

**CENTRO UNIVERSITÁRIO UNIVESC  
CURSO DE CIÊNCIA DA COMPUTAÇÃO  
TRABALHO DE CONCLUSÃO DE CURSO**

**CAMILA MATIAS DOS REIS**

**PET SHOP LOVERS**

**Área: Sistema Gerenciado de Banco de Dados**

**LAGES (SC), DEZEMBRO DE 2013**

**CENTRO UNIVERSITÁRIO UNIVESC  
CURSO DE CIÊNCIA DA COMPUTAÇÃO  
TRABALHO DE CONCLUSÃO DE CURSO**

**CAMILA MATIAS DOS REIS**

**PET SHOP LOVERS**

**Área: Sistema Gerenciado de Banco de Dados**

Projeto apresentado à Banca Examinadora do Trabalho de Conclusão do Curso de Ciência da Computação para análise e aprovação.

Orientador(a): Márcio José Sembay, Msc.

Lages (SC), Dezembro de 2013

## **EQUIPE TÉCNICA**

### **Acadêmica**

Camila Matias dos Reis

### **Professor Orientador**

Prof<sup>o</sup>. Márcio José Sembay, Msc.

### **Coordenador de TCC**

Prof<sup>o</sup>. Márcio José Sembay, Msc.

### **Coordenador do Curso**

Prof<sup>o</sup>. Márcio José Sembay, Msc.

**PET SHOP LOVERS**

CAMILA MATIAS DOS REIS

BANCA EXAMINADORA

---

Prof(a). Márcio José Sembay, Msc.

Orientador(a)

---

Prof(a). Nome

---

Prof(a). Nome

## SUMÁRIO

<b>EQUIPE TÉCNICA .....</b>	<b>3</b>
<b>RESUMO.....</b>	<b>7</b>
<b>ABSTRACT .....</b>	<b>8</b>
<b>AGRADECIMENTOS .....</b>	<b>9</b>
<b>LISTA DE ABREVIATURAS.....</b>	<b>10</b>
<b>LISTA DE FIGURAS.....</b>	<b>11</b>
<b>I. INTRODUÇÃO .....</b>	<b>13</b>
<b>1.1 Justificativa .....</b>	<b>14</b>
<b>1.2 Importância.....</b>	<b>15</b>
<b>1.3 Objetivo do Trabalho.....</b>	<b>16</b>
<b>1.3.1 Objetivo Geral.....</b>	<b>16</b>
<b>1.3.2 Objetivos Específicos .....</b>	<b>16</b>
<b>1.4 Metodologia.....</b>	<b>16</b>
<b>1.4.1 Estudo de Caso.....</b>	<b>16</b>
<b>1.4.2 Cronograma .....</b>	<b>17</b>
<b>1.5 Estrutura do Trabalho.....</b>	<b>17</b>
<b>II. FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>19</b>
<b>2.1 Ergonomia .....</b>	<b>19</b>
<b>2.2 Banco de Dados.....</b>	<b>20</b>
<b>2.2.1 Definições.....</b>	<b>21</b>
<b>2.2.2 Modelo de Dados dos Bancos de Dados.....</b>	<b>22</b>
<b>2.2.3 Modelos Navegacionais .....</b>	<b>23</b>
<b>2.2.4 Modelo Relacional .....</b>	<b>24</b>
<b>2.2.5 Bancos de Dados Orientados a Objetos.....</b>	<b>25</b>
<b>2.2.6 Bancos de Dados Semiestruturados .....</b>	<b>25</b>
<b>2.2.7 Utilização .....</b>	<b>25</b>
<b>2.2.8 Apresentação dos Dados .....</b>	<b>25</b>
<b>2.2.8 Aplicação de Banco de dados.....</b>	<b>26</b>
<b>2.2.10 Aplicativo de Banco de Dados .....</b>	<b>26</b>
<b>2.2.11 Segurança em Banco de Dados.....</b>	<b>27</b>
<b>2.3 Sistema de Informação .....</b>	<b>27</b>
<b>2.3.1 Funcionamento de um SI .....</b>	<b>28</b>
<b>2.3.2 Tipos de Sistema de Informação .....</b>	<b>31</b>
<b>2.3.2.1 Sistema de Processamento de Transações.....</b>	<b>31</b>

2.3.2.2	Sistem de Trabalho do Conhecimento e de Automação de Escritório.....	32
2.3.2.3	Sistema de Apoio Gerencial .....	32
2.3.2.4	Sistema de Suporte da decisão .....	32
2.3.2.5	Sistema de Suporte Executivo .....	33
2.3.2.6	Sistema de Informação Gerencial .....	33
2.4	Microsoft Dot Net Framework 2.0 .....	33
2.5	Microsoft Visual Studio 2010 .....	34
2.6	C# .....	35
2.7	SQL Server Compact Edition.....	38
2.8	Crystal Reports .....	39
2.8.1	Ferramenta Crystal Reports .....	40
2.8.2	As principais vantagens do software .....	40
III.	PROJETO .....	42
3.1	Análise de Requisito .....	42
3.2	Requisitos funcionais .....	42
3.2	Requisitos não funcionais .....	44
3.4	Problemas a Serem Sanados .....	44
3.5	Melhorias.....	44
3.6	Diagrama de caso de uso.....	44
3.7	Diagrama Entidade Relacionamento.....	48
3.8	Diagrama de Classe.....	49
3.9	Interfaces do sistema .....	50
IV	CONCLUSÃO .....	83
V	REFERÊNCIAS BIBLIOGRÁFICAS .....	84
VI	ANEXOS .....	86
ANEXO I	– ARTIGO .....	87

## RESUMO

Este trabalho apresenta um *Sistema Gerencial de Banco de Dados (SGBD)*, sistema o qual foi implementado para o armazenamento de dados e consulta, visando o gerenciamento e administração das funções rotineiras da empresa Clínica Veterinária Filhotes & Mascotes Pet Shop Lovers. Com este sistema a empresa em questão vai poder ter um controle maior de suas atividades como cadastro de clientes, funcionários, produtos, fornecedores, animais, controle de contas e pré-vendas, agendamento de consultas, além de emissão de relatórios, proporcionando assim uma visão maior de quantos clientes e animais foram atendidos, seus fornecedores, além de um controle maior dos lucros e despesas da empresa. Para o desenvolvimento do mesmo foi realizado um estudo de caso aonde foi possível detectar os principais problemas e dificuldades de gestão da empresa. O sistema foi desenvolvido com banco de dados SQL Server Compact Edition, plataforma Microsoft.NET Framework, C#, Microsoft Visual Studio e Crystal Reports (ferramenta para gerar relatórios).

**Palavras chaves:** *SGBD, C#, SQL Server Compact Edition, Microsoft.NET Framework, Microsoft Visual Studio, Crystal Reports.*

## **ABSTRACT**

This paper presents a Database Management System (DBMS) system, which was implemented for the data storage and query, targeting the management and administration of the routine functions of the company Puppies Veterinary Clinic & Pet Shop Pets Lovers . With this system the company in question will be able to have greater control of their activities such as customer base , employees, products , suppliers , animal control, accounts and sales , appointment scheduling , and reporting, thus providing greater insight how many clients and animals were treated , their suppliers , as well as greater control of income and expenses of the company . For its development, we conducted a case study where it was possible to identify the main problems and difficulties in management. The system was developed with the database SQL Server Compact Edition platform Microsoft.NET Framework, C #, Microsoft Visual Studio and Crystal Reports (tool to generate reports).

**Keywords:** DBMS, C #, SQL Server Compact Edition, Microsoft.NET Framework, Microsoft Visual Studio, Crystal Reports.

## AGRADECIMENTOS

Primeiramente a Deus por ter me dado força mediante as dificuldades, por ter zelado por mim nesta caminhada tão longa e difícil e por ter colocado no meu caminho pessoas sem as quais não seria possível ter realizado este sonho.

A minha família e principalmente aos meus pais por todo apoio, força, compreensão e por terem estado sempre ao meu lado em todos os momentos.

Aos amigos por acreditarem no meu potencial e sempre me estimularem com palavras de força e confiança e, principalmente um em especial sem o qual jamais teria conseguido chegar até aqui e que foi muito importante desde o começo de tudo.

Aos professores e mestres que com determinação e paciência se dedicaram para transmitir o máximo de conhecimento possível e, em especial um que me mostrou que eu era capaz de chegar até o final e me ajudou concluir esta fase tão importante de minha vida.

## LISTA DE ABREVIATURAS

CASE - Computer Aided Software Engineering  
CGI - Common Gateway Interface  
FI - Form Interpreters  
SGDB - Sistema gerenciador de banco de dados  
SIAE - Sistema de Informação para Automação de Escritórios  
SIG Sistema de Informação Gerencial  
SIPT - Sistema de Processamento de Transações  
SQL - Structured Query Language  
SSTD - Sistemas de Informação de Suporte a Tomada de Decisão  
TCC - Trabalho de Conclusão de Curso  
TI - Tecnologia da Informação  
UML - Unified Modeling Language  
URL - Uniform Resource Locator  
SI - Sistema de Informação.  
STP - Sistema de Processamento de Transações.  
SAD - Sistema de Apoio à Decisão.  
SAE - Sistema de Automação de Escritórios.  
SIE - Sistema de Informação Executiva.  
SIM - System of Information Management  
RUP - Rational Unified Process  
XML - Extensive Markup Language  
MSVS - Microsoft Visual Studio  
JIT - Just In Time Compiler

## LISTA DE FIGURAS

Figura 01 - Modelo de Ergonomia de Software.....	20
Figura 02 - Visão Geral de um SPT.....	30
Figura 03 – Diagrama de caso de uso com ações pertinentes à registro de animais no sistema.....	45
Figura 04 – Diagrama de caso de uso com ações pertinentes à registro de clientes no sistema.....	45
Figura 05 – Diagrama de caso de uso com ações pertinentes à registro de produtos no sistema.....	46
Figura 06 – Diagrama de caso de uso com ações pertinentes à registro de funcionários no sistema.....	46
Figura 07 – Diagrama de caso de uso com ações pertinentes à registro de fornecedores no sistema.....	47
Figura 08 – Diagrama de caso de uso com ações pertinentes à registro de Contas a Pagar no sistema.....	47
Figura 09 – Diagrama de caso de uso com ações pertinentes à registro de Consultas no sistema.....	47
Figura 10 – Modelo Entidade Relacionamento.....	48
Figura 11– Diagrama de classe do sistema.....	49
Figura 12 – Interface inicial do sistema.....	50
Figura 13 – Menu Arquivo do Sistema.....	51
Figura 14 – Menu Arquivo - Animais.....	51
Figura 15 – Remover Espécie.....	52
Figura 16 – Menu Arquivos - Produtos.....	53
Figura 17 – Remover Categoria.....	53
Figura 18 – Controle de Entrada de Produtos.....	54
Figura 19 – Cadastro de Entrada de Produtos.....	55
Figura 20 – Controle de Saída de Produtos.....	56
Figura 21 – Cadastro de Saída de Produtos.....	57
Figura 22 – Menu de Cadastros.....	58
Figura 23 – Controle de Cadastro de Fornecedores.....	58
Figura 24 – Cadastro de Fornecedores.....	59
Figura 25 – Controle de Cadastro de Produtos.....	60
Figura 26 – Cadastro de Produtos.....	61
Figura 27 – Cadastro de Categoria.....	61
Figura 28 – Controle de Cadastro de Funcionários.....	62
Figura 29 – Cadastro de Funcionário.....	63
Figura 30 – Controle de Cadastro de Clientes.....	64

Figura 31 – Cadastro de Clientes.....	65
Figura 32 – Controle de Cadastro de Animais.....	66
Figura 33 – Busca de clientes cadastrados.....	67
Figura 34 – Cadastro de Animais.....	68
Figura 35 – Busca por arquivos de imagem nos Discos Rígidos.....	69
Figura 36 – Menu de Consulta.....	70
Figura 37 – Controle de Cadastro de Consultas.....	70
Figura 38 – Cadastro de Consultas.....	71
Figura 39 – Agendamento de Consultas.....	71
Figura 40 – Busca de Horário Disponível para Consulta com Veterinário.....	72
Figura 41 – Visualização de Agenda de um Veterinário.....	73
Figura 42 – Cadastro de pré-vendas.....	73
Figura 43 – Busca de Clientes.....	74
Figura 44 – Busca de Produtos.....	74
Figura 45 – Controle de Cadastro de Constatas.....	75
Figura 46 – Cadastro de Contas.....	76
Figura 47 – Menu de Relatórios.....	77
Figura 48 – Filtro de relatório de animais.....	77
Figura 49 – Relatório de Animais.....	78
Figura 50 – Filtro para Relatório de pré-vendas.....	78
Figura 51 – Relatório de Pré-vendas.....	79
Figura 52 – Relatório de Produtos.....	80
Figura 53 – Filtro para Relatório de Contas.....	81
Figura 54 – Relatório de Contas.....	81
Figura 55 – Interface de Aniversariantes.....	82

## I. INTRODUÇÃO

Hoje em dia as empresas estão cada vez mais preocupadas em utilizar as tecnologias existentes no mercado para se informatizar e também para tornar o trabalho mais eficaz, organizado e principalmente com o objetivo de se manter competitiva no mercado.

Tendo em vista esses requisitos este trabalho tem por objetivos demonstrar as técnicas e ferramentas utilizadas para criar um Sistema Gerencial de Banco de Dados (SGBD).

De acordo com (SILBERSCHATZ, 1999), um Sistema Gerenciador de Banco de Dados (SGBD) é constituído por um conjunto de dados associados a um conjunto de programas para acesso a esses dados. O conjunto de dados, comumente chamado banco de dados, contém informações sobre uma empresa em particular. O principal objetivo de um SGBD é proporcionar um ambiente tanto conveniente quanto eficiente para recuperação e armazenamento das informações do Banco de Dados.

Sabemos que é muito importante um sistema administrativo e gerencial, mas mais importante ainda é um SGBD bem estruturado para o armazenamento de informações e dados, principalmente para a consulta dos mesmos.

No desenvolvimento de um software é necessário saber as exigências da organização, para que o mesmo possa ser eficiente auxiliando a empresa nas rotinas e funções. Todo software passa por uma fase de teste, mais só pode ser constatada a eficiência do mesmo a partir do momento que o usuário começa a utilizar o sistema. Conforme a utilização, os usuários vão solicitando modificações até que o sistema possa satisfazer todas as necessidades da empresa.

São vários os recursos tecnológicos encontrados no mercado hoje em dia, e cada vez mais estabelecimentos vêm procurando os mesmo, com o objetivo de trazer benefícios e em busca de agilidade e controle para suas funções diárias. A parte de gerenciamento e administração de uma empresa é de extrema importância, mas os mesmo não seriam eficazes e possíveis sem um SGBD por traz. Em essência, um banco de dados é apenas um sistema

computadorizado de armazenamento de registros. (DATE, 2000). Afinal sabemos que uma empresa por menor que seja possui e, é formada por dados e informações os quais precisam ser armazenados para assim ter um controle maior e poder ter acesso às informações com mais facilidade, visando ter um futuro crescimento. Analisando estes pontos, e sendo o principal problema da empresa em questão a solução foi à implementação de um SGBD para um Pet Shop.

## **1.1 Justificativa**

Um acadêmico do curso de Ciência da Computação ao se formar tem o comprometimento não apenas de simplesmente desenvolver um sistema, mais sim de colocar em pratica todo o conteúdo, ferramentas e aprendizagem adquiridas ao longo do curso. Um profissional desta área quando realiza um trabalho de criação ou implementação de um sistema não pode delimitar-se apenas a programação do mesmo, existe todo um processo que consiste em avaliar, elaborar o projeto de acordo com as necessidades do cliente, executar o projeto e também realizar possíveis manutenções que serão necessárias conforme a utilização do software.

O cliente tem uma visão e ideia de como deseja que seu sistema funcione, e mesmo que sejam atingidos os objetivos iniciais, conforme o usuário utiliza se analisa a necessidade de fazer manutenções ou até mesmo alterações.

De acordo com (DATE, 2000), um sistema de banco de dados é “um sistema computadorizado cuja finalidade geral é armazenar informações e permitir que os usuários busquem e atualizem essas informações quando as solicitar”. Para o autor um sistema de banco de dados é composto por dados, hardware, software e usuários. Ou seja, todos os sistemas possuem um banco de dados, afinal um bom gerenciamento e administração da empresa são feitos de uma série de informações e dados coletados, os quais precisam ser armazenados adequadamente para uma visão mais ampla da situação da empresa e também para um fácil acesso aos mesmos. Por isso este trabalho aborda o SGBD para uma clínica veterinária a qual seu maior problema era a organização, acesso e controle de suas funções, atividades financeiras e rotinas de trabalho.

Para se construir um sistema de qualidade é importante que o profissional seja comunicativo e não apenas um bom programador, mas também um analista e consiga absorver o máximo possível de informações de seu cliente. O analista deve estar inteirado do

assunto e sistema em questão, não só para conseguir programar o desejado, mais também para estar agregando valor ao que o cliente não está sabendo solicitar. Para obter os resultados esperados é necessária uma interação entre analista, cliente (usuário) e a área de atuação para poder ser feito um estudo de caso para desenvolver, analisar, projetar e implantar um sistema de qualidade.

Durante o desenvolvimento de sistemas a conduta a ser adotada deve levar a construção de sistemas efetivos. Sistemas que não sejam apenas eficientes, mas também eficazes; que façam exatamente o que é útil e necessário para os usuários e para a empresa de modo geral.

“O principal objetivo de um SGDB é proporcionar um ambiente tanto conveniente quanto eficiente para a recuperação e armazenamento das informações do banco de dados” (SILBERSCHATZ; KORTH; SUDARSHAN, 1999). Esta foi a maior preocupação durante a implementação deste trabalho criar um SGBD que fosse ao mesmo tempo eficiente no armazenamento, recuperação e alteração dos mesmos, como com um ambiente conveniente e confortável ao usuário proporcionando assim um fácil acesso as funções desejadas.

## **1.2 Importância**

Banco de dados é um conjunto de informações que se relaciona de forma a criar um sentido. Sua utilização é de grande importância para a definição da política de promoções das empresas. Com as informações contidas no banco de dados é possível para a empresa, por exemplo, criar um cartão de fidelidade para oferecer determinado tipo de ação promocional em datas especiais como os aniversários, o tipo de brinde, quer sejam ingressos de cinema ou teatro, presentes, ou seja, ações que visam cativar o cliente.

Os Bancos de Dados, além de manterem todo o volume de dados organizado, também executam tarefas e comandos que podem ser previamente programadas por uma pessoa (um DBA – Data Base Administrator) que define os serviços a serem realizados pelo sistema baseado nas rotinas de sua empresa.

Os bancos de dados se firmaram nas últimas duas décadas como a principal base de informações das companhias e se tornaram vitais para o seu crescimento.

## **1.3 Objetivos do Trabalho**

### **1.3.1 Objetivo Geral**

Desenvolvimento de uma ferramenta de para auxiliar o controle operacional de uma loja do varejo no ramo de Pet Shop, usando uma estrutura que amplie tecnologicamente o processo funcional do usuário tornando suas tarefas ágeis e precisas, auxiliando assim na tomada de decisões da empresa e principalmente no controle dos dados da empresa.

### **1.3.2 Objetivos Específicos**

Os objetivos específicos do sistema tem como finalidade a implantação das seguintes funções:

- a) Cadastrar, armazenar e dessa formar organizar os dados de clientes, fornecedores, produtos, funcionários e animais.
- b) Agendamento de consultas
- c) Controle de contas e pré-vendas
- d) Emissão de relatórios
- e) Função que mostra os aniversariantes do dia.

## **1.4 Metodologia**

### **1.4.1 Estudo de Caso**

O primeiro passo para o início do estudo e implementação do sistema, foi realizada uma entrevista com o cliente para o levantamento dos requisitos necessários e após isso foi feito um levantamento da parte dos funcionários e suas principais dificuldades em relação as rotinas da empresa.

O trabalho foi dividido da seguinte forma:

- a) Levantamento dos dados da empresa
- b) Acompanhamento das rotinas da empresa
- c) Levantamento de erros, dificuldades e o que poderia ser melhorado para se adequar as funções e rotinas da empresa.

### 1.4.2 Cronograma

A tabela a seguir apresenta o cronograma que foi utilizado para a realização do trabalho.

Atividades Realizadas	Jul	Ago	Set	Out	Nov	Dez
Pesquisa						
Criação e elaboração da pré-proposta						
Elaboração da revisão bibliográfica						
Pré-modelagem do software						
Implementação do software						
Fase de testes						
Entrega e defesa do TCC a banca avaliadora						
Correções						
Entrega e defesa do TCC à banca avaliadora						

Tabela 01 – Cronograma

### 1.4.5 Estrutura do trabalho

Etapas que foram cumpridas para o desenvolvimento do trabalho:

- a) Estudo de caso
- b) Pesquisa bibliográfica
- c) Revisão bibliográfica
- d) Modelagem do sistema
- e) Implementação do sistema
- f) Testes

Para conseguir realizar e alcançar os objetivos do trabalho, primeiro foi feito um estudo de caso para identificar as dificuldades e os requisitos que seriam necessários para o sistema, depois foi feita uma pesquisa bibliográfica para dar início ao trabalho, foi feita uma revisão bibliográfica para dar fundamentos e embasamento técnico ao trabalho, explicando os tópicos que foram abordados, após essas etapas foi dado início a modelagem dos sistema que foi elaborada com a ferramenta *Visual studio 2010*.

A última etapa foi a fase de implementação do sistema que também utilizou a ferramenta *Visual Studio 2010* utilizando banco de dados *SQL Server Compact Edition*, e a realização dos testes.

Os dados foram pesquisados em diversas fontes disponíveis como: livros, páginas da Web, trabalhos de conclusão de cursos, entre outros.

*Hardware* utilizado:

- Um *Desktop* para implementação e documentação
- Processador AMD Phenom™ II X4 2,6GHz
- 6GB memória RAM
- HD 500GB

## II. FUNDAMENTAÇÃO TEÓRICA

### 2.1 Ergonomia

A palavra “Ergonomia” vem de das palavras Gregas: “ergon” que significa trabalho, e “nomos” que significa leis. Hoje em dia, a palavra é usada para descrever a ciência de “conceber uma tarefa que se adapte ao trabalhador, e não forçar o trabalhador a adaptar-se à tarefa”. Também é chamada de Engenharia dos Factores Humanos, e ultimamente, também se tem preocupado com a Interface Homem-Computador. As preocupações com a ergonomia estão a tornar-se um factor essencial à medida que o uso de computadores tem vindo a evoluir.

A ergonomia se aplica no *software* no sentido de melhorar o diálogo ou comunicação, entre o programa e o usuário. Considerando que os *softwares* são normalmente utilizados por longos períodos de tempo, é imprescindível que sejam bem projetados, pois do contrário, podem ocasionar problemas de saúde aos usuários.

A ergonomia de software utiliza-se do estudo das características e do comportamento humano para adequar programas e interfaces de *softwares* aos seus usuários, de forma que o seu uso não seja, de nenhuma forma, prejudicial.

