



FACULDADE DE TECNOLOGIA DE AMERICANA – MINISTRO RALPH BIASI

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

João Pedro Maciel

Matheus Cecílio Neves

Rodrigo Sampaio de Santana

GPet

Sistema de Gestão de Petshops e Clínicas Veterinárias

Americana, SP

2020



FACULDADE DE TECNOLOGIA DE AMERICANA – MINISTRO RALPH BIASI**Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**

João Pedro Maciel

Matheus Cecílio Neves

Rodrigo Sampaio de Santana

GPet

Sistema de Gestão de Petshops e Clínicas Veterinárias

Trabalho acadêmico apresentado a disciplina de Laboratório de Engenharia de Software, sob a orientação metodológica do Prof. Dr. Ivan Menerval da Silva.

Área de concentração: Engenharia de Software, Desenvolvimento de Sistemas.

Americana, SP

2020

RESUMO

O presente trabalho acadêmico tem como objetivo principal executar as demandas requisitadas durante o decorrer da disciplina de Laboratório de Engenharia de Software ministrada pelo Prof. Dr. Ivan Menerval da Silva, onde houve a solicitação de desenvolvimento de um projeto para realizar a gestão de Petshops e Clinicas Veterinárias.

Palavras Chave: Software ; Petshop ; Desenvolvimento;

ABSTRACT

The present academic work has a main objective as supply a demand created during the Software Engineer Lab classes taught by Professor Ivan Menerval da Silva, whom requested a development of a project system to manage Pet Shops and Veterinary clinics.

Keywords: Software; Pet Shop; Development;

LISTA DE ILUSTRAÇÕES

Figura 1 – Metodologia XP	10
Figura 2 - Diagrama de Casos de Uso(Projeto).....	17
Figura 3 - Diagrama de Casos de Uso(Desenvolvimento)	18
Figura 4 – Diagrama de Classes do Sistema	19
Figura 5 – Diagrama de Entidade Relacionamento	20
Figura 6 – Modelo Entidade Relacionamento.....	21
Figura 7 – Tabela de Animais.....	22
Figura 8 – Tabela de Atendimentos.....	22
Figura 9 – Tabela de Categorias	23
Figura 10 – Tabela de Clientes.....	23
Figura 11 – Tabela Clientes X Animais	23
Figura 12 – Tabela de Compras	24
Figura 13 – Tabela de Fornecedores	24
Figura 14 – Tabela de Horas X Salários.....	25
Figura 15 – Tabela de Inventário.....	25
Figura 16 – Tabela de Log de Transações	26
Figura 17 – Tabela de Produtos	26
Figura 18 – Tabela de Representante de Vendas	27
Figura 19 – Tabela de Serviços	27
Figura 20 – Tabela de Serviços X Atendimentos.....	28
Figura 21 – Tabela de Serviços X Atendimentos.....	28
Figura 22 – Tabela de Vendas	29
Figura 23 – Tabela de Vendas	29
Figura 24 – Trigger After Insert Atendimento.....	30
Figura 25 – Trigger After Update Atendimento	30
Figura 26 – Trigger After Insert Categoria	30
Figura 27 – Trigger After Insert Compra	31
Figura 28 – Trigger After Insert Produto	31
Figura 29 – Trigger After Update Produto	31
Figura 30 – Trigger After Insert Serviços X Atendimentos	32
Figura 31 – Trigger After Delete Serviços X Atendimentos	32
Figura 32 – Trigger After Insert Venda	32
Figura 33 – Trigger After Update Venda.....	33
Figura 34 – Trigger After Insert Vendas X Produtos	33
Figura 35 – Trigger After Delete Vendas X Produtos	33
Figura 36 – CRUD	34
Figura 37 – Exemplo Create.....	34
Figura 38 – Exemplo Read	35
Figura 39 – Exemplo Update	35
Figura 40 – Exemplo Delete	35
Figura 41 – Layout do Sistema	37
Figura 42 – Logotipo do Sistema	37
Figura 43 – Cabeçalho do Sistema	38
Figura 44 – Barra de Orientação	38
Figura 45 – Menu Lateral.....	38
Figura 46 – Área de conteúdo	39

Figura 47 – Informações do Sistema e Footer.....	39
Figura 48 – Tela de Cadastro	40
Figura 49 – Tela de Visualização	41
Figura 50 – Tela de Alteração	42
Figura 51 – Tela de Remoção	43
Figura 52 – Tela de Finalização	44
Figura 53 – Tela de Login.....	45
Figura 54 – Tela de Inicial	46
Figura 55 – Gerenciamento de Funcionários.....	47
Figura 56 – Gerenciamento de Horas Trabalhadas.....	48
Figura 57 – Holerite	49
Figura 58 – Gerenciamento de Atendimentos	50
Figura 59 – Gerenciamento Serviços Prestados	51
Figura 60 – Histórico de Atendimentos.....	52
Figura 61 – Gerenciamento de Clientes	53
Figura 62 – Gerenciamento de Animais	54
Figura 63 – Gerenciamento de Compras.....	55
Figura 64 – Histórico de Compras	56
Figura 65 – Histórico de Transações.....	57
Figura 66 – Gerenciamento de Fornecedores	58
Figura 67 – Gerenciamento de Representantes.....	59
Figura 68 – Gerenciamento de Categorias	60
Figura 69 – Gerenciamento de Representantes.....	61
Figura 70 – Estoque	62
Figura 71 – Gerenciamento de Serviços	63
Figura 72 – Gerenciamento de Vendas.....	64
Figura 73 – Gerenciamento Carrinho	65
Figura 74 – Histórico de Transações.....	66

LISTA DE TABELAS

Tabela 1 – Requisitos funcionais do projeto.....	11
Tabela 2 – Requisitos não funcionais do projeto.....	12

SUMÁRIO

INTRODUÇÃO	9
2. PROJETO DO SISTEMA	10
2.1 Metodologia Empregada	10
2.2 Levantamento de Requisitos	11
2.2.1 Requisitos Funcionais.....	11
2.2.2 Levantamento de Requisitos	12
2.3 Análise de Mercado.....	12
2.4 Recursos e Ferramentas.....	13
3. MODELAGEM.....	16
3.1 Diagrama de Casos de Uso	16
3.2 Diagrama de Classes	19
3.3 Diagrama de Entidade Relacionamento.....	20
3.4 Modelo Entidade Relacionamento.....	21
4. BANCOS DE DADOS	22
4.1 Tabelas	22
4.1.1 Tabela Animais	22
4.1.2 Tabela Atendimentos	22
4.1.3 Tabela Categorias	23
4.1.4 Tabela Clientes.....	23
4.1.5 Tabela relação Clientes X Animais	23
4.1.6 Tabela Compras	24
4.1.7 Tabela Fornecedor	24
4.1.8 Tabela Horas X Salário.....	25
4.1.9 Tabela Inventário	25
4.1.10 Tabela Log de Transações.....	26
4.1.11 Tabela Produtos	26

4.1.12	Tabela Representantes de Vendas	27
4.1.13	Tabela Serviços.....	27
4.1.14	Tabela Serviços X Atendimento	28
4.1.15	Tabela Usuários	28
4.1.16	Tabela Vendas	29
4.1.17	Tabela Vendas X Produtos.....	29
4.2	Triggers	29
4.2.1	Trigger inserção Atendimento.....	30
4.2.2	Trigger alteração Atendimento.....	30
4.2.3	Trigger inserção Categoria	30
4.2.4	Trigger inserção Compra	31
4.2.5	Trigger inserção Produto	31
4.2.6	Trigger alteração Produto	31
4.2.7	Trigger inserção Serviços X Atendimento.....	32
4.2.8	Trigger remoção Serviços X Atendimento	32
4.2.9	Trigger inserção Venda	32
4.2.10	Trigger alteração Venda	33
4.2.11	Trigger inserção Venda X Produto	33
4.2.12	Trigger remoção Venda X Produto	33
5.	O SISTEMA.....	34
5.1	Layout do Genérico Sistema	37
5.1.1	Layout Tela de Cadastro.....	40
5.1.2	Layout Tela de Visualização	41
5.1.3	Layout Tela de Alteração	42
5.1.4	Layout Tela de Remoção.....	43
5.1.5	Layout Tela de Finalização.....	44
5.2	Características da Tela de Login.....	45

5.3	Características da Tela Principal.....	46
5.4	Características da Tela Gerenciamento de Funcionários.....	47
5.5	Características da Tela Gerenciamento de Horas Trabalhadas.....	48
5.6	Características da Tela Gerenciamento de Atendimentos	50
5.7	Características da Tela de Histórico de Atendimentos.....	52
5.8	Características da Tela Gerenciamento de Clientes	53
5.9	Características da Tela Gerenciamento de Animais	54
5.10	Características da Tela Gerenciamento de Compras	55
5.11	Características da Tela Histórico de Compras.....	56
5.12	Características da Tela Histórico de Transações.....	57
5.13	Características da Tela Gerenciamento de Fornecedores.....	58
5.14	Características da Tela Gerenciamento de Representantes.....	59
5.15	Características da Tela Gerenciamento de Categorias.....	60
5.16	Características da Tela Gerenciamento de Produtos	61
5.17	Características da Tela de Estoque	62
5.18	Características da Tela Gerenciamento de Serviços	63
5.19	Características da Tela Gerenciamento de Vendas.....	64
5.20	Características da Tela Histórico de Vendas	66
6.	CONSIDERAÇÕES FINAIS	67
	REFERÊNCIAS	68

INTRODUÇÃO

O objetivo geral do GPet é fazer o gerenciamento de Petshops, possuindo as seguintes funcionalidades: cadastro de clientes, animais e fornecedores, fluxo de caixa, histórico de entradas e saídas, serviços prestados, e controle de estoque.

