

UNIVERSIDADE FEDERAL DO PARANÁ

PRISCILA ALVES DE LIMA

SOFTWARE PARA GESTÃO DE PET SHOP E CLÍNICA VETERINÁRIA

MATINHOS
2015

PRISCILA ALVES DE LIMA

SOFTWARE PARA GESTÃO DE PET SHOP E CLÍNICA VETERINÁRIA

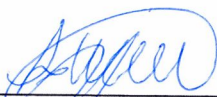
Trabalho de conclusão de curso de Graduação, apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Informática e Cidadania da Universidade Federal do Paraná, campus Matinhos, como requisito parcial para obtenção do título de bacharel em Informática e Cidadania.

Orientadora: Prof.^a Dr.^a Luciana Vieira Castilho Weinert.

MATINHOS
2015

ATA DE AVALIAÇÃO DA DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO

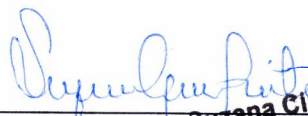
Aos três dias do mês de junho de dois mil e quinze, às 18h30, no Setor Litoral da Universidade Federal do Paraná, reuniu-se a banca avaliadora do trabalho de conclusão de curso, constituída pelo professor Me. David Andrade Silva e pela professora Dra. Suzana Cini Freitas Nicolodi sob a presidência da Orientadora, Professora Dra. Luciana Vieira Castilho Weinert. O Trabalho de Conclusão do Curso de Bacharelado em Informática e Cidadania, da aluna Priscila Alves de Lima, sob o título: "Desenvolvimento de Software para Gestão de Pet Shop e Clínica Veterinária", obteve o conceito APL. A aluna deverá efetuar as correções solicitadas pela banca e entregar a versão final em formato digital via CD-ROOM, até o dia 10 de junho de dois mil e quinze, na assessoria a Câmara do curso de Informática e Cidadania.



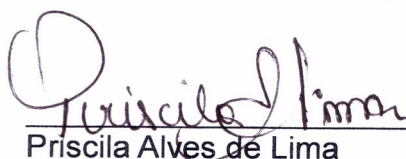
Luciana Vieira Castilho Weinert
Professora Orientadora



David Andrade Silva
Membro da banca avaliadora



Suzana Cini Freitas Nicolodi
Membro da banca avaliadora
Matricula 202246
Setor Litoral - UFPR



Priscila Alves de Lima

Dedicatória

*Aos meus amados “filhos”: Kytta, Athos, Nina,
Grecco (in Memoriam), Félix, Ághata e Snow.
Os quais não sabem falar, mas que possuem
o amor mais sincero.*

AGRADECIMENTOS

Agradeço primeiramente á Deus, pois me deu sabedoria e força para persistir em momentos de dificuldade e permitiu com que esse trabalho fosse concluído.

Obrigada a toda minha família pelo apoio e compreensão.

Agradeço também ao meu esposo Edilson, que de forma especial e carinhosa me deu força e coragem, me apoiando nos momentos de dificuldades.

A Professora Dr.^a Luciana Vieira Castilho Weinert, pela orientação e por todo o conhecimento compartilhado. A você, a minha admiração e o meu agradecimento.

Ao Professor Msc David José de Andrade Silva, pelas orientações, incentivos, apoio incondicional, amizade e conhecimentos transmitidos e também pelo exemplo profissional como pessoa e educador, minha profunda admiração e gratidão.

Agradeço a todos os professores que passaram pela minha formação acadêmica, contribuindo não apenas com conhecimentos, mas também com a minha formação pessoal. Em especial a Professora Dr.^a Suzana Cini Freitas Nicolodi por sua alegria contagiante, obrigada pelos ensinamentos.

Por fim, mas não menos importante agradeço aos amigos que fiz na turma 2010 de Informática e Cidadania. Amigos para todas as horas, amigos para toda a vida.

“O erro da ética até o momento tem sido a crença de que só se deva aplicá-la em relação aos homens”.

(Albert Schweitzer)

RESUMO

Este trabalho tem como objetivo a análise e desenvolvimento de um software para auxiliar na gestão de Pet Shop e clínica veterinária. Devido à expansão desse nicho de mercado, identificou-se a necessidade de elaborar um produto para automatizar a gestão dessas empresas e controlar os procedimentos de atendimento aos seus clientes. Essa ferramenta também permite o controle de mercadorias e serviços para melhorar a logística de trabalho e possibilitar uma maximização de resultados de acordo com a necessidade das demandas. A metodologia de desenvolvimento do aplicativo contemplou a fase de projeto do sistema, em que houve o levantamento de requisitos em conjunto com os usuários e empresários do ramo. Para a implementação utilizou-se a linguagem de programação Delphi com banco de dados relacional Firebird. Também se empregou algumas técnicas como análises de sistemas, construção de diagramas e fluxogramas. Como resultados o sistema possui telas de cadastros, controle de caixa, agendamento de serviços, vendas de produtos e alguns relatórios. Conclui-se que a construção deste software auxiliou e aprimorou os conhecimentos e técnicas adquiridas no curso de Informática e Cidadania, além de contribuir para o uso da tecnologia cooperando com a gestão e desenvolvimento de empresas locais.

Palavras-chave: Análise de Sistemas; Banco de dados; Desenvolvimento de Software; Informática; Pet Shop;

ABSTRACT

This work aims to analyze and develop a software to assist in the management of Pet Shop and veterinary clinic. Due to the expansion of this market niche, we identified the need of a product to automate the management of these business and control procedures of service to its customers. This tool also allows control of goods and services to improve the logistics work and enable maximization of results according to the need of the demands. The application development methodology included the system design phase, in which there was a survey of requirements together with users and business. For implementation we used the Delphi programming language with Firebird relational database. We also employed some techniques as systems analysis, diagrams and flow charts. As a result the system has screens entries, cash management, scheduling services, product sales and some reports. We concluded that the construction of this software improved the knowledge and skills acquired in the course of Informatics and Citizenship, besides contributing to the use of technology cooperating with the management and development of local businesses.

Keywords: System Analysis; Database; Software development; Informatics; Pet Shop; Management system;

LISTA DE FIGURAS

FIGURA 1 -	CENÁRIO <i>PET</i> NO BRASIL.....	16
FIGURA 2 -	INFORMÁTICA: NA INTERSEÇÃO DE QUATRO ÁREAS.....	19
FIGURA 3 -	ONDE O SISTEMA OPERACIONAL SE ENCAIXA.....	23
FIGURA 4 -	GERAÇÕES DE LINGUAGENS DE PROGRAMAÇÃO.....	30
FIGURA 5 -	FORMULÁRIO DE CADASTRO DE ANIMAIS DO SGPET EM AMBIENTE DE DESENVOLVIMENTO DELPHI.....	41
FIGURA 6 -	FORMULÁRIO DE CADASTRO DE ANIMAIS DO SGPET EM AMBIENTE DE APLICAÇÃO.....	41
FIGURA 7 -	TELA DE SENHA DE ACESSO DO SGPET.....	49
FIGURA 8 -	TELA INICIAL DO SGPET.....	50
FIGURA 9 -	TELA DO CADASTRO DE CLIENTES DO SGPET.....	51
FIGURA 10 -	TELA DE ABERTURA DE CAIXA DO SGPET.....	52
FIGURA 11 -	TELA DE FECHAMENTO DE CAIXA DO SGPET.....	53
FIGURA 12 -	TELA DE RECEBIMENTO DE CONTAS DO SGPET.....	54
FIGURA 13 -	TELA DE AGENDAMENTO DE SERVIÇOS DO SGPET.....	55
FIGURA 14 -	TELA DE VENDAS DE PRODUTOS E SERVIÇOS DO SGPET.....	56
FIGURA 15 -	TELA DE ATUALIZAÇÃO DE VERSÃO DO SGPET.....	57

LISTA DE GRÁFICOS

GRÁFICO 1 - COMPARATIVO DA INDÚSTRIA <i>PET</i> NO MUNDO.....	15
---	----

LISTA DE TABELAS

TABELA 1 -	DADOS X INFORMAÇÃO.....	24
------------	-------------------------	----

LISTA DE ABREVIATURAS E SIGLAS

ACID	- <i>Atomicity Consistency Isolation Durability.</i>
BD	- Banco de Dados.
CPU	- <i>Central Processing Unit</i> – Unidade Central de Processamento.
DBAs	- <i>Database administrator.</i>
DER	- Diagrama de Entidades e Relacionamentos.
IA	- Inteligência Artificial.
LP	- Linguagem de Programação.
ONG	- Organizações Não Governamentais.
PSQL	- <i>Procedures Structured Query Language.</i>
SGDB	- Sistema Gerenciamento de Banco de Dados.
SGPET	- <i>Software</i> para Gestão de <i>Pet Shop</i> e Clínica Veterinária.
SO	- Sistema Operacional.
SQL	- <i>Structured Query Language.</i>
TI	- Tecnologia da Informação.
UDFs	- <i>User Defined Function.</i>

SUMÁRIO

MEMORIAL DESCRITIVO DA TRAJETÓRIA DE VIDA E ACADÊMICA	14
1 INTRODUÇÃO	15
2 FUNDAMENTAÇÃO TEÓRICA	18
2.1 INFORMÁTICA.....	18
2.1.1 <i>Hardware</i>	19
2.1.2 <i>Software</i>	20
2.1.2.1 <i>Software</i> de Aplicação	21
2.1.2.2 <i>Software</i> de Sistema	21
2.1.3 Sistema Operacional.....	22
2.1.4 Sistemas de Informação	23
2.1.4.1 Conceitos de Informação	24
2.1.4.2 Conceitos de Sistema	25
2.1.4.3 Sistemas de Informações Gerenciais.....	25
2.2 DESENVOLVIMENTO DE <i>SOFTWARE</i>	26
2.2.1 Métodos de Desenvolvimento de <i>Software</i>	27
2.2.1.1 Método Estruturado.....	27
2.2.1.2 Método Orientado a Objeto.....	27
2.2.2 Modelos de Desenvolvimento de <i>Software</i>	28
2.3 ANÁLISE E PROJETOS DE SISTEMAS	29
2.4 LINGUAGENS DE PROGRAMAÇÃO (LP)	30
2.4.1 Linguagens de Primeira Geração.....	31
2.4.2 Linguagens de Segunda Geração.....	31
2.4.3 Linguagens de Terceira Geração.....	31
2.4.3.1 Linguagens de Alto Nível de Uso Geral.....	32
2.4.3.2 Linguagens Orientadas a Objetos.....	32
2.4.3.3 Linguagens Especializadas.....	33
2.4.4 Linguagens de Quarta Geração	33
2.4.4.1 Linguagens de Consulta	34
2.4.4.2 Geradores de Programas.....	34
2.4.4.3 Linguagens de Prototipação.....	34
2.5 BANCOS DE DADOS.....	35
2.6 LEVANTAMENTO DE REQUISITOS	35
3 METODOLOGIA	37
3.1 PROJETO DO SISTEMA	37
3.1.1 Análise e Levantamento de Requisitos	37
3.1.2 Diagrama de Entidade e Relacionamentos	39
3.1.3 Fluxograma.....	39
3.2 IMPLEMENTAÇÃO	40
3.3 LINGUAGEM DE PROGRAMAÇÃO DELPHI	40

3.3.1 Recursos da Linguagem de Programação <i>Delphi</i>	42
3.4 BANCO DE DADOS FIREBIRD	42
3.4.1 Recursos do <i>Firebird</i>	43
3.5 INTERFACE DO APLICATIVO.....	44
3.6 FUNCIONALIDADES DO APLICATIVO	44
3.7 IMPLANTAÇÃO DO APLICATIVO EM UM <i>PET SHOP</i>	45
3.8 TREINAMENTO E SUPORTE AO USUÁRIO.....	46
3.9 MANUTENÇÃO DO APLICATIVO.....	47
4 RESULTADOS E DISCUSSÕES.....	49
5 CONSIDERAÇÕES FINAIS.....	59
6 PERSPECTIVAS FUTURAS	60
7 REFERÊNCIAS	61
ANEXOS	62
ANEXO 1 – FLUXOGRAMA DO SOFTWARE SGPET.....	62
ANEXO 2 – DIAGRAMA DE ENTIDADE E RELACIONAMENTO.....	63

MEMORIAL DESCRITIVO DA TRAJETÓRIA DE VIDA E ACADÊMICA

Este trabalho reflete minha vivência acadêmica nos três espaços curriculares da Universidade Federal do Paraná – Setor Litoral (UFPR Litoral): Fundamentos Teóricos Práticos (FTP); Interações Culturais e Humanísticas (ICH) e Projeto de Aprendizagem (PAs).

Minha caminhada na UFPR – Litoral começou em 2009, mas devido a alguns contratempos decidi ingressar na turma 2010, de Informática e Cidadania, junto com meu esposo que fazia parte da mesma.