No caso deste projeto, a principal característica da ergonomia de *software* que será ressaltada é a usabilidade, ou seja, facilidade de uso. A usabilidade é subdividida nas seguintes partes.

- **Aprendabilidade:** o sistema buscará ser de fácil utilização, de forma que o usuário possa realizar as tarefas no menor tempo possível. Todos os botões indicam claramente qual a sua função. Os campos dos formulários apresentam descrição objetiva.
- **Eficiência:** uma vez que o usuário saiba como utilizar o sistema, ele apresenta agilidade e eficiência na execução das tarefas
- **Memorabilidade:** com a utilização de botões que expressem claramente suas funções o sistema torna-se de fácil memorização, de forma que o usuário irá se lembrar de como o sistema funciona mesmo após algum tempo sem utilizá-lo.
- **Erros:** o sistema implementará um algoritmo que possua uma baixa taxa de erros e permita que estes sejam facilmente corrigidos, na eventualidade de acontecerem.

- **Ser agradável:** o usuário deve gostar de trabalhar no sistema. A utilização de cores, tipos de fonte e a disposição dos componentes na tela terá de ser adequada ao tipo de usuário. Buscará uma interface simples e objetiva e será voltado para um público adulto e possivelmente restrito ao ambiente acadêmico.

Todos os pontos citados acima são essenciais para o desenvolvimento de um sistema com qualidade. Todos os itens da usabilidade possuem características próprias, o que proporciona um melhor resultado, por isso este projeto pretende levar em conta a questão da ergonomia e usabilidade.

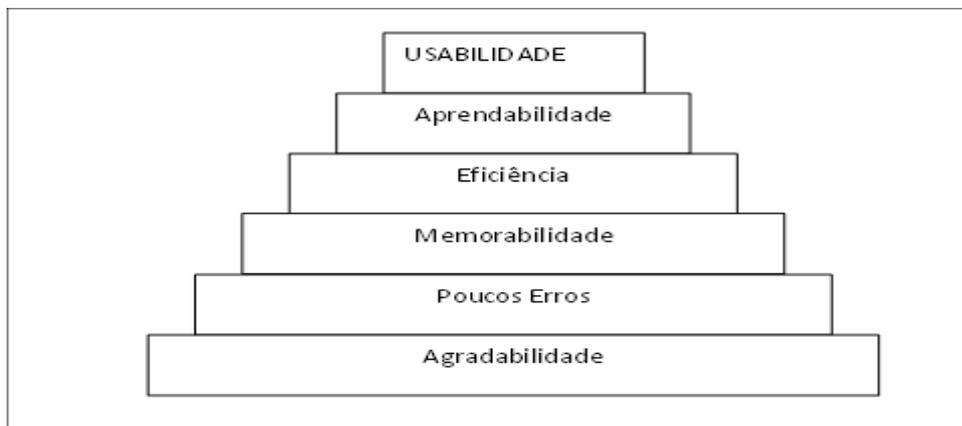


Figura 01: Modelo de Ergonomia de Software  
Fonte: Próprio Autor (2013).

A figura 01 demonstra itens da ergonomia abordados no projeto em desenvolvimento, a utilização da ergonomia nas interfaces de usuário aumentará a usabilidade do sistema, agregando valores ao sistema para os usuários que utilizarem. O sistema se tornará eficaz quando os usuários atingirem seus objetivos, de modo a não dificultar em hipótese alguma a sua navegação.

## 2.2 Banco de Dados

Segundo (DATE,2000) em essência, um banco de dados é apenas um sistema computadorizado de armazenamento de registros.

Bancos de dados, (ou bases de dados), são conjuntos de dados com uma estrutura regular que organizam informação. Um banco de dados normalmente agrupa informações utilizadas para um mesmo fim.

Um banco de dados é usualmente mantido e acessado por meio de um software conhecido como Sistema Gerenciador de Banco de Dados (SGBD). Normalmente um SGBD adota um modelo de dados, de forma pura, reduzida ou estendida. Muitas vezes o termo banco de dados é usado como sinônimo de SGBD.

O modelo de dados mais adotado hoje em dia é o modelo relacional, onde as estruturas têm a forma de tabelas, compostas por linhas e colunas.

### **2.2.1 Definições**

De acordo com (SILBERSCHATZ,1999), um Sistema Gerenciador de Banco de Dados (SGBD) é constituído por um conjunto de dados associados a um conjunto de programas para acesso a esses dados. O conjunto de dados, comumente chamado banco de dados, contém informações sobre uma empresa em particular. O principal objetivo de um SGBD é proporcionar um ambiente tanto conveniente quanto eficiente para recuperação e armazenamento das informações do Banco de Dados.

O termo banco de dados foi criado inicialmente pela comunidade de computação, para indicar coleções organizadas de dados armazenados em computadores digitais, porém o termo é atualmente usado para indicar tanto bancos de dados digitais como bancos de dados disponíveis de outra forma. No Brasil, é mais comum usar o termo base de dados quando se mencionam outros tipos de bancos de dados senão aqueles armazenados em um computador e gerenciados por um SGBD.

Aceitando uma abordagem mais técnica, um banco de dados é uma coleção de registros salvos em um computador em um modo sistemático, de forma que um programa de computador possa consultá-lo para responder questões.

Normalmente um registro está associado a um conceito completo e é dividido em campos, ou atributos, que dão valores a propriedades desses conceitos. Possivelmente alguns registros podem apontar diretamente ou referenciar indiretamente outros registros, o que faz parte da caracterização do modelo adotado pelo banco de dados.

A descrição de quais são os tipos de registros existentes em um banco de dados e ainda quais são os campos de cada registro é conhecida como esquema do banco de dados.

Estritamente falado, o termo banco de dados deve ser aplicado apenas aos dados, enquanto o termo sistema gerenciador de bancos de dados deve ser aplicado ao software com a capacidade de manipular bancos de dados de forma geral. Porém, é comum misturar os dois conceitos.

### **2.2.2 Modelos de Dados dos Bancos de Dados**

A maneira mais prática de classificar bancos de dados é de acordo com a forma que seus dados são vistos pelo usuário, ou seja, seu modelo de dados. Diversos modelos foram e vem sendo utilizados ao longo da história, com vantagens para um ou para outro por determinados períodos.

Atualmente, a classificação mais comum citaria 4 modelos básicos:

Modelos Navegacionais, divididos em:

- Modelo Hierárquico
- Modelo em Redes
- Modelo Relacional
- Modelo Orientado a Objetos

Porém, outros modelos podem ser citados, incluindo:

- Modelo de Entidades e Relacionamentos
- Modelo de lista invertida
- Modelo Relacional Estendido
- Modelo Semiestruturado

Historicamente, o modelo de bancos de dados em rede foi implementado primeiro; porém o primeiro produto comercial usava o modelo de bancos de dados hierárquico, que nada mais é que uma versão simplificada do primeiro. Ambos os modelos foram resultado da busca de usar mais efetivamente os novos dispositivos de memória secundária de acesso direto, que substituíam os cartões perfurados e as fitas magnéticas. Isso aconteceu na década de 1960.

(CODD, 1970) propôs o modelo de bancos de dados relacional que surgiu e ganhou destaque teórico imediato. Porém, a implementação do modelo exigia pesquisas e só na década de 1980 eles iam começar a ganhar o mercado, se estabilizando totalmente como líder do mercado a partir da década de 1990.

Podemos identificar o aparecimento do que pode ser chamado modelo plano (tabular) para fins mais diretos e simples. Nesse caso, os dados estão simplesmente arranjados em uma única matriz bidimensional de elementos de dados na qual todos os membros de uma dada coluna possuem valores de mesmo tipo, e todos os membros de uma linha estão relacionados entre si. Seu melhor exemplo são as planilhas eletrônicas.

O único modelo que foi extensamente tratado de forma teórica foi o modelo relacional. Os modelos pré-existentes foram fruto de implementações, enquanto os modelos subsequentes, como o modelo orientado a objetos, não apresentavam um campo tão rico para novas teorias, mas apresentam grandes desafios para a implementação eficiente das operações necessárias.

### **2.2.3 Modelos Navegacionais**

No modelo em redes, os dados são organizados em registros, que são coleções de itens de dados, e podem ser armazenados ou recuperados de um banco de dados de forma conjunta. É possível que um registro possua uma estrutura interna, e elementos (itens de dados) contínuos podem ser agrupados, que também podem formar outros grupos. Dessa forma, um registro pode ter uma construção hierárquica. Os registros com a mesma estrutura formam um tipo de registro, que podem ser considerados equivalentes a uma tabela fora da primeira forma normal, ou ainda a um objeto complexo. Os tipos de registro possíveis em um banco de dados são definidos em seu esquema.

A principal característica do modelo em redes é permitir a navegação entre os registros, por meio de Conjuntos de Dados, que possui um registro proprietário e registros membros, implementados por meio de ponteiros. Basicamente, registros equivalem a entidades e conjuntos de dados equivalem a descrição dos relacionamentos. Como não há limitação na topologia criada pelos registros e conjuntos, o modelo permite a criação de redes, de onde ganhou o nome.

Um subconjunto particular do modelo de rede, o modelo hierárquico, limita os relacionamentos a uma estrutura de árvore, ao contrário da estrutura aplicada pelo modelo de rede completo.

O modelo em redes foi definido formalmente em 1971, pela *Conference on Data Systems Languages* (CODASYL), de onde ganhou seu outro nome: modelo CODASYL.

### 2.2.4 Modelo Relacional

De acordo com (Date,2000) o modelo relacional é uma teoria matemática desenvolvida para descrever como as bases de dados devem funcionar. Embora esta teoria seja a base para o software de bases de dados relacionais, muito poucos sistemas de gestão de bases de dados seguem o modelo de forma restrita, e todos têm funcionalidades que violam a teoria, desta forma variando a complexidade e o poder. A discussão se esses bancos de dados merecem ser chamados de relacional ficou esgotada com tempo, com a evolução dos bancos existentes.

De acordo com a arquitetura ANSI / SPARC em três níveis, os Bancos de dados relacionais consistem de três componentes:

- Uma coleção de estruturas de dados, formalmente chamadas de relações, ou informalmente tabelas, compondo o nível conceitual;
- Uma coleção dos operadores, a álgebra e o cálculo relacionais, que constituem a base da linguagem SQL; e
- Uma coleção de restrições da integridade, definindo o conjunto consistente de estados de base de dados e de alterações de estados. As restrições de integridade podem ser de quatro tipos:
  - Domínio (ou tipo de dados);
  - Atributo;
  - Relvar;
  - Restrições de base de dados;

Diferentemente dos modelos navegacionais, não existem quaisquer ponteiros, de acordo com o Princípio de Informação: toda informação tem de ser representada como dados; qualquer tipo de atributo representa relações entre conjuntos de dados.

Diferentemente dos bancos de dados em rede, nos bancos de dados relacionais os relacionamentos entre as tabelas não são codificados explicitamente na sua definição. Em vez disso, se fazem implicitamente pela presença de atributos chave. As bases de dados relacionais permitem aos utilizadores (incluindo programadores) escreverem consultas (queries), reorganizando e utilizando os dados de forma flexível e não necessariamente antecipada pelos projetistas originais. Esta flexibilidade é especialmente importante em bases de dados que podem ser utilizadas durante décadas, tornando as bases de dados relacionais muito populares no meio comercial.

Um dos pontos fortes do modelo relacional de banco de dados é a possibilidade de definição de um conjunto de restrições de integridade. Estas definem os conjuntos de estados e mudanças de estado consistentes do banco de dados, determinando os valores que podem e os que não podem ser armazenados.

### **2.2.5 Bancos de Dados Orientados a Objetos**

Segundo (SILBERSCHATZ, 1999), um objeto corresponde a uma entidade no modelo Entidade-Relacionamento. O paradigma orientado a objetos está baseado no encapsulamento de dados e em código relacionado a um objeto dentro de uma única unidade. Conceitualmente todas as interações entre um objeto e o resto do sistema são via mensagens. Então, a interface entre um objeto e o resto do sistema é definida por um conjunto de mensagens permitidas.

Na década de 90, o modelo baseado na orientação a objeto foi aplicado também aos bancos de dados, criando um novo modelo de programação conhecido como bancos de dados orientados a objeto. Os objetos são valores definidos segundo classes, ou tipos de dados complexos, com seus próprios operadores (métodos).

Com o passar do tempo, os sistemas gestores de bancos de dados orientados a objeto e os bancos de dados relacionais baseados na linguagem SQL se aproximaram. Muitos sistemas orientados a objeto são implementados sobre bancos de dados relacionais baseados em linguagem SQL.

O resultado comercial, porém, foi pequeno. Atualmente vários princípios de orientação a objeto foram adotados pelos bancos de dados relacionais, gerando o que pode ser chamado de banco de dados relacional estendido.

### **2.2.6 Bancos de Dados Semiestruturados**

Mais recentemente ainda, apareceram os bancos de dados semiestruturados, onde os dados são guardados e manipulados na forma de XML (ao contrário da forma de tabelas). Novamente, os produtores de bancos de dados relacionais responderam estendendo suas capacidades para tratar dados semiestruturados.

### **2.2.7 Utilização**

Os bancos de dados são utilizados em muitas aplicações, abrangendo praticamente todo o campo dos programas de computador. Os bancos de dados são o método de armazenamento preferencial para aplicações multiusuário, nas quais é necessário haver coordenação entre vários usuários. Entretanto, são convenientes também para indivíduos, e

muitos programas de correio eletrônico e organizadores pessoais baseiam-se em tecnologias padronizadas de bancos de dados.

Um banco de dados é um conjunto de informações com uma estrutura regular. Um banco de dados é normalmente, mas não necessariamente, armazenado em algum formato de máquina lido pelo computador. Há uma grande variedade de bancos de dados, desde simples tabelas armazenadas em um único arquivo até gigantescos bancos de dados com muitos milhões de registros, armazenados em salas cheias de discos rígidos.

Bancos de dados caracteristicamente modernos são desenvolvidos desde os anos da década de 1960. Um pioneiro nesse trabalho foi Charles Bachman.

### **2.2.8 Apresentação dos Dados**

A apresentação dos dados pode ser semelhante à de uma planilha eletrônica, porém os sistemas de gestão de banco de dados possuem características especiais para o armazenamento, classificação e recuperação dos dados.

### **2.2.9 Aplicações de bancos de dados**

Bancos de dados são usados em muitas aplicações, enquanto atravessando virtualmente a gama inteira de software de computador. Bancos de dados são o método preferido de armazenamento para aplicações multiusuárias grandes donde coordenação entre muitos usuários é necessária. Até mesmo usuários individuais os acham conveniente, entretanto, e muitos programas de correio eletrônico e os organizadores pessoais estão baseado em tecnologia de banco de dados standard.

### **2.2.10 Aplicativo de Banco de Dados**

Um Aplicativo de Banco de dados é um tipo de software exclusivo para gerenciar um banco de dados. Aplicativos de banco de dados abrangem uma vasta variedade de necessidades e objetivos, de pequenas ferramentas como uma agenda, até complexos sistemas empresariais para desempenhar tarefas como a contabilidade.

O termo "Aplicativo de Banco de dados" usualmente se refere a softwares que oferecem uma interface para o banco de dados. O software que gerencia os dados é geralmente chamado de sistema gerenciador de banco de dados (SGBD) ou (se for embarcado) de "*database engine*".

Exemplos de aplicativos de banco de dados são Microsoft Visual FoxPro, Microsoft Access, dBASE, FileMaker , (em certa medida) HyperCard, MySQL, PostgreSQL, Microsoft SQL Server e Oracle.

(RESEARCH, 2004), previu que aplicações de banco de dados de código aberto seriam amplamente aceitas em 2006.

### **2.2.11 Segurança em banco de dados**

Os bancos de dados são utilizados para armazenar diversos tipos de informações, desde dados sobre uma conta de e-mail até dados importantes da Receita Federal. Para tal existem diversos tipos, os quais variam em complexidade e sobretudo em segurança. São meios de proteger as informações armazenadas num banco de dados:

- Criptografia
- Senhas
- Backup

## **2.3 Sistema de Informação**

Sistemas de Informação são sistemas que, através de processos de coleta e tratamento de dados, geram e disseminam as informações necessárias aos diversos níveis e processos organizacionais. Lembrando que da grande diversidade de sistemas, podemos separar os sistemas naturais e dos sistemas artificiais (criados pelo homem). Vamos encontrar um grupo de sistemas artificiais, que pela sua natureza, são chamados de Sistemas de Informação, justamente porque seu maior objetivo é Fornecer, Controlar, Prover, Pesquisar e Analisar Informações.

Segundo Bonatto (2004), um exemplo prático é que a maioria das lojas de varejo atualmente utiliza sistemas de informação computadorizados para registrar as compras efetuadas por seus clientes, administrar estoque, pagar funcionários, adquirir novas mercadorias e avaliar tendências das vendas. As operações de uma loja sofreriam em uma interrupção sem o apoio de tais sistemas de informação.

Geralmente, um SI é composto de um subsistema social e de um subsistema automatizado. O primeiro inclui as pessoas, processos, informações e documentos. O segundo consiste dos meios automatizados (máquinas, computadores, redes de comunicação) que interligam os elementos do subsistema social.

Ao contrário do que muitos pensam, as pessoas (juntamente como os processos que executam e com as informações e documentos que manipulam) também fazem parte do SI. O SI é algo maior que um software, pois além de incluir o hardware e o software, também inclui os processos (e seus agentes) que são executados fora das máquinas.

Outra definição de Tecnologia da Informação (TI) é o conjunto de conhecimentos sobre toda a cadeia de valor da informação (tudo o que pode ser feito com a informação):

- Especificação (identificação de necessidades e requisitos);
- Obtenção (origem e captura);
- Armazenamento (classificação, estruturação, armazenamento e recuperação);
- Manipulação (tratamento, formatação e apresentação);
- Distribuição (identificação do destino e transmissão);
- Uso (análise e uso).

Em um Sistema de Informação bem construído, suas principais vantagens são: Otimização do fluxo de informação permitindo mais agilidade e organização; Redução de custos operacionais e administrativos e ganho de produtividade; Mais integridade e veracidade da informação; Mais estabilidade; Mais segurança de acesso à informação.

### **2.3.1 Funcionamento de um SI**

Segundo Stair (1998), um sistema é um conjunto de elementos e componentes que interagem para atingir objetivos. Os próprios elementos e as relações entre eles determinam como o sistema trabalha. Os sistemas têm entradas, mecanismos de processamento, saídas e feedback.

Como qualquer outro sistema, o SI inclui a entrada (input) que envolve a captação ou coleta de fontes de dados brutos de dentro da empresa ou de um ambiente externo. O processamento envolve a conversão dessa entrada bruta em uma forma mais útil e apropriada. A saída (output) envolve a transferência de informação processada às pessoas ou atividades que a usarão (processa os inputs e produz outputs, que são enviados para o usuário ou para outro sistema). Pode conter também um mecanismo de feedback que controla a operação.

Um SI pode ser dividido em 3 partes:

**Entrada:** que recebe todos os problemas e dados da empresa, como dados, informações, regra de negócios, todos eles podem ser interno ou externo (jornais, revistas, pesquisa) (este se enquadra como um SI de BI - *Business Intelligence*)

**Processamento / Controle:** que faz toda o processamento para transformar esses dados, informações e regra de negócios em informação.

**Saída:** que gera os resultados para que possa dar um suporte na tomada de decisões gerenciais da empresa.

**Entrada:**

- Dados;
- Informação;
- Regras de Negócio;

**Processamento/Controle:**

- Tomadores de decisão;
- Auto Controle;

**Saída:**

- Relatórios;
- Gráficos;
- Cálculos;
- Táticas;

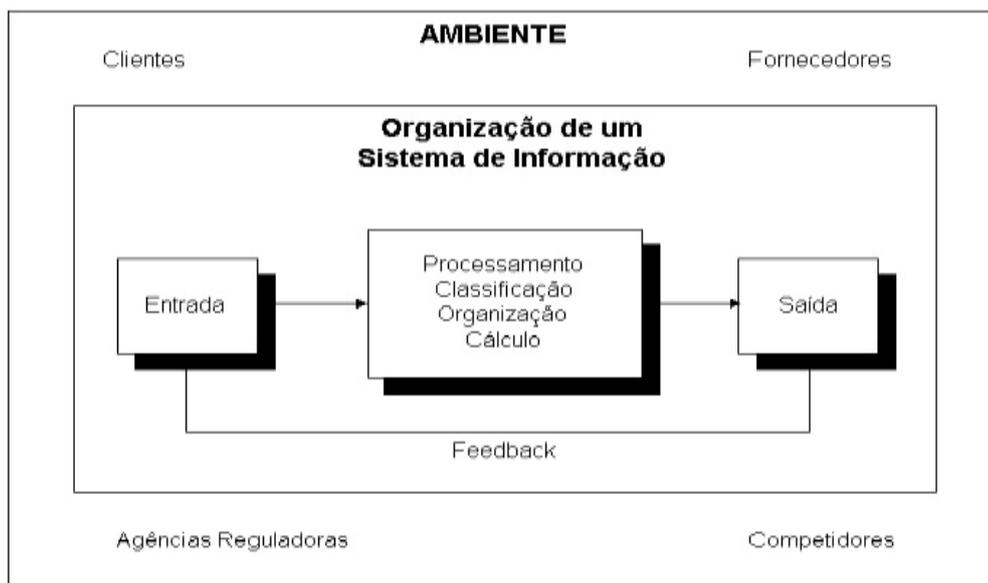


Figura 02: Visão Geral de um SPT

Fonte: Stair (1998).

Segundo (Turban ,McLean e Wetherbe, 2008), um sistema de informação baseado em computador (genericamente chamado de sistema da informação) é um método que utiliza tecnologia de computação para executar algumas de todas as tarefas desejadas. Pode ser composto de apenas um computador pessoal e software, ou incluir milhares de computadores de diversos tamanhos com centenas de impressoras e outros equipamentos, bem como redes de comunicação e banco de dados.

De acordo com O'Brien os SI são relacionados da seguinte forma:

- **Hardware:** computadores e periféricos como: impressora, processadores, monitores, teclados, dispositivos de leitura externo, etc. Juntos, eles aceitam dados e informação, processam-nos e permitem sua visualização.
- **Software:** é um conjunto de programas que permite que o hardware processe dados. Exemplos: software utilitário (sistema operacional); Software aplicativo (conjunto de programas que realizam as funções necessárias para dar suporte às atividades empresariais, como gerar folha de pagamento, emitir nota fiscal, etc.).
- **Pessoas:** são aqueles indivíduos que trabalham com o sistema ou utilizam sua saída . São usuários e operadores de hardware e software.

- **Banco de Dados:** é uma coleção de arquivos, tabelas e outros dados inter-relacionados que armazenam dados e suas respectivas associações.
- **Redes:** é um sistema de ligação que permite o compartilhamento de recursos entre diversos computadores.
- **Procedimentos:** são um conjunto de instruções sobre como combinar os elementos mencionados de forma a processar as informações e gerar saídas desejadas. Também podemos dizer que são as funções que o sistema deve executar.

