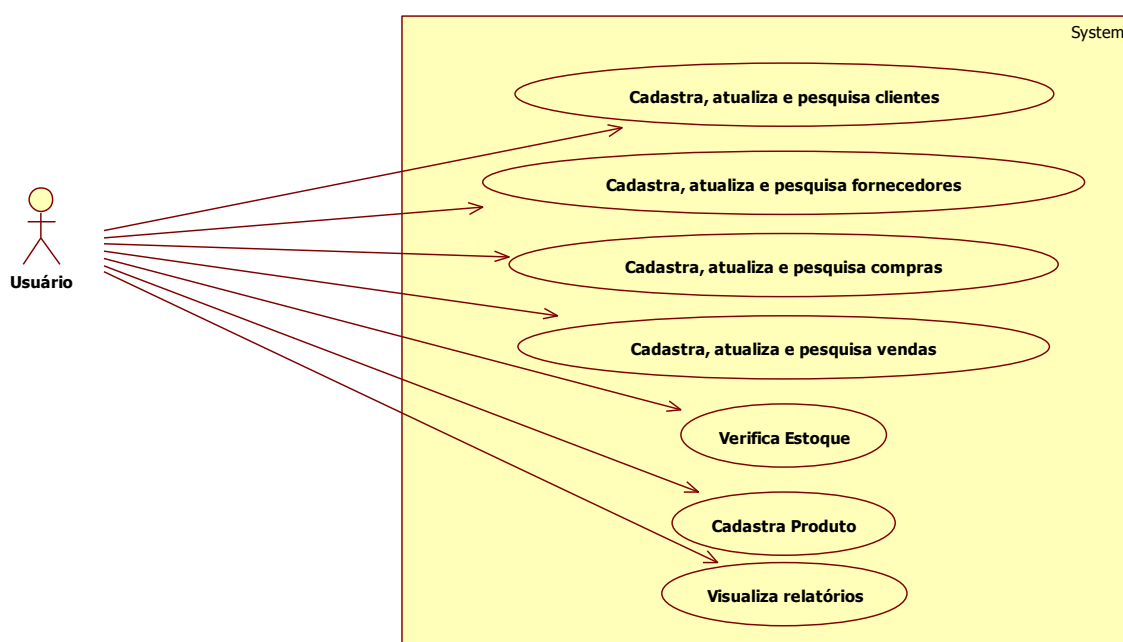
O usuário poderá criar mais de uma empresa no sistema, sendo cada empresa salva em um arquivo diferente, possibilitando assim a utilização do produto em hardwares diferentes.

## 4.5. Diagramas

A seguir serão apresentados os diagramas desenvolvidos com a finalidade de apresentar o modelo do Software desejado. Foram modelados os seguintes diagramas: Caso de uso, classes, objeto, atividade e implantação. Os diagramas apresentados foram desenvolvidos no programa StarUML e Microsoft Office Access 2007.

### 4.5.1. Diagrama de Caso de Uso

O primeiro diagrama que será apresentado será a figura 12 - Diagrama de Caso de Uso 2.



**Figura 12: Diagrama de Caso de Uso 2, fonte: Autoria própria**

Pelo diagrama, é notado que o Software terá apenas um usuário. Ele terá seus dados cadastrados ao se inicializar o programa e será gerado um login e uma senha, para o acesso. Com isto, o programa não abrirá brechas para desentendimentos, pois com apenas um cadastro, a mesma ação não será realizada várias vezes por pessoas diferentes.

Este usuário terá o controle de todas as funções que serão desempenhadas, tendo como meta manter o sistema sempre atualizado e em compatibilidade com a empresa controlada.

A primeira atividade do usuário está relacionada com os clientes. Ele poderá cadastrar novos clientes, sempre que for necessário, além de atualizar dados e pesquisar clientes quando desejar informações específicas a respeito de um cliente já cadastrado. O cadastro de novos clientes só acontecerá com todos os dados devidamente preenchidos, assim como a atualização, que apenas será aceita com todos os dados devidamente alterados. A pesquisa de

cliente será um auxílio do programa ao usuário, a fim de encontrar de maneira mais rápida a informação desejada. O usuário deverá manter as informações sobre os clientes sempre atualizadas, com o intuito de se facilitar a comunicação externa e as vendas. O principal objetivo do sistema nesta área é manter atualizado os clientes do comércio de modo que toda venda a ser realizada possa ser atribuída a um cliente.

Assim como os clientes, os fornecedores também fazem parte da uma empresa de distribuição de frutas. Para isto, o usuário será capaz de cadastrar novos fornecedores, além de atualizar dados, sempre que for necessário e buscar fornecedores, quando for preciso encontrar informações específicas de um fornecedor. O cadastro de novos fornecedores só acontecerá com todos os dados devidamente preenchidos, assim como a atualização, que apenas será aceita com todos os dados devidamente alterados. A pesquisa será um auxílio do programa ao usuário, a fim de encontrar de maneira mais rápida a informação desejada. O usuário deverá manter as informações sobre os fornecedores sempre atualizadas, com o intuito de se facilitar a comunicação externa e as compras. O principal objetivo do sistema nesta área é manter atualizados os fornecedores do comércio de modo que toda compra a ser realizada possa ser atribuída a um fornecedor.

Com fornecedores e clientes cadastrados, o usuário deverá então cadastrar seus produtos, uma parte simples, porém fundamental para o bom funcionamento do sistema. O usuário terá que cadastrar um novo produto sempre que for comercializar algum que ainda não o tenha feito. O sistema deverá manter sempre atualizado os produtos comercializados pelo comércio, com a intenção de não haverem vendas e compras de produtos não cadastrados.

Com fornecedores, clientes e produtos cadastrados, finalmente o usuário será capaz de cadastrar suas compras e vendas, os dois setores funcionaram de forma semelhante, podendo ser cadastradas novas compras e vendas, além de atualizar os dados sempre que for necessário e também será possível realizar buscas sobre compras e vendas já realizadas com a finalidade de encontrar informações específicas. O cadastro de uma nova compra ou venda só acontecerá com todos os dados devidamente preenchidos, assim como a atualização, que apenas será aceita com todos os dados devidamente alterados. A pesquisa será um auxílio do programa ao usuário, a fim de encontrar de maneira mais rápida a informação desejada. O usuário deverá manter as informações sobre as compras e vendas sempre atualizadas, com o intuito de se facilitar o seu controle de estoque. O principal objetivo do sistema nesta área é manter



atualizado o cadastro das compras e vendas realizadas pelo comercia, a fim de se manter um controle de estoque dos produtos comercializados.

Com as vendas e compras registradas, o sistema manterá um controle de estoque, sendo possível o usuário verificar este estoque sempre que desejar. O sistema deverá manter este estoque sempre atualizado com as informações transmitidas pelos dados das compras e vendas realizadas.