Não é segredo para ninguém o meu amor pelos animais e devido a esse amor, o primeiro ICH que frequentei, ainda em 2009, foi sobre Bem Estar Animal. Neste espaço, rico em aprendizagens, surgiu algumas ideias e uma delas foi o tema deste trabalho: Um *Software* para *Pet Shop* e Clínica Veterinária.

Ali, comecei a enxergar que eu poderia unir minhas duas paixões: a tecnologia e os animais. Então, eu e meu marido Edilson Rodrigues Paio, resolvemos trabalhar esta ideia no Projeto de Aprendizado (PA), o qual foi desenvolvido por seis semestres, com os Fundamentos Teóricos Práticos dando todo o suporte que precisamos.

Ao final do curso a ideia iniciada em 2009, em um ICH, evoluiu para um trabalho de conclusão de curso (TCC), que teve como produto final um *software* para *Pet Shop* que está funcionando, em fase de teste.

1 INTRODUÇÃO

A motivação para a escolha e elaboração desse trabalho no seguimento de *pet shop* se deu pelo seu expressivo crescimento. Segundo dados disponíveis no site da associação brasileira da indústria de produtos para animais de estimação (ABINPET¹,2014), são mais de 40 mil *pet shops* no Brasil que movimentaram um mercado de, aproximadamente, R\$ 15,4 bilhões no ano de 2014. Um aumento de 7,3% em relação ao ano anterior. O GRÁFICO 1 ilustra o comparativo da indústria *pet* no mundo.

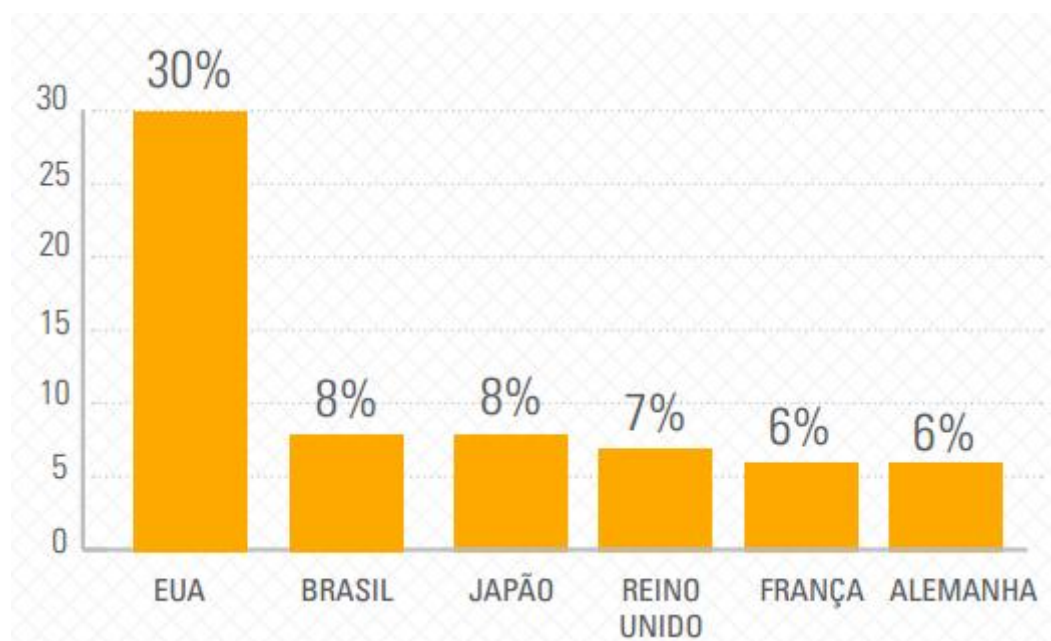


Gráfico 1 - Comparativo da indústria *pet* no mundo.

Fonte: ABINPET (2014)

Elaboração: SEBRAE/2014. <http://www.sebrae2014.com.br>

Com o total de 106,2 milhões de animais de estimação, o Brasil permanece como segundo país do mundo, perdendo apenas para os Estados Unidos, em faturamento e em população de cães, gatos dentre outros animais de estimação conforme ilustra a FIGURA 1.

¹ <http://www.abinpet.org.br/>

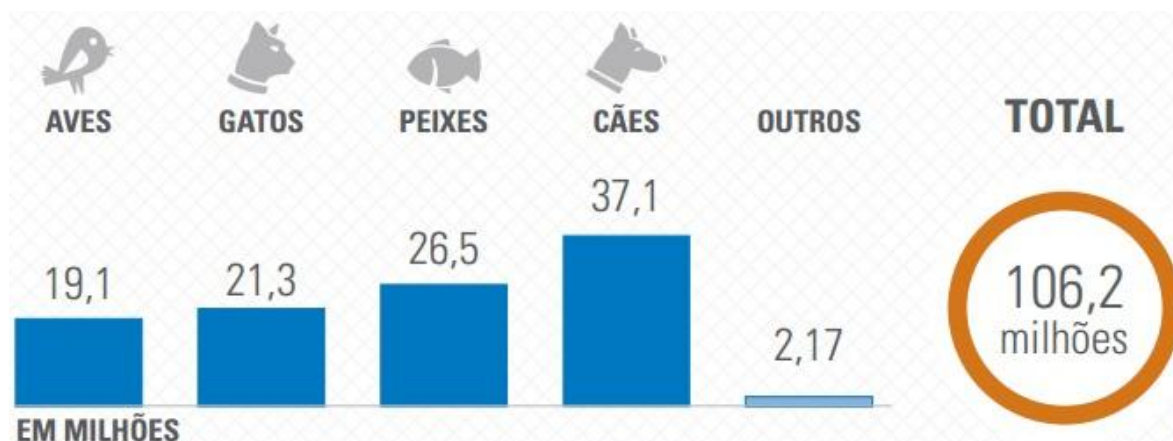


Figura 1 – Cenário *pet* no Brasil

Fonte: ABINPET (2014)

Elaboração: SEBRAE/2014. <http://www.sebrae2014.com.br>

O desenvolvimento deste trabalho foi dividido em três partes precípuas. Na primeira, foi realizado um estudo de campo na área de gestão de *pet shops* e clínicas veterinárias, assim como entrevistas com profissionais e empresários do ramo, médicos veterinários e presidente de ONGs. Nesta fase, foram coletadas informações necessárias para o conhecimento sobre o funcionamento, de modo geral, desse comércio, os produtos comercializados, os serviços oferecidos, como é feito o controle de estoque sem um sistema de gerenciamento, de que forma é utilizada a agenda de clientes/pacientes, como são armazenadas as informações do prontuário médico veterinário para possível consulta posterior, o funcionamento do fluxo de caixa, recebimento de contas e as formas utilizadas para o gerenciamento do estabelecimento comercial. Com base nesse conhecimento, fora efetuada uma análise sobre as vantagens de um *software* para a gestão deste nicho de mercado.

Na segunda, foi efetuado um levantamento de requisitos, organizações e planejamento imprescindíveis para a elaboração e estruturação do projeto. Nesta fase, foram definidas, juntamente com os usuários, quais as principais necessidades e requisitos funcionais e não funcionais a fim de maximizar os resultados obtidos pela utilização de um *software* para controle e gestão desse mercado.

Por fim, na terceira e última fase, foi definida toda a parte prática do desenvolvimento do *software* como: definição da linguagem de programação e banco de dados a serem utilizados; análise do sistema; do diagrama de entidade e

relacionamento; elaboração do fluxo de informações; estudo de casos; implantação do sistema em um *pet shop*; manutenção; suporte; treinamento aos usuários.

2 FUNDAMENTAÇÃO TEÓRICA

Este projeto de desenvolvimento de sistema foi realizado observando o objetivo do curso de Informática e Cidadania que contempla as três fases propostas pelo PPP da Universidade Federal do Paraná - Setor Litoral: conhecer e compreender, compreender e propor e agir (PPC, 2011), além da fundamentação, básica, sobre os três pilares de sustentação estudados ao longo de oito semestres do curso: Gestão de informação, Informática e Cidadania.

2.1 INFORMÁTICA

Informática, de acordo com (MARÇULA, 2008), é o estudo e o tratamento de dados e informações através de recursos básicos do sistema de computador. Como objeto de estudo, é dotada de conceitos distintos e tem por objetivo tratar as informações utilizando-se dos recursos de um computador e outros relacionados com ele.

“Informática é o estudo de tudo o que se relaciona à tecnologia da informação. É uma união de trechos de duas outras palavras, Informação + automática, e foi criada pelos franceses.” (ALCAIDE et AL.,1991).

Segundo (VELLOSO, 2014), o neologismo foi criado na França (*Informatique*) em abril de 1966 como alternativa a *information science* até então acolhido pela comunidade internacional. Em seguida popularizou-se no termo, em inglês, *informatic*. Vale a pena destacar que hoje o termo TI também está sendo amplamente utilizado para denominar essa área. A Informática compromete-se com as áreas de ciências exatas e ciências sociais. Pode-se considerá-la localizada, basicamente, em quatro áreas do conhecimento: Ciências da Computação, Ciências da Informação, Teoria dos sistemas e Cibernética, como ilustrado na FIGURA 2.

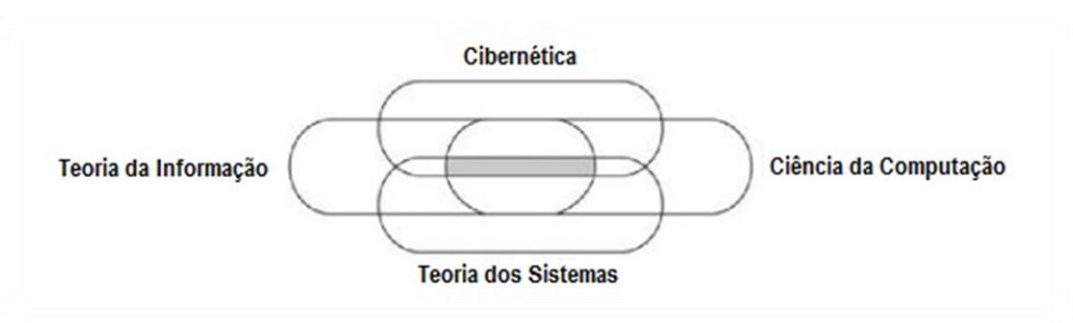


Figura 2 – Informática: na interseção de quatro áreas.

Fonte: VELLOSO (2014)

No que se refere ao computador em si, costuma-se dividi-lo em *hardware*, a plataforma física, e *software*, o conjunto de programas que possibilitam o funcionamento do equipamento. A seguir, serão apresentados alguns conceitos sobre essas duas áreas.

2.1.1 Hardware

Hardware é a parte física de um computador que normalmente é formado pelos seus componentes eletrônicos, como por exemplo: placa mãe, *hard disk* (HD), placa de rede, placas de memória, unidade central de processamento, termo derivado do inglês, *central processing unit* (CPU), dispositivos de entrada e saída como monitores de vídeo, *mouses*, teclados e impressoras ou qualquer outro material em estado físico, que seja necessário para fazer com que o computador desempenhe sua função. (MORIMOTO, 2007).

O *hardware* é basicamente utilizado por computadores e equipamentos eletrônicos. Qualquer equipamento físico que seja uma peça necessária ao funcionamento de um computador, também é chamado de *hardware*. O *hardware* não se limita apenas a computadores pessoais, pois estão disponíveis em automóveis, *tablets*, aparelhos de celular e os mais variados equipamentos eletrônicos.

Existem vários tipos de *hardware* com diferentes objetivos e funcionalidades. O *hardware* de rede, por exemplo, é um equipamento construído com o propósito de possibilitar a comunicação entre dispositivos que estão conectados em rede seja local ou *internet*. Normalmente os *hardwares* possuem *softwares* embarcados, ou seja, embutidos e designados para uma tarefa específica, que executam determinadas funções para o perfeito funcionamento do mesmo. Em outros casos, há *hardwares* que dependem de *softwares* ou programas externos que também dependem dos chamados *drivers*, que são conjuntos de informações dispostas em arquivos com o objetivo de configurar o *software* para que esse utilize o *hardware* de forma correta.

2.1.2 Software

Em um computador, o *software* é classificado como a parte lógica cuja função é fornecer uma sequência de instruções escritas, por intermédio de linguagens de programa, que controlam ou requisitam determinadas tarefas específicas a serem interpretadas e executadas pelo *hardware*. Também é constituído por todos os programas existentes, denominado sistemas, quer sejam produzidos pelo próprio usuário, por desenvolvedores de programas ou pelo fabricante dos componentes que compõe um computador. (TURBAN, 2004).