### 2.3.2 Tipos de Sistema de Informação

Os sistemas, do ponto de vista empresarial, podem ser classificados de acordo com a sua forma de utilização e o tipo de retorno dado ao processo de tomada de decisões. Os sistemas podem ser de contexto operacional ou gerencial, ou seja, Sistemas de Apoio às operações e Sistema de Apoio Gerencial. Na seqüência, estão apresentados os principais sistemas de informação.

#### 2.3.2.1 Sistema de Processamento de Transações (SPT)

Os sistemas de processamento de transações são utilizados no nível operacional da empresa, afirmam LAUDON e LAUDON (2001, p. 31), que “... um sistema de processamento de transações é um sistema computadorizado que executa e registra as transações rotineiras diárias necessárias para a condução dos negócios”. A automatização dos trabalhos repetitivos e rotineiros comuns aos negócios da empresa agiliza e facilita a realização dos trabalhos. Além de oferecer uma gama maior de informações. Como exemplo, pode-se citar a transação das rotinas da folha de pagamento, a computadorização, além de produzir os cheques para pagamento dos colaboradores, pode fornecer relatórios exigidos pelos órgãos federais e estaduais. São também exemplos de SPT, a emissão de notas fiscais e o controle de estoque.

### **2.3.2.2 Sistemas de Trabalho do Conhecimento e de Automação de Escritório (STC e SAE):**

A necessidade do nível de conhecimento da empresa é suprida pelos sistemas de trabalho do conhecimento e de automação de escritório. Segundo BATISTA (2004, p. 24), a definição que se aplica ao STC e SAE é descrita da seguinte forma: "... toda e qualquer tecnologia de informação que possui como objetivo principal aumentar a produtividade pessoal dos trabalhadores que manipulam as informações de escritório". LAUDON e LAUDON (2001, p. 33) também definem: "os sistemas de automação de escritório (SAE) são aplicações de informática projetadas para aumentar a produtividade dos trabalhadores de dados, dando suporte à coordenação e às atividades de comunicação de um escritório típico". Os aplicativos dos escritórios são projetados com base na necessidade de manipulação e gerenciamento de documentos, aumentando assim a produtividade dos envolvidos com a atividade, por exemplo, a editoração eletrônica, arquivamento digital, planilhas de cálculo e outros, favorecem a qualidade e agilidade das tarefas. Os sistemas de trabalho do conhecimento exigem uma visão ampla das pessoas, pois além de saber usar os aplicativos dos escritórios, essas pessoas precisam saber utilizar o que o aplicativo oferece para criar informações novas.

### **2.3.2.3 Sistema de Apoio Gerencial:**

Quando se fala em fornecer informações para a tomada de decisão, toda a empresa deve estar envolvida nesse processo. A complexa relação entre os diversos gerentes de uma organização deve ser facilitada pelos sistemas de apoio gerencial. O'BRIEN (2002, p.29), afirma que "quando os sistemas de informação se concentram em fornecer informação e apoio à tomada de decisão eficaz pelos gerentes, eles são chamados sistemas de apoio gerencial". Entre os vários tipos de sistemas de apoio gerencial, pode-se citar: Sistema de Suporte da Decisão (SSD), Sistema de Suporte Executivo (SSE) e Sistema de Informação Gerencial (SIG).

### **2.3.2.4 Sistema de Suporte da decisão (SSD):**

Os sistemas de suporte da decisão são munidos de grande quantidade de dados e ferramentas de modelagem, permitindo uma flexibilidade, adaptabilidade e capacidade de resposta rápida ao nível gerencial da organização. Nessa contextualização, BATISTA (2004,

p. 25), considera como SSD "... os sistemas que possuem interatividade com as ações do usuário, oferecendo dados e modelos para a solução de problemas semiestruturados e focando a tomada de decisão". Os sistemas de suporte a decisão oferecem recursos cruciais que viabilizam o suporte às decisões de nível gerencial.

#### **2.3.2.5 Sistema de Suporte Executivo (SSE):**

Os sistemas de suporte executivo dão suporte ao nível estratégico da empresa e ajudam a definir os objetivos a serem estabelecidos, utilizando-se de tecnologia avançada para a elaboração de gráficos e relatórios.

Os sistemas de suporte executivo não são projetados para resolver problemas específicos, em vez disso, fornecem uma capacidade de computação e telecomunicações que pode mudar a estrutura dos problemas.

#### **2.3.2.6 Sistema de Informação Gerencial (SIG)**

O sistema de informação gerencial dá suporte às funções de planejamento, controle e organização de uma empresa, fornecendo informações seguras e em tempo hábil para tomada de decisão. OLIVEIRA (2002, p. 59), define que, "o sistema de informação gerencial é representado pelo conjunto de subsistemas, visualizados de forma integrada e capaz de gerar informações necessárias ao processo decisório".

### **2.4 Microsoft Dot Net Framework 2.0**

Segundo (STEIN, STEVEN J., 2009), Microsoft .NET é uma iniciativa da Microsoft em que visa uma plataforma única para desenvolvimento e execução de sistemas e aplicações. Todo e qualquer código gerado para .NET, pode ser executado em qualquer dispositivo ou plataforma que possua um framework: a "Plataforma .NET" (*.NET Framework*). Com ideia semelhante à plataforma Java, o programador deixa de escrever código para um sistema ou dispositivo específico, e passa a escrever para a plataforma .NET.

A plataforma .NET, é executada sobre uma CLR (*Common Language Runtime*) interagindo com uma Coleção de Bibliotecas Unificadas, que juntas são o próprio framework. Esta CLR é capaz de executar, atualmente, mais de vinte diferentes linguagens de programação, interagindo entre si como se fossem uma única linguagem. Estas são: *APL; Boo; Fortran; Pascal; C++; Haskell; Perl; C#; Java; Python; COBOL; Microsoft JScript®;*

*RPG; Component Pascal; Mercury; Scheme; Curriculum; Mondrian; SmallTalk; Eiffel; Oberon; Standard ML; Forth; Oz; Microsoft Visual Basic®; Delphi; J#; Ruby; Lua.*

Arquitetura .NET, A plataforma .NET se baseia em um dos princípios utilizados na tecnologia Java (compiladores JIT), os programas desenvolvidos para ela são duplo-compilados, ou seja são compilados duas vezes, uma na distribuição e outra na execução.

Um programa é escrito em qualquer das mais de vinte linguagens de programação disponível para a plataforma, o código fonte gerado pelo programador é então compilado pela linguagem escolhida gerando um código intermediário em uma linguagem chamada MSIL (*Microsoft Intermediate Language*).

Este novo código fonte gera um arquivo chamado de Assembly, de acordo com o tipo de projeto:

- EXE - Arquivos Executáveis, Programas
- DLL - Biblioteca de Funções
- ASPX - Página Web
- ASMX - Web Service

No momento da execução do programa ele é novamente compilado, desta vez pelo JIT (*Just In Time Compiler*), de acordo com a utilização do programa, por exemplo:

Temos um Web Site desenvolvido em ASP.NET, ao entrar pela primeira vez em uma página o JIT irá compila-la, nas outras vezes que algum outro usuário acessar esta página, ele usará esta compilação.

Também é possível, através de ferramentas específicas, "pré-compilar" o código para que não se tenha o custo da compilação JIT durante a execução.

## **2.5 Microsoft Visual Studio 2010**

Segundo (PELLAND, PARE e HAINES, 2010), Visual Studio é um conjunto completo de ferramentas de desenvolvimento para construção de aplicações Web ASP.NET, serviços Web XML, aplicações desktop e aplicativos móveis. Visual Basic, Visual C# e Visual C++ usam todos o mesmo ambiente de desenvolvimento integrado (IDE), que permite o compartilhamento de ferramentas e facilita a criação de soluções com mistura de linguagens. Além disso, essas linguagens usam a funcionalidade do .NET Framework, que fornece acesso às tecnologias chaves que simplificam o desenvolvimento de aplicativos Web em ASP e serviços Web XML.

O Visual Studio disponibiliza um editor de código que torna a leitura do código mais fácil. Além disso, quando você clica em um símbolo no Visual C# ou Visual Basic, todas as instâncias desse símbolo são realçadas automaticamente.

No Visual C# e Visual C++, A Hierarquia de Chamada permite navegar a partir de um membro para os membros que chamá-lo e para os membros que o chama. Isso é útil na exploração de código orientada a objeto.

Uma janela de threads redesenhada fornece filtragem, pesquisa de chamadas de pilha, expansão e agrupamento.

As janelas de tarefas paralelas ajudam visualizar e depurar código paralelo escrito em C++, C# ou Visual Basic.

Recursos na IDE, tais como adicionar referência e caixa de Ferramentas, agora correspondem à versão do .NET Framework ou do Silverlight que você escolheu para o projeto. Como resultado, tipos, membros e controles de *assemblies* para uma versão anterior do .NET Framework não aparecerão no *IntelliSense* e produzirá erros de compilação. Por exemplo, se seu projeto tem como objetivo o .NET Framework 2.0 e você usa recursos de linguagem que o .NET Framework 2.0 não suporta, como tipos anônimo, a IDE sinalizará o código como errado. Para obter mais informações, consulte *Direcionando-se a um .NET Framework Específico*.

Portanto este ambiente de desenvolvimento e o conhecimento da linguagem C#, linguagem utilizada para desenvolver o software, foram realizados estudos específicos de como realizar o desenvolvimento do aplicativo. E com isso chegar ao resultado demonstrado no capítulo VI deste documento.

## 2.6 C#

Segundo (STELLAMN, 2011), C# (CSharp) é uma linguagem de programação orientada a objetos criada pela Microsoft, faz parte da sua plataforma .Net. A companhia baseou C# na linguagem C++ e Java.

A linguagem C# foi criada junto com a arquitetura .NET. Embora existam várias outras linguagens que suportam essa tecnologia (como VB.NET, C++, J#), C# é considerada a linguagem símbolo do .NET pelas seguintes razões:

Foi criada praticamente do zero para funcionar na nova plataforma, sem preocupações de compatibilidade com código de legado. O compilador C# foi o primeiro a ser desenvolvido. A maior parte das classes do .NET Framework foram desenvolvidas em C#.

A criação da linguagem, embora tenha sido feita por vários desenvolvedores, é atribuída principalmente a Anders\_Hejlsberg, hoje um Distinguished Engineer na Microsoft. Anders Hejlsberg era desenvolvedor de compiladores na Borland, e entre suas criações mais conhecidas estão o Turbo Pascal e o Delphi.

C# (pronuncia-se "cê chárp" em português ou "cí charp" em inglês) é, de certa forma, a linguagem de programação que mais diretamente reflete a plataforma .NET sobre a qual todos os programas .NET executam. C# está de tal forma ligado a esta plataforma que não existe o conceito de código não-gerenciado (unmanaged code) em C#. Suas estruturas de dados primitivas são objetos que correspondem a tipos em .NET. A desalocação automática de memória por *garbage collector* além de várias de suas abstrações tais como classes, interfaces, delegados e exceções são nada mais que a exposição explícita recursos do ambiente .NET.

Quando comparada com C e C++, a linguagem é restrita e melhorada de várias formas incluindo:

Ponteiros e aritmética sem checagem só podem ser utilizados em uma modalidade especial chamada modo inseguro (*unsafe mode*). Normalmente os acessos a objetos são realizados através de referências seguras, as quais não podem ser invalidadas e normalmente as operações aritméticas são checadas contra sobrecarga (overflow).

Objetos não são liberados explicitamente, mas através de um processo de coleta de lixo (*garbage collector*) quando não há referências aos mesmos, prevenindo assim referências inválidas.

Destrutores não existem. O equivalente mais próximo é a interface *Disposable*, que juntamente com a construção *using block* permitem que recursos alocados por um objeto sejam liberados prontamente. Também existem finalizadores, mas como em Java sua execução não é imediata.

Como no Java, não é permitida herança múltipla, mas uma classe pode implementar várias interfaces abstratas. O objetivo principal é simplificar a implementação do ambiente de execução.

C# é mais seguro com tipos que C++. As únicas conversões implícitas por default são conversões seguras, tais como ampliação de inteiros e conversões de um tipo derivado para

um tipo base. Não existem conversões implícitas entre inteiros e variáveis lógicas ou enumerações. Não existem ponteiros nulos (*void pointers*) (apesar de referências para *Object* serem parecidas). E qualquer conversão implícita definida pelo usuário deve ser marcada explicitamente, diferentemente dos construtores de cópia de C++.

- A sintaxe para a declaração de vetores é diferente ("int[] a = new int[5]" ao invés de "int a[5]").
- Membros de enumeração são colocados em seu próprio espaço de nomes (*namespace*)
- C# não possui modelos (*templates*), mas C# 2.0 possui genéricos (*generics*).
- Propriedades estão disponíveis, as quais permitem que métodos sejam chamados com a mesma sintaxe de acesso a membros de dados.
- Recursos de reflexão completos estão disponíveis
- Apesar de C# ser frequentemente tido como similar a Java, existem uma série de diferenças importantes, tais como:
  - Java não implementa propriedades nem sobrecarga de operadores.
  - Java não implementa um modo inseguro que permita a manipulação de ponteiros e aritmética sem checagem.
  - Java possui exceções checadas, enquanto exceções em C# são não checadas como em C++.
  - Java não implementa o goto como estrutura de controle, mas C# sim.
  - Java utiliza-se de comentários Javadoc para gerar documentação automática a partir de arquivos fonte. C# utiliza comentários baseados em XML para este propósito.
  - C# suporta indexadores e delegados.

Ao contrário das outras linguagens de programação, nenhuma implementação de C# atualmente inclui qualquer conjunto de bibliotecas de classes ou funções. Ao invés disso, C# está muito vinculada ao framework .Net, do qual C# obtém suas classes ou funções de execução. O código é organizado em um conjunto de *namespaces* que agrupam as classes com funções similares. Por exemplo: *System.Drawing* para gráficos, *System.Collections* para estrutura de dados e *System.Windows.Forms* para o sistema Windows Form.

Um nível de organização superior é fornecido pelo conceito de montador (*assembly*). Um montador pode ser um simples arquivo ou múltiplos arquivos ligados juntos (como em *al.exe*) que podem conter muitos *namespaces* ou objetos. Programas que precisam de classes para realizar uma função em particular podem se referenciar a montadores como

System.Drawing.dll e System.Windows.Forms.dll assim como a biblioteca core (conhecida como mscorlib.dll na implementação da Microsoft).

## 2.7 SQL Server Compact Edition

Segundo (TIFFANY, ROB, 2003), o *SQL Server Compact 3.5* é um mecanismo de banco de dados em processo e de pequeno volume, que permite aos desenvolvedores criar aplicativos robustos para áreas de trabalho do Windows e dispositivos móveis.

O *SQL Server Compact 3.5 SP2* tem vários recursos, incluindo:

- Suporte para o *Transact-SQL Editor* no *Visual Studio 2010* que pode ser utilizado para executar solicitações T-SQL sem texto e para exibir planos de exibição de solicitações para solicitações otimizadas.
- Exposição do recurso de Controle de Alterações como uma interface de programação de aplicativos (API), especificamente `System.Data.SqlServerCe.SqlCeChangeTracking`. A nova API de Controle de Alterações fornece a capacidade de configurar, habilitar e desabilitar o controle de alterações em uma tabela e de acessar os dados de controle de alterações da tabela.
- Novo suporte de versões de *assembly* e estrutura de diretórios que permite que os aplicativos do *SQL Server Compact* que utilizam implantação privada carreguem os *assemblies do SQL Server Compact* de dentro da pasta de aplicativos, caso sejam de uma versão mais recente do que a que se encontra no cache de *assembly* global (GAC).
- Suporte aprimorado para implantação da versão de 64 bits do *SQL Server Compact* utilizando a implantação *ClickOnce*.
- *Synchronization Services* para ADO.NET v1.0 que oferece a possibilidade de sincronizar dados de fontes diferentes em arquiteturas de duas camadas, de N camadas e baseadas em serviços.

O *SQL Server Compact 3.5 SP2* pode ser redistribuído gratuitamente mediante um contrato de licença de redistribuição. Os desenvolvedores de aplicativos que redistribuem o *SQL Server Compact 3.5 SP2* podem se registrar opcionalmente no site de redistribuição do *SQL Server Compact*. O registro ajudará os desenvolvedores a obter informações sobre os patches críticos de segurança e os hot fixes do *SQL Server Compact* que também podem ser aplicados às instalações do cliente.

As instalações existentes do *SQL Server Compact 3.5* ou do *SQL Server Compact 3.5 SP1* no computador são atualizadas para a versão de lançamento do *SQL Server Compact 3.5 SP2* instalando o *SQL Server Compact 3.5 SP2* usando o arquivo MSI.

É compatível com os seguintes sistemas operacionais: *Windows 7*; *Windows Server 2003 R2 (32-Bit x86)*; *Windows Server 2003 R2 x64 editions*; *Windows Server 2003 Service Pack 2*; *Windows Server 2003 Service Pack 2 x64 Edition*; *Windows Server 2008 R2*; *Windows Server 2008 Service Pack 2*; *Windows Vista Service Pack 2*; *Windows XP Service Pack 3*.

Para o computador de desenvolvimento é necessário que tenha 2 a 3 MB de espaço disponível no disco rígido. 2 a 3 MB de espaço adicional devem estar disponíveis também para arquivos de configuração temporários.

Visual Studio 2010 e Visual Studio 2008 SP1 para suporte ao designer do Visual Studio para aplicativo de desenvolvimento para desktop Windows. Para aplicativos de desenvolvimento para dispositivos do Windows Mobile, utilize o Visual Studio 2008 SP1. Suporte do Visual Studio 2005 SP1 para cenários baseados em código para desktop Windows e dispositivos móveis.

.NET Framework 2.0, 3.0, 3.5 ou 4 para desenvolvimento de aplicativos gerenciados.

Microsoft .NET Framework: o visualizador da Ajuda usado pelos Manuais Online exigem o Microsoft .NET Framework versão 2.0. Windows Installer: o Windows Installer 3.0 é necessário para instalar os Manuais Online do *SQL Server Compact 3.5 SP2*. O Windows Server 2003 Service Pack 2 e o Windows XP Service Pack 2 incluem o Windows Installer 3.0. Para todos os outros sistemas operacionais, é necessário primeiro baixar e instalar o Windows Installer 3.0.

## **2.8 Crystal Reports**

Sabendo da necessidade de criar um sistema direcionado a SGBD e ao mesmo tempo em que fosse administrativo e gerencial que pudesse ter um controle melhor dos cadastros, contas pagas, pré-vendas, etc. Resolvi utilizar esta ferramenta Crystal Reports, que permite a criação de relatórios já que uma empresa por menor que seja precisa se manter competitiva no mercado e para isso ter uma visão mais ampla da entrada e saída de produtos, principalmente das despesas, além de fornecer a informação da quantidade de clientes que possui e animais que são atendidos, proporcionando também um fácil acesso aos principais dados dos seus clientes.

### 2.8.1 Ferramenta Crystal Reports

Segundo (SPANBAUER; SCOTT; MCRAE, 2002), o Crystal Reports é um software capaz de gerar relatórios. Através desse software é possível encontrar informações de bancos de dados ou mesmo arquivos de texto recuperados e processados na forma de gráficos ou mesmo tabelas.

A geração de relatórios através do Crystal Reports possibilita aos usuários grande rapidez de desenvolvimento, uma das razões é que não necessita de auxílio de programadores para desenvolver os variados tipos de relatórios. Trata-se de uma ferramenta basicamente visual e com assistentes que guiam o desenvolvedor do relatório em todas as etapas do processo.

Com o Crystal Reports os usuários finais conseguem consultar os relatórios com ótima visualização, além do mais é possível realizar modelagens de negócios no relatório, e ainda executar decisões instantâneas a partir do próprio relatório.

### 2.8.2 As principais vantagens do software:

De acordo com (SPANBAUER; SCOTT; MCRAE, 2002), as principais vantagens do Crystal Reports são:

- Criação de relatórios profissional com valor bastante acessível;
- Os usuários finais podem explorar os relatórios interativamente;
- Possibilidade de criação de relatórios com ótima aparência e excelente visualização;
- Economia de tempo na criação de relatórios.

Os recursos que permitem escrever, visualizar, explorar e compartilhar relatórios a partir da área de trabalho do *Crystal Reports*:

**Suporte do Crystal Reports 2008:** Com esta opção é possível usar botões de classificação *on-report* e parâmetros interativos para reduzir a seleção de dados e assim receber informações mais importantes. Ainda é possível interagir com gráficos altamente dinâmicos em um relatório. Também é possível obter uma excelente visualização dos dados através de referências cruzadas *on-report*, códigos de barra e layouts de página.

**Visualização offline e interativa:** Aqui é possível abrir arquivos .rtf do *Crystal Reports* direto da área de trabalho, mesmo quando estiver trabalhando *offline*. Também é possível explorar as informações do relatório.

**Integração com o servidor da SAP *BusinessObjects*:** É possível usar o *Crystal Reports Viewer* para fazer o upload, atualizar e ainda salvar relatórios hospedados em SAP *BusinessObjects Enterprise*, *SAP BusinessObjects Edge* e *Crystal Reports Server* versão XI R2 com *Service Pack 2* ou mais recente.

**Páginação contínua e busca completa:** Nesta opção é possível rolar para baixo e visualizar todos os registros no arquivo de relatório. Também é possível buscar dados no próprio documento, incluindo as visualizações ocultas.

**Navegação rápida:** É possível navegar com muita facilidade pelos relatórios usando a árvore de grupo incorporada e as funções de zoom, pesquisa e avanço e retrocesso de página.

**Habilitação para e-mail e Web:** Com esta opção é possível usar a função de e-mail e assim, compartilhar arquivos a partir do *Crystal Reports Viewer*. Além de tudo, o usuário pode acessar conteúdo em hiperlink, como páginas da Web, dentro do próprio software.

**Saída de arquivos em vários formatos:** Existe a possibilidade de exportar relatórios em .rpt para os formatos PDF, Microsoft Excel e Microsoft Word e assim permitir o compartilhamento com outras ferramentas.

**Suporte da comunidade:** Com este suporte o usuário tem acesso ao painel da comunidade, que inclui informações interessantes para usuários do *Crystal Reports Viewer*.

**Serviço de atualização:** Existe a possibilidade de usar a função de serviço de atualização automática para realizar o upgrade para versões mais recentes.

### III. PROJETO

No intuito de projetar um sistema, é necessário utilizar uma metodologia de desenvolvimento que guie a estruturação do projeto, de tal forma a apresentar o modelo que deverá ser desenvolvido.

Segundo LARMAN (2000) o desenvolvimento de uma aplicação necessita de uma descrição do problema e dos seus requisitos. Para identificar o que o problema e o que o sistema precisa fazer, deve-se realizar uma análise, a qual enfatiza uma investigação do problema e de como a solução será definida. Porém, é necessário ter descrições detalhadas da solução lógica e de como ela atende aos requisitos e restrições existentes, para tanto se realiza o projeto.