A motivação para o desenvolvimento do sistema foi gerada pelo projeto da disciplina de Laboratório de Engenharia de Software, ministrada pelo Prof. Dr. Ivan Menerval da Silva, que tinha como necessidade a criação de um sistema para Petshops.

Este projeto foi dividido em quatro capítulos:

O **primeiro** conceitua a criação do sistema e a metodologia tomada como base, além de exibir o levantamento de requisitos e os recursos usados.

O **segundo** exibe as idealizações do projeto em formas de Diagramas UML (*Unified Model Language*).

O **terceiro** capítulo exibe a estrutura do banco de dados, juntamente com as automações de banco.

O **quarto** exprime as funcionalidade do sistema e suas características específicas. Por fim as Considerações Finais do grupo em relação ao projeto.

2. PROJETO DO SISTEMA

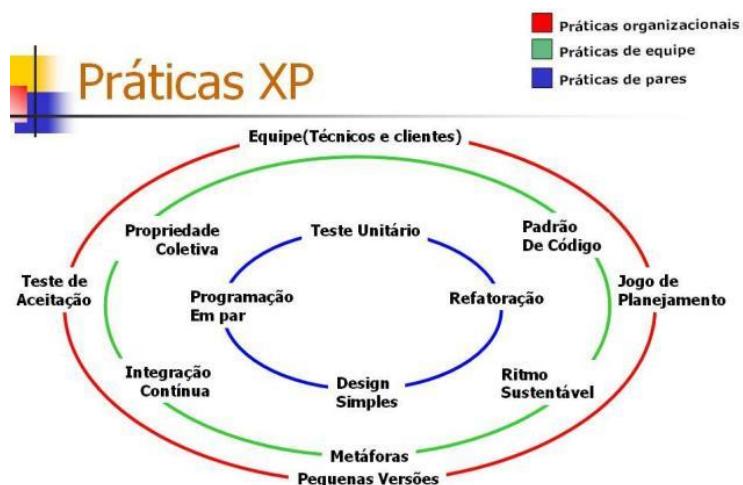
O Presente projeto foi desenvolvido no decorrer das aulas de Laboratório de Engenharia de Software, ministradas pelo Prof. Dr. Ivan Menerval da Silva, que concedeu aos alunos como proposta da disciplina o desenvolvimento de um projeto de Sistema de Gestão de Petshops e Clínicas Veterinárias, com o enfoque principal em executar princípios básicos da Engenharia de Software como, Levantamento de Requisitos, Desenvolvimento de Diagramas e Desenvolvimento de Software.

2.1 Metodologia Empregada

O conceito utilizado para o desenvolvimento do sistema baseia-se nas metodologias ágeis, que na prática resultam em um projeto que destina-se a melhorias contínuas a fim de proporcionar melhores resultados. A ramificação mais adequada das metodologias ágeis para descrever o desenvolvimento do presente projeto é a XP¹ (*Extreme Programming*), onde é possível concluir o projeto de forma objetiva, rápida e eficiente.

A XP é uma boa escolha metodológica quando nos deparamos com projetos que podem sofrer constantes mudanças de requisitos, projetos que não são totalmente claros quanto ao seu fim ou ainda para uma maior interação dos envolvidos com o projeto, além é claro do tempo empregado.

Figura 1 – Metodologia XP



Fonte: Kent Beck(1999).

¹ *Extreme Programming* ou Programação Extrema em tradução literal, é o nome dado ao processo de desenvolvimento com foco em agilidade e qualidade de projeto.

Na prática todo o desenvolvimento do sistema ocorreu de forma rápida, simples e eficaz onde foi possível obter resultados mais expressivos a partir da padronização de código, testes de desempenho e implementação de melhorias contínuas afim de que o projeto solucione a necessidade reportada durante a aula, gerenciar um Petshop e ou Clínica Veterinária.

2.2 Levantamento de Requisitos

A engenharia de requisitos (RE – *Requirements Engineering*) é o processo de descobrir, analisar, documentar e verificar requisitos de um sistema. Um requisito pode ser definido como uma descrição dos serviços fornecidos pelo sistema e as suas restrições operacionais (SOMMERVILLE, 2007). Tradicionalmente, os requisitos são divididos em dois tipos: requisitos funcionais e requisitos não funcionais.

2.2.1 Requisitos Funcionais

Os requisitos funcionais descrevem o que o sistema deve fazer, isto é, definem a funcionalidade desejada do software (SOMMERVILLE, 2007). A Tabela 1 apresenta os requisitos funcionais deste projeto.

Tabela 1 – Requisitos funcionais do projeto.

Identificação	Requisito Funcional	Prioridade
RF001	Manter Fornecedores	Essencial
RF002	Manter Clientes	Essencial
RF003	Manter Funcionários	Essencial
RF004	Manter Produtos	Essencial
RF005	Manter Serviços	Essencial
RF006	Manter Animais	Essencial
RF007	Manter Movimentações	Essencial

Fonte: Autoral (2020).

2.2.2 Levantamento de Requisitos

“Os requisitos não funcionais são aqueles não diretamente relacionados às funções específicas fornecidas pelo sistema” (SOMMERVILLE, 2007). A Tabela 2 apresenta os requisitos não funcionais deste projeto.

Tabela 2 – Requisitos não funcionais do projeto.

Identificação	Requisito não funcional	Categoria	Prioridade
RNF001	Interface Intuitiva	Formatação	Essencial
RNF002	Exibir nome do dono junto ao cadastro do animal	Formatação	Importante
RNF003	Exibir especificações de uso dos produtos(Descrição)	Formatação	Importante
RNF004	Exibir históricos de compras, vendas e serviços prestados	Formatação	Importante
RNF005	Ser integral aos dados cadastrados	Desempenho	Importante
RNF006	Exibir carga horária dos funcionários	Formatação	Desejável
RNF007	Exibir data dos registros	Segurança	Importante

Fonte: Autoral (2020).

2.3 Análise de Mercado

Durante as aulas foi proposta uma análise de mercado para que fosse possível identificar as características necessárias para que o software execute a função proposta. Abaixo estão alguns dos softwares encontrados:

Nuvem Gestor – É um software que faz o gerenciamento de estoque, vendas e materiais usados no cuidado de cada espécie e raça. 100% em nuvem totalmente integrado a sua loja ou clínica veterinária algumas de suas funcionalidades são:

- Pedidos e Atendimentos
- Controle de estoque
- Emissor de NF-e e NFC-e
- Cadastro de Clientes, produtos e serviços
- Suporte gratuito

- Sem taxa de implantação
- Sem contrato de fidelidade
- Site gratuito

Petshop Control – O Petshop Control foi construído a partir dos cinco princípios de gestão de um negócio pet: Financeiro, Estoque, Relacionamento, Vendas e Administrativo algumas de suas funcionalidades são:

- Pedidos e Atendimentos
- Controle de estoque
- Emissor de NF-e e NFC-e
- Cadastro de Clientes, produtos e serviços
- Suporte gratuito
- Sem taxa de implantação
- Sem contrato de fidelidade
- Site gratuito

Vetus – O Vetus é uma plataforma de Gestão para Petshop, Clínica Veterinária e Hospital Veterinário algumas de suas funcionalidades são:

- Pedidos e Atendimentos
- Controle de estoque
- Emissor de NF-e e NFC-e
- Cadastro de Clientes, produtos e serviços
- Suporte gratuito
- Sem taxa de implantação
- Sem contrato de fidelidade
- Site gratuito

2.4 Recursos e Ferramentas

Esta seção contempla as ferramentas de programação e os conceitos necessários para o desenvolvimento do sistema:

Adobe PhotoShop CS6: Adobe Photoshop é um software caracterizado como editor de imagens bidimensionais do tipo *raster* (possuindo ainda algumas capacidades de edição típicas dos editores vectoriais) desenvolvido pela Adobe Systems. (Adobe Photoshop C S6, 2018).

Astah UML: É uma ferramenta que suporta os requerimentos da UML 2.x para construção de diagramas de classe, caso de uso, sequência, máquina de estado, atividade, componente, dentre outros (ASTAH, 2018).

Bootstrap: É o mais popular *framework*² CSS utilizado para auxiliar no desenvolvimento de sites responsivos e/ou mobile (W3SCHOOLS, 2020).

CSS: O *Cascading Style Sheets* (CSS) é uma "folha de estilo" composta por "camadas" e utilizada para definir a apresentação (aparência) em páginas da internet que adotam para o seu desenvolvimento linguagens de marcação (como XML, HTML e XHTML). O CSS define como serão exibidos os elementos contidos no código de uma página da internet e sua maior vantagem é efetuar a separação entre o formato e o conteúdo de um documento. (TECMUNDO, 2009).

HeidiSQL: Permite que você veja e edite dados e estruturas de computadores que executam um dos sistemas de banco de dados MariaDB, MySQL, Microsoft SQL³ (*Structured Query Language*) ou PostgreSQL, facilitando a estruturação de banco de dados para aplicação em softwares, aplicativos, aplicações web (HEIDSQL, 2017).