Para Turban (2004), o *hardware* de um computador não pode realizar um simples ato sem instruções. Essas instruções são conhecidas como *software* ou programas de computador. O *software* está no centro de todas as aplicações do computador e consiste em sequências de instruções escrita por programadores, seguidas e executadas na unidade central de processamento denominada CPU.

Segundo o autor (TURBAN, 2004), existem dois importantes tipos de *software*, o *software* de aplicação e o *software* de sistema, que juntos se complementam para executarem tarefas que vão desde o processamento de informações até aquelas que são responsáveis pelo funcionamento de um computador.

2.1.2.1 *Software* de Aplicação

O *Software* de aplicação é um conjunto de instruções, escrito em linguagem de programação, que direcionam o *hardware* a executar determinadas atividades no processamento de informações, fornecendo ao usuário algumas funcionalidades como: produção de texto, elaboração de planilhas eletrônicas, tocador de música e vídeo. Além de atender a necessidades pessoais, há também aqueles que auxiliam organizações, como programas para confecção de folha de pagamento, gestão para aumentar a produtividade da empresa, processamento de vendas, controlar as contas a pagar, a receber ou simplesmente para controlar o estoque, contribuindo assim nas tomadas de decisões (TURBAN, 2004).

2.1.2.2 *Software* de Sistema

Ainda conforme Turban (2004), o *software* de sistema independe de qualquer tipo de aplicação ou programa. São eles que dão suporte e controlam o *hardware* do computador gerenciando suas atividades de processamento. Um bom exemplo é o programa de inicialização de um computador, um *software* de sistema que prepara e disponibiliza todos os dispositivos para o processamento. Esses *softwares* podem ser agrupados em três categorias funcionais importantes:

- Programas de controle de sistemas: Programas que controlam o uso de *hardwares*, *software* e recursos de dados durante sua execução. Um exemplo é o sistema operacional.
- Programas de suporte ao sistema: Gerenciam operações como desempenho e monitoramento de segurança, programas utilitários de sistema, além de um específico controle de usuários do sistema de computador.

- Programas de desenvolvimento de sistemas: Auxiliam usuários no desenvolvimento de programas e aplicações que processam informações para usuários.

2.1.3 Sistema Operacional

Existem atualmente vários sistemas operacionais, tais como: *Windows*², *Linux*³, *Mac OS*⁴, *Android*⁵, *Unix*⁶, *Solaris*⁷ etc. Um sistema operacional (SO) é composto por vários programas que manipulam o *hardware* - parte física - do computador. Esses programas fornecem rotinas básicas para o controle desses dispositivos. Também oferecem rotinas que permitem o gerenciamento de processos simultâneos ou em nível de prioridade, ou seja, determinando a execução de programas necessários para comandar os componentes integrados ao computador seguindo uma lista em ordem de necessidade para a inicialização do computador mantendo a integridade do mesmo. (TANENBAUM, 2009).

Para Tanenbaum (2009) é difícil definir o que é, exatamente, um sistema operacional, pois depende muito do perfil do usuário e o objetivo de sua operacionalidade. Alega o autor que alguns sistemas operacionais são desenvolvidos para gerenciar servidores, outros para computadores de grande porte, computadores com multiprocessadores, computadores pessoais, computadores portáteis, celulares, *tablets*, sistemas operacionais embarcados entre outros.

Segundo o mesmo autor (TANENBAUM, 2009), o trabalho de um SO é fornecer aos programas do usuário um modelo de computador melhor, mais simples e mais limpo e lidar com o gerenciamento de todos os recursos dos dispositivos de *hardware*.

² <http://windows.microsoft.com/pt-br/windows/home>. Site oficial do Sistema Operacional Microsoft Windows.

³ <http://www.linux.com/>. Site oficial do Sistema Operacional Linux.

⁴ <https://www.apple.com/mac/>. Site oficial do Sistema Operacional Mac OS.

⁵ <https://www.android.com/>. Site oficial do Sistema Operacional Mobile Android.

⁶ www.unix.org/. Site oficial do Sistema Operacional Unix.

⁷ <http://www.oracle.com/us/sun/index.htm>. Site do Sistema Operacional Solaris produzido pela Oracle.

“O Sistema Operacional é a peça mais básica de software e opera em modo núcleo (também chamado modo supervisor). Nesse modo ele tem acesso completo a todo o hardware e pode executar qualquer instrução que a máquina seja capaz de executar.” (TANENBAUM, 2009, p.22).

A FIGURA 3 ilustra a dimensão de atuação do sistema operacional:

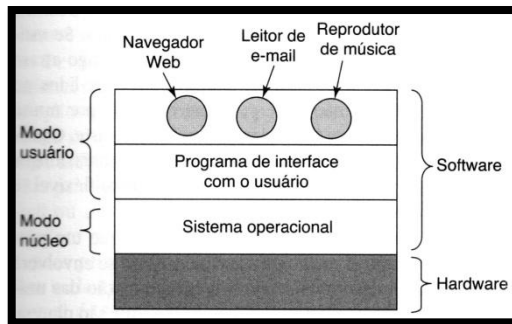


Figura 3 – Onde o sistema operacional se encaixa

Fonte: TANENBAUM (2009)

2.1.4 Sistemas de Informação

Segundo Marçula, (2008), os sistemas de informação desempenham um papel fundamental na maioria das organizações, pois a informação pode fornecer recursos que são extremamente importantes para que as empresas possam se desenvolver tornando-se mais competitivas. A informação também pode contribuir para o conhecimento das tendências do mercado além de inteirar-se sobre os passos da concorrência. Com base nesse conhecimento, é possível para o administrador desenvolver um planejamento de trabalho e tomar decisões no contexto de um cenário autêntico.

“Um sistema eficiente tem dimensões organizacional e humana, além dos componentes técnicos. Ele existe para responder as necessidades organizacionais, incluindo problemas apresentados pelo ambiente externo criado por tendências políticas, demográficas, econômicas e sociais”. (LAUDON, 1999).

2.1.4.1 Conceitos de Informação

De acordo com Stair (2011), informação é um dos recursos valiosos em uma organização. Além de fornecer um conjunto de fatos que, organizados, fornecem subsídios para administração por ser baseada na representação real de dados obtidos sobre um determinado assunto ou fato. É de suma importância para apoiar uma tarefa específica ou auxiliar na tomada de decisão, viabilizando, assim, o crescimento da empresa.

Para Stair (2011), a informação não pode ser confundida com dados. Dados representam fatos de um mundo real como total de horas trabalhadas em uma semana, número de produtos em estoque, quantidade do produto vendido no dia. A informação é quando esses dados são organizados de forma significativa e pode ser muito valiosa por contribuir para que pessoas e suas organizações possam desempenhar tarefas de forma mais eficiente e eficaz, tomando decisões e alcançando metas.

A (TABELA 1) ilustra a informação obtida através de dados.

DADO X INFORMAÇÃO	
DADO	INFORMAÇÃO
Data de nascimento do animal: 20/07/2010	Idade: 4 anos
Soma de Preço Unitário x Quantidade	Valor Total da Fatura: R\$ 2.500,00
Medição x Métrica da temperatura corporal humana = 40°	Febre
Taxa de glicemia humana = 270 mg/dL	Diabetes

Tabela 1 – Dado x Informação.
Fonte: elaborado pelo autor (2015).

2.1.4.2 Conceitos de Sistema

Em breves palavras, Marçula (2008), descreve que praticamente, qualquer coisa, como uma organização, por exemplo, pode ser vista como um sistema. Ele classifica os sistemas como simples, complexos, dinâmicos, estáticos, temporários ou permanentes e também pode ser classificado como sistema de ensino, sistema de computador etc.

“Um sistema é um Conjunto de elementos que interagem para realizar objetivos. Os próprios elementos e o relacionamento entre eles determinam como o sistema funciona. Os sistemas têm entrada, mecanismos de processamento, saídas e realimentação.”
(STAIR, 2011).

Estabelecendo uma conexão com a Informática, os sistemas eletrônicos visam a abranger a complexidade dos processos que se propõe a ordenar e transformar os dados em informações gerenciáveis e passíveis de análise para a tomada de decisões.

2.1.4.3 Sistemas de Informações Gerenciais

Os SIGs são sistemas que permitem aos administradores atingirem metas através de controles, organização e planejamento, pois o sistema fornece informações, por meio de relatórios, que dão uma visão básica das operações da empresa auxiliando os profissionais nas tomadas de decisões. Esses sistemas, normalmente integram os dados da corporação fornecendo um aspecto mais amplo e relacionado sobre a situação da empresa. (MARÇULA, 2008).

Para Batista (2004), sistema de informação gerencial:

“É o conjunto de tecnologias que disponibilizam os meios necessários à operação do processamento dos dados disponíveis. É um sistema voltado para a coleta, armazenagem, recuperação e processamento de informações usadas ou desejadas por um ou mais

executivos no desempenho de suas atividades. É o processo de transformação de dados em informações que são utilizadas na estrutura decisória da empresa que proporcionam a sustentação administrativa para otimizar os resultados esperados” (BATISTA, 2004).

Assim, é possível definir que sistemas de informação são mecanismos de apoio à gestão, desenvolvidos com base na tecnologia de informação e com suporte da informática para atuar como condutores das informações que visam facilitar, agilizar e otimizar o processo decisório nas organizações. A gestão empresarial precisa cada dia mais do apoio de sistemas, pois estes dão segurança, agilidade e versatilidade para a empresa no momento em que se processam as informações.

2.2 DESENVOLVIMENTO DE SOFTWARE

Segundo Marçula (2008), um *software* não é fabricado, mas projetado e desenvolvido onde se tem o maior custo na parte de projeto. Na maioria das vezes o *software* é desenvolvido sob medida, de acordo com a necessidade de controle e gerenciamento da empresa ou organização. Existem estudos que indicam que a maior parte do tempo gasto para desenvolvimento ocorre após o *software* já ter sido entregue devido a ajustes em rotinas e implantação de novos módulos para adequação dos requisitos necessários.

Para que um *software* tenha controle e garantia de qualidade, segundo opinião de Marçula (2008), deve-se, além da habilidade do programador, analisar, projetar e documentar de forma adequada para que modificações futuras possam ser interpretadas por outros programadores podendo assim ser realizadas de forma mais rápida, prática e sem maiores problemas.

2.2.1 Métodos de Desenvolvimento de *Software*

De acordo com Marçula (2008), existem muitos métodos para desenvolvimento de *software*, mas dois deles são os mais utilizados: método estruturado e o método orientado a objeto.

2.2.1.1 Método Estruturado

De acordo com Marçula (2008), o método estruturado é um método de desenvolvimento tradicional que foca nas funções que o sistema realizará, ou seja, compreendem o sistema como um conjunto de programas que executam funções utilizando dados. Dessa forma, no processo de modelagem do sistema a preocupação é descobrir os processos que devem ser realizados pelos programas, para depois descobrir quais dados são necessários para isso e qual a relação entre eles (criação dos bancos de dados).

2.2.1.2 Método Orientado a Objeto

De acordo com Marçula (2008), o método orientado a objeto vê o sistema de *software* como uma coleção de objetos que interagem entre si, representados pelos processos (Operações) e atributos (Dados). Enquanto o método estruturado tem o foco nas funções, o método orientado a objeto tem o foco no objeto, ou melhor, nas classes de objetos que possuem comportamento próprio, características representadas por seus atributos e podem ser compartilhadas no desenvolvimento da aplicação.

2.2.2 Modelos de Desenvolvimento de *Software*

De acordo com Marçula (2008), existem vários modelos de desenvolvimento de *software* e aborda alguns como exemplo:

- **Modelo em Cascata:** Criado em 1970 por W. W. Royce, também é conhecido como ciclo de vida clássico. É composto pelas fases de análise de requisitos, projeto, codificação, testes e manutenção. Nesse modelo cada fase deve ser aprovada para avançar à fase seguinte.
- **Modelo de Prototipação:** Apresenta três fases distintas: Coleta de requisitos, prototipação (projeto rápido) e avaliação do protótipo. Cria-se um protótipo de acordo com os requisitos do usuário submetendo os resultados a uma avaliação para que seja construído outro protótipo. Esse processo continua até a obtenção do produto final.
- **Modelo Incremental:** Nesse modelo, o *software* é desenvolvido por meio de aprimoramentos incrementais, uma mistura do modelo em cascata com o modelo de prototipação. Primeiramente é desenvolvida uma versão inicial do *software*, seguindo as fases do modelo em cascata, onde o usuário já pode ir trabalhando. A partir dessa primeira versão, o desenvolvedor vai incrementando funções ao *software*, criando uma segunda versão e assim sucessivamente até a satisfação do usuário.
- **Modelo Espiral:** Também é uma combinação do modelo em cascata com o modelo de prototipação. A ideia é que cada etapa do processo seja considerada um ciclo a ser realizado completamente, mas que envolva planejamento, análise de riscos, desenvolvimento, validação e planejamento do próximo ciclo.
- **Modelo de Desenvolvimento Concorrente:** É um modelo onde várias partes do *software* são desenvolvidas ao mesmo tempo. Não é um modelo adequado para qualquer tipo de *software*, mas pode ser usado com sistemas mais complexos que permite dividir seu desenvolvimento em partes.
- **Modelo de Desenvolvimento Baseado em Componentes:** Esse modelo parte do princípio que já existam partes de *software* criadas anteriormente (conhecidas como componentes) e se concentra em integrá-los e não desenvolvê-los.