Com base no estudo realizado definiu-se um sistema de gerenciamento personalizado para a clínica e loja de animais de estimação. Terá como principais características o gerenciamento de seus clientes e animais, de pré-venda de produtos e de assistência veterinária.

O sistema deverá sistematizar o gerenciamento do estabelecimento, informatizando os processos de atendimento e manutenção; isso de forma que o usuário interaja facilmente com um sistema confiável adaptado ao ambiente em questão.

#### 3.1 Análise de Requisito

Os requisitos funcionais abrangem todas as funcionalidades que o sistema deve possuir. De acordo com o estudo feito, foi definido que para o sistema proposto devem ser atribuídos os seguintes itens como requisitos:

#### 3.2 Requisitos funcionais

RF-001	Cadastrar Animal
RF-002	Buscar Animal
RF-003	Alterar Animal Cadastrado
RF-004	Excluir Animal
RF-005	Cadastrar Cliente
RF-006	Buscar Cliente
RF-007	Alterar Cliente Cadastrado
RF-008	Excluir Cliente
RF-009	Cadastrar Produto

RF-010	Buscar Produto
RF-011	Alterar Produto Cadastrado
RF-012	Excluir Produto
RF-013	Cadastrar Funcionário
RF-014	Buscar Funcionário
RF-015	Alterar Funcionário Cadastrado
RF-016	Excluir Funcionário
RF-017	Cadastrar Fornecedor
RF-018	Buscar Fornecedor
RF-019	Alterar Fornecedor Cadastrado
RF-020	Excluir Fornecedor
RF-021	Registrar Entrada de Produtos
RF-022	Registrar Saída de Produtos
RF-023	Cadastrar pré-venda
RF-024	Agendar Consulta
RF-025	Cadastrar Consulta
RF-026	Notificar Aniversariantes do Dia
RF-027	Cadastrar Contas a Pagar
RF-028	Alterar Contas a Pagar
RF-029	Excluir Contas a Paga
RF-030	Pesquisar Contas Cadastradas
RF-031	Pagar Conta
RF-032	Emitir Relatório de Contas
RF-033	Emitir Relatório de Clientes
RF-034	Emitir Relatório de Produtos
RF-035	Emitir Relatório de Animais
RF-036	Emitir Relatório de pré-vendas
RF-037	Cadastrar Espécie de Animais
RF-038	Cadastrar Categoria de Produtos

### 3.3 Requisitos não funcionais

- RNF-01 Sistema operacional Windows XP ou superior;
- RNF-02 Espaço livre em HD 400 MB;
- RNF-03 Memória RAM 512 MB;
- RNF-04 Processador de 1.2 GHz (32 ou 64 bits);
- RNF-05 Deve estar instalado na máquina o Crystal Reports Runtime;
- RNF-06 O banco de dados deve ser o SQL CE;

### 3.4 Problemas a Serem Sanados

- Controle de pré-vendas de produtos;
- Controle de Consultas;
- Visualizar relatórios diversificados, gerados através de filtros do sistema;
- Manter dados de compras realizadas;
- Manter cadastro de clientes.
- Manter cadastro de produtos.
- Controle de visitas a clientes.
- Controle financeiro de contas a pagar.

### 3.5 Melhorias

- Evitar a perda de informações;
- Maior agilidade no atendimento ao cliente.
- Controlar as pré-vendas por períodos.
- Armazenar histórico das pré-vendas, compras, contas e consultas.

### 3.6 Diagrama de caso de uso

Segundo Booch et. all., (2000), diagramas de casos de uso são importantes para visualizar, especificar e documentar o comportamento de um elemento. Esses diagramas fazem com que sistemas, subsistemas e classes fiquem acessíveis e compreensíveis, por apresentarem uma visão externa sobre como esses elementos podem ser utilizados no contexto.

Para ter uma visão geral do funcionamento do sistema foram produzidos diagramas de casos de uso que mostra de forma ampla a interação entre os atores e as funcionalidades do sistema. Este diagrama é de fácil compreensão e ajuda o cliente a perceber se o projeto está de acordo com seus interesses.

A seguir é possível visualizar os diagramas do sistema:

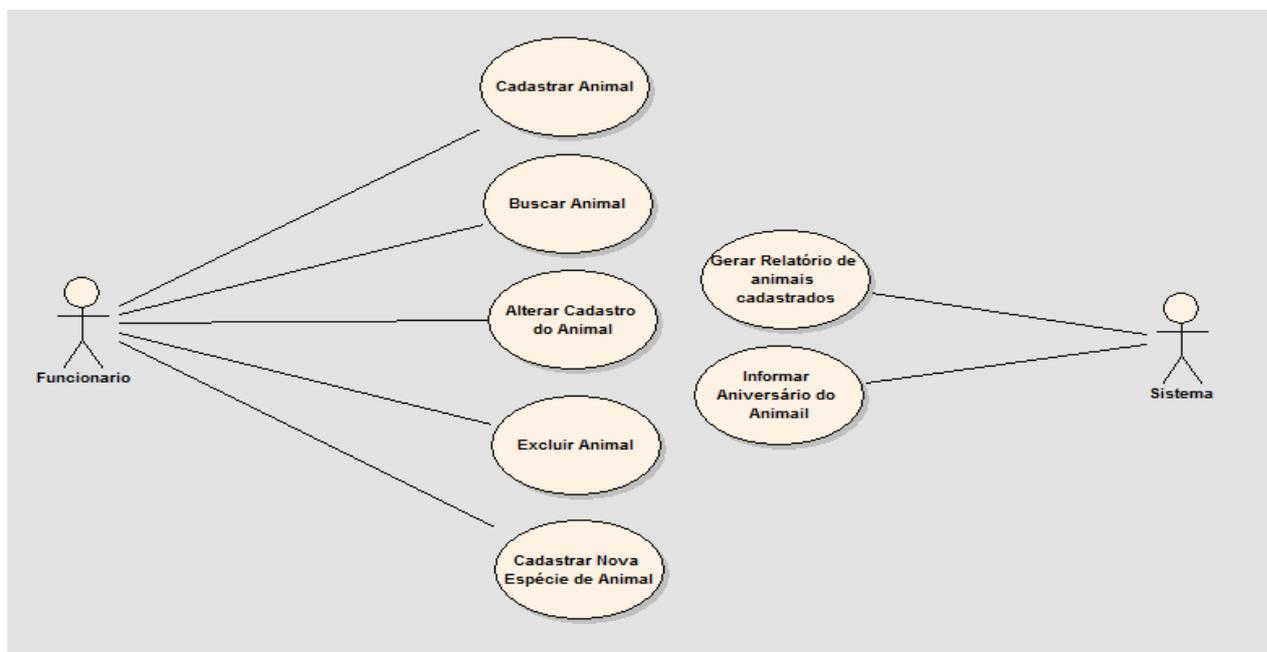


Figura 03 – Diagrama de caso de uso com ações pertinentes à registro de animais no sistema.

Fonte: Próprio Autor (2013).

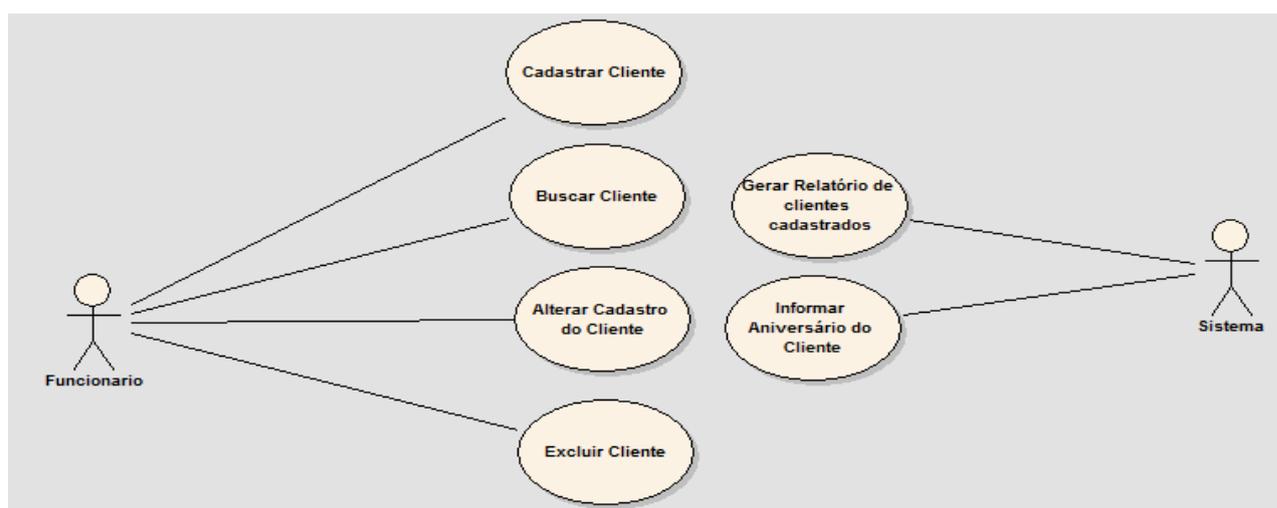


Figura 04 – Diagrama de caso de uso com ações pertinentes à registro de clientes no sistema.

Fonte: Próprio Autor (2013).

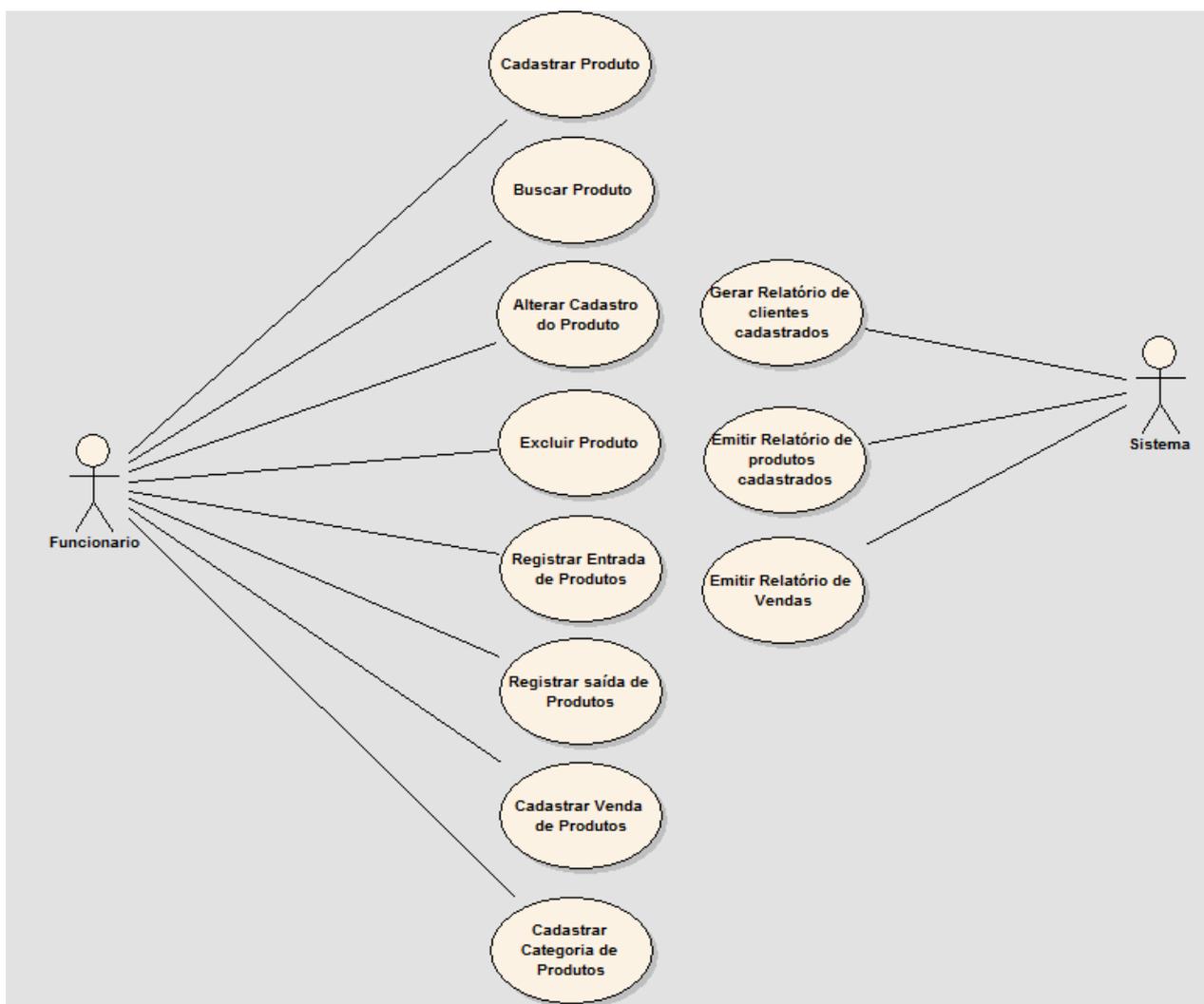


Figura 05 – Diagrama de caso de uso com ações pertinentes à registro de produtos no sistema.

Fonte: Próprio Autor (2013).



Figura 06 – Diagrama de caso de uso com ações pertinentes à registro de funcionários no sistema.

Fonte: Próprio Autor (2013).

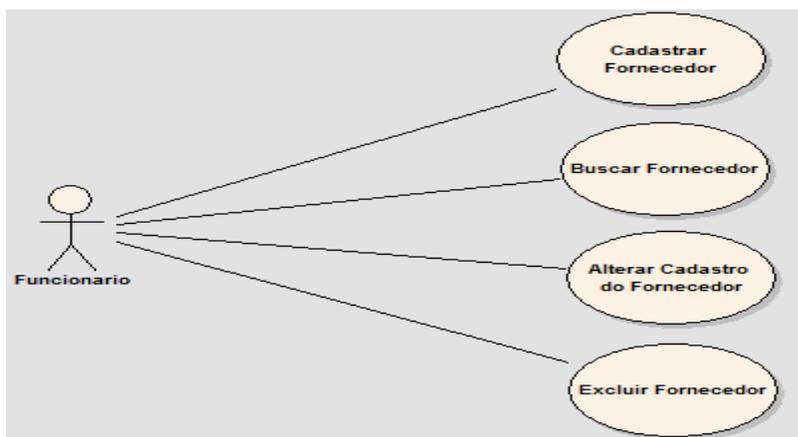


Figura 07 – Diagrama de caso de uso com ações pertinentes à registro de fornecedores no sistema.

Fonte: Próprio Autor (2013).

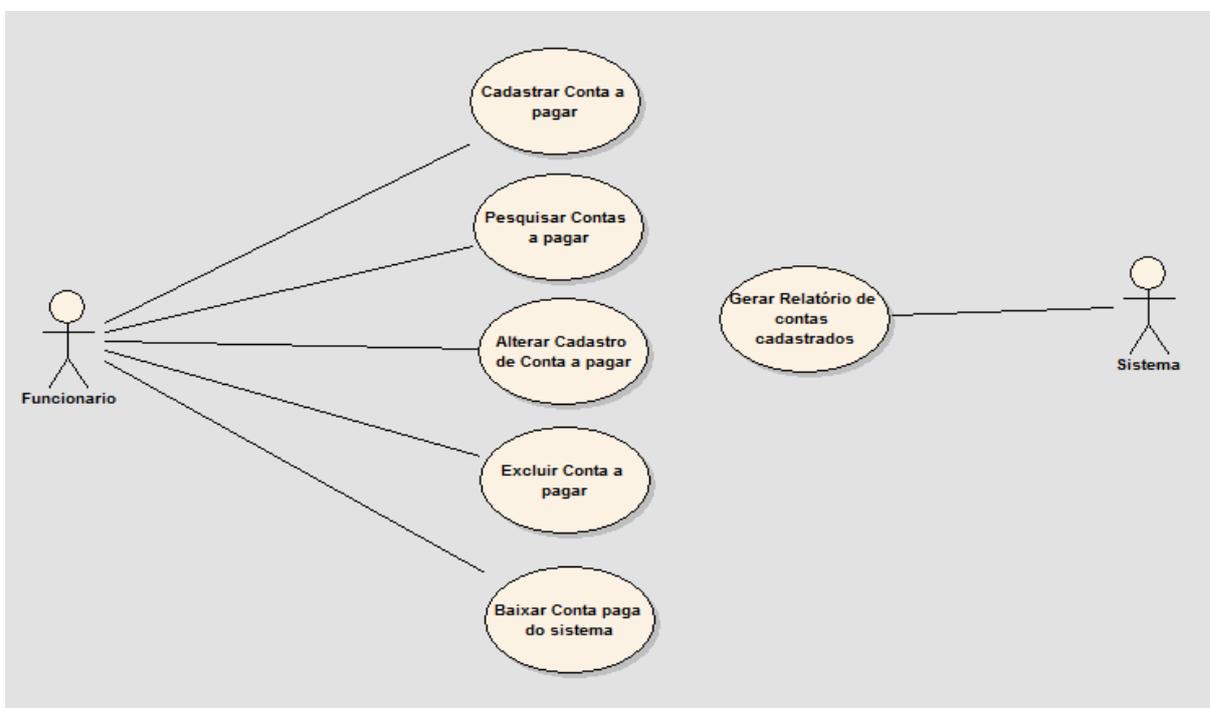


Figura 08 – Diagrama de caso de uso com ações pertinentes à registro de Contas a Pagar no sistema.

Fonte: Próprio Autor (2013).

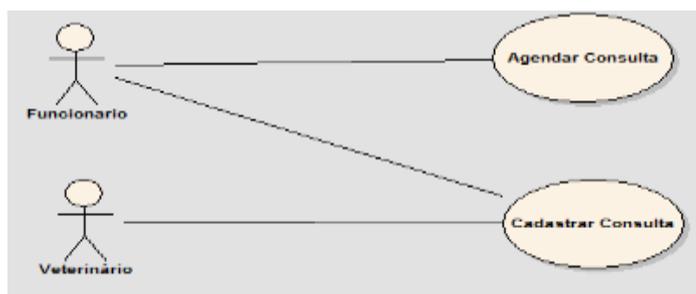


Figura 09 – Diagrama de caso de uso com ações pertinentes à registro de Consultas no sistema.

Fonte: Próprio Autor (2013).

### 3.7 Diagrama Entidade Relacionamento

Diagrama entidade relacionamento é um modelo diagramático que descreve o modelo de dados de um sistema com alto nível de abstração. Ele é a principal representação do Modelo de Entidades e Relacionamentos. Sua maior aplicação é para visualizar o relacionamento entre tabelas, no qual as relações são constituídas através da associação de um ou mais atributos destas tabelas que farão parte do projeto físico do banco de dados. A Figura 04 apresenta o diagrama que foi construído com a ajuda da ferramenta SQL Server Management Studio:

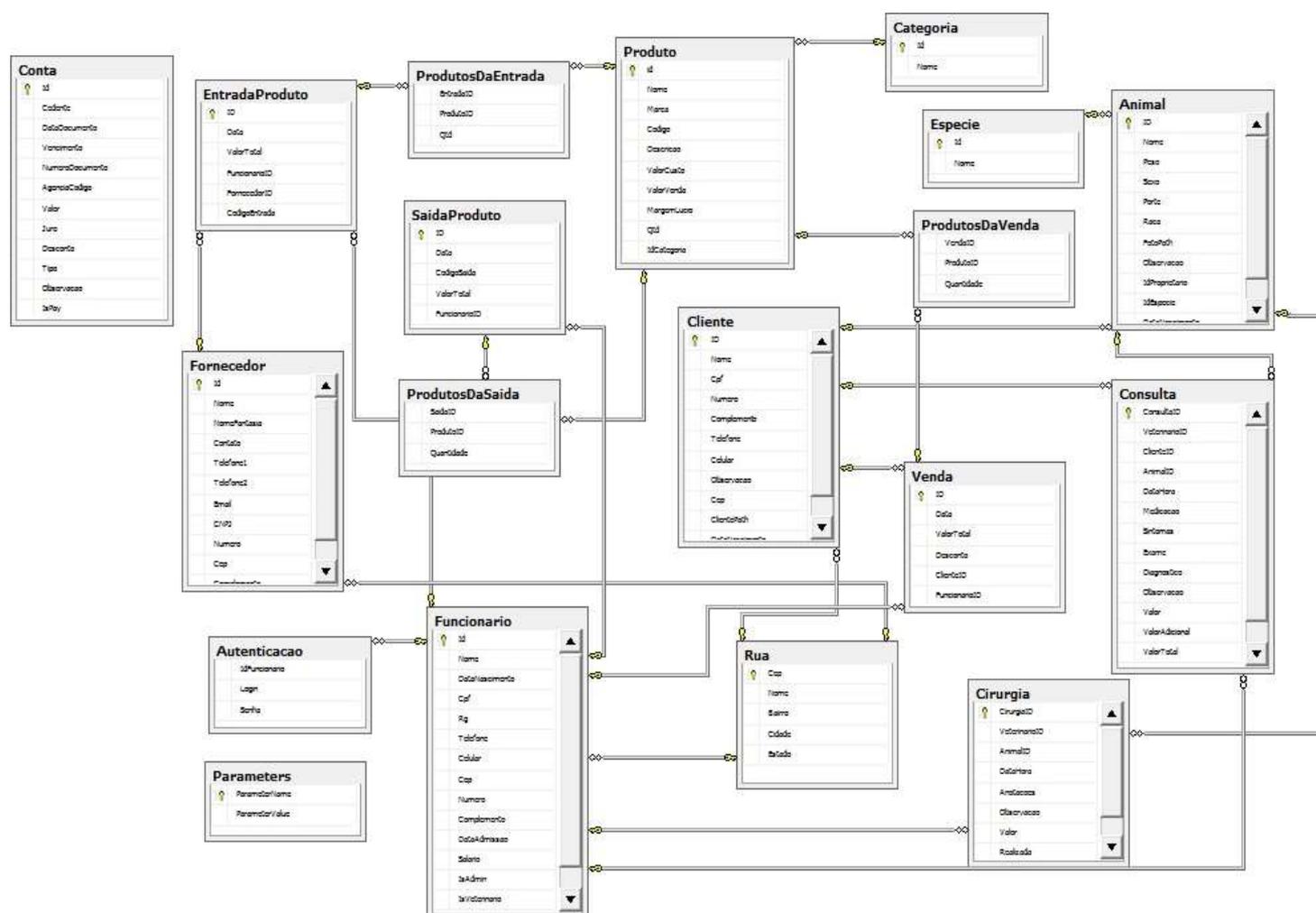


Figura 10 – Modelo Entidade Relacionamento.

Fonte: Próprio Autor (2013).

### 3.8 Diagrama de Classe

Conforme Booch et. all., (2000), os diagramas de classe são os diagramas mais encontrados com maior frequência na modelagem de sistemas orientados a objetos. Um diagrama de classes mostra um conjunto de dados, interfaces e colaborações e seus relacionamentos.

Segundo FURLAN (1998 apud MIOTTO, 2006) trata-se de uma estrutura lógica estática em uma superfície de duas dimensões mostrando uma coleção de elementos declarativos de modelo como classes, tipos e seus respectivos conteúdos e relações.