HTML: HTML é uma das linguagens que utilizamos para desenvolver websites. O acrônimo HTML vem do inglês e significa *Hypertext Markup Language* ou em português Linguagem de Marcação de Hipertexto. O HTML é a linguagem base da internet. Foi criada para ser de fácil entendimento por seres humanos e também por máquinas, como por exemplo o Google ou outros sistemas que percorrem a internet capturando informação (TABLELESS, 2011).

JavaScript: é uma linguagem leve, interpretada e baseada em objetos com funções de primeira classe, mais conhecida como a linguagem de script para páginas Web, mas usada também em vários outros ambientes sem *browser*, tais como node.js, Apache CouchDB e Adobe Acrobat (MDN WEB DOCS, 2019).

² Framework - é um template com diversas funções que podem ser usadas pelo desenvolvedor e ocasiona em ganho de tempo de desenvolvimento.

³ SQL - *Structured Query Language* - ou Linguagem de Consulta Estruturada, é a linguagem de pesquisa declarativa padrão para banco de dados relacional (base de dados relacional).

MySQL: O MySQL é um sistema de gerenciamento de banco de dados relacional (RDBMS) de software livre suportado pela Oracle, baseado em SQL (*Structured Query Language*). O MySQL é executado em praticamente todas as plataformas, incluindo Linux , UNIX e Windows .

PHP: é uma linguagem de script *open source* de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML (PHP, 2020).

Visual Studio Code: É um editor de código-fonte leve, mas poderoso, que roda na sua área de trabalho e está disponível para Windows, macOS e Linux. Ele vem com suporte interno para JavaScript, TypeScript e Node.js e possui um rico ecossistema de extensões para outras linguagens (como C ++, C #, Java, Python, PHP, Go) e tempos de execução (como .NET e Unity) (Visual Studio Code, 2020).

3. MODELAGEM

Na fase da modelagem é feita a documentação do aplicativo, se trata de diagramas que facilitam na compreensão do projeto de forma padronizada.

A documentação deste trabalho utilizará a linguagem de modelagem (UML - *Unified Modeling Language*⁴ para modelar os casos de uso e o diagrama de classe.

3.1 Diagrama de Casos de Uso

Os diagramas de caso de uso descrevem um cenário de funcionalidades do ponto de vista do usuário, catalogando os requisitos funcionais do sistema. Dentro do Diagrama são retratados os atores (representado pelos bonecos), as funcionalidades (representadas pelos balões com a ação escrita por dentro) e as relações (representadas pelas linhas).

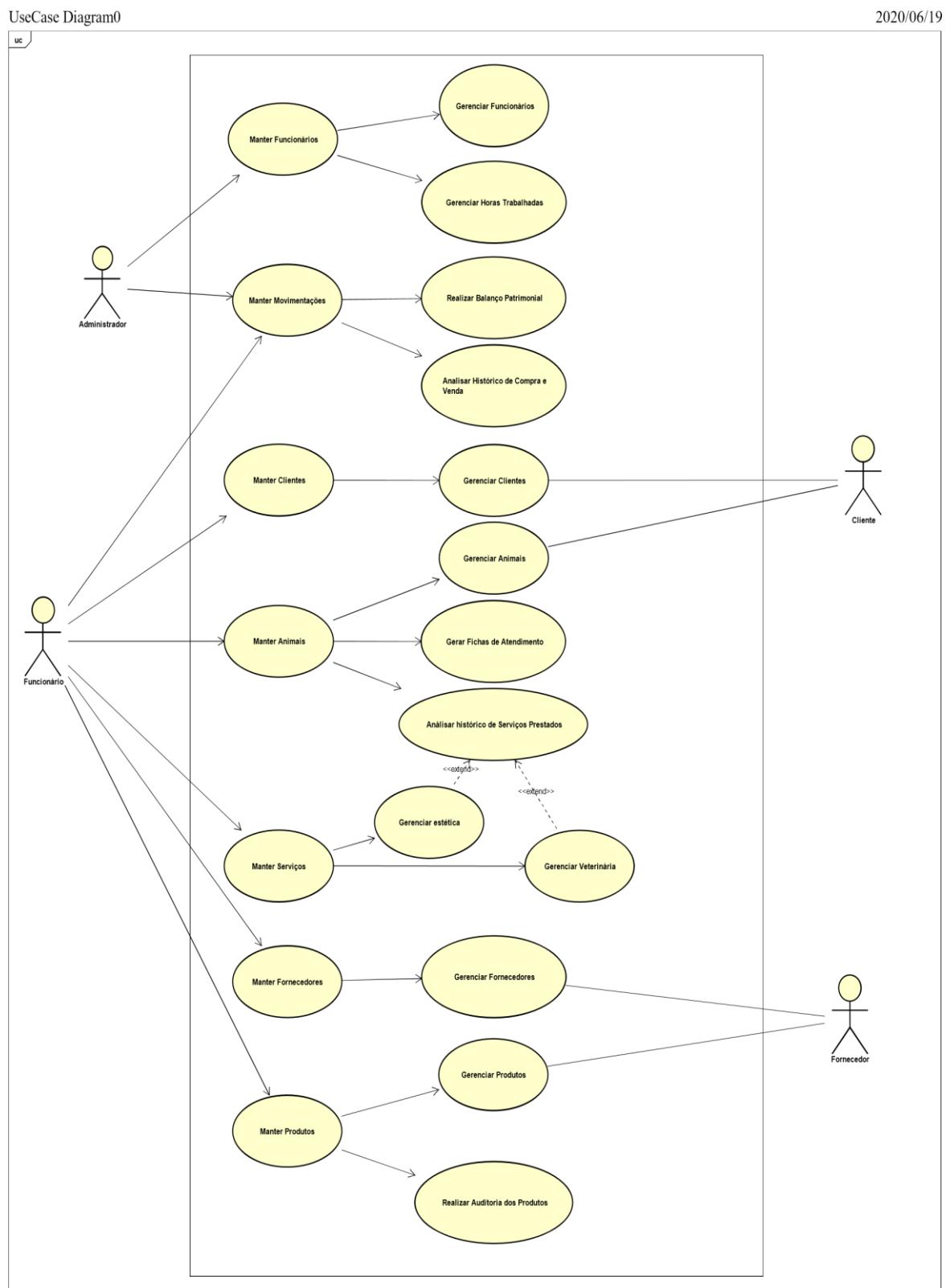
Os atores que interagem diretamente com o sistema são: o(a) Administrador(a) e os(as) Funcionários(as).

- **Administrador(a)** é o ator que representa usuário de maior hierarquia do sistema, onde somente este usuário possui permissão para realizar o cadastro de novos funcionários, editar, excluir ou atualizar os mesmos.
- **Funcionário(a)** representa o ator que realiza as funções não administrativas do sistema , ou seja, é a aquele que utiliza o sistema diariamente realizando atendimentos a clientes e animais, gerenciando produtos, fornecedores e representantes de vendas entre outros.

Em especial para o Diagrama de casos de uso do GPet foram desenvolvidas duas versões do diagrama de casos de uso, onde é possível ver a diferença prática entre o projeto e o desenvolvimento. Sendo assim a Figura 1 apresenta os casos de uso do projeto do sistema, enquanto a Figura 2 exibe os casos de usos pós conclusão do desenvolvimento.

⁴*Unified Modeling Language* ou Linguagem Unificada de Modelagem (UML) é uma linguagem padrão para modelagem e documentar os sistemas orientados a objetos.

Figura 2 - Diagrama de Casos de Uso(Projeto)