O administrador também será capaz de verificar relatórios sobre todas as atividades por ele desenvolvidas, com a intenção de se ter uma melhor visualização sobre os dados de interesse. O sistema deverá oferecer relatórios atualizados e de fácil visualização ao usuário.

#### 4.5.2. Diagrama de Classes

Em seguida é apresentado o próximo diagrama na figura 13– Diagrama de Classes 2.

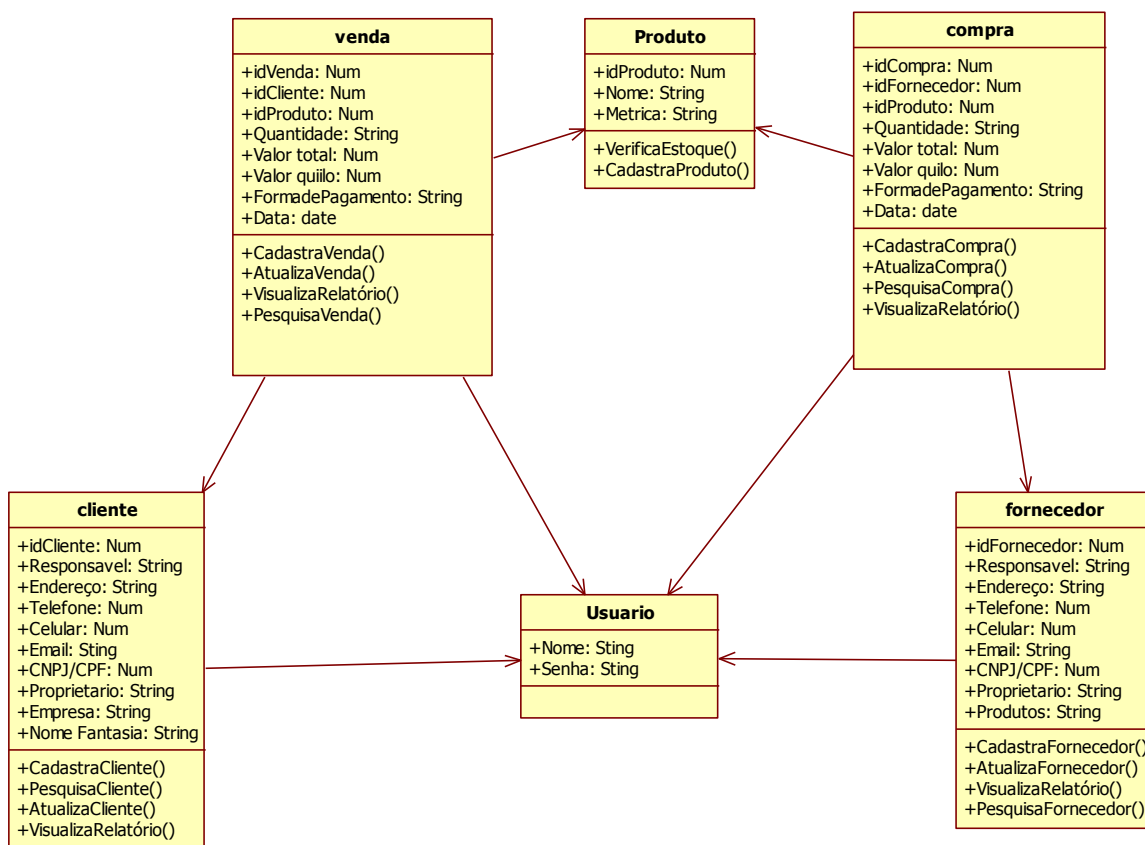


Figura 13: Diagrama de classes 2, fonte: Autoria própria

O diagrama apresentado demonstra todas as classes que farão parte do sistema, sendo elas usuário, cliente, fornecedor, compra, venda e produto. Dentro de cada classe foi especificado seus atributos além das operações que deverão ser realizadas pelo usuário nesta classe.

A classe usuário é representada pela pessoa que utilizará o sistema, sendo seus atributos o nome e a senha, que serão cadastrados quando o sistema for inicializado.

Os clientes são representados na classe cliente, tendo como atributos o “idCliente”, que será fornecido pelo sistema, “responsável”, “endereço”, “telefone”, “celular”, “e-mail”, “CNPJ/CPF”, “proprietário”, “empresa” e “nome fantasia”, que serão preenchidos pelo usuário. Nesta classe as operações possíveis de serem realizadas são, cadastrar, atualizar e pesquisar clientes, além de uma possível visualização de relatório, sendo possível a verificação de todos os clientes cadastrados.

Na classe dos fornecedores, os atributos necessários para o cadastro serão: “idFornecedor”, oferecido pelo sistema, “responsável”, “endereço”, “telefone”, “celular”, “e-mail”, “CNPJ/CPF”, “proprietário” e “produtos”, que serão preenchidos pelo usuário. Assim como na classe clientes, nesta classe as operações possíveis serão as mesmas, cadastrar, atualizar e pesquisar fornecedores, além de uma possível visualização de relatório, sendo possível a verificação de todos os fornecedores cadastrados.

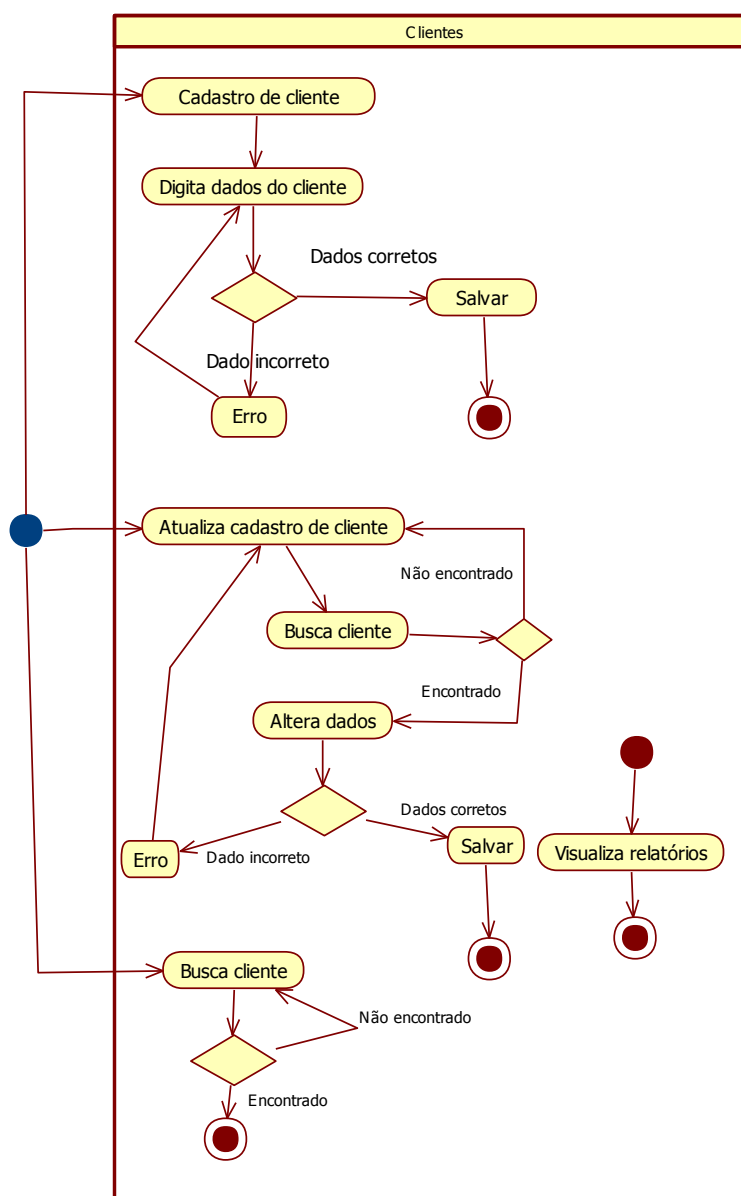
A classe produto possui menos atributos, “idProduto”, oferecido pelo sistema, “nome” e “métrica”, fornecidos pelo usuário. As operações disponíveis serão o cadastro de produtos e a verificação de estoques.