A Figura 05 apresenta o diagrama que foi construído com a ajuda da ferramenta Visual Studio 2010

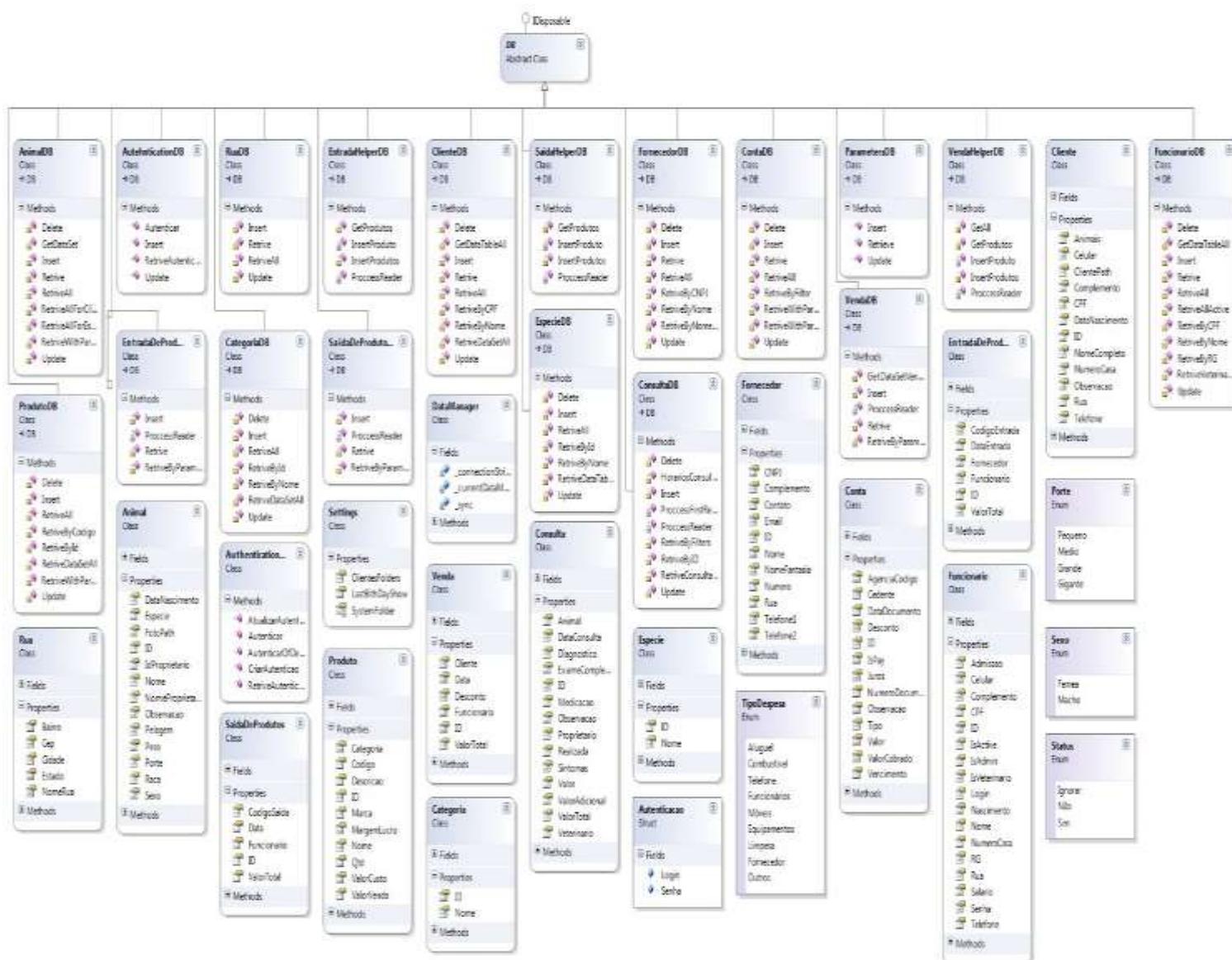


Figura 11– Diagrama de classe do sistema.

Fonte: Próprio Autor (2013).

### 3.8 Interfaces do sistema

Neste capítulo serão apresentadas as interfaces do sistema e trechos do código fonte do sistema.



Figura 12 – Interface inicial do sistema.

Fonte: Próprio Autor (2013).

A figura 12 apresenta a interface inicial do sistema de onde é possível acessar todos os módulos disponíveis no software. Nesta etapa é possível acessar todas as funcionalidades do sistema através do menu presente na parte superior da tela. Os módulos estão distribuídos em grupos no menu: Arquivos, Cadastros, Consultas, Pré-vendas, Contas, Relatórios e Aniversariantes.



Figura 13 – Menu Arquivo do Sistema.

Fonte: Próprio Autor (2013).

A figura 13 apresenta a o menu Arquivo selecionado no sistema. Neste menu é possível acessar propriedades auxiliares para cadastro de Produtos e Animais, além da opção Sair do sistema.



Figura 14 – Menu Arquivo - Animais.

Fonte: Próprio Autor (2013).

A figura 14 apresenta o menu Arquivo\Animais do sistema onde é possível remover espécies de animais cadastradas no sistema.



Figura 15 – Remover Espécie.

Fonte: Próprio Autor (2013).

A figura 15 apresenta a tela de remoção de espécie de animais cadastrados no sistema. Ao remover uma espécie, todos os registros de animais vinculados a esta espécie serão automaticamente excluídos do sistema.

Abaixo é apresentado trecho do código fonte responsável pela remoção de espécie do sistema:

```
private void btnRemove_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Deseja realmente excluir " + cmbEspecies.SelectedItem +
        "\r\nOBS: Ao remover esta espécie todos os animais relacionados a ela serão automaticamente excluídos do sistema!",
        "Atenção", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
    {
        try
        {
            foreach (Animal animal in Animal.RetrieveAllForEspecie((Especie)cmbEspecies.SelectedItem))
                animal.Delete();
            Especie.Delete(((Especie)cmbEspecies.SelectedItem).ID);
            MessageBox.Show("Espécie excluída com sucesso!", "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Information);
            CarregarEspecies();
            if (cmbEspecies.Items.Count != 0)
                cmbEspecies.SelectedIndex = 0;
            else
                this.Close();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```



Figura 16 – Menu Arquivos - Produtos.

Fonte: Próprio Autor (2013).

A figura 16 apresenta o menu Arquivo\Produtos do sistema onde é possível remover categorias de produtos cadastradas no sistema, realizar entrada e saídas do sistema.



Figura 17 – Remover Categoria.

Fonte: Próprio Autor (2013).

A figura 17 apresenta a tela de remoção de Categoria de produtos cadastrados no sistema. Ao remover uma categoria, todos os registro de produtos vinculados a esta categoria será automaticamente excluído do sistema.

A seguir é apresentado o trecho do código fonte responsável pela remoção de categorias de produtos do sistema:

```

private void btnRemove_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Deseja realmente excluir " + cmbCategoria.SelectedItem +
        " do sistema?\r\nOBS: Ao remover esta categoria todos os produtos relacionados a ela serão automaticamente excluídos do sistema!",
        "Atenção", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
    {
        try
        {
            Categoria.Delete(((Categoria)cmbCategoria.SelectedItem).ID);
            MessageBox.Show("Categoria excluída com sucesso!", "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Information);
            CarregarCategorias();
            if (cmbCategoria.Items.Count != 0)
                cmbCategoria.SelectedIndex = 0;
            else
                this.Close();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
}

```

Funcionário	Fornecedor	Data	Valor Total
Vendedor	Fornecedor de Ração	13/11/2013 13:55	R\$ 1125.00

Figura 18 – Controle de Entrada de Produtos.

Fonte: Próprio Autor (2013).

A figura 18 apresenta a tela de Controle de Entrada de Produtos, onde é possível buscar as entradas realizadas no sistema através dos filtros disponíveis, cadastrar novas entradas de produtos e visualizar entrada cadastradas.

A seguir é apresentado um trecho do código fonte responsável pela pesquisa de entradas no sistema:

```

private void Pesquisar()
{
    try
    {
        entradas = EntradaDeProdutos.RetrieveByParameters(ckFuncionario.Checked, (Funcionario)cmbFuncionarios.SelectedItem,
            ckFornecedor.Checked, fornec, ckDataInicial.Checked, dtDataInicial.Value, ckDataFinal.Checked, dtDataFinal.Value);
        LoadDataGrid();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Controle de Entrada de Produtos", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
}

```

Figura 19 – Cadastro de Entrada de Produtos.

Fonte: Próprio Autor (2013).

A figura 19 apresenta a tela de cadastro de entradas de produtos, onde seleciona-se o funcionário, o fornecedor e uma lista de produtos com suas respectivas quantidades para a entrada no estoque.

A seguir é apresentado um trecho do código fonte responsável pela entrada de produtos no sistema:

```

private void btnSave_Click(object sender, EventArgs e)
{
    if (ValidEntrada())
    {
        try
        {
            entrada.Save();
            MessageBox.Show("Entrada de produtos realizada com sucesso!", "Cadastro de Entrada de Produtos", MessageBoxButtons.OK, MessageBoxIcon.Information);
            ClearAll();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Ocorreu um erro ao salvar entrada no Banco de Dados. Desc.: " + ex.Message, "Atenção", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
}

```

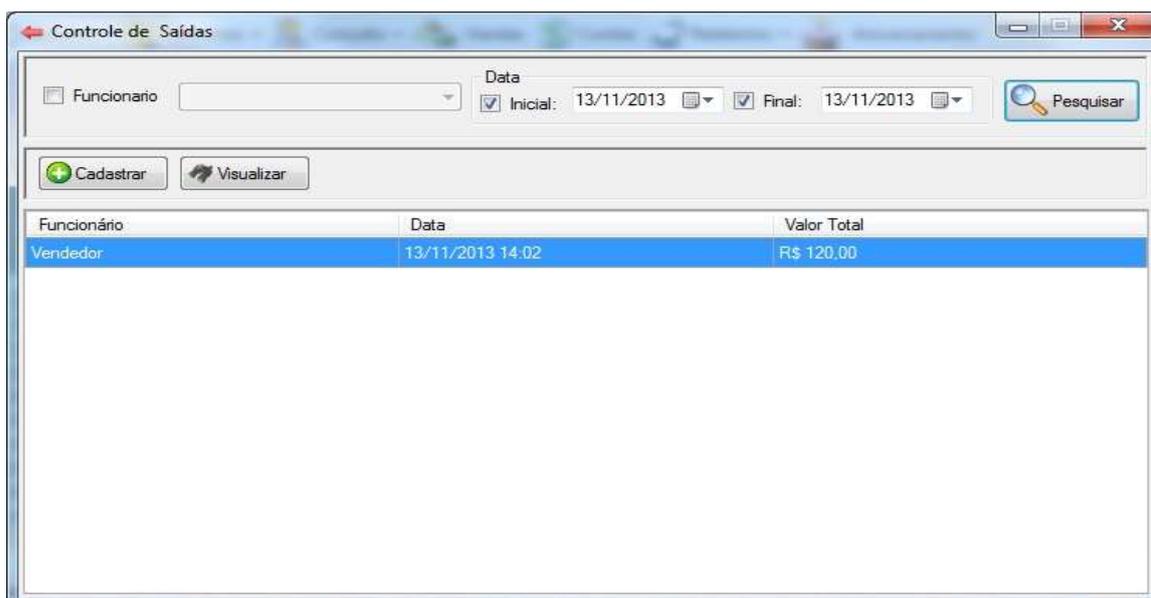


Figura 20 – Controle de Saída de Produtos.

Fonte: Próprio Autor (2013).

A figura 20 apresenta a tela de Controle de Saída de Produtos, onde é possível buscar as saídas realizadas no sistema através dos filtros disponíveis, cadastrar novas saídas de produtos e visualizar saídas cadastradas.

Abaixo é apresentado um trecho do código fonte responsável pelo controle de saídas de produtos.

```
private void Pesquisar()
{
    try
    {
        saidas = SaidaDeProdutos.RetrieveByParameters(ckFuncionario.Checked,
            (Funcionario)cmbFuncionarios.SelectedItem, ckDataInicial.Checked,
            dtDataInicial.Value, ckDataFinal.Checked, dtDataFinal.Value);
        LoadDataGrid();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Controle de Entrada de Produtos", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

The screenshot shows a software window titled "Saída de Produtos". At the top, there are three input fields: "Data" (13/11/2013), "Funcionário" (Vendedor), and "Codigo da saída" (123654789). Below these is a section for "Produtos" with a "Codigo do Produto" field and a search icon. A table titled "Lista de produtos" contains one row with the following data:

Código	Nome	R\$ Custo	Qty.	R\$ Total
123456789	Qualquer	R\$ 15,00	8	R\$ 120,00

At the bottom of the window, there is a "Remover" button, a summary "R\$ Total R\$ 120,00", and "Salvar" and "Fechar" buttons.

Figura 21 – Cadastro de Saída de Produtos.

Fonte: Próprio Autor (2013).

A figura 21 apresenta a tela de cadastro de saída de produtos, onde seleciona-se o funcionário e uma lista de produtos com suas respectivas quantidades para a saída no estoque.

A seguir é apresentado um trecho do código fonte responsável pelo cadastro de saídas de produto do sistema:

```
private void btnSave_Click(object sender, EventArgs e)
{
    if (ValidEntrada())
    {
        try
        {
            saida.Save();
            MessageBox.Show("Saída de produtos realizada com sucesso!", "Cadastro de Saída de Produtos", MessageBoxButtons.OK, MessageBoxIcon.Information);
            ClearAll();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Ocorreu um erro ao salvar Saída no Banco de Dados. Desc.: " + ex.Message, "Atenção", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```



Figura 22 – Menu de Cadastros.

Fonte: Próprio Autor (2013).

A figura 22 apresenta o menu de cadastros do sistema, onde é possível realizar o cadastro de animais, clientes, funcionários, produtos e fornecedores.



Figura 23 – Controle de Cadastro de Fornecedores.

Fonte: Próprio Autor (2013).

A figura 23 apresenta a tela de controle dos cadastros de fornecedores. Nesta etapa é possível adicionar, editar e excluir fornecedores do sistema, além de disponibilizar um filtro de pesquisa com o intuito de facilitar a localização do registro no sistema.

A seguir é apresentado um trecho do código fonte responsável pelo controle de fornecedores do sistema.

```
private void btnPesquisar_Click(object sender, EventArgs e)
{
    try
    {
        LoadSelected();
        LoadGrid();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Figura 24 – Cadastro de Fornecedores.

Fonte: Próprio Autor (2013).

A figura 24 apresenta a tela de cadastro e edição de fornecedores no sistema. Nesta tela é possível salvar os dados pertinentes a cada fornecedor no sistema.

A seguir é apresentado um trecho do código fonte responsável pelo cadastro de fornecedores no sistema:

```
private void btnConfirm_Click(object sender, EventArgs e)
{
    try
    {
        GerarRua();
        rua.Save();
        GerarFornecedor();
        fornecedor.Save();
        MessageBox.Show(fornecedor.Nome + " salvo com sucesso! ID: " + fornecedor.ID, "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Information);
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}
```

ID	Código	Nome	Marca	Categoria	R\$ Custo	R\$ Venda	Quantidade
1	123456789	Qualquer	asdfasdf	Outros	R\$ 15,00	R\$ 30,00	71

Figura 25 – Controle de Cadastro de Produtos.

Fonte: Próprio Autor (2013).

A figura 25 apresenta a interface de controle de cadastro de produtos. Neste módulo é possível adicionar, editar e remover registro de produtos no sistema, além de disponibilizar alguns filtros de pesquisa para facilitar a localização do registro de desejado.

A seguir é apresentado um trecho do código fonte responsável pelo controle de produtos no sistema:

```
public void LoadProdutos()
{
    try
    {
        if (rbId.Checked)
            LoadById();
        else if (rbCodigo.Checked)
            LoadByCodigo();
        else if (rbFiltros.Checked)
            LoadByFiltros();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);
        produtos = new List<Produto>();
    }
    LoadGrid();
}
```

Figura 26 – Cadastro de Produtos.

Fonte: Próprio Autor (2013).

A figura 26 apresenta a interface de cadastro de produtos do sistema. Nesta etapa é possível cadastrar informações pertinentes a cada produto a ser cadastrado. Também é possível cadastrar uma nova categoria de produtos ao clicar no botão adicionar, conforme a figura 27.

Figura 27 – Cadastro de Categoria.

Fonte: Próprio Autor (2013).

A seguir é apresentado um trecho do código fonte responsável pelo cadastro de novos produtos no sistema:

```
private void btnConfirm_Click(object sender, EventArgs e)
{
    try
    {
        GerarProduto();
        produto.Save();
        MessageBox.Show(produto.Nome + " salvo com sucesso! ID: " + produto.ID, "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Information);
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}
```

A seguir é apresentado um trecho do código fonte responsável pelo cadastro de categoria de produtos no sistema:

```
private void btnAdd_Click(object sender, EventArgs e)
{
    try
    {
        ValidarCategoria();
        Categoria.Save();
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}
```

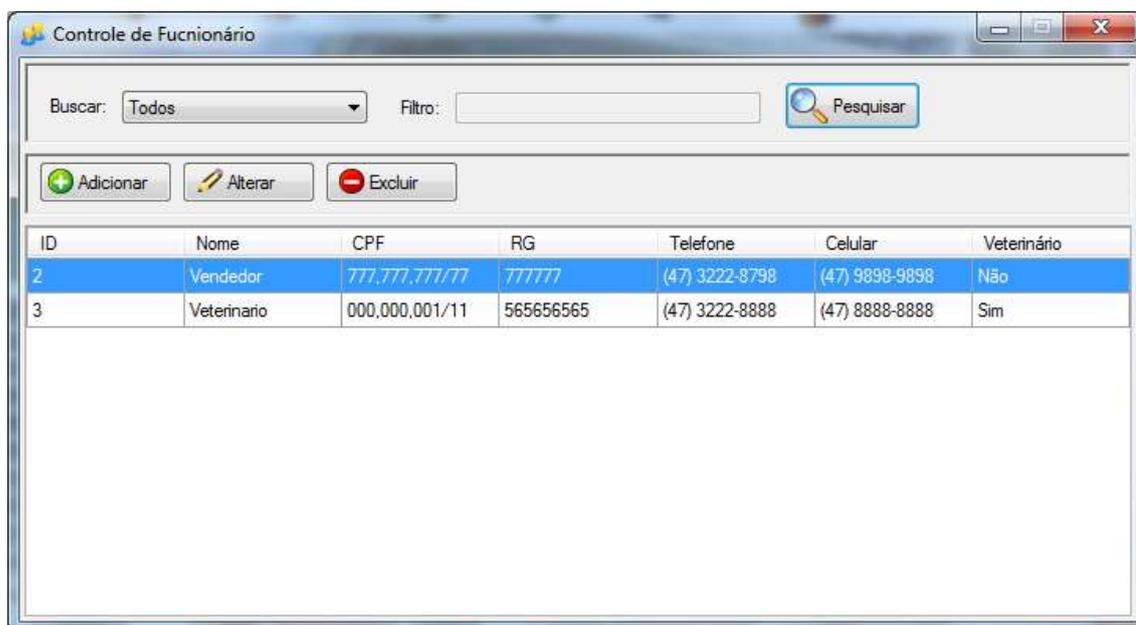


Figura 28 – Controle de Cadastro de Funcionários.

Fonte: Próprio Autor (2013).

A figura 28 apresenta a interface de controle de cadastros de funcionários. Nesta etapa é possível inserir, alterar e excluir registros de funcionários no sistema, além de disponibilizar filtros para pesquisa, para facilitar a localização dos registros.

A seguir é apresentado um trecho do código fonte do controle de funcionário no sistema:

```
private void btnPesquisar_Click(object sender, EventArgs e)
{
    try
    {
        LoadSelected();
        LoadGrid();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Figura 29 – Cadastro de Funcionário.

Fonte: Próprio Autor (2013).

A figura 29 apresenta a interface de cadastro de funcionários. Nesta etapa é possível cadastrar informações pertinentes a registros de funcionários no sistema.

A seguir é apresentado um trecho do código fonte responsável pelo cadastro de novos funcionários no sistema:

```
private void btnConfirm_Click(object sender, EventArgs e)
{
    try
    {
        GerarRua();
        rua.Save();
        GerarFuncionario();
        funcionario.Save();
        MessageBox.Show(funcionario.Nome + " salvo com sucesso! ID: " + funcionario.ID, "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Information);
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}
```

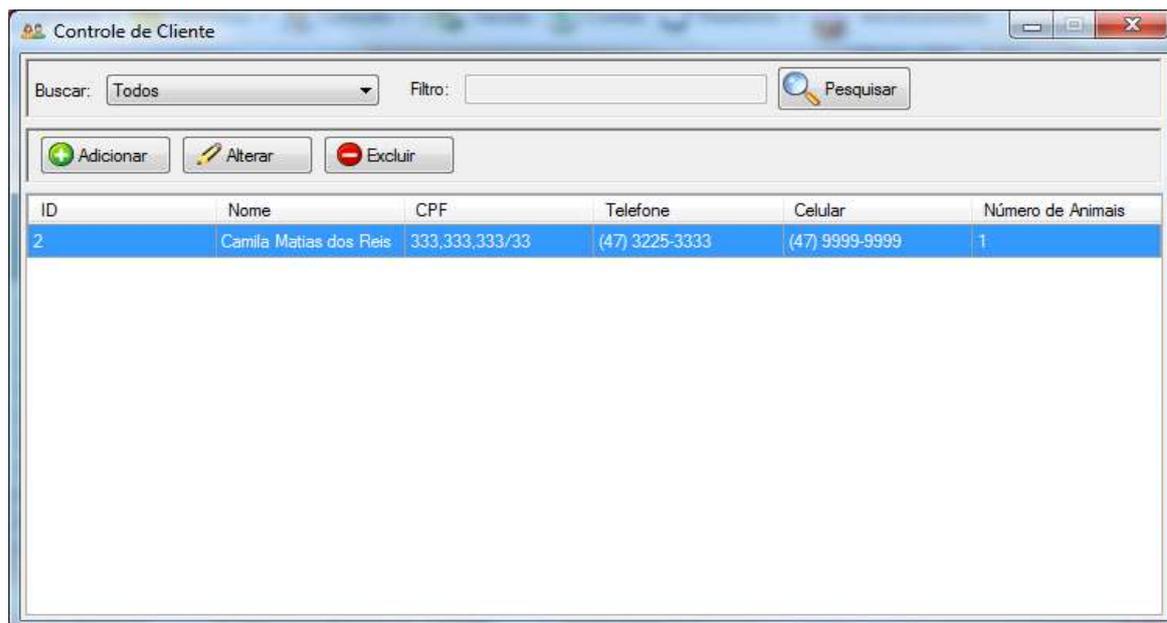


Figura 30 – Controle de Cadastro de Clientes.

Fonte: Próprio Autor (2013).

A figura 30 apresenta a interface de controle de Cadastro de Clientes. Nesta etapa é possível inserir editar e excluir registros de clientes do sistema, além de disponibilizar filtros de pesquisa para facilitar a localização de registros previamente cadastrados.

A seguir é apresentado um trecho do código fonte responsável pelo controle de clientes no sistema:

```
private void btnPesquisar_Click(object sender, EventArgs e)
{
    try
    {
        LoadSelected();
        LoadGrid();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

**Cadastro de Cliente**

**Dados Pessoais**

Nome:

CPF:  Tel.:  Cel.:  Data Nasc.:

**Endereço**

CEP:

Rua:  Numero:

Bairro:  Cidade:  Estado:

Complemento:

Observação:

Animais

Nome	Espécie	Raça	Pelagem
Lucky	Cachorro	YorkShire	Preta e amarela

Figura 31 –Cadastro de Clientes.