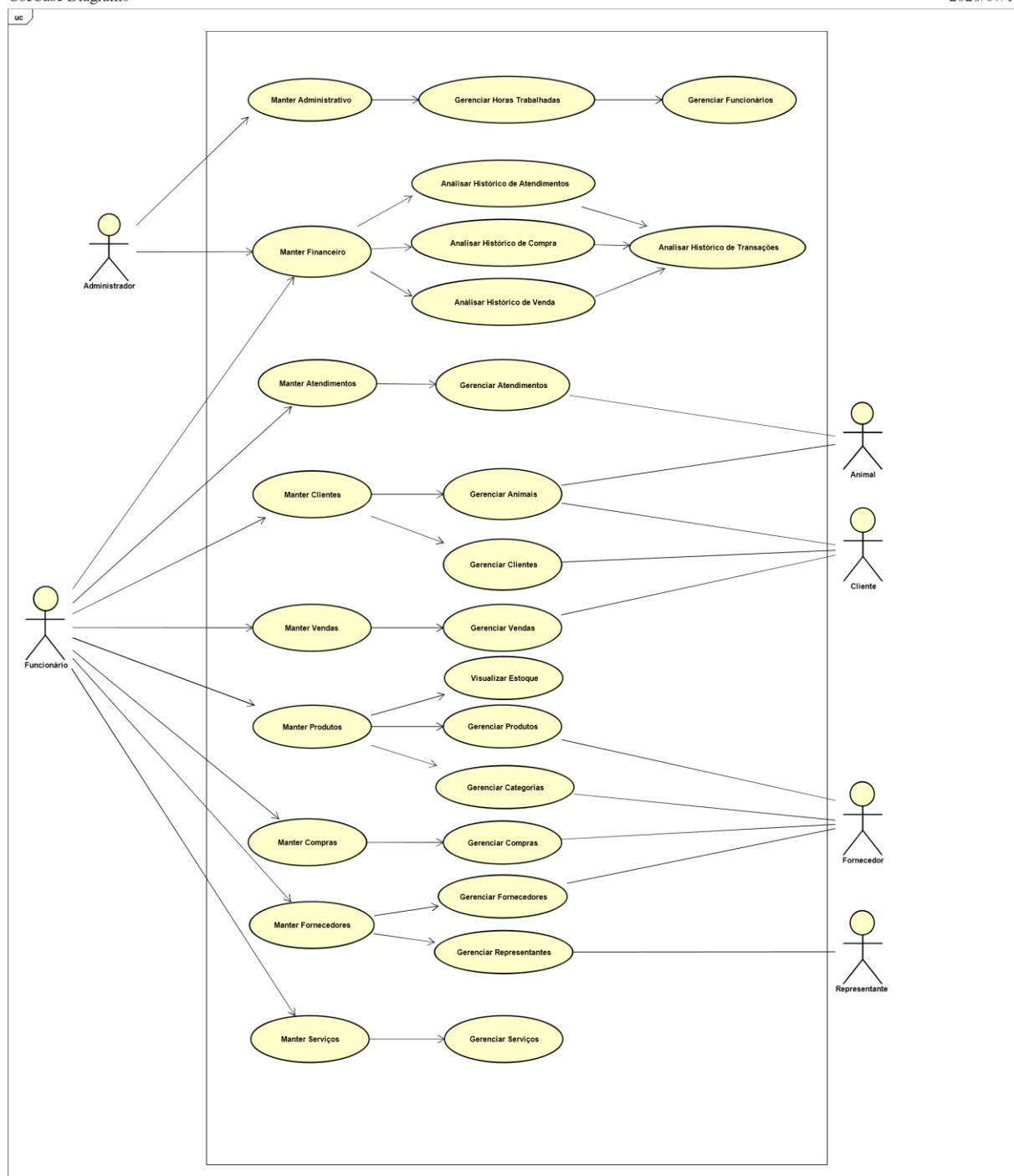


Fonte: Autoral (2020).

Figura 3 - Diagrama de Casos de Uso(Desenvolvimento)

UseCase Diagram0

2020/07/13

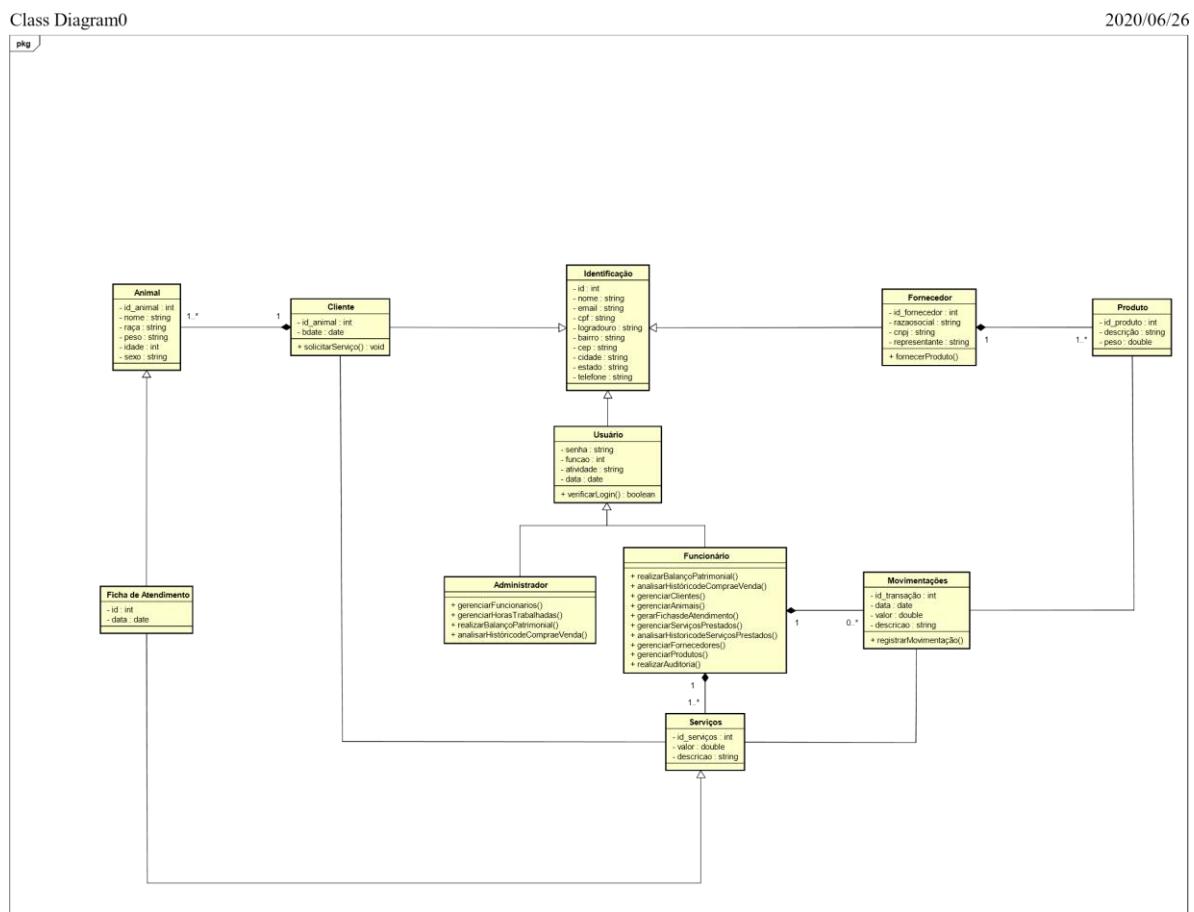


Fonte: Autoral (2020).

3.2 Diagrama de Classes

O diagrama de classe é responsável por oferecer uma representação da estrutura e relações das classes como também as operações solicitadas pelos atores que servem de modelo para os objetos. A Figura 4 representa o Diagrama de Classes do Sistema.

Figura 4 – Diagrama de Classes do Sistema

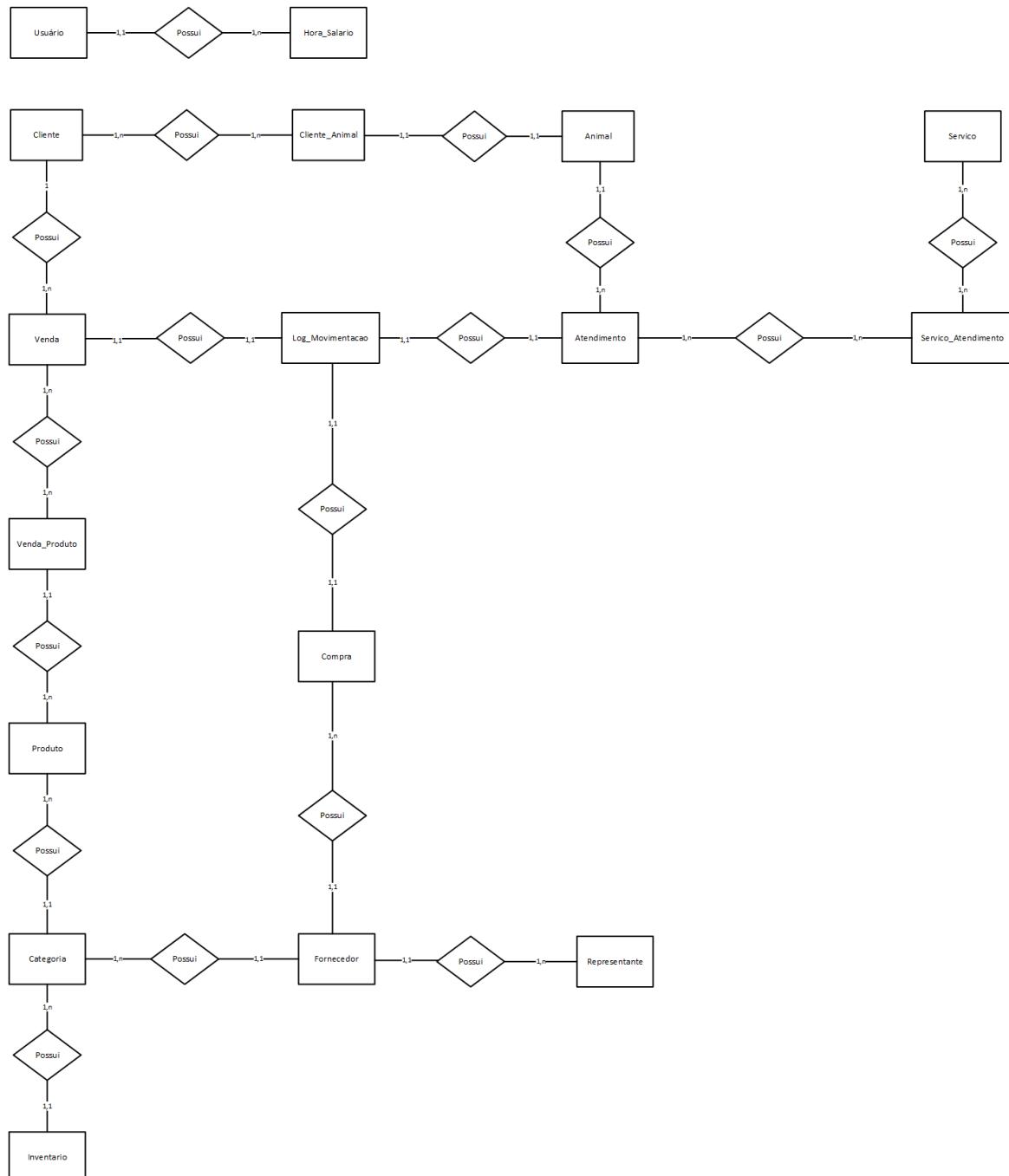


Fonte: Autoral (2020).