- Modelo de Métodos Formais: Baseia-se na produção de uma especificação formal matemática transformando-a, por meio de métodos matemáticos, em um programa.

2.3 ANÁLISE E PROJETOS DE SISTEMAS

Para Wazlawick (2011), a fase de concepção de uma aplicação vai exigir do analista uma visão inicial e geral do sistema a ser desenvolvido. Essa visão pode ser obtida a partir de entrevistas, documentos e sistemas. A partir dessa visão será possível analisar e determinar as diretrizes que serão necessárias para a elaboração do projeto.

Segundo Marconi (2007), são inúmeros os benefícios que a análise de projetos oferece a uma empresa dentre os quais se destacam:

- Diferenciais competitivos e novas técnicas uma vez que toda a metodologia está sendo planejada.
- Antecipação de problemas permitindo uma ação corretiva antes que eles aconteçam.
- Molda o sistema de modo que atenda ao mercado consumidor e ao cliente usuário.
- Aumenta o gerenciamento de todas as fases implementadas já que os requisitos estão sendo estruturados e disponibilizados.
- Facilita e orienta a revisão da estrutura do projeto decorrente de modificações.
- Aperfeiçoa a alocação de pessoas, equipamentos e materiais necessários.
- Documenta e facilita a estimativa de melhorias e novas implementações no projeto.

2.4 LINGUAGENS DE PROGRAMAÇÃO (LP)

De acordo com Marçula (2008), a linguagem de programação é utilizada para desenvolver e criar programas, também denominados sistemas de computador, e pode ser definida como um conjunto de palavras, comandos e regras usadas para instruir um sistema de computador a realizar determinadas tarefas específicas. A LP, também conhecida como compiladores, normalmente transformam essas instruções em chamados código Fonte.

O programador escreve os comandos que vão realizar as tarefas desejadas na linguagem de programação de alto nível, por ser próxima da linguagem humana, usando palavras-chaves e *sintaxes* próprias que passa por um processo de compilação transformando essas instruções em código Fonte e conseqüentemente em linguagem de máquina que será processada pelo computador criando assim o programa de computador. (MARÇULA, 2008).

Segundo Marçula (2008), essas linguagens de Programação são categorizadas em “gerações” tendo relação com a época em que foi criada e a proximidade que a linguagem utilizada para escrever comandos tem com a linguagem natural humana. Conforme FIGURA 4, as LPs estão divididas em quatro gerações;

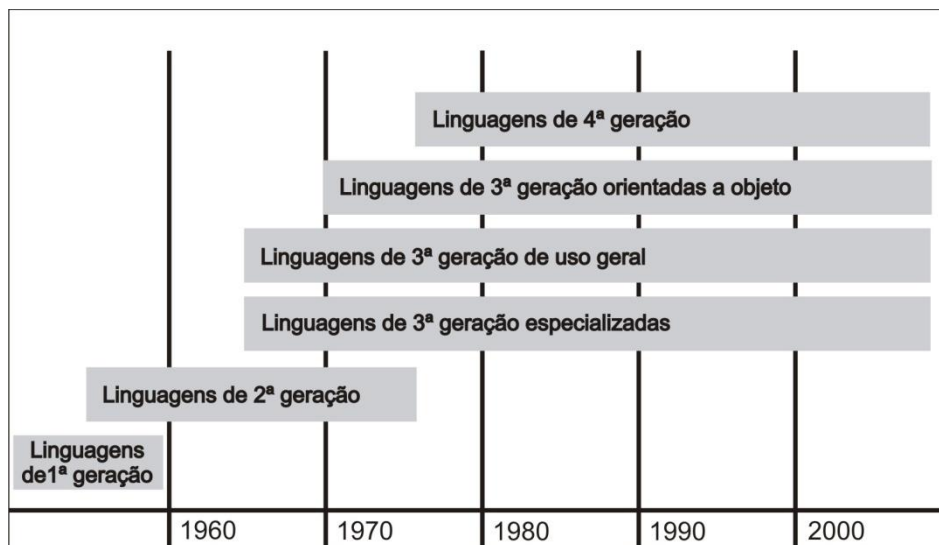


Figura 4 – Gerações de Linguagens de Programação.
Fonte: PRESSMAN (1995).

2.4.1 Linguagens de Primeira Geração

São tratadas como linguagens de baixo nível onde exigem que o programador escreva o comando praticamente no nível de linguagem de máquina com instruções escritas em código hexadecimal, cujo comando tem correspondência direta com o comando em micro código da CPU. Exemplo: Linguagem Assembly (MARÇULA, 2008).

2.4.2 Linguagens de Segunda Geração

São linguagens que apresentam avanços em reação a linguagem da geração anterior. Os comandos são dados por intermédio de palavras que são utilizadas no dia a dia (normalmente verbos em inglês). Destacam-se as principais (MARÇULA, 2008):

- *Fortran*: Reconhecida como a primeira linguagem de programação. É ideal para aplicações matemáticas embora seja inadequada para criação de *softwares* básicos.
- *Cobol*: Primeira linguagem voltada para criação de aplicações comerciais.
- *Basic*: Nasceu na década de 1970 e foi desenvolvida para o ensino de programação.
- *Algol*: Voltada para aplicações científicas, foi precursora das linguagens de terceira geração.

2.4.3 Linguagens de Terceira Geração

Também conhecidas como linguagens de programação estruturadas ou modernas seguem técnicas de programação estruturada e são classificadas em três categorias (MARÇULA, 2008) a seguir:

2.4.3.1 Linguagens de Alto Nível de Uso Geral

São linguagens baseadas em *Algol* e apresentam um amplo uso. São exemplos desse tipo de linguagem (MARÇULA, 2008):

- *Pascal*: Podendo ser usada para criação de aplicativos para áreas de engenharia, científicas e *software* básico, foi a primeira linguagem estruturada usada para criar e ensinar técnicas de programação.
- *PL/1*: Criada pela *IBM*⁸ foi a primeira linguagem verdadeiramente de amplo aspecto utilizada para desenvolvimento de aplicações comerciais.
- *C*: Linguagem desenvolvida para criar Sistemas Operacionais e aplicações comerciais.
- *Ada*: Linguagem criada pelo Departamento de Defesa dos Estados Unidos para sistemas de computação em tempo real para uso militar. Também é utilizado para outras aplicações.

2.4.3.2 Linguagens Orientadas a Objetos

São linguagens que permitem a implementação de modelos de análise e projeto orientados a objeto. São exemplos desse tipo de linguagem (MARÇULA, 2008):

- *Smalltalk*: Linguagem criada para explicar conceitos de orientação de objetos.
- *C++*: Derivada da linguagem *C* é usada para criação de uma ampla gama de aplicações.
- *Java*: linguagem de aplicação criada pela *Sun*⁹ apresenta vantagem de poder ser executada em qualquer sistema de computador.

⁸ <http://www.ibm.com/br/pt/>. Site oficial da IBM.

⁹ Sun é um Sistema desenvolvido pela empresa Oracle. <http://www.oracle.com/us/sun/index.htm>

- Linguagens de *Script (Internet)*: Melhoram as funcionalidades das páginas da *internet*. Alguns exemplos são: *JavaScript*, *VBScript* e *JScript*(*Microsoft*¹⁰).
- Linguagens *Internet*. Na maioria dos casos, são utilizadas para criação de páginas dinâmicas na *internet*. Alguns exemplos são: *PHP*, *ASP*, *JavaServer Pages*, *Perl* e *Phyton*.

2.4.3.3 Linguagens Especializadas

Criada para aplicações especiais apresentam formas de sintaxe de comandos bem diferentes. Alguns exemplos são (MARÇULA, 2008):

- LISP: Trabalha com a tradução e documentação de texto através de símbolos e listas. São usadas quase que exclusivamente para aplicações de inteligência artificial (IA).
- PROLOG: linguagem orientada a objeto e que também é utilizada em IA para desenvolvimento de sistemas especialistas.
- FORTH: Utilizada para desenvolvimento de *software* para microprocessadores.

2.4.4 Linguagens de Quarta Geração

São linguagens, até o momento, que apresentam o mais alto nível através da automatização de muitas tarefas. Podem ser classificadas em três categorias: Linguagens de consulta, geradores de programas e linguagem de prototipação. (MARÇULA, 2008).

¹⁰ <http://www.microsoft.com/>. Site oficial da Microsoft.

2.4.4.1 Linguagens de Consulta

São linguagens criadas para trabalhar em conjunto com banco de dados permitindo que esses dados sejam manipulados. Em alguns aspectos elas não podem nem ser consideradas linguagens de programação por não traduzirem os comandos para linguagem de máquina. Exemplo: SQL (*Structured Query Language*). (MARÇULA, 2008).

2.4.4.2 Geradores de Programas

São linguagens que permitem ao programador criar programas através dos modelos gerados pelos projetos. É uma linguagem de programação de nível muito elevado onde o trabalho de programação é muito pequeno. Algumas ferramentas denominadas *CASE (Computer-Aid System Engineering)* implementam esse tipo de linguagem. Exemplos de ferramentas *CASE*: *Rational Rose (IBM)*, *System Architect (Popkin)* e *ERWin (CA – Computer Associates)*. (MARÇULA, 2008).

2.4.4.3 Linguagens de Prototipação

São linguagens que facilitam a criação da *interface* com o usuário, conhecidas como linguagens visuais ou linguagens orientadas a evento. O programador tem a sua disposição um conjunto de componentes e se preocupa somente em programar o que deve ser executado quando o usuário interagir com a *interface* do programa. (MARÇULA, 2008).

Essas linguagens de programação proporcionam a prototipação, ou seja, permitem que sejam criados esboços visuais do programa final e com isso o usuário pode identificar problemas de operação, estética etc. São exemplos dessa linguagem: *Delphi (Embarcadero⁴)*, *Visual Basic* e *C# (Microsoft)*.

2.5 BANCOS DE DADOS

Segundo Rangel (2003), um banco de dados (BD) é o arquivo ou conjunto de arquivos onde se dá armazenamento de dados que serão disponibilizados para inclusão, alteração, exclusão e consulta por programas que atuam sobre ele. Há dois tipos de BD: os operacionais e os analíticos. Os operacionais são usados no dia a dia, quando existir a necessidade de coletar, armazenar e modificar dados. Têm a características de armazenar dados dinâmicos que se modificam constantemente e refletem a informação instantaneamente. Como exemplo, podemos citar o caso de controle de estoques, em que o estoque atual deve ser preciso e representar fielmente a quantidade de produtos existentes fisicamente a todo o instante.

O banco de dados analítico, diferentemente dos operacionais, é usado para acompanhar dados históricos. Esse tipo de banco de dados é chamado de “*Data Warehouse*”. Sua principal função é fornecer informações gerenciais para auxiliar na tomada de decisões. Geralmente são atualizados periodicamente com dados fornecidos pelos bancos de dados operacionais. (RANGEL, 2003).

2.6 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos é a etapa do desenvolvimento de sistemas de informação responsável por colher as ideias do usuário, identificando e modelando as necessidades do negócio a serem atendidas pelos sistemas de informação, e é, portanto, uma atividade cada vez mais relevante em um cenário cada vez mais dinâmico.

A tarefa de levantar requisitos para desenvolver um aplicativo não é nada fácil. Segundo Melo (2010), esse desenvolvimento leva em consideração o fato de estarmos lidando com um ser humano que expressa suas ideias, geralmente fora de ordem e sem muita explicação. É difícil encontrarmos usuários que nos forneça a ‘receita do bolo’ pronta para o desenvolvimento de uma aplicação. Normalmente as ideias são bem distorcidas, confusas e, por não conhecerem o modo que o computador processa as informações, acabam sendo problemáticas.

Para Melo (2010) cabe ao analista pegar todo esse emaranhado de ideias e necessidades, organizá-las de forma que sejam documentadas, transformando numa linguagem que possa ser compreendida por ambas as partes, para que assim o aplicativo chegue o mais próximo possível do desejado, segundo a necessidade de usabilidade do cliente.

3 METODOLOGIA

Neste capítulo será descrito o andamento geral do projeto. Serão apresentadas as metodologias empregadas, assim como os métodos utilizados na sua elaboração e desenvolvimento.