Fonte: Próprio Autor (2013).

A figura 31 apresenta a interface de cadastro de clientes. Nesta etapa é possível cadastrar informações pertinentes a clientes cadastrados do sistema além de poder inserir, editar e excluir animais nos quais o cliente é proprietário.

A seguir é apresentado um trecho do código fonte responsável pelo cadastro de novos clientes no sistema:

```
private void btnConfirmarCliente_Click(object sender, EventArgs e)
{
    try
    {
        GerarRua();
        rua.Save();
        GerarCliente();
        cliente.Save();
        MessageBox.Show(cliente.NomeCompleto + " salvo com sucesso! ID: " + cliente.ID, "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Information);
        SetValidationText();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}
```

ID	Nome	Proprietário	Espécie	Raça	Pelagem
2	Lucky	Camila Matias dos ...	Cachorro	York-Shire	Preta e amarela

Figura 32 – Controle de Cadastro de Animais.

Fonte: Próprio Autor (2013).

A figura 32 apresenta a interface de cadastro de animais. Nesta etapa é possível inserir, editar e remover registros de animais do sistema. Além de disponibilizar filtros de pesquisa para facilitar a localização dos registros previamente cadastrados.

Para inserir um novo registro de animal no sistema é necessário, obrigatoriamente, selecionar um cliente previamente cadastrado, sendo assim, ao clicar no botão adicionar será apresentada uma tela de seleção de clientes cadastrados, conforme apresenta a figura 33.

A seguir é apresentado um trecho do código fonte responsável pelo controle de animais cadastrados no sistema:

```
public void LoadAnimais()
{
    string nome = txtNome.Text;
    string raca = txtRaca.Text;
    Porte porte = (Porte)Enum.Parse(typeof(Porte), cmbPorte.SelectedIndex.ToString());
    Sexo sexo = (Sexo)Enum.Parse(typeof(Sexo), cmbSexo.SelectedIndex.ToString());
    bool byEspecie = ckEspecie.Checked;
    bool byCliente = ckCliente.Checked;
    Especie especie = null;
    if (byEspecie)
        especie = (Especie)cmbEspecie.SelectedItem;

    try
    {
        animais = Animal.RetrieveWithParameters(nome, raca, porte, sexo, byEspecie, especie, byCliente, cliente);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);
        animais = new List<Animal>();
    }
    LoadGrid();
}
```

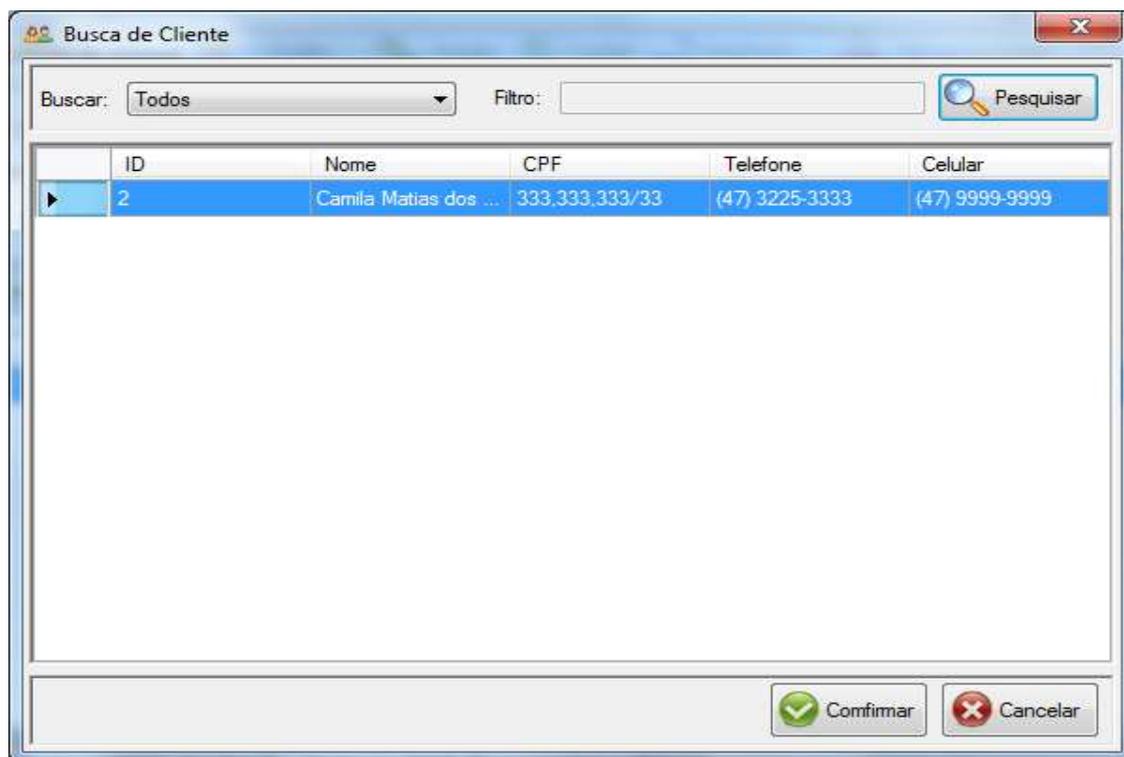


Figura 33 – Busca de clientes cadastrados.

Fonte: Próprio Autor (2013).

A seguir é apresentado um trecho do código fonte responsável pela busca de clientes cadastrados no sistema:

```
private void btnPesquisar_Click(object sender, EventArgs e)
{
    try
    {
        LoadSelected();
        LoadGrid();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

**Cadastro de Animal**

**Dados do animal**

Nome:

Espécie:

Raça:

Pelagem:

Nascimento:  Peso (KG):

Porte:  Sexo:

Observação:

Cadastro completo.

Figura 34 – Cadastro de Animais.

Fonte: Próprio Autor (2013).

A figura 34 apresenta a interface de cadastro de animais no sistema. Nesta etapa é possível cadastrar informações pertinentes a registros de animais a serem cadastrados no sistema. Também é possível adicionar uma imagem do animal através de uma busca por arquivos em diretórios do disco rígido, conforme apresenta a figura 35.

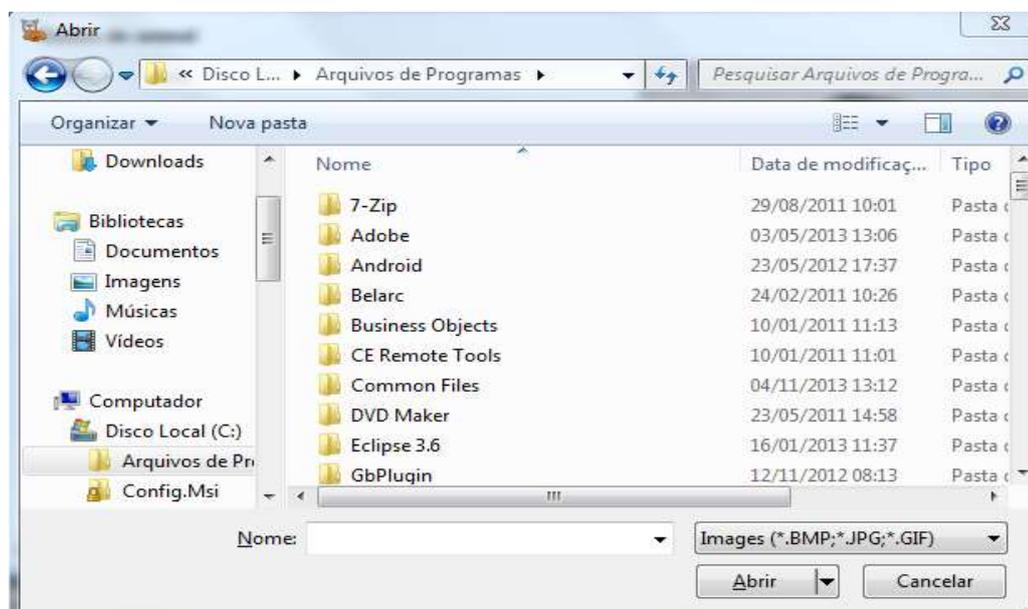


Figura 35 – Busca por arquivos de imagem nos Discos Rígidos.

Fonte: Próprio Autor (2013).

A seguir é apresentado um trecho do código fonte responsável pelo cadastro de animais no sistema:

```
private void btnConfirmarCliente_Click(object sender, EventArgs e)
{
    try
    {
        GerarAnimal();
        if (!String.IsNullOrEmpty(imagePath))
        {
            string fileName = String.Empty;
            if (String.IsNullOrEmpty(animal.FotoPath))
                fileName = GerarNomeDoArquivo(imagePath);
            else
                fileName = animal.FotoPath;
            string fullPath = Settings.ClientesFolders + proprietario.ClientePath + "\\\" + fileName;
            if (File.Exists(fullPath))
                File.Delete(fullPath);

            Image img = ResizeImage(picPhoto.Image, picPhoto.Size.Width, picPhoto.Size.Height);
            img.Save(fullPath, System.Drawing.Imaging.ImageFormat.Jpeg);
            img.Dispose();
            img = null;
            animal.FotoPath = fileName;
        }
        animal.Save();
        MessageBox.Show(animal.Nome + " salvo com sucesso! ID: " + animal.ID, "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Information);
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}
```

A seguir é apresentado um trecho do código fonte responsável pela busca de arquivos de imagem de animais no sistema operacional:

```
private void btnAddPhoto_Click(object sender, EventArgs e)
{
    imagePath = String.Empty;
    OpenFileDialog f = new OpenFileDialog();
    f.Filter = "Images (*.BMP;*.JPG;*.GIF)|*.BMP;*.JPG;*.GIF|All files (*.*)|*.*";
    if (DialogResult.OK == f.ShowDialog())
    {
        try
        {
            imagePath = f.FileName;
            if (picPhoto.Image != null)
                picPhoto.Image.Dispose();
            picPhoto.Image = System.Drawing.Image.FromFile(imagePath);
        }
        catch
        {
            imagePath = string.Empty;
            MessageBox.Show("Arquivo de imagem inválido!", "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
    }
}
```



Figura 36 – Menu de Consulta.

Fonte: Próprio Autor (2013).

A figura 36 apresenta a interface de seleção do menu de consulta na tela principal do sistema. Nesta etapa é possível pesquisar consultas cadastradas, cadastrar nova consulta, agendar consulta para um veterinário previamente cadastrado e visualizar a agenda de um veterinário.

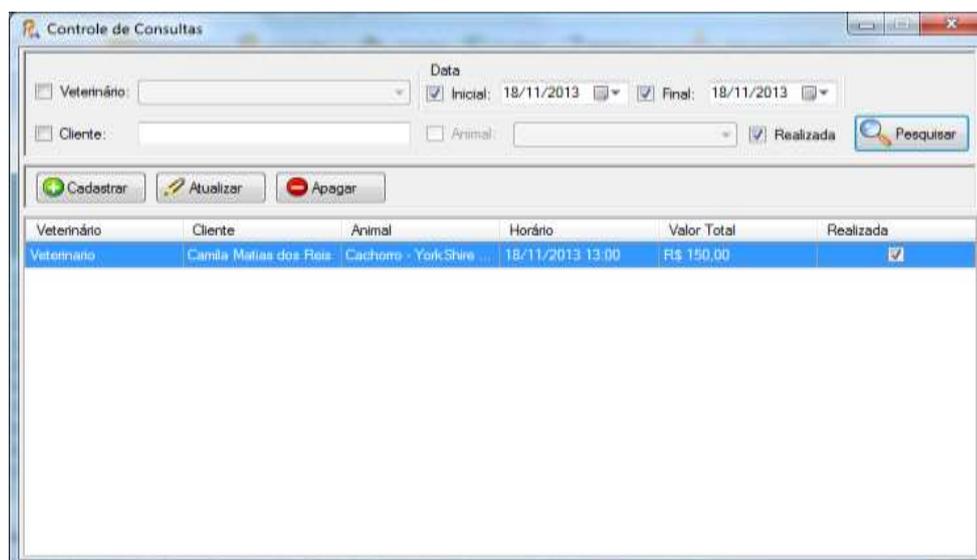


Figura 37 – Controle de Cadastro de Consultas.

Fonte: Próprio Autor (2013).

A figura 37 apresenta a interface de controle de consultas cadastradas no sistema. Nesta etapa é possível cadastrar, atualizar e excluir registros previamente cadastrados no sistema, além de disponibilizar filtros de pesquisa para facilitar a localização dos registros cadastrados.

A interface 'Cadastro de Consulta' apresenta os seguintes campos e botões:

- Dados Veterinário:** Campo 'Veterinário' com o valor 'Veterinario', campo 'Data' com '18/11/2013' e campo 'Hora' com '13:00'. Botão 'Buscar Horário'.
- Dados Cliente:** Campo 'CPF' com '333.333.333/33' e botão 'Pesquisar', campo 'Cliente' com 'Camila Matias dos Reis', campo 'Animal' com 'Cachorro - YorkShire - Lucky' e botão 'Adicionar'.
- Dados da Consulta:** Campo 'Valor' com '150.00', campo 'Valor Adicional' com '0.00' e campo 'Valor Total' com '150.00'. Botões 'Medicações', 'Exames', 'Sintomas' e 'Diagnóstico'.
- Botões 'Salvar' e 'Fechar' na base da janela.

Figura 38 – Cadastro de Consultas.

Fonte: Próprio Autor (2013).

A figura 38 apresenta a interface de cadastro de consulta. Nesta etapa é possível cadastrar informações pertinente a consulta mp sistema.

A interface 'Agendar Consulta' apresenta os seguintes campos e botões:

- Dados Veterinário:** Campo 'Veterinário' com o valor 'Veterinario', campo 'Data' com '18/11/2013' e campo 'Hora' com '17:00'. Botão 'Buscar Horário'.
- Dados Cliente:** Campo 'CPF' com '333.333.333/33' e botão 'Pesquisar', campo 'Cliente' com 'Camila Matias dos Reis', campo 'Animal' com 'Cachorro - YorkShire - Lucky' e botão 'Adicionar'.
- Botões 'Agendar' e 'Cancelar' na base da janela.

Figura 39 – Agendamento de Consultas.

Fonte: Próprio Autor (2013).

A figura 39 apresenta a interface de agendamento de consultas. Nesta etapa é possível agenda um horário para consulta com um veterinário previamente cadastrado no sistema,

relacionando este agendamento a um cliente e um animal também cadastrados. Ao clicar no botão Buscar Horário, serão listados os horários disponíveis do veterinário para o agendamento da consulta, conforme apresenta a figura 40.

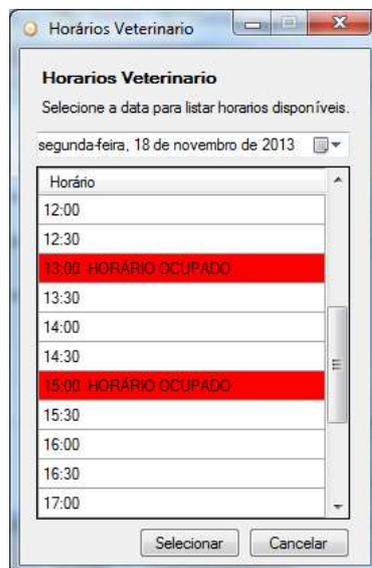


Figura 40 – Busca de Horário Disponível para Consulta com Veterinário.

Fonte: Próprio Autor (2013).

A seguir é apresentado um trecho do código fonte responsável pelo agendamento de novas consultas no sistema:

```
private void btnAgendar_Click(object sender, EventArgs e)
{
    try
    {
        if (_consulta == null)
            _consulta = new Consulta();
        _consulta.Veterinario = (Funcionario)cmbVeterinario.SelectedItem;
        _consulta.Animal = (Animal)cmbAnimal.SelectedItem;
        _consulta.Proprietario = cliente;
        _consulta.DataConsulta = Horario;
        _consulta.Save();
        MessageBox.Show("Consulta agendada com sucesso para \r\n\t" + _consulta.DataConsulta.ToString("dd/MM/yyyy HH:mm"),
            "Angedar Consulta", MessageBoxButtons.OK, MessageBoxIcon.Information);
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Erro ao agendar consulta. Descrição: " + ex.Message, "Angedar Consulta", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

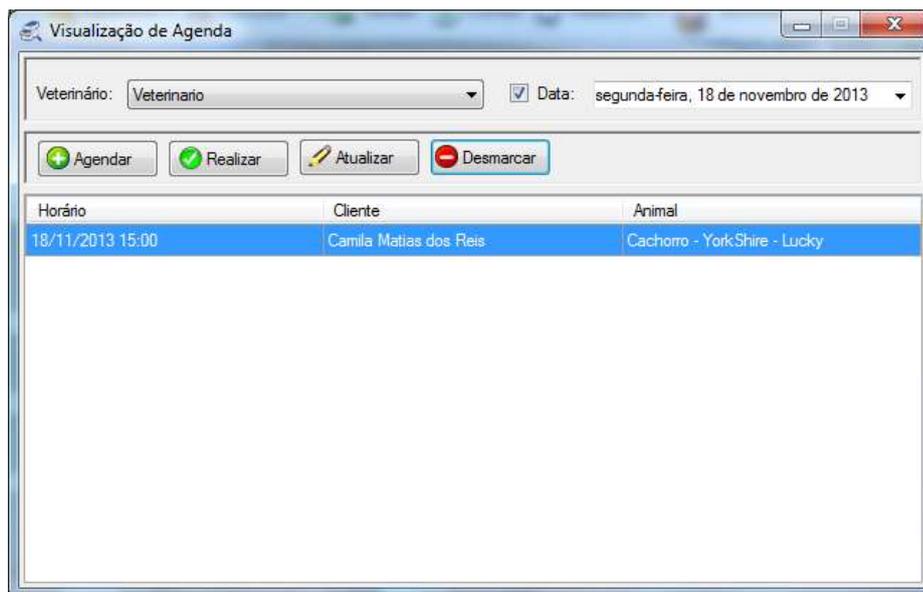


Figura 41 – Visualização de Agenda de um Veterinário.

Fonte: Próprio Autor (2013).

A figura 41 apresenta a interface e visualização da agenda de um veterinário, Nesta etapa é possível selecionar um veterinário e uma data específica para visualizar as consultas agendadas para o veterinário cadastrado no sistema. Também é possível agendar nova consulta, realizar consulta previamente agendada, atualizar registro de consulta cadastrada e desmarcar consultas.

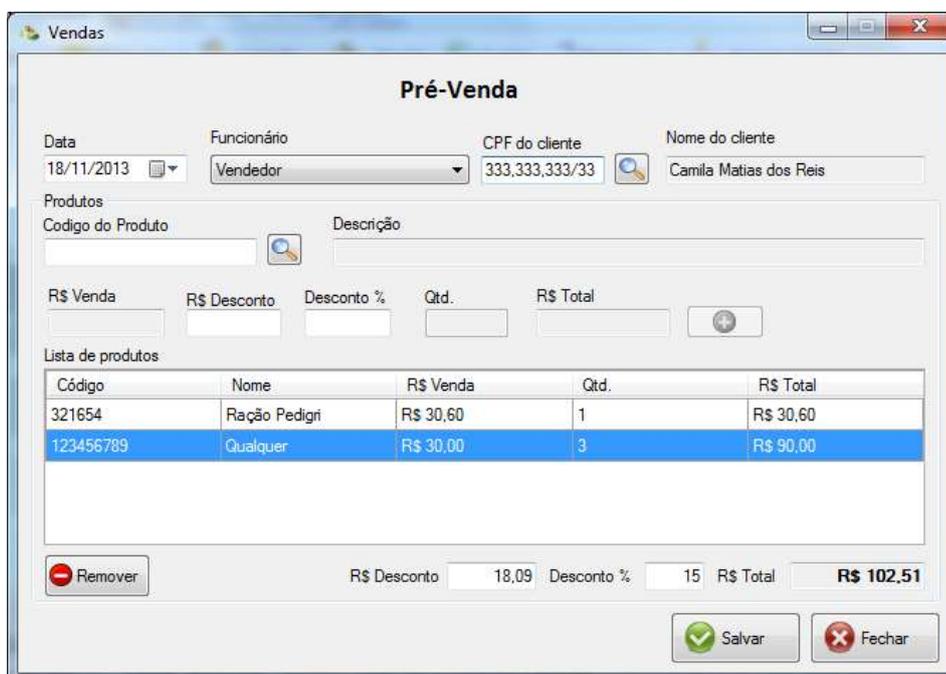


Figura 42 – Cadastro de pré-vendas.

Fonte: Próprio Autor (2013).

A figura 42 apresenta a interface de cadastro de pré-vendas no sistema. Nesta etapa é possível cadastrar registros de pré-vendas no sistema, selecionando um vendedor, um cliente e uma lista de produtos com suas respectivas quantidades previamente cadastrados no sistema, além de disponibilizar cálculos de desconto por valor e porcentagem para cada produto e no total da pré-venda. É possível clicar nos botões de pesquisa de cliente e de produtos para facilitar a localização dos respectivos registros conforme apresenta as figuras 43 e 44.

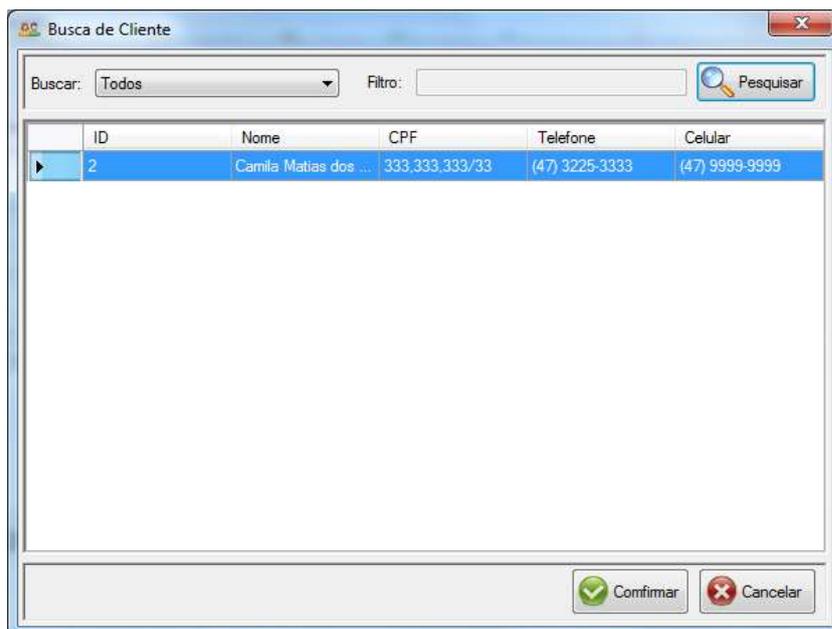


Figura 43 – Busca de Clientes.