3.3 Diagrama de Entidade Relacionamento

O Diagrama entidade relacionamento é a representação gráfica do banco de dados sendo assim exibe as tabelas(entidades) e suas relações(chaves-estrangeiras). A Figura 5 exibe o DER.

Figura 5 – Diagrama de Entidade Relacionamento

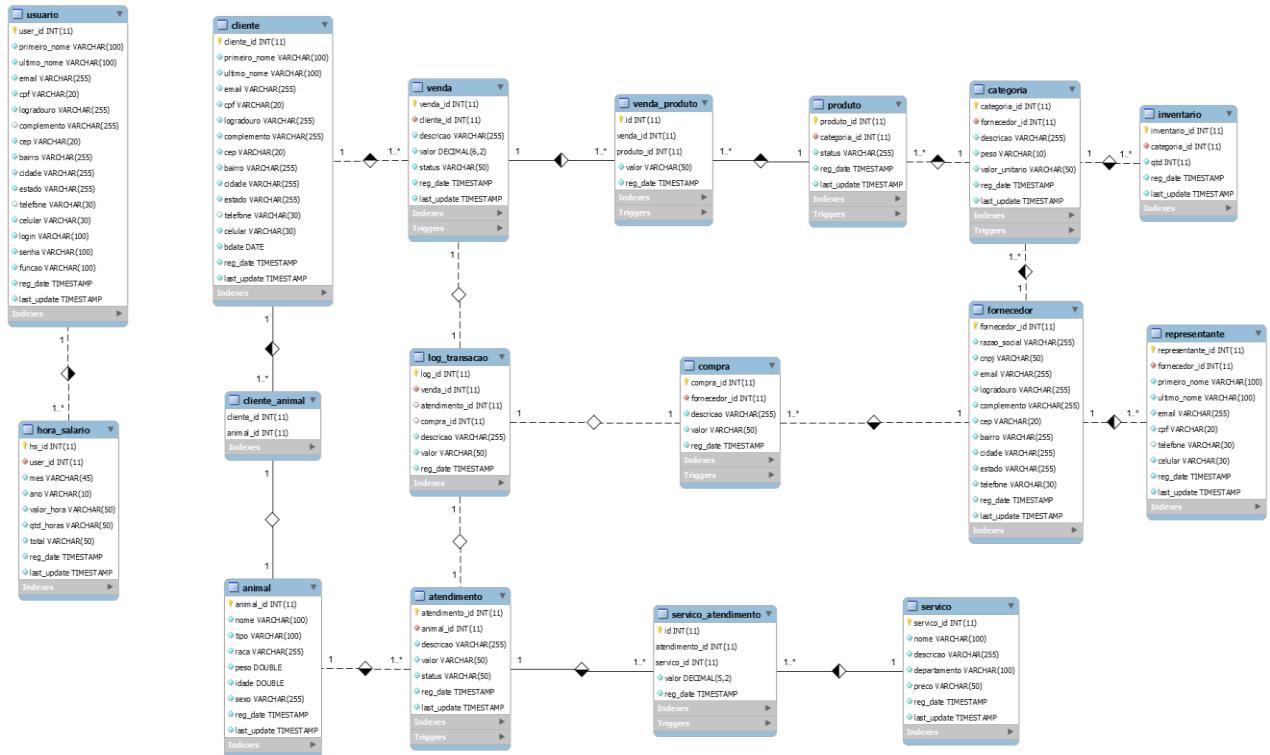


Fonte: Autoral (2020).

3.4 Modelo Entidade Relacionamento

O Modelo Entidade Relacionamento descreve as tabelas do banco de dados, com suas características (atributos) e como elas se relacionam entre si (relacionamentos). A Figura 6 exibe o MER do sistema.

Figura 6 – Modelo Entidade Relacionamento



Fonte: Autoral (2020).

4. BANCOS DE DADOS

O banco de dados possui todas as informações pertinentes ao sistema, ou seja, possui os dados de clientes, animais, entre outros.

4.1 Tabelas

Abaixo estão os códigos de criação das tabelas utilizadas pelo sistema para possibilitar a execução das ações de gestão do Petshop e ou Clínica Veterinária.

4.1.1 Tabela Animais

Possui os dados referentes aos animais.

Figura 7 – Tabela de Animais

```
CREATE TABLE IF NOT EXISTS animal (
    animal_id int(11) NOT NULL AUTO_INCREMENT,
    nome varchar(100) NOT NULL,
    tipo varchar(100) NOT NULL,
    raca varchar(255) NOT NULL,
    peso double NOT NULL,
    idade double NOT NULL,
    sexo varchar(255) NOT NULL,
    reg_date timestamp NOT NULL DEFAULT current_timestamp(),
    last_update timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
    PRIMARY KEY (animal_id)
)
```

Fonte: Autoral (2020).

4.1.2 Tabela Atendimentos

Possui os dados referentes aos atendimentos.

Figura 8 – Tabela de Atendimentos

```
CREATE TABLE IF NOT EXISTS atendimento (
    atendimento_id int(11) NOT NULL AUTO_INCREMENT,
    animal_id int(11) NOT NULL,
    descricao varchar(255) NOT NULL,
    valor varchar(50) NOT NULL DEFAULT '',
    status varchar(50) NOT NULL,
    reg_date timestamp NOT NULL DEFAULT current_timestamp(),
    last_update timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
    PRIMARY KEY (atendimento_id),
    KEY fk_atendimento_animal (animal_id),
    CONSTRAINT fk_atendimento_animal FOREIGN KEY (animal_id) REFERENCES animal (animal_id) ON DELETE CASCADE ON UPDATE CASCADE
)
```

Fonte: Autoral (2020).

4.1.3 Tabela Categorias

Possui os dados referentes as categorias de produto.

Figura 9 – Tabela de Categorias

```
CREATE TABLE IF NOT EXISTS categoria (
    categoria_id int(11) NOT NULL AUTO_INCREMENT,
    fornecedor_id int(11) NOT NULL,
    descricao varchar(255) NOT NULL,
    peso varchar(10) NOT NULL DEFAULT '',
    valor_unitario varchar(50) NOT NULL DEFAULT '0',
    reg_date timestamp NOT NULL DEFAULT current_timestamp(),
    last_update timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
    PRIMARY KEY (categoria_id),
    KEY fk_fornecedor_tipo (fornecedor_id),
    CONSTRAINT `fk_fornecedor_tipo` FOREIGN KEY (fornecedor_id) REFERENCES fornecedor (fornecedor_id) ON DELETE CASCADE ON UPDATE CASCADE
)
```

Fonte: Autoral (2020).

4.1.4 Tabela Clientes

Possui os dados referentes aos donos dos animais.

Figura 10 – Tabela de Clientes

```
CREATE TABLE IF NOT EXISTS cliente (
    cliente_id int(11) NOT NULL AUTO_INCREMENT,
    primeiro_nome varchar(100) NOT NULL,
    ultimo_nome varchar(100) NOT NULL,
    email varchar(255) NOT NULL,
    cpf varchar(20) NOT NULL,
    logradouro varchar(255) NOT NULL,
    complemento varchar(255) NOT NULL,
    cep varchar(20) NOT NULL,
    bairro varchar(255) NOT NULL,
    cidade varchar(255) NOT NULL,
    estado varchar(255) NOT NULL,
    telefone varchar(30) DEFAULT NULL,
    celular varchar(30) NOT NULL,
    bdate date NOT NULL,
    reg_date timestamp NOT NULL DEFAULT current_timestamp(),
    last_update timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
    PRIMARY KEY (cliente_id)
)
```

Fonte: Autoral (2020).

4.1.5 Tabela relação Clientes X Animais

Essa tabela é responsável pela relação entre clientes e seus animais.

Figura 11 – Tabela Clientes X Animais

```
CREATE TABLE IF NOT EXISTS cliente_animal (
    cliente_id int(11) NOT NULL,
    animal_id int(11) NOT NULL,
    PRIMARY KEY (cliente_id,animal_id),
    KEY fk_animal (animal_id),
    CONSTRAINT fk_animal FOREIGN KEY (animal_id) REFERENCES animal (animal_id) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT `fk_cliente` FOREIGN KEY (cliente_id) REFERENCES cliente (cliente_id) ON DELETE CASCADE ON UPDATE CASCADE
)
```

Fonte: Autoral (2020).

4.1.6 Tabela Compras

Essa tabela é responsável pelo armazenamento de dados de compras de fornecedores.