3.1 PROJETO DO SISTEMA

O Sistema foi projetado baseado em uma análise e levantamento de requisitos levando em consideração, principalmente, as expectativas e necessidades dos gestores da empresa. Após esse levantamento elaborou-se um diagrama de entidade e relacionamento, um fluxograma e, a partir de então, definiu-se a linguagem de programação e banco de dados a serem utilizados no desenvolvimento do *software*.

3.1.1 Análise e Levantamento de Requisitos

Como em todo projeto de sistema, o levantamento de requisitos tem um papel fundamental para o desenvolvimento de uma ferramenta que seja realmente funcional e atenda às necessidades reais da empresa, assessorando para uma gestão e administração baseada em informações mais precisas. Para esse propósito foram realizadas reuniões com os gestores da empresa a fim de analisar e identificar as necessidades de controle e gestão e obter informações sobre como deveria ser e o que deveria ter um aplicativo para atender, de forma geral, as expectativas.

As reuniões para tratar do levantamento dos requisitos foram efetuadas no ambiente de trabalho do *pet shop*, por ser entendido que isso contribuiria para uma análise baseada no funcionamento do ambiente comercial. Essa estratégia serviu para uma melhor compreensão da realidade do dia a dia de um *pet shop*,

contribuindo para o desenvolvimento de um aplicativo com o propósito de criar uma afinidade na relação sistema/usuário/*pet shop*.

Num primeiro momento foram identificadas algumas situações que exigiram atenção para a elaboração de um projeto que atendesse à necessidade da empresa a fim de agilizar o atendimento aos clientes. Um ponto importante abordado foi a necessidade de elaborar e organizar uma agenda de serviços necessária para marcar um horário de atendimento para cada *pet*¹¹ contendo informações de: o serviço a ser executado; a data e horário de início do serviço; qual funcionário que executará o trabalho; se é para buscar e levar o *pet*; informações sobre temperamento (arredio, calmo, agressivo); se está vacinado, desverminado e despulgado; horário previsto para finalização do serviço e valor cobrado.

Devido à precisão das informações sobre vacinação e medicamentos administrados pelo *pet*, houve a ideia de incluir tais campos de controle no cadastro de animais para que seja consultado e atualizado durante o agendamento dos serviços, podendo ficar gravados num histórico para acompanhamento, tanto do *pet shop* quanto do médico veterinário que por ventura tenha a clínica veterinária interligada utilizando o mesmo sistema. Também foi relatada a necessidade de incluir fotos do *pet* no cadastro de animais para que seja mais fácil a identificação do mesmo. Portanto, foi decidido que seria interessante implementar uma galeria de fotos.

Continuando no cadastro de animais, foi solicitada, pelos administradores do *pet shop*, a inclusão de informações como: castração; sexo; idade; adoção; espécie; cor; raça; temperamento; vacinação e medicamentos; uso de medicamentos contínuos; histórico de doenças; último atendimento médico, exames e serviços realizados.

No que concerne à necessidade de um melhor gerenciamento sobre as contas a receber, se fez importante construir um módulo para armazenar os dados referentes às vendas a prazo efetuadas aos clientes que pagam mensalmente, pois, o controle até então, se fazia por meio de anotações em livros que acabavam dificultando o acesso às informações por essas serem lançadas em folhas de controles diários juntamente com a contabilidade do caixa.

¹¹ Significa animal de estimação em inglês. O termo é também utilizado no Brasil.

As contas a pagar também foram alvo da análise e levantamento de requisitos devido à necessidade de organizar, de forma sistêmica, os débitos da empresa para que fossem visualizados, através do aplicativo, por ordem de data de vencimento tornando o controle mais preciso e disponibilizando rapidamente a informação por meio de relatórios.

Com o controle de contas a pagar e receber houve a importância de estruturarmos um módulo para lançamentos de caixa onde seriam informados a abertura e o fechamento do mesmo, disponibilizando o saldo atual durante o período da movimentação de entradas e saídas do caixa. Com essas informações será possível analisar, ao final de um período, os dias de maior movimento contribuindo para tomadas de decisões.

3.1.2 Diagrama de Entidade e Relacionamentos

O diagrama de entidade e relacionamento (DER) utilizado para a modelagem do banco de dados, a partir dos requisitos levantados, discutidos e analisados, colaborou para a otimização do desenvolvimento das funções do sistema oferecendo uma nuance das informações a serem adquiridas através do conjunto de dados relacionados e processados pelo aplicativo. O diagrama ilustrado está disponível no (ANEXO 1).

3.1.3 Fluxograma

Fluxograma é uma sequência lógica de procedimentos inter-relacionados cujo objetivo é orientar a execução de uma tarefa ou atividade. Os fluxogramas servem para tornar o processo decisório e executivo simples, direto e claro. São frequentemente utilizados no desenvolvimento de sistemas para modelar de dados, no meio científico e empresarial. Contribuem para sistematizar rotinas e tarefas estruturando a execução das atividades, assim como a sequência das ações necessárias para atingir um objetivo.

Após a fase de levantamento de requisitos elaborou-se as tabelas do banco de dados responsáveis pelo armazenamento dos dados de acordo com a modelagem baseada no fluxograma definido como ilustrado no (ANEXO 2).

3.2 IMPLEMENTAÇÃO

A fase de implementação, ou seja, o ciclo de vida de um *software* consiste na decisão sobre quais ferramentas serão utilizadas para a elaboração do aplicativo, bem como sua documentação baseada em levantamentos de requisitos, estudos, análises, diagramas, fluxogramas e, posteriormente, as etapas de desenvolvimento, utilização e manutenção do sistema como produto final.

3.3 LINGUAGEM DE PROGRAMAÇÃO DELPHI

Delphi foi a linguagem de programação adotada para ser utilizada como ferramenta para o desenvolvimento do *software* SGPET. Desenvolvido em 1995 pela empresa *Borland Software Corporation*⁹, o nome *Delphi* veio de uma inspiração na cidade de *Delfos*, na Grécia, e seu ambiente de desenvolvimento, viabiliza a produtividade facilitando a programação, sendo capaz de permitir a criação de aplicativos para diversos sistemas operacionais. Atualmente o *Delphi* pertence à *Embarcadero Technologies*¹⁰, uma empresa norte americana produtora de *software*, fundada em outubro de 1993 por *Stephen Wong* e *Stuart Browning*.

Considerado uma linguagem de prototipação de alto nível pertencente à quarta geração, o *Delphi* também permite o desenvolvimento orientado a objetos. Seu ambiente de trabalho definido como RAD (Rapid Application Development) possibilita ao desenvolvedor elaborar o protótipo da interface gráfica do aplicativo em tempo de desenvolvimento, tornando o trabalho mais rápido e menos complicado. Esse protótipo é a definição final dos formulários que serão utilizados pelo usuário.

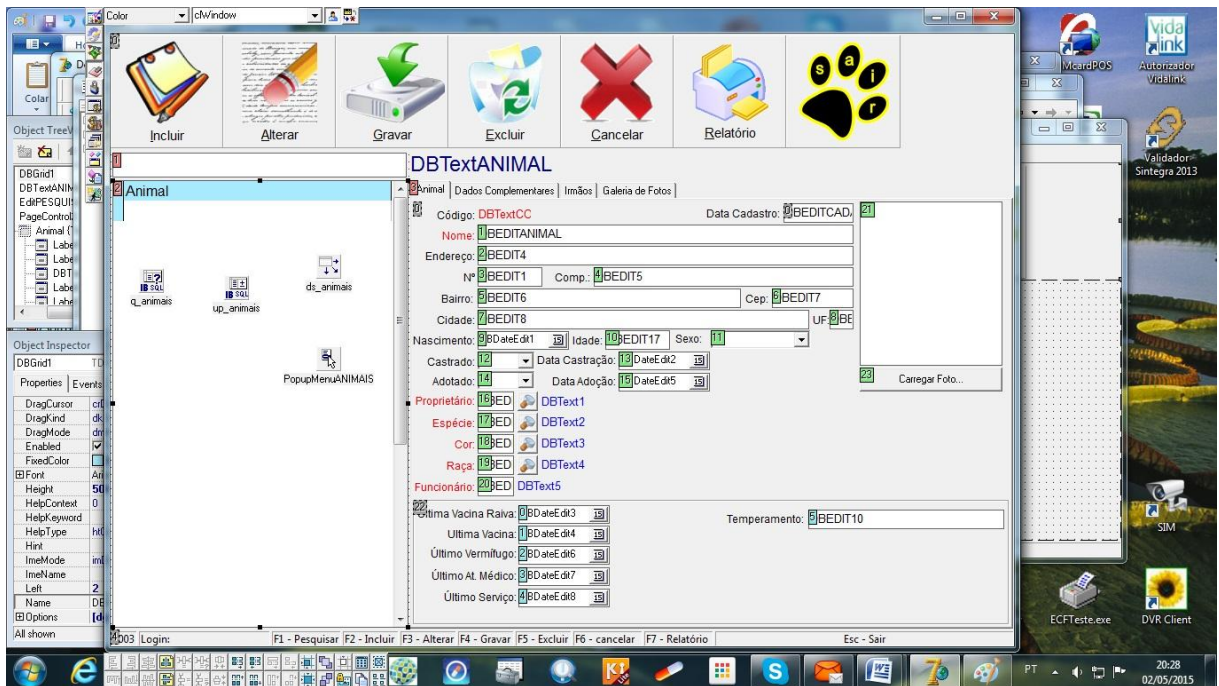


Figura 5 – Formulário de cadastro de animais do SGPET em ambiente de desenvolvimento Delphi.
 FONTE: elaborado pelo autor (2015)

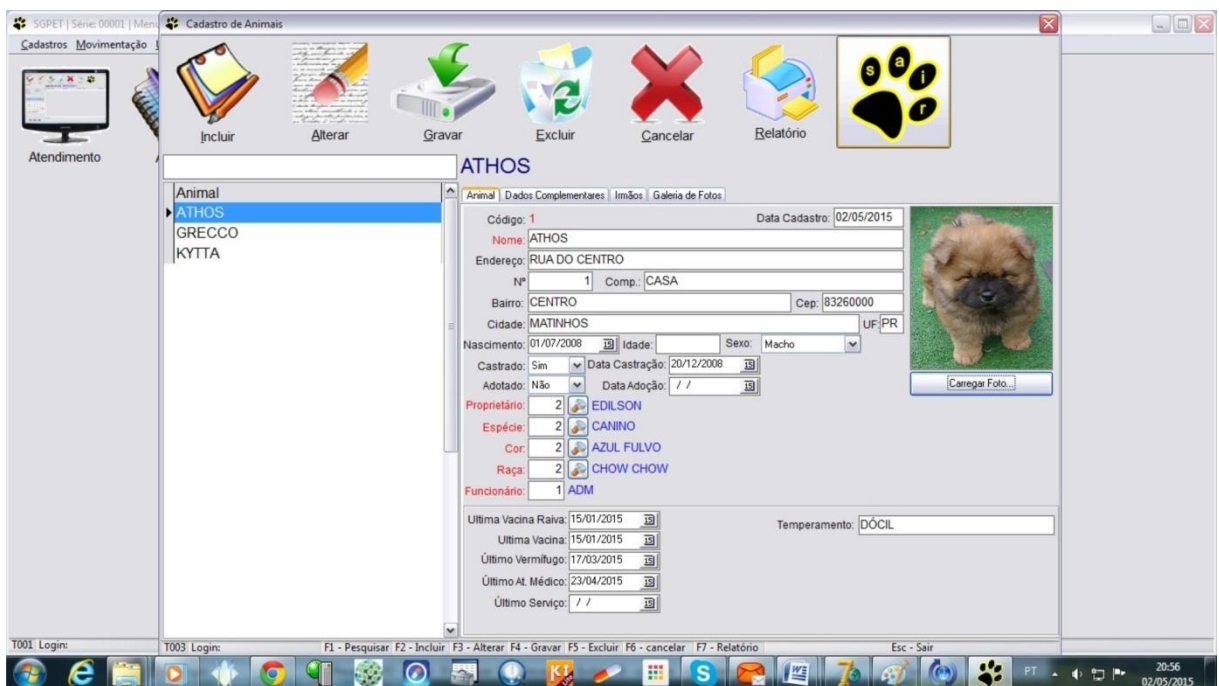


Figura 6 – Formulário de cadastro de animais do SGPET em ambiente de aplicação.
 FONTE: elaborado pelo autor (2015)

3.3.1 Recursos da Linguagem de Programação *Delphi*

A linguagem de programação *Delphi* possui inúmeros recursos que podem auxiliar o desenvolvedor de *software* na sua jornada de produção de um aplicativo. Na sua versão mais recente, XE7, o *Delphi* possui recursos muito interessantes como desenvolvimento de aplicativos para plataformas 32 e/ou 64 *bits* e Sistema Operacional *Mobile Android*¹¹, *IOS*¹² e *Windows Mobile*¹³, assim como outros recursos herdados de versões anteriores. Além dos citados, os recursos mais relevantes são:

:

- Desenvolvimento *Web* com *Intraweb* e *Extraweb*.
- Desenvolvimento de aplicações *Client/Servidor*.
- Desenvolvimento de aplicações para *Desktop*.
- Desenvolvimento de aplicações em 3 camadas.
- Programação Orientada a Objeto.
- Suporte para desenvolvimento de aplicações *.NET*.
- Conectividade com diversos bancos de dados.