A classe compra possui os seguintes atributos, “idCompra”, novamente oferecida pelo sistema, “idFornecedor”, “idProduto”, “quantidade”, “valor total”, “valor quilo”, “formaDePagamento” e “data” por parte do usuário. Aqui será possível cadastrar, pesquisar e atualizar compras, além de visualizar relatórios sempre que necessário.

A classe venda possui atributos semelhantes à compra, “idVenda”, novamente oferecida pelo sistema, “idCliente”, “idProduto”, “quantidade”, “valor total”, “valor quilo”, “formaDePagamento” e “data” por parte do usuário. Também sendo possível cadastrar, pesquisar e atualizar vendas, além de visualizar relatórios sempre que necessário.

#### 4.5.3. Diagrama de Atividade

Para um melhor entendimento o diagrama de atividades foi dividido em partes, o primeiro a ser apresentado será na figura 14 – Diagrama de atividade cliente.



**Figura 14: Diagrama de atividade cliente, fonte: Autoria própria**

A figura 14 apresenta as atividades que o usuário desempenhará na classe cliente, como dito anteriormente ele será capaz de cadastrar um novo cliente, caso algum dado esteja errado o sistema indicará o erro e solicitará ao cliente a correção. Somente com todos os dados devidamente cadastrados o sistema salvará o cliente cadastrado.

Na classe clientes o usuário também poderá atualizar cadastro de clientes, tendo primeiro que buscar o cliente e assim atualizar seus dados, novamente, o sistema somente atualizará os dados se os mesmos estiverem corretos, em caso de erro as atualizações não serão salvas.

O Software também será capaz de buscas clientes e fornecer relatórios dos clientes para o usuário.

O próximo diagrama é apresentado na figura 15 – Diagrama de atividade fornecedor.

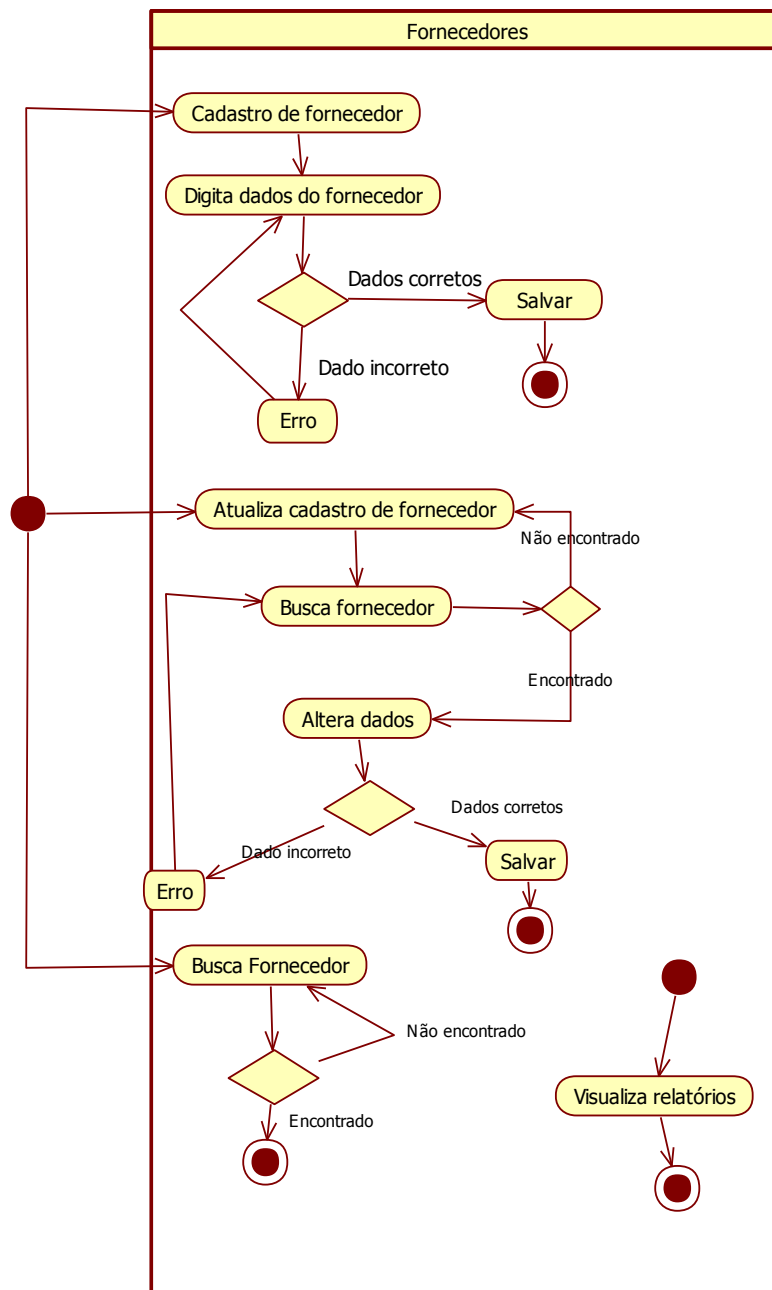


Figura 15: Diagrama de atividade fornecedor, fonte: Autoria própria

Sendo as atividades realizadas pelo usuário nesta classe semelhantes às realizadas no diagrama anterior. Cadastrar um novo fornecedor, tendo o usuário que digitar todos os dados necessários, e em caso de erro o sistema indicará o erro e não cadastrará o novo fornecedor.

Com o cliente cadastrado é possível atualizar o cadastro de fornecedores, tendo primeiro que buscar o fornecedor e assim atualizar seus dados, novamente, o sistema somente atualizará os dados se os mesmos estiverem corretos.

O Software também será capaz de buscar fornecedores e fornecer relatórios para o usuário.

A figura 16 – Diagrama de Atividade Compras, é apresentada abaixo.