Figura 12 – Tabela de Compras

```
CREATE TABLE IF NOT EXISTS compra (
    compra_id int(11) NOT NULL AUTO_INCREMENT,
    fornecedor_id int(11) NOT NULL,
    descricao varchar(255) NOT NULL,
    valor varchar(50) NOT NULL DEFAULT '0',
    reg_date timestamp NOT NULL DEFAULT current_timestamp(),
    PRIMARY KEY (compra_id),
    KEY fk_fornecedor_compra (fornecedor_id),
    CONSTRAINT fk_fornecedor_compra FOREIGN KEY (fornecedor_id) REFERENCES fornecedor (fornecedor_id) ON DELETE CASCADE ON UPDATE CASCADE
)
```

Fonte: Autoral (2020).

4.1.7 Tabela Fornecedor

Essa tabela é responsável pelo armazenamento de dados de fornecedores de produtos.

Figura 13 – Tabela de Fornecedores

```
CREATE TABLE IF NOT EXISTS fornecedor (
    fornecedor_id int(11) NOT NULL AUTO_INCREMENT,
    razao_social varchar(255) NOT NULL,
    cnpj varchar(50) NOT NULL,
    email varchar(255) NOT NULL,
    logradouro varchar(255) NOT NULL,
    complemento varchar(255) NOT NULL,
    cep varchar(20) NOT NULL,
    bairro varchar(255) NOT NULL,
    cidade varchar(255) NOT NULL,
    estado varchar(255) NOT NULL,
    telefone varchar(30) NOT NULL,
    reg_date timestamp NOT NULL DEFAULT current_timestamp(),
    last_update timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
    PRIMARY KEY (fornecedor_id)
)
```

Fonte: Autoral (2020).

4.1.8 Tabela Horas X Salário

Essa tabela é responsável pelo armazenamento de dados de trabalho dos funcionários.

Figura 14 – Tabela de Horas X Salários

```
CREATE TABLE IF NOT EXISTS hora_salario (
    hs_id int(11) NOT NULL AUTO_INCREMENT,
    user_id int(11) NOT NULL,
    mes varchar(45) NOT NULL,
    ano varchar(10) NOT NULL,
    valor_hora varchar(50) NOT NULL DEFAULT '0',
    qtd_horas varchar(50) NOT NULL DEFAULT '',
    total varchar(50) NOT NULL DEFAULT '',
    reg_date timestamp NOT NULL DEFAULT current_timestamp(),
    last_update timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
    PRIMARY KEY (hs_id),
    KEY fk_usuario (user_id),
    CONSTRAINT fk_usuario FOREIGN KEY (user_id) REFERENCES usuario (user_id) ON DELETE CASCADE ON UPDATE CASCADE
)
```

Fonte: Autoral (2020).

4.1.9 Tabela Inventário

Essa tabela é responsável dados referentes aos produtos do sistema e seus dados são inseridos ou alterados de forma automática por meio de triggers que serão especificadas posteriormente.

Figura 15 – Tabela de Inventário

```
CREATE TABLE IF NOT EXISTS inventario (
    inventario_id int(11) NOT NULL AUTO_INCREMENT,
    categoria_id int(11) NOT NULL,
    qtd int(11) NOT NULL,
    reg_date timestamp NOT NULL DEFAULT current_timestamp(),
    last_update timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
    PRIMARY KEY (inventario_id),
    KEY fk_inventario_categoria (categoria_id),
    CONSTRAINT fk_inventario_categoria FOREIGN KEY (categoria_id) REFERENCES categoria (categoria_id) ON DELETE CASCADE ON UPDATE CASCADE
)
```

Fonte: Autoral (2020).

4.1.10 Tabela Log de Transações

Essa tabela possui os dados referentes a todas as transações financeira adicionadas no sistema, uma vez que utiliza de triggers que serão posteriormente explicadas para inserir seus registros e atualizar de forma automática após a execução da ação da ação no sistema.

Figura 16 – Tabela de Log de Transações

```
CREATE TABLE IF NOT EXISTS log_transacao (
    log_id int(11) NOT NULL AUTO_INCREMENT,
    venda_id int(11) DEFAULT NULL,
    atendimento_id int(11) DEFAULT NULL,
    compra_id int(11) DEFAULT NULL,
    descricao varchar(255) NOT NULL,
    valor varchar(50) NOT NULL DEFAULT '',
    reg_date timestamp NOT NULL DEFAULT current_timestamp(),
    PRIMARY KEY (log_id),
    KEY fk_atendimento_transacao (atendimento_id),
    KEY fk_compra_transacao (compra_id),
    KEY fk_venda_transacao (venda_id),
    CONSTRAINT fk_atendimento_transacao FOREIGN KEY (atendimento_id) REFERENCES atendimento (atendimento_id) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_compra_transacao FOREIGN KEY (compra_id) REFERENCES compra (compra_id) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_venda_transacao FOREIGN KEY (venda_id) REFERENCES venda (venda_id) ON DELETE CASCADE ON UPDATE CASCADE
)
```

Fonte: Autoral (2020).

4.1.11 Tabela Produtos

A tabela de produtos possui somente os códigos do produto, identificação de da categoria e status, sendo o status utilizado como base para atualização automática da tabela de estoque possuindo dois status: vendido e estoque.

Ao cadastrar um produto é gerado automaticamente seu status como em estoque.

Após adicionar um produto ao carrinho de venda, o status do produto é alterado para vendido e é dada baixa no estoque.

Figura 17 – Tabela de Produtos

```
CREATE TABLE IF NOT EXISTS produto (
    produto_id int(11) NOT NULL AUTO_INCREMENT,
    categoria_id int(11) NOT NULL,
    status varchar(255) NOT NULL,
    reg_date timestamp NOT NULL DEFAULT current_timestamp(),
    last_update timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
    PRIMARY KEY (produto_id),
    KEY fk_produto_categoria (categoria_id),
    CONSTRAINT fk_produto_categoria FOREIGN KEY (categoria_id) REFERENCES categoria (categoria_id) ON DELETE CASCADE ON UPDATE CASCADE
)
```

Fonte: Autoral (2020).

4.1.12 Tabela Representantes de Vendas

A tabela de representantes possui os dados dos representantes de vendas que prestam serviço aos fornecedores possuindo o código de identificação do fornecedor como atributo.

Figura 18 – Tabela de Representante de Vendas

```
CREATE TABLE IF NOT EXISTS representante (
    representante_id int(11) NOT NULL AUTO_INCREMENT,
    fornecedor_id int(11) NOT NULL,
    primeiro_nome varchar(100) NOT NULL,
    ultimo_nome varchar(100) NOT NULL,
    email varchar(255) NOT NULL,
    cpf varchar(20) NOT NULL,
    telefone varchar(30) DEFAULT NULL,
    celular varchar(30) NOT NULL,
    reg_date timestamp NOT NULL DEFAULT current_timestamp(),
    last_update timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
    PRIMARY KEY (representante_id),
    KEY fk_fornecedor (fornecedor_id),
    CONSTRAINT fk_fornecedor FOREIGN KEY (fornecedor_id) REFERENCES fornecedor (fornecedor_id) ON DELETE CASCADE
    ON UPDATE CASCADE
)
```

Fonte: Autoral (2020).

4.1.13 Tabela Serviços

A tabela de serviços possui em seus registros os dados de todos os serviços prestados pelo Petshop e ou Clínica Veterinária.

Figura 19 – Tabela de Serviços

```
CREATE TABLE IF NOT EXISTS servico (
    servico_id int(11) NOT NULL AUTO_INCREMENT,
    nome varchar(100) NOT NULL,
    descricao varchar(255) NOT NULL,
    departamento varchar(100) NOT NULL,
    preco varchar(50) NOT NULL DEFAULT '0',
    reg_date timestamp NOT NULL DEFAULT current_timestamp(),
    last_update timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
    PRIMARY KEY (servico_id)
)
```

Fonte: Autoral (2020).

4.1.14 Tabela Serviços X Atendimento

A tabela em questão faz a relação entre os atendimentos e os respectivos serviços prestados.

Figura 20 – Tabela de Serviços X Atendimentos

```
CREATE TABLE IF NOT EXISTS servico_atendimento (
    id int(11) NOT NULL AUTO_INCREMENT,
    atendimento_id int(11) NOT NULL,
    servico_id int(11) NOT NULL,
    valor decimal(5,2) NOT NULL DEFAULT 0.00,
    reg_date timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
    PRIMARY KEY (id,atendimento_id,servico_id),
    KEY fk_atendimento_servico (atendimento_id),
    KEY fk_servico_atendimento (servico_id),
    CONSTRAINT fk_atendimento_servico FOREIGN KEY (atendimento_id) REFERENCES atendimento (atendimento_id) ON DELETE CASCADE ON
    UPDATE CASCADE,
    CONSTRAINT fk_servico_atendimento FOREIGN KEY (servico_id) REFERENCES servico (servico_id) ON DELETE CASCADE ON UPDATE CASCADE
)
```

Fonte: Autoral (2020).

4.1.15 Tabela Usuários

A tabela de usuários possui os dados dos funcionários e usuários do sistema.

Figura 21 – Tabela de Serviços X Atendimentos

```
CREATE TABLE IF NOT EXISTS usuario (
    user_id int(11) NOT NULL AUTO_INCREMENT,
    primeiro_nome varchar(100) NOT NULL,
    ultimo_nome varchar(100) NOT NULL,
    email varchar(255) NOT NULL,
    cpf varchar(20) NOT NULL,
    logradouro varchar(255) NOT NULL,
    complemento varchar(255) DEFAULT NULL,
    cep varchar(20) NOT NULL,
    bairro varchar(255) NOT NULL,
    cidade varchar(255) NOT NULL,
    estado varchar(255) NOT NULL,
    telefone varchar(30) DEFAULT NULL,
    celular varchar(30) NOT NULL,
    login varchar(100) NOT NULL,
    senha varchar(100) NOT NULL,
    funcao varchar(100) NOT NULL,
    reg_date timestamp NOT NULL DEFAULT current_timestamp(),
    last_update timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp(),
    PRIMARY KEY ('user_id')
)
```

Fonte: Autoral (2020).