3.4 BANCO DE DADOS FIREBIRD

*Firebird*¹⁸ é um sistema gerenciador de banco de dados que roda em *Linux*¹⁴, *Unix*¹⁵, *Windows*¹⁶ e *MacOS*¹⁷. Na sua versão mais recente, 2.5, foi o sistema gerencial de banco de dados (SGDB) escolhido para ser o banco de dados no desenvolvimento desse projeto. A opção por esse banco se dá pelo fato de ele ser de código aberto, não tendo a necessidade de adquirir uma licença para utilizá-lo. Desse modo, não onera o usuário e também por ser um banco de dados confiável que suporta sistemas com centenas de usuários simultâneos e centenas de *gigabyte* de armazenamento de dados, além de um suporte totalmente gratuito.

3.4.1 Recursos do *Firebird*

Abaixo, segue alguns dos principais recursos desse poderoso sistema de gerenciamento de banco de dados chamado *Firebird*:

- Suporte a *Stored Procedures* e *Triggers*
- Transações compatíveis com ACID
- Integridade Referencial
- Multi *Generational Architecture*
- Consome poucos recursos de processamento
- Linguagem nativa para *Stored Procedures* e *Triggers* (PSQL)
- Suporte para Funções Externas (UDFs)
- Praticamente não necessita de DBAs especializados
- Quase nenhuma configuração – Instalação simples.
- Grande comunidade de usuários e vários lugares para se obter suporte gratuito
- Versão *embedded* do SGBD - perfeita para criação de catálogos em CDROM, aplicações "demo" ou *stand-alone*
- Dezenas de ferramentas de terceiros, incluindo aplicações gráficas de administração, replicação de dados, etc.
- *Careful writes* – Recuperação rápida dispensando o uso de *log* de transações.
- Formas diversas de acesso ao banco de dados: nativo/API, *dbExpress*, ODBC, OLEDB, .Net *provider*, JDBC nativo tipo 4, *Python* module, PHP, Perl, etc.
- Suporte nativo para os maiores sistemas operacional: *Windows*, *Linux*, *Solaris*¹⁸ e *MacOS*.
- Backups incrementais
- *Builds* de *64bits* disponíveis
- Total controle de cursores em PSQL
- Tabelas de Monitoramento
- *Triggers* de conexão e transação
- Tabelas temporárias
- *TraceAPI* - saiba o que está acontecendo no servidor.

3.5 INTERFACE DO APLICATIVO

Um aplicativo precisa ser agradável, simples, eficiente, eficaz e confiável. Certamente uma interface objetiva, clara e intuitiva colabora muito para esses pré-requisitos. Para o desenvolvimento do *Software* SGPET, levaram-se em consideração alguns fatores relevantes no intuito de atender usuários menos experientes sem, contudo, prejudicar o desempenho operacional do sistema.

Algumas funcionalidades na *interface* do *software* foram definidas para simplificar e agilizar a usabilidade como, por exemplo: botões maiores e com ícones sugestivos nos formulários de cadastros que também contam com a interligação desses botões às teclas de Função de acesso rápido como: F1, F2, F3 etc., facilitando e agilizando na parte operacional; mudança de um campo para outro através da tecla *Enter*; campos com alinhamento vertical e simétrico; Fonte de letras maiores que o padrão, auxiliando na visualização; resolução de tela de 1024x720.

Foram definidas nas telas zonas funcionais com o máximo de clareza possível a fim de colaborar para uma *interface* agradável, “mais leve”, possibilitando ao usuário trabalhar mais tempo sem ficar entediado ou aborrecido.

Para os botões que compõem o sistema, foram pensadas imagens com tendências lúdicas para deixar o ambiente de trabalho do sistema mais informal e agradável “quebrando” um pouco do visual “pesado” e “frio” que normalmente alguns sistemas carregam. Outro aspecto estudado foram as cores utilizadas em todo o aplicativo. Tomou-se o cuidado para utilizar, no máximo, cinco cores em uma tela, na sua maior parte, com tendências neutras, evitando assim uma confusão visual tornando a usabilidade menos cansativa. Além disso, procurou-se relacionar algumas dessas cores a uma função distinta como, por exemplo, o nome do campo em vermelho alertando para que o campo seja, obrigatoriamente, preenchido.

3.6 FUNCIONALIDADES DO APLICATIVO

O *Software* SGPET contempla várias funcionalidades necessárias para o controle e administração das informações como, por exemplo:

- Cadastros de: animais, caixas, classificação, cliente, convênio, cores, espécies, fabricantes, forma de pagamento, fornecedores, funcionários, grupos, lojas, médicos veterinários, produtos e serviços, raças e tributação.
- Vendas: vendas de produtos e serviços ao consumidor.
- Agendamento de serviços e consultas.
- Controle de abertura, movimentação e fechamento de caixa.
- Relatórios
- Comissão de funcionários sobre vendas e serviços.
- Controle de vencimento de produtos.
- Aviso de revacinação.
- Prontuário médico veterinário.
- Fotos de identificação do Pet.
- Contas a Pagar.
- Contas a Receber.
- Estoque: entrada, saída, movimentação e balanço.
- Vendas: consulta e estorno de vendas e serviços.
- Backup dos dados e informações.

3.7 IMPLANTAÇÃO DO APLICATIVO EM UM *PET SHOP*

Foi estabelecido que a implantação do *Software* SGPET seria em uma empresa do ramo na cidade de Matinhos, litoral do Paraná, devido o mesmo ser objeto de estudos para realização de projeto de aprendizagem (PA), disciplina presente ao longo dos oito semestres do curso de Informática e Cidadania, conforme proposta no PPP (projeto Político Pedagógico) da Universidade Federal do Paraná.

O processo de implantação do SGPET iniciou-se logo após o desenvolvimento do primeiro módulo do aplicativo que contemplou as partes de: cadastros de animais, caixa, classificação, clientes, convênios, cores de animais, espécies, fabricantes, formas de pagamento, fornecedores, funcionários, grupos de

preços, lojas, médicos veterinários, produtos e serviços, raças, tributação e atualização de versão.

Concomitantemente ao processo de implantação foi iniciado o treinamento ao usuário, orientando sobre a utilização das partes referentes ao módulo inicial.

Alguns contratemplos ocorreram nessa primeira fase, dificultando o andamento inicial da implantação. Tal dificuldade se deu por conta de que, alguns dias depois do início da implantação e a segunda etapa do treinamento, o computador no qual foi realizada a instalação do sistema veio a ter problemas com o processador que, subitamente, por razões de superaquecimento, “queimou” ocasionando um atraso de aproximadamente três meses no andamento dessa fase inicial.

3.8 TREINAMENTO E SUPORTE AO USUÁRIO

O treinamento aos usuários do sistema foi aplicado em várias etapas com intervalos de um mês. Essa decisão levou em consideração o fato de usuários menos experientes terem dificuldade em absorver todo o conteúdo das funções do sistema de uma única vez. Portanto, devido a essa estratégia, observou-se um melhor rendimento do usuário no processo de aquisição do conhecimento da ferramenta. O treinamento foi dividido nas cinco etapas seguintes:

- Primeira etapa: instrução de como efetuar *login* de acesso ao aplicativo e como efetuar uma inclusão, alteração e exclusão de dados em todos os módulos de cadastros que compõem o sistema; explicar a relação sistema/computador; os cuidados que devem ser tomados para a preservação e integridade dos dados e informações contidas no banco de dados assim como efetuar cópia de segurança diariamente, também chamado de *backup* e a aquisição de um *no-Break*, equipamento composto por uma bateria interna responsável por alimentar o computador em caso de queda de energia¹².

¹² A falta de energia, ocasionalmente, corrompe o banco de dados e até mesmo danifica o computador.

- Segunda etapa: revisão do treinamento da primeira etapa e sanar as dúvidas remanescentes; apresentação e abordagem dos módulos de: abertura, movimentação e fechamento de caixa assim como o módulo de agendamento de serviços.
- Terceira etapa: Nessa etapa, além da revisão das etapas anteriores, o controle de estoque, entradas, saídas, movimentações e balanço de estoque e o módulo de venda de produtos e serviços foram definidos como objetivos desse treinamento.
- Quarta etapa: Contas a pagar e receber, comissão de funcionários, relatórios e revisões de etapas anteriores finalizaram a parte de orientação das funções do sistema.
- Quinta etapa: esclarecimento de dúvidas gerais e finalização com a definição de como seria realizada as atualizações de versões que contemplariam alterações e novas funcionalidades no *software*. Após, houve uma avaliação de novos requisitos onde foram identificadas a necessidade da emissão de cupom fiscal e nota fiscal de serviços (NFS-e), gerando assim novas implementações.

A utilização de ferramentas de acesso remoto também contribuiu para dar suporte ao usuário. Por algumas vezes foi necessário fazer uso dessas ferramentas para acessar o computador do usuário via *internet* e avaliar erros sistêmicos ou dificuldades de utilização oferecendo uma solução ou auxílio para uso correto da aplicação.

A comunicação por email e telefone também foi importante para viabilizar o esclarecimento de dúvidas em relação aos módulos abordado em cada etapa na impossibilidade de uma visita fora das etapas previstas para treinamento.

3.9 MANUTENÇÃO DO APLICATIVO

Conforme acordo estabelecido com os empresários do *pet shop* e firmado pelo termo de consentimento livre e esclarecido, o sistema SGPET será disponibilizado para utilização, sem custo, com direito a manutenção e novas

implementações que porventura vierem a ser inseridas em novas versões. Esse acordo será mantido enquanto as atividades do *pet shop* estiverem sendo exercidas de forma legal, estando com o cadastro nacional de pessoa jurídica (CNPJ) ativa e em conformidade com a legislação nacional.

A manutenção do aplicativo será efetuada conforme a necessidade de implementação de outros módulos, de acordo com a observação de novos requisitos, que contemplam mudanças propostas para a adequação das funcionalidades do sistema, permitindo que o aplicativo tenha uma otimização no gerenciamento das informações, oferecendo assim mais recursos aos usuários.

No intuito de aprimorar a aplicação, será realizado um acompanhamento, uma vez ao mês, chamado de visita técnica, com a finalidade de obter informações sobre a usabilidade, eficiência e eficácia do sistema. Com base nessa análise, será rediscutido, se houver necessidade, possíveis alterações que venham viabilizar melhorias na funcionalidade do *software*.

Caso sejam reportados *bugs* ou anomalias, essas falhas deverão ser analisadas a fim de reproduzir e identificar o erro que será devidamente documentado e posteriormente passará a ser desenvolvida uma solução de correção sendo disponibilizada ao cliente/usuário em forma de nova versão. Uma versão contempla toda e qualquer alteração ou implementação, seja ela grande ou pequena, efetuada no sistema. Essa versão poderá ser atualizada *online* pelo usuário através do módulo do aplicativo definido como atualização de versão.

4 RESULTADOS E DISCUSSÕES

Como resultado do desenvolvimento do SGPET, foram obtidos diversos módulos que contribuíram para o objetivo proposto. As telas de cadastros, com campos para coletas de dados, definem a parte fundamental para o processamento de informações tão importante para a gestão da empresa. Nos parágrafos seguintes serão disponibilizadas ilustrações das telas de alguns módulos do aplicativo obtidos como resultados da análise e desenvolvimento de *software*.

A tela de acesso ao sistema, com senha criptografada no banco de dados, permite ao usuário acessar o SGPET com segurança. Cada acesso efetuado pelo usuário é gravado no banco de dados para efeito de controle e possível verificação posterior. Estuda-se a possibilidade de programar um módulo de controle ponto. Para esse fim, será necessário verificar se a legislação atual permite.

A FIGURA 7 ilustra a tela de acesso ao sistema SGPET.



Figura 7 – Tela de senha de acesso do SGPET.
FONTE: elaborado pelo autor (2015)

Na tela inicial do sistema, conforme FIGURA 8, temos um *layout* simples com botões de acesso rápido aos módulos mais utilizados no sistema. Na parte superior ficam os menus de acesso aos demais módulos do SGPET.