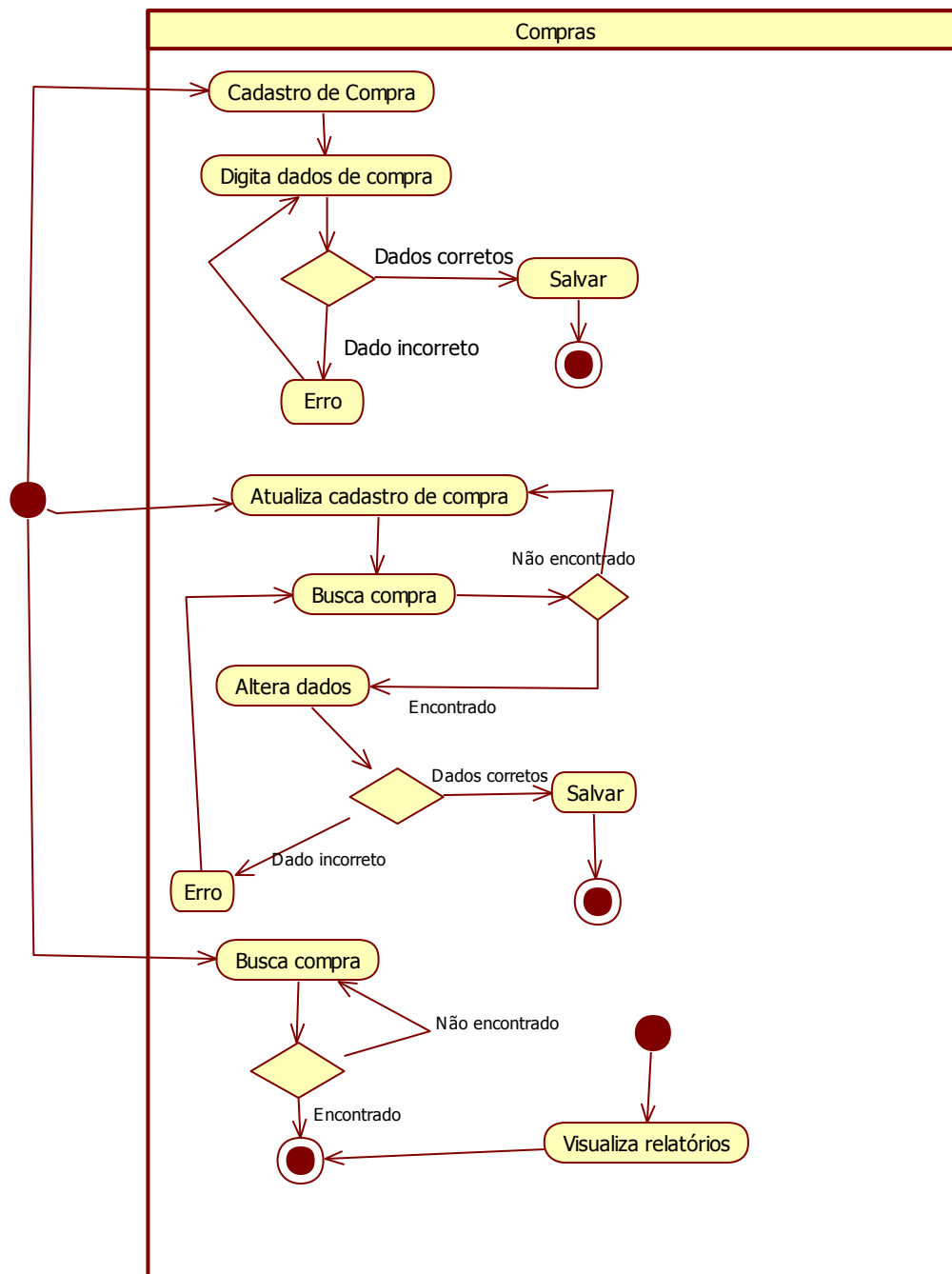


Figura 16: Diagrama de atividade compras, fonte: Autoria própria

Novamente as atividades desempenhadas pelo usuário são semelhantes, sendo ele responsável por cadastrar uma nova compra, digitando os campos necessários, sendo salva a compra somente com todos os dados necessário. Em caso de erro a compra não será salva.

Com a compra efetuada, será possível atualizar cadastro de compra, tendo primeiro que buscar a compra e assim atualizar seus dados, novamente, o sistema somente atualizará os dados se os mesmos estiverem corretos.

O Software também será capaz de buscar compras anteriormente realizadas e fornecer relatórios para o usuário.

A figura 17 – Diagrama de atividade vendas, é apresentada abaixo.