4.1.16 Tabela Vendas

A tabela de vendas possui os dados das vendas realizadas.

Figura 22 – Tabela de Vendas

```
CREATE TABLE IF NOT EXISTS venda (
    venda_id int(11) NOT NULL AUTO_INCREMENT,
    cliente_id int(11) NOT NULL,
    descricao varchar(255) NOT NULL,
    valor decimal(6,2) NOT NULL DEFAULT 0.00,
    status varchar(50) NOT NULL,
    reg_date timestamp NOT NULL DEFAULT current_timestamp(),
    last_update timestamp NOT NULL DEFAULT current_timestamp(),
    PRIMARY KEY (venda_id),
    KEY fk_cliente_venda (cliente_id),
    CONSTRAINT fk_cliente_venda FOREIGN KEY (cliente_id) REFERENCES cliente (cliente_id) ON DELETE CASCADE ON UPDATE CASCADE
)
```

Fonte: Autoral (2020).

4.1.17 Tabela Vendas X Produtos

Tabela responsável pela relação da Venda com seu Produtos podendo ser definida como a tabela de carrinho de compras.

Figura 23 – Tabela de Vendas

```
CREATE TABLE IF NOT EXISTS venda_produto (
    id int(11) NOT NULL AUTO_INCREMENT,
    venda_id int(11) NOT NULL,
    produto_id int(11) NOT NULL,
    valor varchar(50) NOT NULL DEFAULT '',
    reg_date timestamp NOT NULL DEFAULT current_timestamp(),
    PRIMARY KEY (id,produto_id,venda_id),
    KEY fk_produto_venda (venda_id),
    KEY fk_venda_produto (produto_id),
    CONSTRAINT fk_produto_venda FOREIGN KEY (venda_id) REFERENCES venda (venda_id) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_venda_produto FOREIGN KEY (produto_id) REFERENCES produto (produto_id) ON DELETE CASCADE ON UPDATE CASCADE
)
```

Fonte: Autoral (2020).

4.2 Triggers

As trigger do sistema realizam diversas funções no banco de acordo com o que foi alterado do mesmo, ou seja, toda vez que houver uma alteração(*insert, update ou delete*) em uma tabela especificada elas realizarão um ação pré-determinada.

Abaixo estão os códigos de criação das triggers utilizadas pelo banco de dados.

4.2.1 Trigger inserção Atendimento

Trigger responsável por inserir os dados de todos os atendimentos prestados e seus respectivos valores na tabela Log de Transação.

Figura 24 – Trigger After Insert Atendimento

```
DELIMITER //
CREATE TRIGGER atendimento_after_insert AFTER INSERT ON atendimento FOR EACH ROW begin
    insert into log_transacao(atendimento_id,descricao,valor,reg_date)
    values(new.atendimento_id, concat('Atendimento realizado no Animal: ',new.animal_id), new.valor ,new.
.reg_date);
end//
DELIMITER ;
```

Fonte: Autoral (2020).

4.2.2 Trigger alteração Atendimento

Trigger responsável por realizar alterações nos valores de atendimentos em caso de adição ou retirada de serviços prestados na tabela Log de Transação.

Figura 25 – Trigger After Update Atendimento

```
DELIMITER //
CREATE TRIGGER atendimento_after_update AFTER UPDATE ON atendimento FOR EACH ROW BEGIN
UPDATE log_transacao
    SET valor = NEW.valor
    WHERE atendimento_id = new.atendimento_id;
END//
DELIMITER ;
```

Fonte: Autoral (2020).

4.2.3 Trigger inserção Categoria

Trigger responsável por inserir categorias no inventario.

Figura 26 – Trigger After Insert Categoria

```
DELIMITER //
CREATE TRIGGER categoria_after_insert AFTER INSERT ON categoria FOR EACH ROW BEGIN
    INSERT INTO inventario(categoria_id,qtd)
    VALUES (NEW.categoria_id,0);
END//
DELIMITER ;
```

Fonte: Autoral (2020).

4.2.4 Trigger inserção Compra

Trigger responsável por inserir os dados de todas as compras e seus respectivos valores na tabela Log de Transação.

Figura 27 – Trigger After Insert Compra

```
DELIMITER //
CREATE TRIGGER compra_after_insert AFTER INSERT ON compra FOR EACH ROW begin
    insert into log_transacao(compra_id,descricao,valor,reg_date)
    values(new.compra_id, concat('Compra de produtos do fornecedor: ',new.fornecedor_id), new.valor ,new.
    .reg_date);
end//
DELIMITER ;
```

Fonte: Autoral (2020).

4.2.5 Trigger inserção Produto

Trigger responsável por adicionar produtos ao inventario.

Figura 28 – Trigger After Insert Produto

```
DELIMITER //
CREATE TRIGGER produto_insert AFTER INSERT ON produto FOR EACH ROW BEGIN
    UPDATE inventario
    SET qtd = (SELECT COUNT(p.status) qtd
    FROM produto p
    WHERE p.status = 'Estoque'
    AND categoria_id = NEW.categoria_id
    ORDER BY categoria_id)
    WHERE categoria_id = new.categoria_id;
END//
DELIMITER ;
```

Fonte: Autoral (2020).

4.2.6 Trigger alteração Produto

Trigger responsável por remover produtos ao inventario.

Figura 29 – Trigger After Update Produto

```
DELIMITER //
CREATE TRIGGER produto_update AFTER UPDATE ON produto FOR EACH ROW begin
    UPDATE inventario
    SET qtd = (SELECT COUNT(p.status) qtd
    FROM produto p
    WHERE p.status = 'Estoque'
    ORDER BY categoria_id)
    WHERE categoria_id = new.categoria_id;
end//
DELIMITER ;
```

Fonte: Autoral (2020).

4.2.7 Trigger inserção Serviços X Atendimento

Trigger responsável por alterar o valor do atendimento ao adicionar um serviço prestado.

Figura 30 – Trigger After Insert Serviços X Atendimentos

```
DELIMITER //
CREATE TRIGGER servicio_atendimento_add AFTER INSERT ON servicio_atendimento FOR EACH ROW BEGIN
    UPDATE atendimento a
    SET valor = (SELECT SUM(valor)
                 FROM servicio_atendimento sa
                 WHERE sa.atendimento_id = new.atendimento_id)
    WHERE a.atendimento_id = new.atendimento_id;
END//
DELIMITER ;
```

Fonte: Autoral (2020).

4.2.8 Trigger remoção Serviços X Atendimento

Trigger responsável por alterar o valor do atendimento ao remover um serviço prestado.

Figura 31 – Trigger After Delete Serviços X Atendimentos

```
DELIMITER //
CREATE TRIGGER servicio_atendimento_drop AFTER DELETE ON servicio_atendimento FOR EACH ROW BEGIN
    UPDATE atendimento a
    SET valor = (SELECT SUM(valor)
                 FROM servicio_atendimento sa
                 WHERE sa.atendimento_id = old.atendimento_id)
    WHERE a.atendimento_id = old.atendimento_id;
END//
DELIMITER ;
```

Fonte: Autoral (2020).

4.2.9 Trigger inserção Venda

Trigger responsável por adicionar as vendas e seus respectivos valores na tabela Log de Transações.

Figura 32 – Trigger After Insert Venda

```
DELIMITER //
CREATE TRIGGER venda_after_insert AFTER INSERT ON venda FOR EACH ROW begin
    insert into log_transacao(venda_id,descricao,valor,reg_date)
    values(new.venda_id, concat('Venda para o Cliente: ',new.cliente_id), new.valor ,new.reg_date);
end//
DELIMITER ;
```

Fonte: Autoral (2020).

4.2.10 Trigger alteração Venda

Trigger responsável por alterar o valor da venda na tabela Log de Transação.

Figura 33 – Trigger After Update Venda

```
DELIMITER //
CREATE TRIGGER venda_after_update AFTER UPDATE ON venda FOR EACH ROW BEGIN
    UPDATE log_transacao
    SET valor = NEW.valor
    WHERE venda_id = new.venda_id;
END//
DELIMITER ;
```

Fonte: Autoral (2020).

4.2.11 Trigger inserção Venda X Produto

Trigger responsável por alterar o status do produto para Vendido.

Figura 34 – Trigger After Insert Vendas X Produtos

```
DELIMITER //
CREATE TRIGGER venda_produto_add AFTER INSERT ON venda_produto FOR EACH ROW BEGIN
    UPDATE produto
    SET STATUS = 'Vendido'
    WHERE produto_id = new.produto_id;

    UPDATE venda v
    SET valor = (SELECT SUM(valor)
                 FROM venda_produto vp
                 WHERE vp.venda_id = new.venda_id)
    WHERE v.venda_id = new.venda_id;
END//
DELIMITER ;
```

Fonte: Autoral (2020).

4.2.12 Trigger remoção Venda X Produto

Trigger responsável por alterar o status do produto para Estoque.