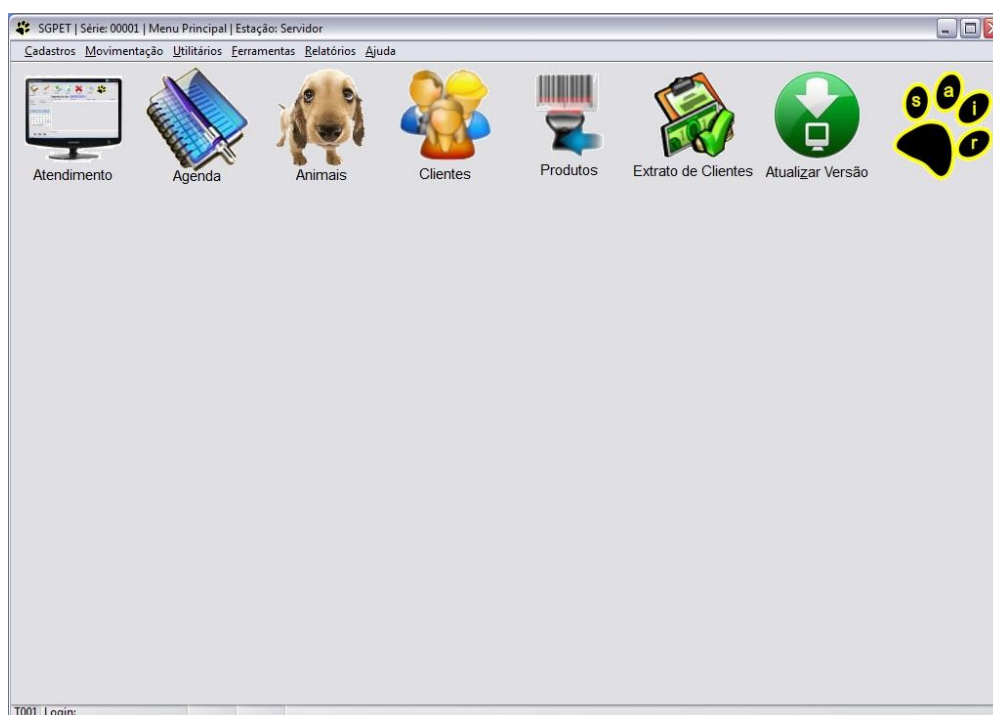


Figura 8 – Tela inicial do SGPET.

FONTE: elaborado pelo autor (2015)

O cadastro de clientes, também chamados de tutores, proprietários dos *pets*, disponibiliza ao usuário e seus administradores algumas informações pertinentes ao controle e gerenciamento do sistema SGPET. Esse cadastro é relacionado com o cadastro de animais de estimação a fim de vincular o Tutor ao *pet*. A FIGURA 9 ilustra essa tela.

Figura 9 – Tela do cadastro de clientes do SGPET.
 FONTE: elaborado pelo autor (2015)

Para efeito de controle de entradas e saídas de recursos financeiros, temos na FIGURA 10, a abertura do caixa. Esse procedimento deverá ser executado todos os dias assim que iniciar as atividades da empresa. Vale lembrar que se não for realizada a abertura de caixa, os módulos vinculados a movimentação financeira, tais como vendas de produtos, agendamento de serviços, recebimento de contas e lançamentos de entradas e saídas do caixa, não serão liberados para utilização por não estarem em conformidade com as regras de negócios do *software*.



Figura 10 – Tela de abertura de caixa do SGPET.
FONTE: elaborado pelo autor (2015)

O fechamento de caixa, conforme ilustra a FIGURA 11, permite ao usuário/administrador do SGPET fazer uma leitura rápida da posição financeira atual das entradas e saídas do caixa. Proporciona também o acesso detalhado dos lançamentos efetuados no caixa, possibilitando uma análise mais detalhada desses lançamentos.

Fechamento de Caixa

CAIXA PRINCIPAL
Caixa aberto em 24/05/2015 - 20:32:12
Operador: ADM

Confirmar Cancelar Relatório

Resumo Lançamentos (Resumo) Lançamentos (Detalhado)

Saldo de abertura: 100,00
Total de Créditos: 0,00
Total de Débitos: 0,00
Saldo Atual: 100,00

Ticket Médio:

F12 - Confirmar F7 - Relatório ESC - Sair

Figura 11 – Tela de fechamento de caixa do SGPET.
FONTE: elaborado pelo autor (2015)

O módulo de contas a receber, conforme ilustrado na FIGURA 12 disponibilizará ao usuário/administrador do SGPET uma forma de controlar o recebimento de contas de clientes pendentes de pagamento, também muito conhecido como “fiado”. Embora com todo o recurso tecnológico oferecido através dos cartões de crédito e débito, essa forma de pagamento ainda é muito utilizada no meio comercial, principalmente em cidades menores. Esse módulo calcula juros e multas das contas atrasadas além de gerar relatórios da posição financeira das contas pendentes e possibilita uma visualização dos extratos de toda movimentação, de todos ou de um determinado cliente.

Extrato de Contas a Receber

Marcar Todos
 Desmarcar Todos
 Inverter Seleção
 PARCIAL
 TOTAL
 Baixa Parcial
 Baixa Total
 Cancelar
 Relatório

Cliente:
 Cupom:
 Data inicial:
 Data final:

Cliente: **CLIENTES EM GERAL**

Cupom	Data Venda	AT	Vencimento	Vlr Parcela	Desc.	Dt. Abatimento	Vlr Abatido	Juros Rec.	Juros Rest.	Juros Atuais	Reajuste	Total	F12-Sel

Total Vencido: 0,00
Total a Vencer: 0,00
 Total das Notas: 0,00
 Juros Restante: 0,00
 Juros (R\$): 0,00
 Valor Reajuste: 0,00
 Total Com Juros: 0,00
 Selecionados: 0
 Valor Selecionados: 0,00

F2 - Marcar Todos
 F3 - Itens
 F4 - Desmarcar Todos
 F5 - Inverter Seleção
 F6 - Cancelar
 F7 - Relatório
 Ctrl+F3 - Alterar Vencimento
 F9 - Baixa Parcial
 F10 - Baixa Total
 F11 - Imprimir Extrato
 F12 - Selecionar Registro
 Esc - Sair

Figura 12 – Tela de recebimento de contas do SGPET.
 FONTE: elaborado pelo autor (2015)

O módulo de agendamento de serviços permite um melhor controle e ganho de tempo nos serviços oferecidos pelo pet shop, além de melhorar o atendimento aos clientes oferecendo agilidade e qualidade. Esse módulo proporcionou, por intermédio do controle da agenda, que um menor número de *pets* fique a espera de atendimento no local, evitando dessa forma o estresse do animal causado nessas circunstâncias. A FIGURA 13 ilustra a tela de agendamento de serviços.

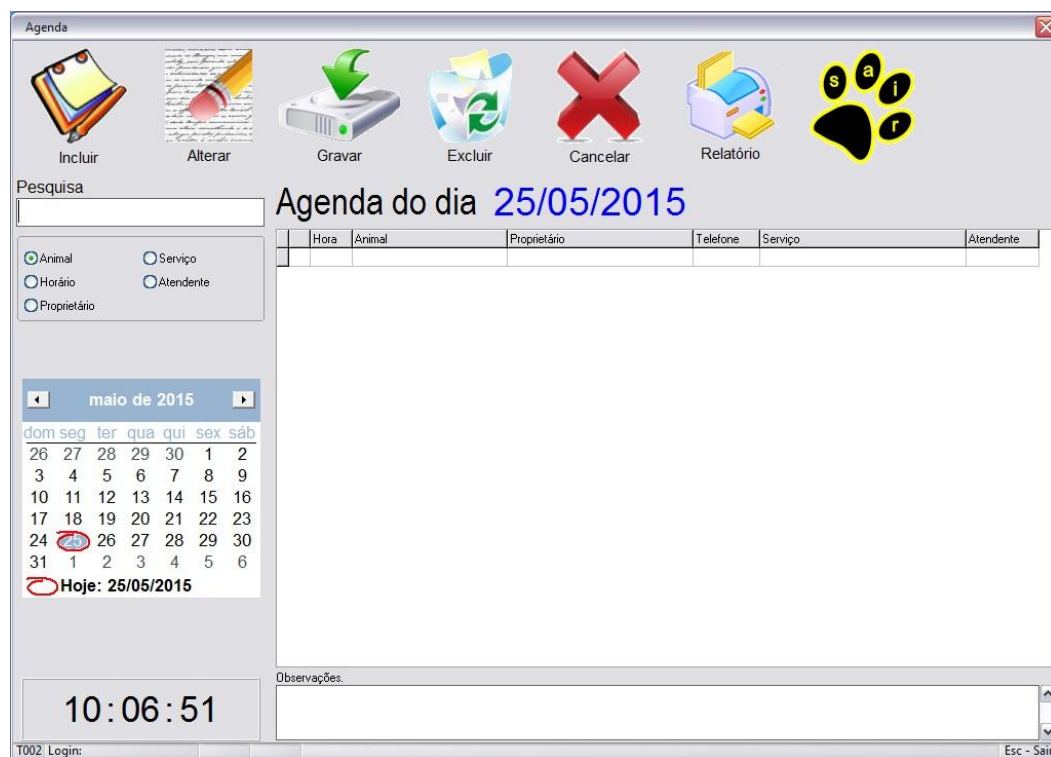


Figura 13 – Tela de agendamento de serviços do SGPET.
 FONTE: elaborado pelo autor (2015)

Todas as vendas e serviços prestados aos clientes serão efetuados no módulo denominado vendas, ilustrado pela FIGURA 14. As vendas controlam o estoque retirando a quantidade do produto vendido, determinando assim uma posição atual do mesmo. Também efetua o lançamento, no caixa, das formas de pagamento selecionadas na conclusão da operação, além de contabilizar as comissões dos atendentes. Nas vendas a prazo, é realizado o lançamento no controle das contas a receber dos clientes. Esse módulo é um vínculo de relacionamento com vários outros, tornando o sistema “amarrado” nas operações realizadas, obtendo assim um controle relacional sobre vários dados, disponibilizando rapidamente as informações de forma mais precisa.

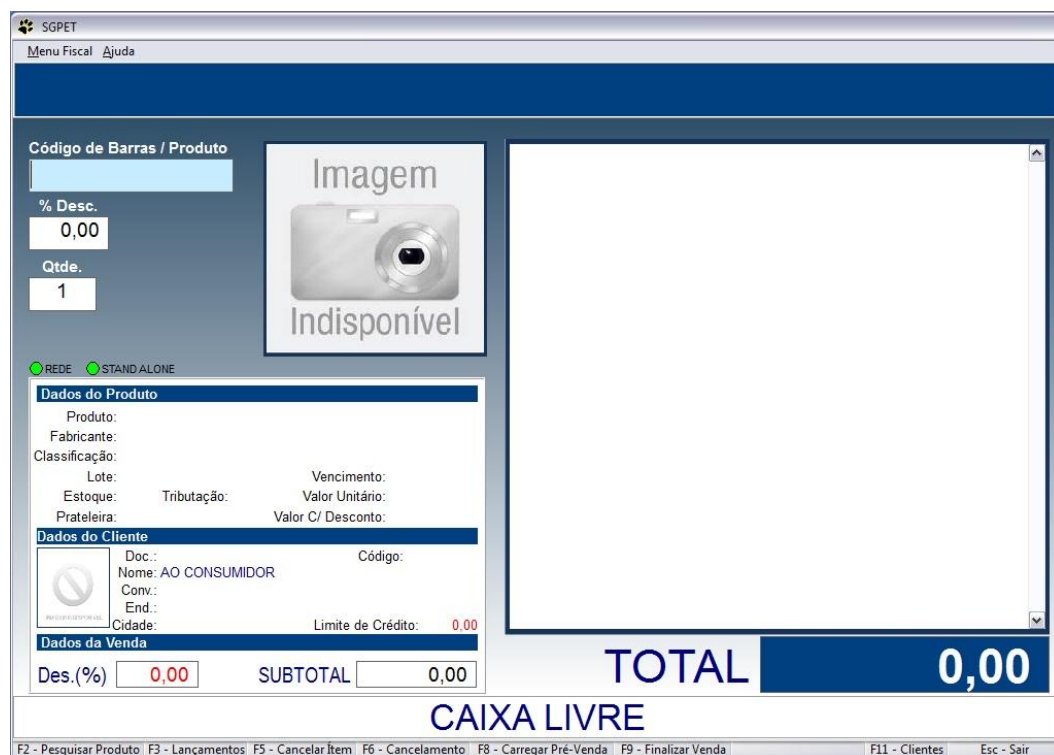


Figura 14 – Tela de vendas de produtos e serviços do SGPET.
 FONTE: elaborado pelo autor (2015)

A Atualização de versão, representada na FIGURA 15, dispensará o suporte em virtude da facilidade que o sistema oferece para a execução desse procedimento. Quando houver uma nova versão disponível, o próprio usuário/administrador do SGPET poderá executar a ferramenta de atualização da versão bastando clicar sobre o botão correspondente e o sistema se encarregará de aplicar as alterações disponíveis na nova versão.