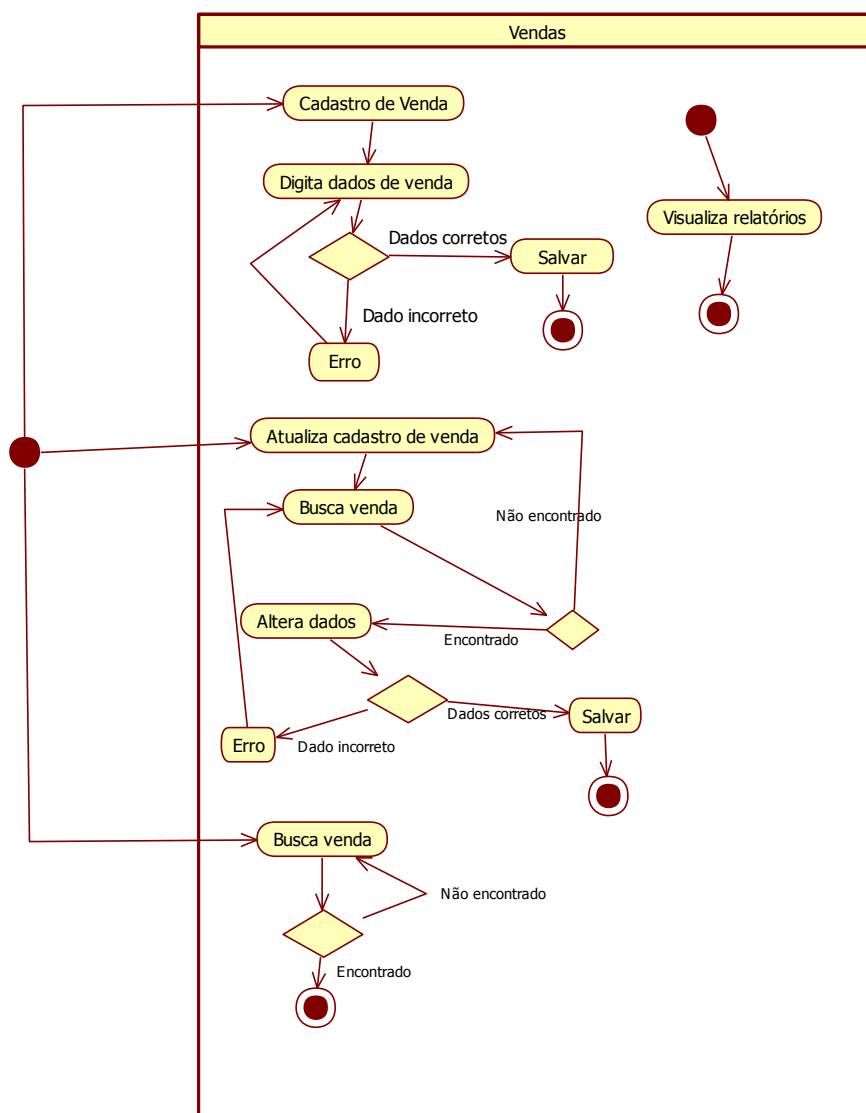


Figura 17: Diagrama de atividade vendas, fonte: Autoria própria

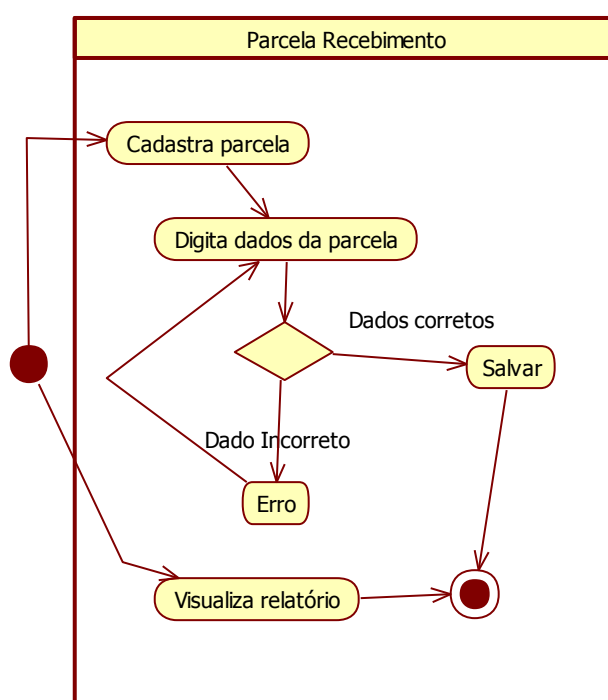
Como as atividades desempenhadas pelo usuário na classe compras, aqui o usuário é responsável por cadastrar uma nova venda, tendo que digitar todos os dados solicitados, caso

algum campo não seja preenchido o sistema indicará o erro e retornará a tela de cadastro, sendo a venda salva apenas com todos os dados devidamente preenchidos.

Com a venda cadastrada o sistema será capaz de atualizar o cadastro de vendas, tendo primeiro que buscar a venda e assim atualizar seus dados, novamente, o sistema somente atualizará os dados se os mesmos estiverem corretos.

O Software também será capaz de buscar vendas anteriormente realizadas e fornecer relatórios para o usuário.

As compras e vendas estão relacionadas a parcelas de recebimento e pagamento respectivamente, primeiro é apresentada as atividades referentes a recebimento na figura 18 – Diagrama de atividade recebimento.

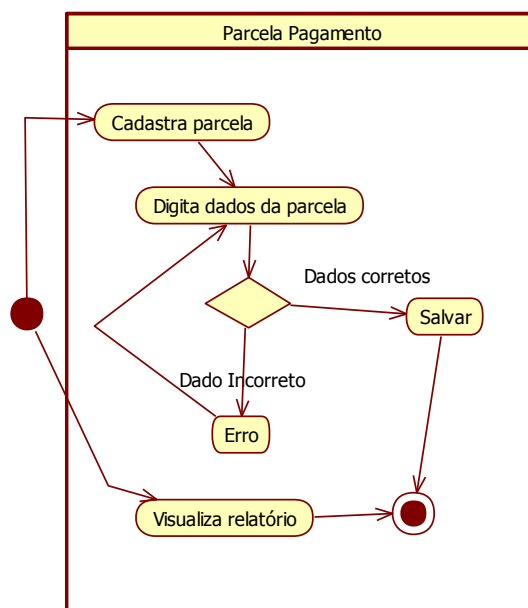


**Figura 18: Diagrama de atividade recebimento, fonte: Autoria própria**

Sendo as atividades do usuário, cadastrar uma nova parcela, digitando todos os dados solicitados, em caso de erro o sistema voltará à tela de cadastro, e com todos os dados devidamente preenchidos a parcela será salva.

O usuário também será capaz de visualizar relatório, referente a tais parcelas.

E na figura 19 – Diagrama de atividade pagamento, apresentada abaixo, demonstra as atividades referentes ao pagamento de parcelas das vendas efetuadas.



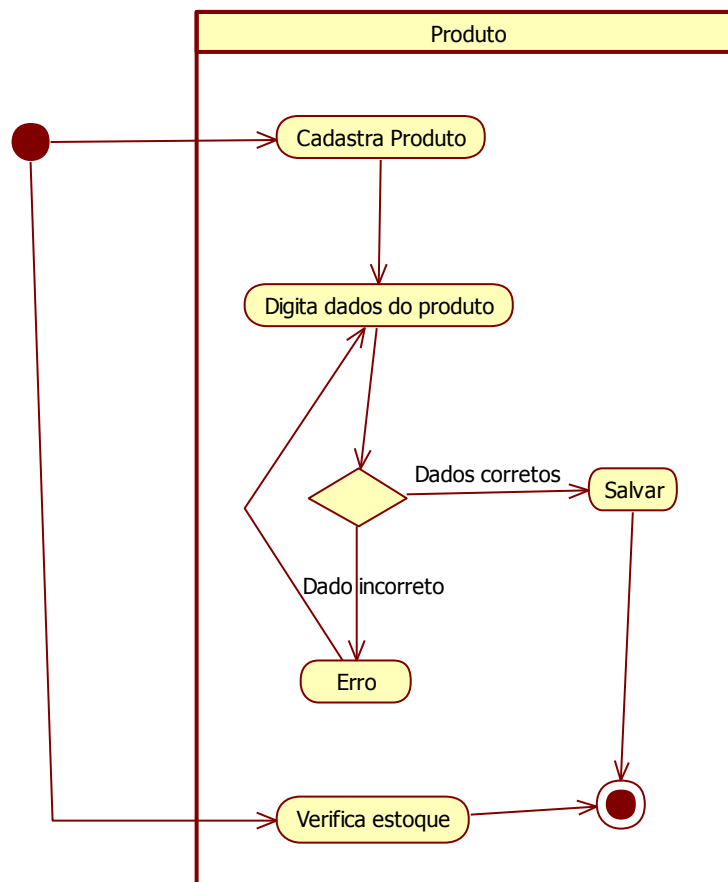
**Figura 19: Diagrama de atividade pagamento, fonte: Autoria própria**

Sendo as atividades do usuário, cadastrar uma nova parcela, digitando todos os dados solicitados, em caso de erro o sistema voltará à tela de cadastro, e com todos os dados devidamente preenchidos a parcela será salva.