Fonte: Próprio Autor (2013).

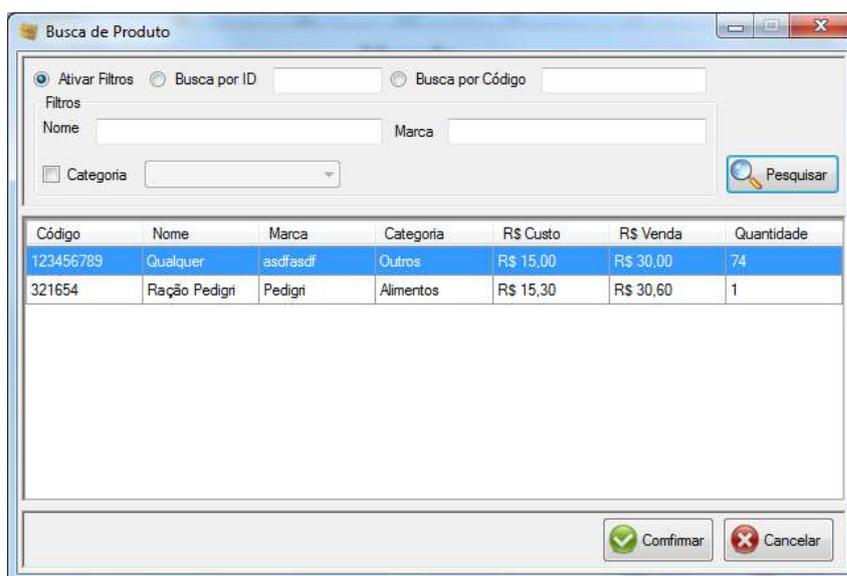


Figura 44 – Busca de Produtos.

Fonte: Próprio Autor (2013).

A seguir é apresentado um trecho do código fonte responsável pelo cadastro de pré-vendas no sistema:

```
private void btnSave_Click(object sender, EventArgs e)
{
    if (ValidVenda())
    {
        try
        {
            var aux = String.Format("{0:C}", 0).Replace("0", "").Replace(",", "").Replace(".", "").Replace(" ", "");
            venda.ValorTotal = Convert.ToDecimal(txtValorTotalEntrada.Text.Replace(aux, ""));
            venda.Save();
            MessageBox.Show("Saída de produtos realizada com sucesso!", "Cadastro de Vendas", MessageBoxButtons.OK, MessageBoxIcon.Information);
            ClearAll();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Ocorreu um erro ao Savar Venda no Banco de Dados. Desc.: " + ex.Message, "Atenção", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```

Cedente	Número do documento	Vencimento	Valor
Nova conta	123456789	07/11/2013	R\$ 550,00
Outra conta		07/11/2013	R\$ 300,00

Figura 45 – Controle de Cadastro de Constas.

Fonte: Próprio Autor (2013).

A figura 45 apresenta a interface de controle de cadastro de contas no sistema. Nesta etapa é possível inserir, alterar, e excluir registros de contas no sistema, além de disponibilizar filtros para pesquisa facilitando a localização dos mesmos. Também é possível marcar contas selecionada como paga selecionando-as no grid e clicando no botão pagar.

Figura 46 – Cadastro de Contas.

Fonte: Próprio Autor (2013).

A figura 46 apresenta a interface de cadastro de contas, onde é possível cadastrar informação pertinentes a contas a serem registradas no sistema.

A seguir é apresentado o trecho do código fonte responsável pelo cadastro de contas no sistema:

```
private void btnConfirm_Click(object sender, EventArgs e)
{
    try
    {
        GerarConta();
        if (payForm)
        {
            conta.Pagar();
            MessageBox.Show("Conta paga com sucesso!", "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            conta.Save();
            MessageBox.Show("Conta salva com sucesso!", "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }