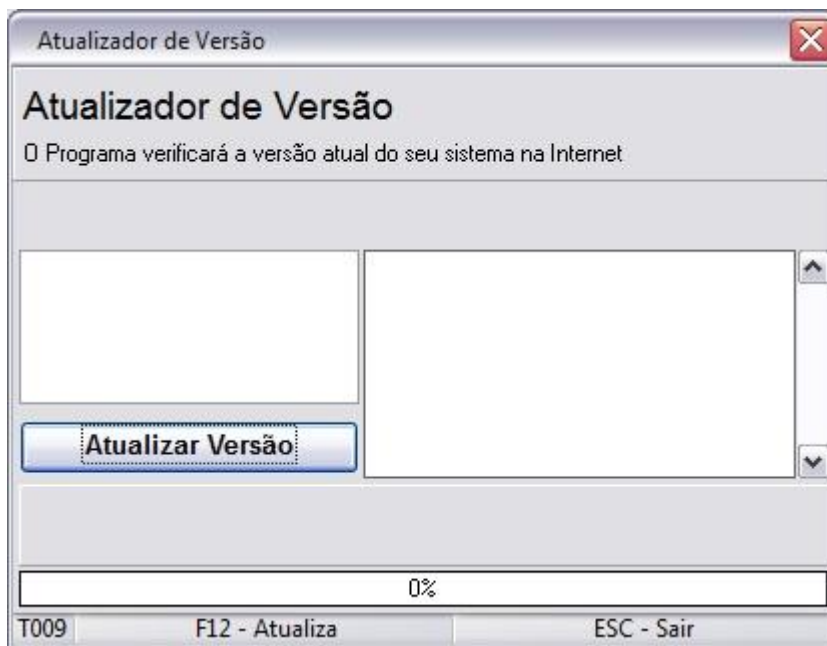


Figura 15 – Tela de atualização de versão do SGPET.
FONTE: elaborado pelo autor (2015)

Durante o período em que foram executadas as etapas do processo de desenvolvimento do *software* para SGPET, observou-se a importância da interação com o cliente para a avaliação e definição dos requisitos. Esse estreitamento de relação contribuiu para um desenvolvimento mais preciso, resultando em um *software* simples e sem módulos e campos desnecessários. Em outras palavras, a funcionalidade do aplicativo ficou bem adequada às necessidades da empresa.

Percebeu-se que, após a implantação do sistema, houve uma melhoria no controle e gestão do *pet shop*, principalmente no que se refere a organização e metodologia de trabalho. O agendamento de serviços contribuiu diretamente para essa melhoria otimizando o tempo e melhorando os resultados.

Com as informações obtidas através do sistema, observou-se um aperfeiçoamento no controle de estoque possibilitando uma análise mais precisa das demandas, que por sua vez proporcionou uma rápida tomada de decisão para reposição de mercadorias e serviços. Havia, anteriormente, perda de produtos ocasionados pela dificuldade de controle de vencimento, visto que alguns medicamentos na área veterinária, como vacinas, tem prazo curto de validade e quando vencidos, gera um ônus para a empresa, problema que foi sanado com o módulo de controle de estoque e vencimento de produtos.

Uma das questões relatadas pelos empresários era a dificuldade de identificação e informação sobre os *pets*. Essa questão também foi solucionada com o módulo de cadastro de animais que, além de relacionar o *pet* ao seu tutor, disponibilizam informações como fotos, raça, cor, sexo, temperamento, assim como, a data da última vacina, se o animal faz uso de algum medicamento contínuo ou tem algum problema de saúde relevante. Essas informações ficam disponíveis aos usuários, a qualquer momento, facilitando o atendimento.

Durante os primeiros dias após a implantação do sistema, foi relatada a dificuldade de adaptação dos usuários, menos experientes com tecnologias, ao aplicativo, situação prevista e esperada, mas superada gradativamente por intermédio de suporte/treinamento e pela prática da utilização da ferramenta no cotidiano.

Até o presente momento os resultados obtidos estão dentro da expectativa dos usuários. Com saldo final positivo, o *software* foi avaliado com um desempenho satisfatório alcançando sucesso no objetivo proposto pelo projeto.

O módulo que corresponde aos controles e gestão de clínica veterinária também não foi finalizado. O fato de não conseguirmos uma parceria com uma clínica veterinária, aqui no litoral, que dispunha de um computador para a implementação e implantação dos módulos correspondentes tornou inviável a construção e desenvolvimento dessa etapa.

5 CONSIDERAÇÕES FINAIS

A formação de Bacharel em Informática e Cidadania na UFPR - litoral vai muito além do domínio da técnica. Graças à sua metodologia interdisciplinar, o profissional tem uma melhor percepção da sua influência na sociedade. Este ambiente, de interações pedagógicas, foi fundamental para a construção deste trabalho, pois tivemos contato com diversas áreas e com outros cursos, o que enriqueceu nossa formação.

Com o conhecimento adquirido nos três eixos curriculares (FTP, ICH e PA) aprendemos a interagir e a dialogar com a comunidade. Essa percepção é de grande valia para o profissional da área de informática, pois é preciso saber analisar e identificar as verdadeiras necessidades do seu cliente, para que a informática venha ajudar a solucionar os problemas do dia a dia, e não a se tornar mais um.

O *software* para *pet shop* e clínica veterinária, resultado deste trabalho, está instalado em um *pet shop* na cidade de Matinhos – PR, e está em fase de testes.

O aplicativo é de fácil uso e atende de forma satisfatória ao que foi proposto. Algumas ferramentas não ficaram prontas no prazo esperado, mas serão implementadas futuramente.

6 PERSPECTIVAS FUTURAS

Embora o SGPET esteja implantado e sendo utilizado em um Pet Shop na cidade de Matinhos, litoral do Paraná, planeja-se trabalhar no intuito de aprimorar este software para que sejam agregados novos módulos e funções a fim de atender, de forma mais completa, às necessidades e demandas que o mercado de trabalho exige, a cada dia, com seu crescimento e expansão.

Os próximos passos para novas análises e implementações será o módulo de clínica veterinária, onde se deseja construir módulos mais específicos para o atendimento de animais, bem como o controle de medicamentos e um módulo de sugestão de diagnóstico baseado em dados de observação de sintomas nos animais.

Pretende-se no período em curto-médio prazo, com o aprimoramento e inovação desta ferramenta, comercializá-la atendendo a *pet shops* e clínicas veterinárias da região litorânea assim como demais regiões e cidades do Paraná, abrangendo até mesmo outros estados do Brasil, conforme demanda e crescimento desse mercado varejista.

7 REFERÊNCIAS

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

CAMPOS, Vicente Falconi. Controle da Qualidade Total (no estilo Japonês). Belo Horizonte: Fundação Christiano Ottoni, 1992.

LAUDON, Kenneth C. Sistemas de Informação, Rio de Janeiro: LTC – Livros Técnicos e Científicos S.A., 1999.

LIMA, Renata de Almeida. Como a relação entre clientes e fornecedores internos a organização pode contribuir para a garantia da qualidade: o caso de uma empresa automobilística. Ouro Preto: UFOP, 2006.

MARÇULA, Marcelo. Informática conceitos e Aplicações, 3ª Edição, São Paulo: Érica, 2008.

MELO, Ana C. Desenvolvendo aplicações com UML 2.2: do conceito à implementação, 3ª Edição, Rio de Janeiro: Brasport, 2010.

MORIMOTO, Carlos E. Hardware, o Guia Definitivo, São Paulo: GDH Press e Sul Editores, 2007.

PRESSMAN, Roger S. Engenharia de Software, 3ª Edição, São Paulo: McGraw Hill, 1995.

RANGEL, Alexandre. Interbase 7 – Desenvolvendo e administrando banco de dados. Rio de Janeiro: Alta Books, 2003.

SILVA, Edna L.; **MENEZES**, Estera M. Metodologia da Pesquisa e Elaboração de Dissertações, 3ª Edição, Florianópolis: Laboratório de Ensino a Distância da UFSC, 2005.

SILVA, Jane Azevedo da. Apostila de Controle da Qualidade I. Juiz de Fora: UFJF, 2006.

TANENBAUM, Andrew S. Sistemas Operacionais Modernos. 3ª Edição. São Paulo: Pearson Education do Brasil, 2009.

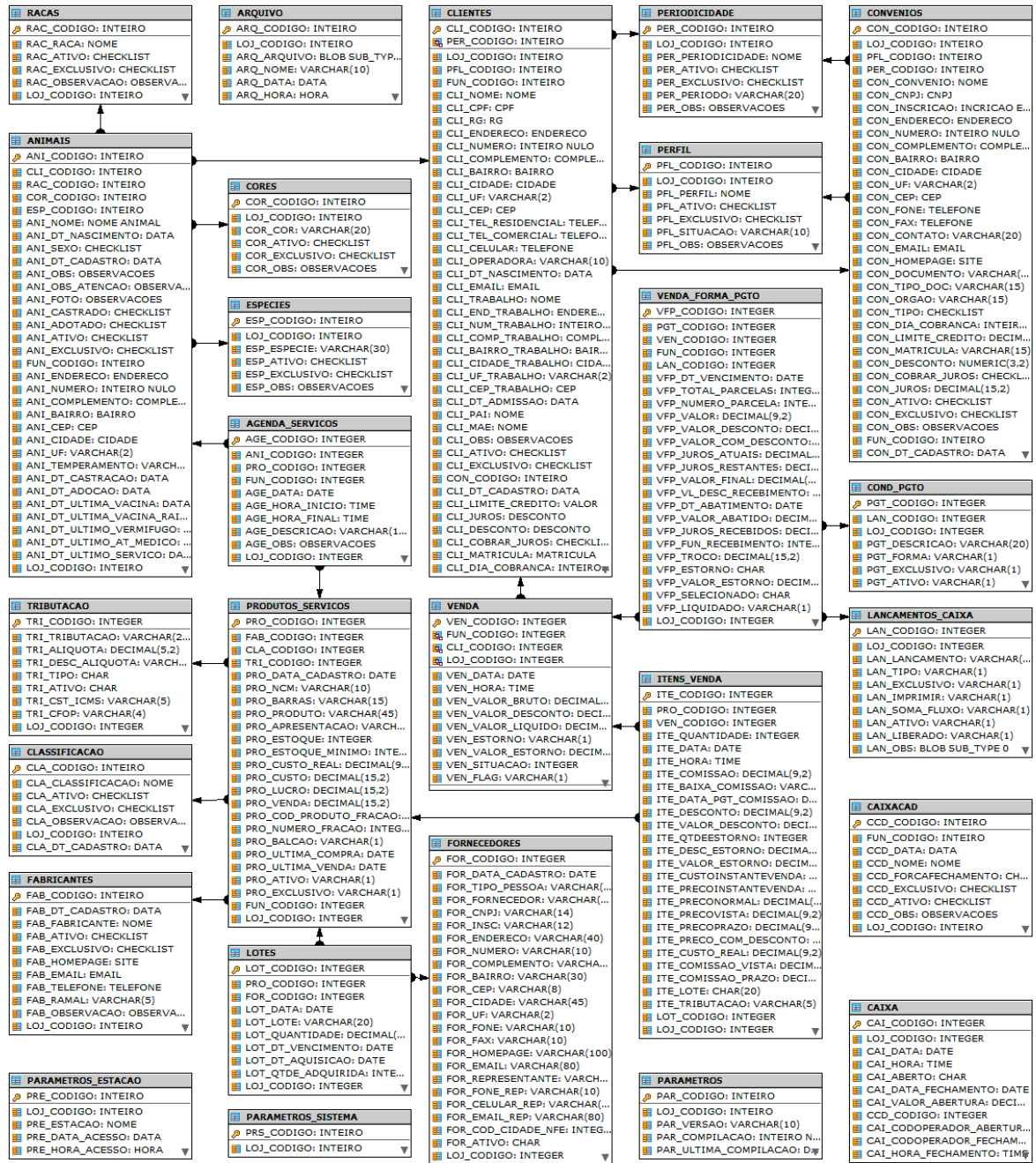
WAZLAWICK, Raul S. Análise e Projeto de Sistemas de Informação Orientados a Objetos, 2ª Edição, Rio de Janeiro: Elsevier, 2011.

WERKEMA, M. C. C. As ferramentas da qualidade no gerenciamento de processos. Belo Horizonte : Fundação Christiano Ottoni, UFMG, 1995.

VELLOSO, Fernando de C. Informática: Conceitos Básicos, 9ª Edição, Rio de Janeiro: Elsevier, 2014

ANEXOS

ANEXO 1 – FLUXOGRAMA DO SOFTWARE SGPET.



ANEXO 2 – DIAGRAMA DE ENTIDADE E RELACIONAMENTO.