O usuário também será capaz de visualizar relatório, referente a tais parcelas.

E o último diagrama de atividade apresentado será a figura 20 – Diagrama de Atividade Produto, abaixo.





**Figura 20: Diagrama de atividade produto, fonte: Autoria própria**

Na classe produto o usuário poderá cadastrar novos produtos, digitando os dados solicitados, em caso de erro o sistema retornará a tela inicial de cadastro, e com todos devidamente preenchidos o sistema salvará o produto.

Também será possível verificar o estoque, conferindo a quantidade de produto que ele ainda possui para comercializar.

#### 4.5.4. Diagrama de Objetos

O diagrama de objeto é apresentado na figura 21 – Diagrama de objeto 2, abaixo.

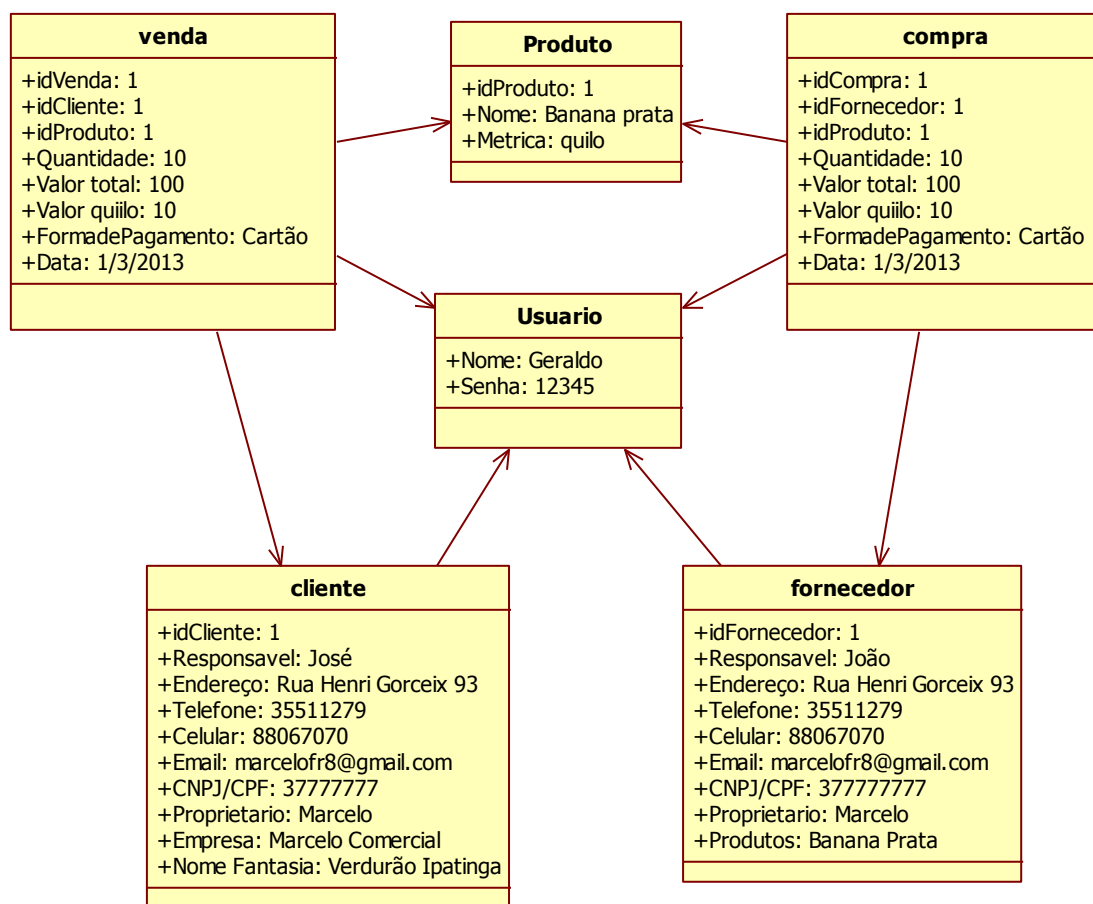


Figura 21: Digrama de objeto 2, fonte: Autoria própria

Como citado anteriormente, tal diagrama é uma instancia do diagrama de classe, porém apresentando atributos possíveis para as classes denominadas em determinado ponto no tempo.

Para isto foram criados exemplos de objetos a serem utilizados na empresa de distribuição de frutas.

Foi criado o objeto cliente José, com todos os dados devidamente preenchidos e recebendo o “idCliente” “1” gerado automaticamente pelo sistema.

Também foi criado o objeto fornecedor João, “idFornecedor” “1”, que seria um fornecedor da empresa em questão.

Para que fossem criadas compras e vendas, foi gerado o produto banana prata, que recebeu a identificação de “idProduto” “1”, para que enfim pudessem ser criados os objetos compra, realizada junto com o fornecedor 1 João, e uma venda, concretizada pelo cliente 1 João.

#### 4.5.5. Diagrama de Implantação

O diagrama de implantação também será dividido em partes para uma melhor apresentação, primeiro serão apresentadas as partes referentes a fornecedor, seguido por cliente, compra, venda e produto.

#### 4.5.5.1. Fornecedor

O layout do cadastro de fornecedores é apresentado na figura 22 – Implantação fornecedor, apresentada abaixo.

Figura 22: Implantação fornecedor, fonte: Autoria própria

A figura 22 acima apresenta como funcionarão os cadastrados de novos fornecedores, o usuário deverá cadastrar todos dados solicitados, sendo fornecido somente o “idFornecedor” por parte do sistema, em ordem cronológica. Os botões na parte inferior esquerda da figura apresentam as opções do usuário de adicionar, desfazer, salvar dados e visualizar relatório.

Na direita da figura, serão apresentados todos os fornecedores já cadastrados, em ordem de cadastro, e na parte inferior o usuário poderá passar para o fornecedor seguinte, anterior, inicial ou final, além da opção buscar, onde poderá encontrar qualquer fornecedor cadastrado.

O relatório poderá ser visto na figura 23 – Relatório fornecedor, abaixo.

| Fornecedor   |             |          |          |                      |                 |              |
|--------------|-------------|----------|----------|----------------------|-----------------|--------------|
| idFornecedor | Responsavel | Telefone | Celular  | Endereço             | CPF/CNPJ        | Proprietário |
| 1            | Joao        | 35511279 | 88067070 | Rua Henri Gorceix 93 | 376216178022222 | Luis         |
| 2            | José        | 35520741 | 88067070 | Rua Henri Gorceix 93 | 376216178022222 | Carlos       |

**Figura 23: Relatório fornecedor, fonte: Autoria própria**

O relatório deverá fornecer os dados de todos os fornecedores de maneira a facilitar a visualização do usuário.

#### 4.5.5.2. Cliente

A implantação dos clientes é mostrada na figura 24 – Implantação cliente, abaixo.