        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Aviso", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}
```



Figura 47 – Menu de Relatórios.

Fonte: Próprio Autor (2013).

A figura 47 apresenta a seleção do menus de relatórios, onde é possível selecionar relatórios de animais, pré-vendas, produtos, clientes e contas na tela inicial do sistema.



Figura 48 – Filtro de relatório de animais.

Fonte: Próprio Autor (2013).

A figura 48 apresenta a interface de filtro para relatório de animais, onde pode-se selecionar um cliente específico para servir de parâmetro de busca para registro no sistema.

Nome	Cliente	Sexo	Peso	Raça	Especie	Data Nascimento
Ludy	Camilla Maria das R	Macho	6.30	York Shire	Cachorro	18/11/2011

Figura 49 – Relatório de Animais.

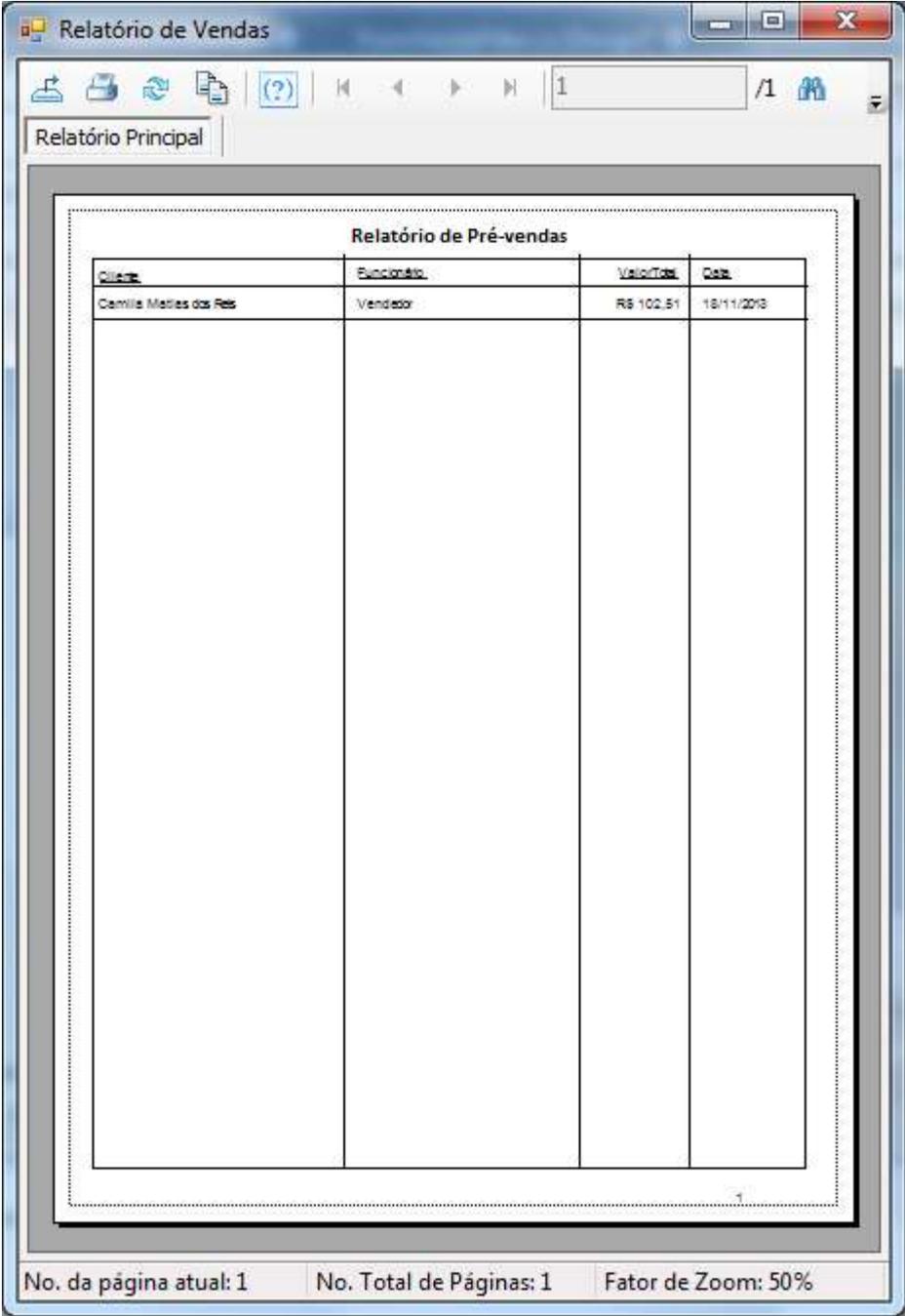
Fonte: Próprio Autor (2013).

A figura 49 apresenta a interface de apresentação do relatório de animais. Nesta etapa é possível navegar através dos registros contidos no relatório além de disponibilizar a função de imprimir o mesmo.

Figura 50 – Filtro para Relatório de pré-vendas.

Fonte: Próprio Autor (2013).

A figura 50 apresenta a interface de filtro para o relatório de pré-vendas. Nesta etapa é possível selecionar uma data, um funcionário e um cliente específico para emissão do relatório de pré-vendas.



Cliente	Funcionário	Valor Total	Data
Camilla Matias dos Reis	Vendedor	R\$ 102,51	18/11/2013

Figura 51 – Relatório de Pré-vendas.

Fonte: Próprio Autor (2013).

A figura 51 apresenta a interface de apresentação do relatório de pré-vendas. Nesta etapa é possível navegar através dos registros contidos no relatório além de disponibilizar a função de imprimir o mesmo.

Relatório de Produtos

Relatório Principal

Nome	Marca	Categoria	Código	Valor de Cust.	Valor de Venda	Qtd.
Qualquer	esotof	Outros	123456789	15,00	30,00	74
ração Feógi	Feógi	Alimentos	021684	15,00	30,00	1

No. da página atual: 1      No. Total de Páginas: 1      Fator de Zoom: 50%

Figura 52 – Relatório de Produtos.

Fonte: Próprio Autor (2013).

A figura 52 apresenta a interface de apresentação do relatório de produtos. Nesta etapa é possível navegar através dos registros contidos no relatório além de disponibilizar a função de imprimir o mesmo.



Figura 53 – Filtro para Relatório de Contas.

Fonte: Próprio Autor (2013).

A figura 53 apresenta a interface de filtro para o relatório de contas. Nesta etapa é possível selecionar uma data, um funcionário e um cliente específico para emissão do relatório de contas.

Cedente	Tipo	Juro	Descomb	Valor	Valor Total	Vencimento
Novo conta	Alugue	R\$ 12,00	R\$ 3,50	R\$ 550,00	R\$ 565,50	07/11/2013
Outra conta	Telefone	R\$ 10,00	R\$ 10,00	R\$ 300,00	R\$ 300,00	07/11/2013

Soma Total das contas: R\$ 565,50

No. da página atual: 1    No. Total de Páginas: 1    Fator de Zoom: 50%

Figura 54 – Relatório de Contas.

Fonte: Próprio Autor (2013).

A figura 54 apresenta a interface de apresentação do relatório de contas. Nesta etapa é possível navegar através dos registros contidos no relatório além de disponibilizar a função de imprimir o mesmo.

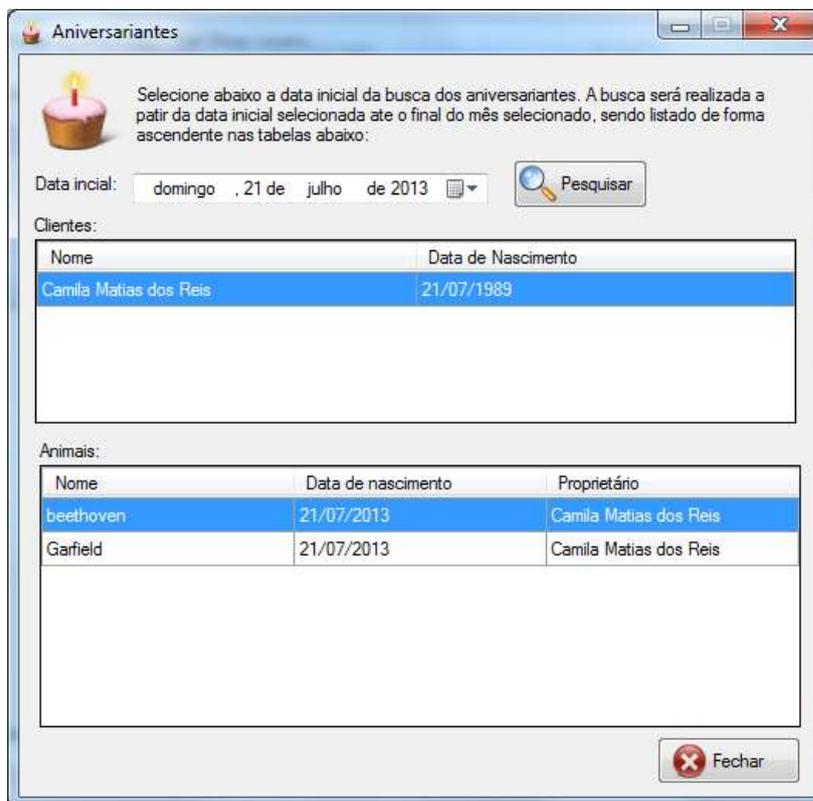


Figura 55 – Interface de Aniversariantes.

Fonte: Próprio Autor (2013).

A figura 55 apresenta a interface de apresentação de aniversariantes do dia com base na data selecionada no filtro disponível na tela. São apresentados os animais e clientes aniversariantes.

## IV CONCLUSÃO

No início deste trabalho tínhamos uma empresa que sua principal dificuldade e problema eram o armazenamento e consulta aos seus dados como clientes, animais, despesas e pré-vendas até a não existência de relatórios, entre outros requisitos. Os quais são de extrema importância para um crescimento e uma visão geral da situação da empresa.

Com o desenvolvimento deste Sistema Gerencial de Banco de Dados (SGBD), foi possível solucionar vários problemas com relação aos dados e informações que não haviam um lugar específicos para serem armazenados, os quais acabavam sendo perdidos prejudicando assim um futuro crescimento e o controle das atividades da empresa.

O sistema foi desenvolvido com objetivo de poder oferecer uma melhor organização para os dados e informações coletados, fazendo com que se possa ter um controle maior das rotinas da mesma, proporcionando assim mais conforto aos usuários e tornando o trabalho mais eficaz.

Foi possível observar durante a coleta de dados a importância de um analista que se envolva nas atividades da empresa, para poder observar as reais necessidades da mesma, juntamente com as pessoas que iram interagir com o sistema.

Para uma empresa apresentar um crescimento satisfatório, fazendo de forma correta a administração de seu empreendimento, necessita estar bem orientada e acompanhada por ferramentas funcionais que tornem seu dia a dia mais produtivo e lucrativo. Sendo assim nota-se a real importância de um sistema administrativo e com um banco de dados bem estruturado, suprimindo suas necessidades. Portanto pode-se dizer que o sistema Pet Shop Lovers além de um SGBD, ele poderá ser classificado também dentro de dois tipos de sistemas um SPT e um SIG.

## V REFERÊNCIAS BIBLIOGRÁFICAS

BATISTA, Emerson de Oliveira. **Sistema de Informação: o uso consciente da tecnologia para o gerenciamento.** São Paulo: Saraiva, 2004.

BONATTO, S.J. **Introdução aos Sistemas de Informação nas Empresas.**

“[[http:// www.univel.br/selmo/siaulas.html](http://www.univel.br/selmo/siaulas.html) ]” Acesso em: 15/08/13.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário.** Rio de Janeiro: Campus, 2000.

DATE, C. J. (2000). **Introdução a Sistemas de Banco de Dados.** Editora Campus Ltda. Rio de Janeiro.

FURLAN, José Davi. **Modelagem de objetos através da UML: the unified modeling language** São Paulo: Makron Books, 1998.

GARCIA, Elias; GARCIA, Osmarina Pedro Garcia. **A importância do sistema de informação gerencial para a gestão empresarial.** Revista Ciências Sociais em Perspectiva, do Centro de Ciências Sociais Aplicadas de Cascavel, Cascavel, v.2 , n.1, p. 21-32, 1 sem. 2003.

LARMAN, Craig. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos.** Porto Alegre: Bookman, 2000.

LAUDON, Kenneth C.; LAUDON, Jane P. **Gerenciamento de Sistemas de Informação.** 3.ed. Rio de Janeiro: LTC, 2001.

O'BRIEN, James A. **Sistemas de informação e as decisões gerenciais na era da internet.** Tradução de Cid Knipel Moreira. São Paulo: Saraiva, 2002.

OLIVEIRA, Djalma de Pinho Rebouças. **Sistemas, organizações e métodos: uma abordagem gerencial.** 13. Ed São Paulo, 2002, Paulo, 2002.

OLIVEIRA, Djalma de Pinho Rebouças de. **Sistemas de informação gerenciais: estratégias, táticas, operacionais**. 8. ed., São Paulo: Atlas,1992.

PELLAND, Patrice; PARE, Pascal; HAINES, Ken; **Moving to Microsoft Visual Studio 2010**; Ed. Microsoft Press. (2010).

SILBERSCHATZ, A. et al. (1999). **Sistema de Banco de Dados**. 3. Ed. Editora Makron Books, São Paulo.

SPANBAUER, Scott; HOWE, Jill; MCRAE, William; **Crystal Reports 9 Essentials**; Ed. Course Technology Ptr. (2002).

STAIR, Ralph M. **Princípios de sistemas de informação**. Rio de Janeiro: LTC, 1998.

STEIN, Steven. J; Microsoft .NET Framework 3.5 – Windows Forms Application Development; Ed. Microsoft Press. (2009).

STELLAMN, Andrew; **Use a Cabeça! C#** Ed. ALTA BOOKS. (2011).

TIFFANY, Rob; **SQL Server CE Database Development with the .NET Compact Framework (Expert's Voice)** Ed. Trade Paper, Book (2003).

**VI ANEXOS**

**ANEXO I – ARTIGO**

## SISTEMA DE GERENCIADO DE BANCO DE DADOS PET SHOP LOVERS

**Camila Matias dos Reis<sup>1</sup>**

camilamatias.reis@gmail.com

**Márcio José Sembay<sup>2</sup>**

m\_sembay@yahoo.com.br

<sup>1</sup>Acadêmico do Curso de Ciência da Computação,

<sup>2</sup>Professor do Curso de Ciência da Computação

CENTRO UNIVERSITÁRIO UNIVESC

Lages, SC, Brasil.

### RESUMO

Este trabalho apresenta um *Sistema Gerencial de Banco de Dados (SGBD)*, sistema o qual foi implementado para o armazenamento de dados e consulta, visando o gerenciamento e administração das funções rotineiras da empresa Clínica Veterinária Filhotes & Mascotes Pet Shop Lovers. Com este sistema a empresa em questão vai poder ter um controle maior de suas atividades como cadastro de clientes, funcionários, produtos, fornecedores, animais, controle de contas e vendas, agendamento de consultas, além de emissão de relatórios, proporcionando assim uma visão maior de quantos clientes e animais foram atendidos, seus fornecedores, além de um controle maior dos lucros e despesas da empresa. Para o desenvolvimento do mesmo foi realizado um estudo de caso aonde foi possível detectar os principais problemas e dificuldades de gestão da empresa. O sistema foi desenvolvido com banco de dados SQL Server Compact Edition, plataforma Microsoft.NET Framework, C#, Microsoft Visual Studio e Crystal Reports (ferramenta para gerar relatórios).

**Palavras chaves:** *SGBD, C#, SQL Server Compact Edition, Microsoft.NET Framework, Microsoft Visual Studio, Crystal Reports*

### MANAGEMENT DATABASE SYSTEM PET SHOP LOVERS

### ABSTRACT

This paper presents a Database Management System (DBMS) system, which was implemented for the data storage and query, targeting the management and administration of the routine functions of the company Puppies Veterinary Clinic & Pet Shop Pets Lovers. With this system the company in question will be able to have greater control of their activities such as customer base, employees, products, suppliers, animal control, accounts and sales, appointment scheduling, and reporting, thus providing greater insight how many clients and animals were treated, their suppliers, as well as greater control of income and expenses of the company. For its development, we conducted a case study where it was possible to identify the main problems and difficulties in management. The system was developed with the database SQL Server Compact Edition platform Microsoft.NET Framework, C#, Microsoft Visual Studio and Crystal Reports (tool to generate reports).

**Keywords:** *DBMS, C#, SQL Server Compact Edition, Microsoft.NET Framework, Microsoft Visual Studio, Crystal Reports.*

## 1. Introdução

Hoje em dia as empresas estão cada vez mais preocupadas em utilizar as tecnologias existentes no mercado para se informatizar e também para tornar o trabalho mais eficaz, organizado e principalmente com o objetivo de se manter competitiva no mercado.

Tendo em vista esses requisitos este trabalho tem por objetivos demonstrar as técnicas e ferramentas utilizadas para criar um Sistema Gerencial de Banco de Dados (SGBD).

Um Sistema Gerenciador de Banco de Dados (SGBD) é constituído por um conjunto de dados associados a um conjunto de programas para acesso a esses dados. O conjunto de dados, comumente chamado banco de dados, contém informações sobre uma empresa em particular. O principal objetivo de um SGBD é proporcionar um ambiente tanto conveniente quanto eficiente para recuperação e armazenamento das informações do Banco de Dados. (SILBERSCHATZ, 1999).

Sabemos que é muito importante um sistema administrativo e gerencial, mas mais importante ainda é um SGBD bem estruturado para o armazenamento de informações e dados, principalmente para a consulta dos mesmos.

No desenvolvimento de um software é necessário saber as exigências da organização, para que o mesmo possa ser eficiente auxiliando a empresa nas rotinas e funções. Todo software passa por uma fase de teste, mais só pode ser constatada a eficiência do mesmo a partir do momento que o usuário começa a utilizar o sistema. Conforme a utilização, os usuários vão solicitando modificações até que o sistema possa satisfazer todas as necessidades da empresa.

São vários os recursos tecnológicos encontrados no mercado hoje em dia, e cada vez mais estabelecimentos vêm procurando os mesmo, com o objetivo de trazer benefícios e em busca de agilidade e controle para suas funções diárias. A parte de gerenciamento e administração de uma empresa é de extrema importância, mas os mesmo não seriam eficazes e possíveis sem um SGBD por traz. Em essência, um banco de dados é apenas um sistema computadorizado de armazenamento de registros. (DATE, 2000). Afinal sabemos que uma empresa por menor que seja possui e, é formada por dados e informações os quais precisam ser armazenados para assim ter um controle maior e poder ter acesso às informações com mais facilidade, visando ter um futuro crescimento. Analisando estes pontos, e sendo o principal problema da empresa em questão a solução foi à implementação de um SGBD para um Pet Shop.

### **1.1 Objetivos**

Desenvolver um sistema no qual seja direcionado para a área de SGBD, em vista de uma melhor organização dos dados da empresa, no qual foi possível caracterizar a deficiência do armazenamento e consulta de dados, além de colocar funções básicas de administração e gerenciamento, procurando assim tornar a rotina da empresa mais organizada, ágil e precisa.

### **1.2 Objetivos específicos**

Os objetivos específicos do sistema tem como finalidade a implantação das seguintes funções:

- a) Cadastrar, armazenar e dessa formar organizar os dados de clientes, fornecedores, produtos, funcionários e animais.
- b) Agendamento de consultas
- c) Controle de contas e vendas
- d) Emissão de relatórios
- e) Função que mostra os aniversariantes do dia.

Isto fará com que o empresário ajuste sua empresa de forma que ele esteja enquadrado em um nível técnico que o tornará cada vez mais bem informado e produtivo, pois terá dados processados que irão facilitar o controle funcional de seu estabelecimento.

### **1.3 Relevância**

Banco de dados é um conjunto de informações que se relaciona de forma a criar um sentido. Sua utilização é de grande importância para a definição da política de promoções das empresas. Com as informações contidas no banco de dados é possível para a empresa, por exemplo, criar um cartão de fidelidade para oferecer determinado tipo de ação promocional em datas especiais como os aniversários, o tipo de brinde, quer sejam ingressos de cinema ou teatro, presentes, ou seja, ações que visam cativar o cliente.

Os Bancos de Dados, além de manterem todo o volume de dados organizado, também executam tarefas e comandos que podem ser previamente programadas por uma pessoa (um DBA – Data Base Administrator) que define os serviços a serem realizados pelo sistema baseado nas rotinas de sua empresa.

Os bancos de dados se firmaram nas últimas duas décadas como a principal base de informações das companhias e se tornaram vitais para o seu crescimento.

#### **1.4 Justificativa**

Um acadêmico do curso de Ciência da Computação ao se formar tem o comprometimento não apenas de simplesmente desenvolver um sistema, mais sim de colocar em prática todo o conteúdo, ferramentas e aprendizagem adquiridas ao longo do curso. Um profissional desta área quando realiza um trabalho de criação ou implementação de um sistema não pode delimitar-se apenas a programação do mesmo, existe todo um processo que consiste em avaliar, elaborar o projeto de acordo com as necessidades do cliente, executar o projeto e também realizar possíveis manutenções que serão necessárias conforme a utilização do software.

O cliente tem uma visão e ideia de como deseja que seu sistema funcione, e mesmo que sejam atingidos os objetivos iniciais, conforme o usuário utiliza se analisa a necessidade de fazer manutenções ou até mesmo alterações.

De acordo com (DATE, 2000), um sistema de banco de dados é “um sistema computadorizado cuja finalidade geral é armazenar informações e permitir que os usuários busquem e atualizem essas informações quando as solicitar”. Para o autor um sistema de banco de dados é composto por dados, hardware, software e usuários. Ou seja, todos os sistemas possuem um banco de dados, afinal um bom gerenciamento e administração da empresa são feitos de uma série de informações e dados coletados, os quais precisam ser armazenados adequadamente para uma visão mais ampla da situação da empresa e também para um fácil acesso aos mesmos. Por isso este trabalho aborda o SGBD para uma clínica veterinária a qual seu maior problema era a organização, acesso e controle de suas funções, atividades financeiras e rotinas de trabalho.

Para se construir um sistema de qualidade é importante que o profissional seja comunicativo e não apenas um bom programador, mas também um analista e consiga absorver o máximo possível de informações de seu cliente. O analista deve estar inteirado do assunto e sistema em questão, não só para conseguir programar o desejado, mais também para estar agregando valor ao que o cliente não está sabendo solicitar. Para obter os resultados esperados é necessária uma interação entre analista, cliente (usuário) e a área de atuação para poder ser feito um estudo de caso para desenvolver, analisar, projetar e implantar um sistema de qualidade.

Durante o desenvolvimento de sistemas a conduta a ser adotada deve levar a construção de sistemas efetivos. Sistemas que não sejam apenas eficientes, mas também

eficazes; que façam exatamente o que é útil e necessário para os usuários e para a empresa de modo geral.

“O principal objetivo de um SGDB é proporcionar um ambiente tanto conveniente quanto eficiente para a recuperação e armazenamento das informações do banco de dados” (SILBERSCHATZ; KORTH; SUDARSHAN, 1999). Esta foi a maior preocupação durante a implementação deste trabalho criar um SGBD que fosse ao mesmo tempo eficiente no armazenamento, recuperação e alteração dos mesmo, como com um ambiente conveniente e confortável ao usuário proporcionando assim um fácil acesso as funções desejadas.

### **1.5 Metodologia de Construção**

Para início do estudo, uma entrevista com o proprietário do estabelecimento comercial para levantamento dos requisitos básicos do software e posteriormente feito um levantamento da parte dos usuários e suas principais dificuldades as rotinas da empresa.

O trabalho foi dividido da seguinte forma:

- a) Levantamento dos dados da empresa;
- b) Acompanhamento das rotinas da empresa no dia a dia.
- c) Levantando de erros e o que poderia ser melhorado para se adequar aos requisitos da empresa;

Para fazer um sistema conforme a necessidade do cliente, o analista precisa se incorporar na área desejada, sendo necessário utilizar as várias técnicas apresentadas durante o curso. Um bom sistema se constrói não apenas com um bom programador e sim com um profissional comunicativo e que seja capaz de interpretar o que o usuário esta solicitando. Em muitos casos o usuário solicita algo que o analista se inteirando do assunto e da área de atuação percebe que existe necessidade de agregar valor ao que o usuário não esta sabendo solicitar, pois é preciso uma interação total entre analista, usuário e campo de atuação para que na fase da implantação do sistema sejam alcançados os objetivos principais.

É de grande importância para os estabelecimentos comerciais, possuírem uma ferramenta para controlar seus dados funcionais e operacionais da empresa, de forma que auxilie a efetuar o controle do estabelecimento com informações mais precisas e dinâmicas, de maneira que obtenha resultados positivos no seu desenvolvimento.

Este sistema tem por finalidade auxiliar os lojistas a controlar suas transações comerciais, ou seja, controlar os dados do estabelecimento, desde a compra até a venda. Possuindo um controle de cadastro de seus clientes, funcionários e produtos. O sistema possui um controle das transações, que envolve as partes de cadastros (clientes, produtos, funcionários), estoque e vendas, que é o controle de funcionamento da loja, possui também

um controle do setor financeiro, que é a parte que cuida do controle de depósitos e das contas a pagar.

## **2. Banco de Dados**

Bancos de dados, (ou bases de dados), são conjuntos de dados com uma estrutura regular que organizam informação. Um banco de dados normalmente agrupa informações utilizadas para um mesmo fim.

Um banco de dados é usualmente mantido e acessado por meio de um software conhecido como Sistema Gerenciador de Banco de Dados (SGBD). Normalmente um SGBD adota um modelo de dados, de forma pura, reduzida ou estendida. Muitas vezes o termo banco de dados é usado como sinônimo de SGDB.

O modelo de dados mais adotado hoje em dia é o modelo relacional, onde as estruturas têm a forma de tabelas, compostas por linhas e colunas.

### **2.1 Definições**

O termo banco de dados foi criado inicialmente pela comunidade de computação, para indicar coleções organizadas de dados armazenados em computadores digitais, porém o termo é atualmente usado para indicar tanto bancos de dados digitais como bancos de dados disponíveis de outra forma. No Brasil, é mais comum usar o termo base de dados quando se mencionam outros tipos de bancos de dados senão aqueles armazenados em um computador e gerenciados por um SGBD.

Aceitando uma abordagem mais técnica, um banco de dados é uma coleção de registros salvos em um computador em um modo sistemático, de forma que um programa de computador possa consultá-lo para responder questões.

Normalmente um registro está associado a um conceito completo e é dividido em campos, ou atributos, que dão valores a propriedades desses conceitos. Possivelmente alguns registros podem apontar diretamente ou referenciar indiretamente outros registros, o que faz parte da caracterização do modelo adotado pelo banco de dados.

A descrição de quais são os tipos de registros existentes em um banco de dados e ainda quais são os campos de cada registro é conhecida como esquema do banco de dados.

Estritamente falado, o termo banco de dados deve ser aplicado apenas aos dados, enquanto o termo sistema gerenciador de bancos de dados deve ser aplicado ao software com a capacidade de manipular bancos de dados de forma geral. Porém, é comum misturar os dois conceitos.

## 2.2 Modelos de Dados dos Bancos de Dados

A maneira mais prática de classificar bancos de dados é de acordo com a forma que seus dados são vistos pelo usuário, ou seja, seu modelo de dados. Diversos modelos foram e vem sendo utilizados ao longo da história, com vantagens para um ou para outro por determinados períodos.

Atualmente, a classificação mais comum citaria 4 modelos básicos:

Modelos Navegacionais, divididos em:

- Modelo Hierárquico
- Modelo em Redes
- Modelo Relacional
- Modelo Orientado a Objetos

Porém, outros modelos podem ser citados, incluindo:

- Modelo de Entidades e Relacionamentos
- Modelo de lista invertida
- Modelo Relacional Estendido
- Modelo Semiestruturado

Historicamente, o modelo de bancos de dados em rede foi implementado primeiro; porém o primeiro produto comercial usava o modelo de bancos de dados hierárquico, que nada mais é que uma versão simplificada do primeiro. Ambos os modelos foram resultado da busca de usar mais efetivamente os novos dispositivos de memória secundária de acesso direto, que substituíam os cartões perfurados e as fitas magnéticas. Isso aconteceu na década de 1960.

Em 1970 E.F. Codd propôs o modelo de bancos de dados relacional que surgiu e ganhou destaque teórico imediato. Porém, a implementação do modelo exigia pesquisas e só na década de 1980 eles iam começar a ganhar o mercado, se estabilizando totalmente como líder do mercado a partir da década de 1990.

Podemos identificar o aparecimento do que pode ser chamado modelo plano (tabular) para fins mais diretos e simples. Nesse caso, os dados estão simplesmente arranjados em uma única matriz bidimensional de elementos de dados na qual todos os membros de uma dada coluna possuem valores de mesmo tipo, e todos os membros de uma linha estão relacionados entre si. Seu melhor exemplo são as planilhas eletrônicas.

O único modelo que foi extensamente tratado de forma teórica foi o modelo relacional. Os modelos pré-existentes foram fruto de implementações, enquanto os modelos subsequentes, como o modelo orientado a objetos, não apresentavam um campo tão rico para

novas teorias, mas apresentam grandes desafios para a implementação eficiente das operações necessárias.

### **2.3 Modelos Navegacionais**

No modelo em redes, os dados são organizados em registros, que são coleções de itens de dados, e podem ser armazenados ou recuperados de um banco de dados de forma conjunta. É possível que um registro possua uma estrutura interna, e elementos (itens de dados) contínuos podem ser agrupados, que também podem formar outros grupos. Dessa forma, um registro pode ter uma construção hierárquica. Os registros com a mesma estrutura formam um tipo de registro, que podem ser considerados equivalentes a uma tabela fora da primeira forma normal, ou ainda a um objeto complexo. Os tipos de registro possíveis em um banco de dados são definidos em seu esquema.

A principal característica do modelo em redes é permitir a navegação entre os registros, por meio de Conjuntos de Dados, que possui um registro proprietário e registros membros, implementados por meio de ponteiros. Basicamente, registros equivalem a entidades e conjuntos de dados equivalem a descrição dos relacionamentos. Como não há limitação na topologia criada pelos registros e conjuntos, o modelo permite a criação de redes, de onde ganhou o nome.

Um subconjunto particular do modelo de rede, o modelo hierárquico, limita os relacionamentos a uma estrutura de árvore, ao contrário da estrutura aplicada pelo modelo de rede completo.

O modelo em redes foi definido formalmente em 1971, pela Conference on Data Systems Languages (CODASYL), de onde ganhou seu outro nome: modelo CODASYL.

### **2.4 Modelo Relacional**

O modelo relacional é uma teoria matemática desenvolvida por Edgar Frank Codd para descrever como as bases de dados devem funcionar. Embora esta teoria seja a base para o software de bases de dados relacionais, muito poucos sistemas de gestão de bases de dados seguem o modelo de forma restrita, e todos têm funcionalidades que violam a teoria, desta forma variando a complexidade e o poder. A discussão se esses bancos de dados merecem ser chamados de relacional ficou esgotada com tempo, com a evolução dos bancos existentes.

De acordo com a arquitetura ANSI / SPARC em três níveis, os Bancos de dados relacionais consistem de três componentes:

- Uma coleção de estruturas de dados, formalmente chamadas de relações, ou informalmente tabelas, compondo o nível conceitual;
- Uma coleção dos operadores, a álgebra e o cálculo relacionais, que constituem a base da linguagem SQL;
- Uma coleção de restrições da integridade, definindo o conjunto consistente de estados de base de dados e de alterações de estados. As restrições de integridade podem ser de quatro tipos:
  - Domínio (ou tipo de dados);
  - Atributo;
  - Relvar;
  - Restrições de base de dados;

Diferentemente dos modelos navegacionais, não existem quaisquer ponteiros, de acordo com o Princípio de Informação: toda informação tem de ser representada como dados; qualquer tipo de atributo representa relações entre conjuntos de dados.

Diferentemente dos bancos de dados em rede, nos bancos de dados relacionais os relacionamentos entre as tabelas não são codificados explicitamente na sua definição. Em vez disso, se fazem implicitamente pela presença de atributos chave. As bases de dados relacionais permitem aos utilizadores (incluindo programadores) escreverem consultas (queries), reorganizando e utilizando os dados de forma flexível e não necessariamente antecipada pelos projetistas originais. Esta flexibilidade é especialmente importante em bases de dados que podem ser utilizadas durante décadas, tornando as bases de dados relacionais muito populares no meio comercial.

Um dos pontos fortes do modelo relacional de banco de dados é a possibilidade de definição de um conjunto de restrições de integridade. Estas definem os conjuntos de estados e mudanças de estado consistentes do banco de dados, determinando os valores que podem e os que não podem ser armazenados.

## **2.5 Bancos de Dados Orientados a Objetos**

Na década de 90, o modelo baseado na orientação a objeto foi aplicado também aos bancos de dados, criando um novo modelo de programação conhecido como bancos de dados orientados a objeto. Os objetos são valores definidos segundo classes, ou tipos de dados complexos, com seus próprios operadores (métodos).

Com o passar do tempo, os sistemas gestores de bancos de dados orientados a objeto e os bancos de dados relacionais baseados na linguagem SQL se aproximaram. Muitos sistemas orientados a objeto são implementados sobre bancos de dados relacionais baseados em linguagem SQL.

O resultado comercial, porém, foi pequeno. Atualmente vários princípios de orientação a objeto foram adotados pelos bancos de dados relacionais, gerando o que pode ser chamado de banco de dados relacional estendido.

## **2.6 Bancos de Dados Semiestruturados**

Mais recentemente ainda, apareceram os bancos de dados semiestruturados, onde os dados são guardados e manipulados na forma de XML (ao contrário da forma de tabelas). Novamente, os produtores de bancos de dados relacionais responderam estendendo suas capacidades para tratar dados semiestruturados.

## **2.7 Utilização**

Os bancos de dados são utilizados em muitas aplicações, abrangendo praticamente todo o campo dos programas de computador. Os bancos de dados são o método de armazenamento preferencial para aplicações multiusuário, nas quais é necessário haver coordenação entre vários usuários. Entretanto, são convenientes também para indivíduos, e muitos programas de correio eletrônico e organizadores pessoais baseiam-se em tecnologias padronizadas de bancos de dados.

Um banco de dados é um conjunto de informações com uma estrutura regular. Um banco de dados é normalmente, mas não necessariamente, armazenado em algum formato de máquina lido pelo computador. Há uma grande variedade de bancos de dados, desde simples tabelas armazenadas em um único arquivo até gigantescos bancos de dados com muitos milhões de registros, armazenados em salas cheias de discos rígidos.

Bancos de dados caracteristicamente modernos são desenvolvidos desde os anos da década de 1960. Um pioneiro nesse trabalho foi Charles Bachman.

## **2.8 Apresentação dos Dados**

A apresentação dos dados pode ser semelhante à de uma planilha eletrônica, porém os sistemas de gestão de banco de dados possuem características especiais para o armazenamento, classificação e recuperação dos dados.

## **2.9 Aplicações de bancos de dados**

Bancos de dados são usados em muitas aplicações, enquanto atravessando virtualmente a gama inteira de software de computador. Bancos de dados são o método

preferido de armazenamento para aplicações multiusuárias grandes donde coordenação entre muitos usuários é necessária. Até mesmo usuários individuais os acham conveniente, entretanto, e muitos programas de correio eletrônico e os organizadores pessoais estão baseado em tecnologia de banco de dados standard.

## **2.10 Aplicativo de Banco de Dados**

Um Aplicativo de Banco de dados é um tipo de software exclusivo para gerenciar um banco de dados. Aplicativos de banco de dados abrangem uma vasta variedade de necessidades e objetivos, de pequenas ferramentas como uma agenda, até complexos sistemas empresariais para desempenhar tarefas como a contabilidade.

O termo "Aplicativo de Banco de dados" usualmente se refere a softwares que oferecem uma interface para o banco de dados. O software que gerencia os dados é geralmente chamado de sistema gerenciador de banco de dados (SGBD) ou (se for embarcado) de "database engine".

Exemplos de aplicativos de banco de dados são Microsoft Visual FoxPro, Microsoft Access, dBASE, FileMaker , (em certa medida) HyperCard, MySQL, PostgreSQL, Microsoft SQL Server e Oracle.

## **2.11 Segurança em banco de dados**

Os bancos de dados são utilizados para armazenar diversos tipos de informações, desde dados sobre uma conta de e-mail até dados importantes da Receita Federal. Para tal existem diversos tipos, os quais variam em complexidade e sobretudo em segurança. São meios de proteger as informações armazenadas num banco de dados:

- Criptografia
- Senhas
- Backup

## **3 Sistema de Informação**

Sistemas de Informação são sistemas que, através de processos de coleta e tratamento de dados, geram e disseminam as informações necessárias aos diversos níveis e processos organizacionais. Lembrando que da grande diversidade de sistemas, podemos separar os sistemas naturais e dos sistemas artificiais (criados pelo homem). Vamos encontrar um grupo de sistemas artificiais, que pela sua natureza, são chamados de Sistemas de Informação,

justamente porque seu maior objetivo é Fornecer, Controlar, Prover, Pesquisar e Analisar Informações.

Geralmente, um SI é composto de um subsistema social e de um subsistema automatizado. O primeiro inclui as pessoas, processos, informações e documentos. O segundo consiste dos meios automatizados (máquinas, computadores, redes de comunicação) que interligam os elementos do subsistema social.

Ao contrário do que muitos pensam, as pessoas (juntamente como os processos que executam e com as informações e documentos que manipulam) também fazem parte do SI. O SI é algo maior que um software, pois além de incluir o hardware e o software, também inclui os processos (e seus agentes) que são executados fora das máquinas.

#### **4. O Software**

O sistema foi desenvolvido com objetivo de auxiliar nos processos operacionais e financeiros rotineiros na área administrativa da empresa de Pet Shop.

Com a implantação do sistema, a administração da empresa poderá realizar um melhor controle dos produtos disponíveis em estoque, das consultas, do cadastro de cliente e seus respectivos animais e também um controle financeiro mantendo os cadastros das vendas e contas a pagar, resultando em agilidade para realização de balanços e auditorias internas.

O usuário poderá acessar todos os módulos disponíveis no sistema:

- a) Cadastros de Clientes, Fornecedores, Animais, Produtos e Fornecedores
- b) Venda
- c) Controle de consultas
- d) Relatórios
- e) Aniversariantes

O sistema possui filtros de pesquisa variados para facilitar o acesso aos dados cadastrados de maneira mais específica, mostrando apenas dados relativos às condições aplicadas nos filtros.

O sistema também dispõe de um modulo para emissão de relatórios, permitindo gerar relatórios variados dos módulos do sistema onde é possível exportar para arquivo em vários formatos (\*.rpt, \*.pdf, \*.csv, \*.xls, \*.xlsx, \*.doc, \*.rtf, \*.xml), imprimir ou visualizar dados cadastrados no sistema. Alguns módulos são possíveis aplicar filtros para os relatórios mais específicos.

A figura 01 apresenta a tela do menu principal do sistema.



Figura 01 – Tela do Menu Principal do Sistema

Fonte: Próprio Autor (2013).

## 5. Conclusões

No início deste trabalho tínhamos uma empresa que sua principal dificuldade e problema eram o armazenamento e consulta aos seus dados como clientes, animais, despesas e vendas até a não existência de relatórios, entre outros requisitos. Os quais são de extrema importância para um crescimento e uma visão geral da situação da empresa.

Com o desenvolvimento deste Sistema Gerencial de Banco de Dados (SGBD), foi possível solucionar vários problemas com relação aos dados e informações que não haviam um lugar específicos para serem armazenados, os quais acabavam sendo perdidos prejudicando assim um futuro crescimento e o controle das atividades da empresa.

O sistema foi desenvolvido com objetivo de poder oferecer uma melhor organização para os dados e informações coletados, fazendo com que se possa ter um controle maior das rotinas da mesma, proporcionando assim mais conforto aos usuários e tornando o trabalho mais eficaz.

Foi possível observar durante a coleta de dados a importância de um analista que se envolva nas atividades da empresa, para poder observar as reais necessidades da mesma, juntamente com as pessoas que iram interagir com o sistema.

Para uma empresa apresentar um crescimento satisfatório, fazendo de forma correta a administração de seu empreendimento, necessita estar bem orientada e acompanhada por ferramentas funcionais que tornem seu dia a dia mais produtivo e lucrativo. Sendo assim nota-se a real importância de um sistema administrativo e com um banco de dados bem estruturado, suprimindo suas necessidades. Portanto pode-se dizer que o sistema Pet Shop Lovers além de um SGBD, ele poderá ser classificado também dentro de dois tipos de sistemas um SPT e um SIG.

## 6. Referências Bibliográficas

BATISTA, Emerson de Oliveira. **Sistema de Informação: o uso consciente da tecnologia para o gerenciamento.** São Paulo: Saraiva, 2004.

DATE, C. J. (2000). **Introdução a Sistemas de Banco de Dados.** Editora Campus Ltda. Rio de Janeiro.

FURLAN, José Davi. **Modelagem de objetos através da UML: the unified modeling language** São Paulo: Makron Books, 1998.

LAUDON, Kenneth C.; LAUDON, Jane P. **Gerenciamento de Sistemas de Informação.** 3.ed. Rio de Janeiro: LTC, 2001.

O'BRIEN, James A. **Sistemas de informação e as decisões gerenciais na era da internet.** Tradução de Cid Knipel Moreira. São Paulo: Saraiva, 2002.

OLIVEIRA, Djalma de Pinho Rebouças de. **Sistemas de informação gerenciais: estratégias, táticas, operacionais.** 8. ed., São Paulo: Atlas, 1992.

STAIR, Ralph M. **Princípios de sistemas de informação.** Rio de Janeiro: LTC, (1998).

SILBERSCHATZ, A. et al. **Sistema de Banco de Dados.** 3. Ed. Editora Makron Books, São Paulo. (1999).