**Figura 24: Implantação cliente, fonte: Autoria própria**

A figura 24 apresenta como funcionarão os cadastrados de novos clientes, o usuário deverá cadastrar todos dados solicitados, sendo fornecido somente o “idCliente” por parte do sistema, em ordem cronológica. Os botões na parte inferior esquerda da figura apresentam as opções do usuário de adicionar, desfazer, visualizar relatório e salvar dados.

Na direita da figura, serão apresentados todos os clientes já cadastrados, em ordem de cadastro, e na parte inferior o usuário poderá passar para o cliente seguinte, anterior, inicial ou final, além da opção buscar, onde poderá encontrar qualquer cliente cadastrado.

O relatório poderá ser visto na figura 25 – Relatório cliente, abaixo.

| Cliente   |                    |          |          |                      |               |              |                   |                   |
|-----------|--------------------|----------|----------|----------------------|---------------|--------------|-------------------|-------------------|
| idCliente | Pessoa Responsável | Telefone | Celular  | Endereço             | CPF/CNPJ      | Proprietário | Empresa:          | Nome Fantasia:    |
| 1         | João               | 35511279 | 88067070 | Rua Henri Gorceix 93 | 387182901922  | João         | Comercial João    | Verdurão Ipatinga |
| 2         | Marcelo            | 35520741 | 88067070 | Rua Hneri Gorceix 93 | 3818182183838 | Marcelo      | Marcelo Comercial | Verdurão do Vale  |

quinta-feira, 4 de abril de 2013

Página 1 de 1

**Figura 25: Relatório cliente, fonte: Autoria própria**

O relatório deverá fornecer os dados de todos os clientes de maneira a facilitar a visualização do usuário.

#### 4.5.5.3. Compra

O modo de cadastro de compras é mostrado na figura 26 – Implantação compra, abaixo.

**Figura 26: Implantação compra, fonte: Autoria própria**

Assim como nos diagramas anteriores no cadastrado de novas compras, o usuário deverá cadastrar todos dados solicitados, sendo fornecido somente o “idCompra” por parte do sistema, em ordem cronológica.

Como neste caso cada compra deverá estar ligada a um fornecedor e um produto, para isto o sistema deverá auxiliar o usuário no cadastramento de tais itens, como mostrado na figura 26 acima, oferecendo o “idFornecedor” juntamente com a pessoa responsável, o mesmo deverá ser feito para o “IdProduto”.

Os demais botões da parte inferior funcionam da mesma maneira apresentada anteriormente.

O relatório poderá ser visto na figura 27 – Relatório compra, abaixo.

| Compra   |              |            |             |            |            |                    |           |
|----------|--------------|------------|-------------|------------|------------|--------------------|-----------|
| idCompra | idFornecedor | Quantidade | Valor Total | Valor Kilo | Data       | Forma de Pagamento | idProduto |
| 9        | 7            | 300        | 200         | 300        | 10/04/2013 | cartão             | 3         |
| 8        | 2            |            |             |            |            |                    |           |
| 1        | 1            | 100        | 200         | 2          | 31/10/2012 | Cartão             | 1         |
| 2        | 2            | 200        | 400         | 2          | 03/04/2013 | dinheiro           | 2         |

**Figura 27: Relatório compra, fonte: Autoria própria**

O relatório deverá fornecer os dados de todas as compras efetuadas, de maneira a facilitar a visualização do usuário.

#### 4.5.5.4 Venda

O modo de cadastro de vendas é mostrado na figura 28 – Implantação venda, abaixo.

Vendas

segunda-feira, 29 de abril de 201304:45

idVenda:

2

idCliente:

1

idProduto:

1

Quantidade:

100

Valor total:

200

Valor Kilo:

2

Data:

04/04/2013

Forma de pagamento:

Cheque

IdCliente

1

IdProduto

1

Adicionar

Desfazer

Salvar

Relat

Buscar

Hoje

**Figura 28: Implantação venda, fonte: Autoria própria**

Assim como no diagrama anterior no cadastrado de novas vendas, o usuário deverá cadastrar todos dados solicitados, sendo fornecido somente o “idVenda” por parte do sistema, em ordem cronológica. Porém, como neste caso cada compra deverá estar ligada a um fornecedor e um produto, para isto o sistema deverá auxiliar o usuário no cadastramento de tais itens, oferecendo o “idFornecedor” juntamente com a pessoa responsável, o mesmo

deverá ser feito para o “IdProduto”. O sistema também fornecerá um calendário ao usuário na inserção do campo “Data” de maneira a facilitar o preenchimento.

Os demais botões da parte inferior funcionam da maneira apresentada anteriormente.

O relatório poderá ser visto na figura 29 – Relatório venda, abaixo.

| Venda   |           |           |            |             |            |            |                    |
|---------|-----------|-----------|------------|-------------|------------|------------|--------------------|
| idVenda | idCliente | idProduto | Quantidade | Valor total | Valor Kilo | Data       | Forma de pagamento |
| 2       | 1         | 1         | 100        | 200         | 2          | 04/04/2013 | Cheque             |
| 3       | 2         | 2         | 200        | 400         | 2          | 21/03/2013 | Dinheiro           |

16:46:07 Página 1 de 1

**Figura 29: Relatório venda, fonte: Autoria própria**

O relatório deverá fornecer os dados de todas as vendas efetuadas, de maneira a facilitar a visualização do usuário.

#### 4.5.5.5. Produto

O modo de cadastro de produtos é mostrado na figura 30 – Implantação produtos, abaixo.

Produtos

quinta-feira, 4 de abril de 2013  
06:03

idProduto: 1

Produto: Banana Prata

Métrica: kilo

Adicionar

Desfazer

Salvar

Estoque

Buscar

**Figura 30: Implantação produtos, fonte: Autoria própria**

O cadastro de produtos é feito de uma maneira mais simples, sendo necessária apenas a inclusão do produto e da métrica a ser utilizada, contudo o programa só salvará produtos devidamente preenchidos. Os botões na parte inferior possuem as mesmas funções dos anteriores, com exceção do botão estoque, que será por onde o usuário verificará o estoque da empresa.

## **5. Discussão dos resultados**

O estudo de caso demonstrou a definição, modelagem e descrição de um sistema informatizado desenvolvido para uma micro empresa de distribuição de frutas. O projeto demonstrou que o Software deverá possuir um manuseio simples, em virtude do pequeno tamanho do comércio e, portanto, da falta de pessoas capazes de operar sistemas muito complexos. Para isto a análise dos requisitos funcionais foi desenvolvida de uma maneira que pudesse facilitar ao máximo a utilização do Software por parte do usuário, como por exemplo, sendo sempre oferecidos os números de identificações gerados pelo sistema sempre que solicitados.

Por ser uma empresa pequena, o sistema foi desenvolvido para ser utilizado por apenas uma pessoa, o que facilitaria o controle, reduzindo o número de possíveis erros.

Os diagramas descritos foram escolhidos com objetivo de melhor apresentar as visões estáticas e dinâmicas do Software desejado. Eles foram representados através das definições estabelecidas pela análise dos requisitos, demonstrando assim as necessidades do cliente e definindo o modo de operação do sistema.

Os Softwares escolhidos para o desenvolvimento do diagrama foram o StarUML, por ser um programa gratuito e capaz de modelar uma grande quantidade de diagramas diferentes, possibilitando assim a realização de quase todos os diagramas necessários para a realização do estudo. E o outro o Microsoft Office Access 2007, escolhido para modelar o diagrama de implantação, por poder apresentar uma melhor visualização do layout do sistema a ser oferecido.

Uma dificuldade encontrada foi a indisponibilidade da versão em português do Software para modelagem dos diagramas, porém pelo fato do programa seguir a linguagem UML, uma linguagem padronizada, foi possível entender todos os termos técnicos presentes, não sendo esta uma barreira na elaboração dos diagramas.

O fato de os programas utilizados serem de acesso gratuito facilitou o desenvolvimento da pesquisa.

## **6. Conclusão**

A evolução dos recursos trazidos pelas Tecnologias de Informação é algo notório. Assim como mostrado anteriormente neste trabalho, elas vêm sendo cada vez mais importantes e diferenciais na vida das pessoas e dos comerciantes. Desta forma, um comércio que possui como objetivo a sua expansão deve possuir um Sistema de Informação eficiente



em sua empresa, contribuindo principalmente na gestão. Os Sistemas de Informação auxiliam a empresa a coletar, transmitir, armazenar e processar dados de modo a fornecer informações importantes.

O objetivo principal deste trabalho foi definir, modelar e descrever um sistema informatizado destinado à gestão de uma micro empresa de distribuição de frutas. Através de um estudo de caso em uma empresa do ramo e seguindo as etapas propostas pela Engenharia de Software, foi possível a modelagem e a descrição do sistema informatizado, com ajuda do gerente da empresa estudada para definir as principais atividades que deveriam ser administradas pelo Software a ser modelado.

Com o trabalho foi possível perceber a importância do bom gerenciamento das informações em uma pequena empresa, através da quantidade de dados que se fizeram necessárias para controlar as compras e vendas de produto nesta empresa. O uso de um Software especializado para controlar tais atividades tornaria o trabalho do gerente desta empresa muito mais eficiente, fornecendo dados em uma velocidade maior e mais precisa.

O próximo passo seria o desenvolvimento do Software, que utilizaria o presente trabalho para a elaboração do Software em questão. Sendo seguida pela parte de teste e finalmente a implantação do sistema na empresa.

## 7. Bibliografia

ABEPRO. *Associação Brasileira de Engenharia de Produção*. 2008. <http://www.abepro.org.br/interna.asp?p=399&m=424&ss=1&c=362> (acesso em 23 de 1 de 2013).

Bio, Sérgio Rodrigues. *Sistemas de Informação: em enfoque gerencial*. 2º Edição. São Paulo: Atlas, 2008.

Booch, Grady; James Rumbaugh; e Ivar, Jacobson. *UML: guia do usuário*. 6º Reimpressão. Rio de Janeiro: Elsevier, 2005.

Costa, Carlos Alberto. *A aplicação da linguagem de modelagem unificada (UML) para o suporte ao projeto de sistemas computacionais dentro de um modelo de referência*. *Revista Gestão & Produção*, volume 8, 2001.

Foina, Paulo Rogério. *Tecnologia de Informação planejamento e Gestão*. 2º Edição. São Paulo: Atlas, 2006.

Ganga, Gilberto Miller Devós. *Trabalho de Conclusão de curso (TCC) na engenharia de produção: um guia prático de conteúdo e forma*. São Paulo: Atlas, 2012.

Guedes, Gilleanes T. A. *UML 2, uma abordagem prática*. 2º edição. São Paulo: Novatec, 2011

Group, Object Management. *OMG*. 2013. <http://www.omg.org/spec/UML/2.5/Beta1/> (acesso em 11 de Março de 2013).

*IBGE*. 2001. <http://www.ibge.gov.br/home/> (acesso em 23 de 11 de 2012).

Laudon, Kenneth C.; Laudon, Jane P. *Sistemas de Informação Gerenciais*. 7º Edição. São Paulo: Pearson Prentice Hall, 2007.

Laurindo, Fernando José Barbin. *Tecnologia da informação: Planejamento e gestão de estratégias*. 1º Edição. São Paulo: Atlas, 2008.

Lima, Adilson da Silva. *UML 2.3 do requisito à solução*. 1º edição. São Paulo: Érica, 2011.

Magela, Rogério. *Engenharia de Software Aplicada: Fundamentos*. 1º Edição. Rio de Janeiro: Alta Books, 2006.

Melo, Ana Cristina. *Desenvolvendo aplicações com UML 2.2 do conceito à implementação*. 3º edição. Rio de Janeiro: Brasport, 2010.

Miguel, Paulo Augusto Cauchick. *Metodologia de pesquisa em engenharia de produção e gestão de operações*. 2º Edição. Rio de Janeiro: Elsevier, 2012.

Moraes, Giseli Diniz de Almeida; Terence, Ana Cláudia Fernandes; Filho, Edmundo Escrivão. “A tecnologia da informação como suporte à gestão estratégica da informação na pequena empresa.” *Revista de Gestão da Tecnologia e Sistemas de Informação*, 2004: 27-43.

Pfleeger, Shari Lawrence. *Engenharia de Software: teoria e prática*. 2º Edição. São Paulo: Prentice Hall, 2004.

Pressman, Roger S. *Software engineering: a practitioner's approach 7th ed*. New York: McGraw-Hill, 2010.

Rezende, Denis Aleides; Abreu, Aline França de. *Tecnologia da Informação aplicada a Sistemas de Informação Empresariais*. 7º Edição. São Paulo: Atlas, 2010.

Ribeiro, Vinicius Gadis; Zabadal Jorge Rodolfo Silva. *Pesquisa em computação: uma abordagem metodológica para trabalhos de conclusão de curso e projetos de iniciação científica*. Porto Alegre: UniRitter, 2010.

Sebrae. 2012. <http://www.sebrae.com.br/> (acesso em 23 de 11 de 2012).

Sommerville, Ian. *Engenharia de software*. 8º edição. São Paulo: Pearson Addison-Wesley, 2007.

Tonsig, Sérgio Luiz, *Engenharia de Software-Análise e Projeto de Sistemas*. 2º edição. Rio de Janeiro: Ciência Moderna Ltda, 2008

Turban, Efraim; Jr., R. Kelly Rainer; Potter, Richard E. *Introdução a Sistemas de Informação: Uma Abordagem Gerencial*. 4º Edição. Rio de Janeiro: Elsevier, 2007.

Wazlawick, Raul Sidnei. *Metodologia de Pesquisa para Ciência da Computação*. Rio de Janeiro: Elsevier, 2008.