Figura 35 – Trigger After Delete Vendas X Produtos

```
DELIMITER //
CREATE TRIGGER venda_produto_drop AFTER DELETE ON venda_produto FOR EACH ROW BEGIN
    UPDATE produto
    SET STATUS = 'Estoque'
    WHERE produto_id = old.produto_id;

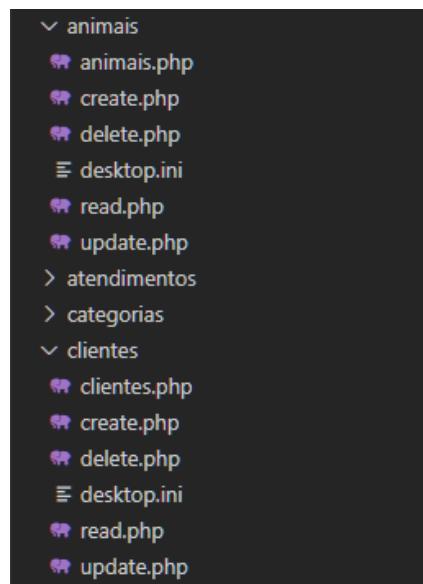
    UPDATE venda v
    SET valor = (SELECT SUM(valor)
                 FROM venda_produto vp
                 WHERE vp.venda_id = old.venda_id)
    WHERE v.venda_id = old.venda_id;
END//
DELIMITER ;
```

Fonte: Autoral (2020).

5. O SISTEMA

O sistema baseia-se no conceito de CRUD⁵ (*Create, Read, Update, Delete*), onde a interface foi produzida a partir das funções básicas de bancos de dados relacionais. Além disso, o sistema utiliza a ideia de padronização onde suas telas de utilização possuem um padrão pré-definido estrutural como exibido nas Figuras 36.

Figura 36 – CRUD



Fonte: Autoral (2020).

Figura 37 – Exemplo Create

```

clientes > create.php > ...
1  <?php
2  require_once "../config.php";
3  require_once "../login.php";
4  if ($_SERVER["REQUEST_METHOD"] == "POST") {
5      $sql = "INSERT INTO cliente (primeiro_nome,ultimo_nome,email,cpf,logradouro,complemento,cep,bairro,cidade,estado,telefone,celular,bdate)
6      VALUES ('$_POST[primeiro_nome]','$_POST[ultimo_nome]','$_POST[email]','$_POST[cpf]','$_POST[logradouro]','$_POST[complemento]','$_POST[cep]','$_POST[bairro]',
7      '$_POST[cidade]','$_POST[estado]','$_POST[telefone]','$_POST[celular]',CONVERT('$_POST[bdate]', DATE))";
8      $stmt = $conn->prepare($sql);
9      $stmt->execute();
10     if ($stmt->error) {
11         echo "Error!" . $stmt->error;
12         exit();
13     } else [
14         header("location: clientes.php");
15         exit();
16     ]
17 }
18 ?>
  
```

Fonte: Autoral (2020).

⁵ CRUD – *Create, Read, Update e Delete* ou Inserção, Leitura, Alteração e Remoção são as quatro operações básicas que existem em bancos de dados relacionais.

Figura 38 – Exemplo Read

```

<?php
require_once './logins.php';
require_once './config.php';
if (isset($_GET['cliente_id'])) {
    try {
        $sql = "SELECT c.cliente_id, c.primeiro_nome, c.ultimo_nome, c.email, c.cpf, c.logradouro, c.complemento, c.cep,
                c.bairro, c.cidade, c.estado, c.telefone, c.celular, c.bdate, c.reg_date, c.last_update
        FROM cliente
        WHERE c.cliente_id = '".$_GET['cliente_id']."' ";
        $stmt = $conn->prepare($sql);
        $stmt->execute();
        $stmt->fetch(PDO::FETCH_ASSOC);
        $linha = $stmt->fetch(PDO::FETCH_ASSOC);
        $primeiro_nome = $linha['primeiro_nome'];
        $ultimo_nome = $linha['ultimo_nome'];
        $email = $linha['email'];
        $cpf = $linha['cpf'];
        $logradouro = $linha['logradouro'];
        $complemento = $linha['complemento'];
        $bairro = $linha['bairro'];
        $cidade = $linha['cidade'];
        $estado = $linha['estado'];
        $telefone = $linha['telefone'];
        $celular = $linha['celular'];
        $bdate = $linha['bdate'];
        $reg_date = $linha['reg_date'];
        $last_update = $linha['last_update'];
    } catch (PDOException $e) {
        echo "Error: " . $e->getMessage();
    }
    if ($telefone == ''){
        $telefone = 'Não Informado';
    }
    if ($celular == ''){
        $celular = 'Não Informado';
    }
    if ($complemento == ''){
        $complemento = 'Não Informado';
    }
    $conn = null;
}
?>
```

Fonte: Autoral (2020).

Figura 39 – Exemplo Update

```

<?php
require_once './logins.php';
require_once './config.php';
if (isset($_GET['cliente_id'])) {
    try {
        $sql = "SELECT c.cliente_id, c.primeiro_nome, c.ultimo_nome, c.email, c.cpf, c.logradouro, c.complemento, c.cep,
                c.bairro, c.cidade, c.estado, c.telefone, c.celular, c.bdate, c.reg_date, c.last_update
        FROM cliente
        WHERE c.cliente_id = '".$_GET['cliente_id']."' ";
        $stmt = $conn->prepare($sql);
        $stmt->execute();
        $linha = $stmt->fetch(PDO::FETCH_ASSOC);
        $primeiro_nome = $linha['primeiro_nome'];
        $ultimo_nome = $linha['ultimo_nome'];
        $email = $linha['email'];
        $cpf = $linha['cpf'];
        $logradouro = $linha['logradouro'];
        $complemento = $linha['complemento'];
        $bairro = $linha['bairro'];
        $cidade = $linha['cidade'];
        $estado = $linha['estado'];
        $telefone = $linha['telefone'];
        $celular = $linha['celular'];
        $bdate = $linha['bdate'];
        $reg_date = $linha['reg_date'];
        $last_update = $linha['last_update'];
    } catch (PDOException $e) {
        echo "Error: " . $e->getMessage();
    }
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        $sql = "UPDATE cliente SET primeiro_nome = '".$_POST['primeiro_nome'].', ultimo_nome = '".$_POST['ultimo_nome'].', email = '".$_POST['email'].', logradouro = '".$_POST['logradouro'].',
                complemento = '".$_POST['complemento'].', bairro = '".$_POST['bairro'].', cidade = '".$_POST['cidade'].', estado = '".$_POST['estado'].', telefone = '".$_POST['telefone'].',
                celular = '".$_POST['celular'].', bdate = '".$_POST['bdate'].', reg_date = '".$_POST['reg_date'].', last_update = '".$_POST['last_update'].'
                WHERE cliente_id = '".$_GET['cliente_id']."' ";
        $stmt = $conn->prepare($sql);
        $stmt->execute();
        $linha = $stmt->fetch(PDO::FETCH_ASSOC);
        $cliente_id = $linha['cliente_id'];
        $primeiro_nome = $linha['primeiro_nome'];
        $ultimo_nome = $linha['ultimo_nome'];
        $email = $linha['email'];
        $cpf = $linha['cpf'];
        $logradouro = $linha['logradouro'];
        $complemento = $linha['complemento'];
        $bairro = $linha['bairro'];
        $cidade = $linha['cidade'];
        $estado = $linha['estado'];
        $telefone = $linha['telefone'];
        $celular = $linha['celular'];
        $bdate = $linha['bdate'];
        $reg_date = $linha['reg_date'];
        $last_update = $linha['last_update'];
    } catch (PDOException $e) {
        echo "Error: " . $e->getMessage();
    }
}
?>
```

Fonte: Autoral (2020).

Figura 40 – Exemplo Delete

```

<?php
require_once './logins.php';
require_once './config.php';
if (isset($_GET['cliente_id'])) {
    try {
        $sql = "SELECT c.cliente_id, c.primeiro_nome, c.ultimo_nome, c.email, c.cpf, c.logradouro, c.complemento, c.cep,
                c.bairro, c.cidade, c.estado, c.telefone, c.celular, c.bdate, c.reg_date, c.last_update
        FROM cliente
        WHERE c.cliente_id = '".$_GET['cliente_id']."' ";
        $stmt = $conn->prepare($sql);
        $stmt->execute();
        $linha = $stmt->fetch(PDO::FETCH_ASSOC);
        $cliente_id = $linha['cliente_id'];
        $primeiro_nome = $linha['primeiro_nome'];
        $ultimo_nome = $linha['ultimo_nome'];
        $email = $linha['email'];
        $cpf = $linha['cpf'];
        $logradouro = $linha['logradouro'];
        $complemento = $linha['complemento'];
        $bairro = $linha['bairro'];
        $cidade = $linha['cidade'];
        $estado = $linha['estado'];
        $telefone = $linha['telefone'];
        $celular = $linha['celular'];
        $bdate = $linha['bdate'];
        $reg_date = $linha['reg_date'];
        $last_update = $linha['last_update'];
    } catch (PDOException $e) {
        echo "Error: " . $e->getMessage();
    }
}
?>

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $sql = "DELETE
            FROM cliente
            WHERE cliente_id = '".$_GET['cliente_id']."' ";
    $stmt = $conn->prepare($sql);
    $stmt->execute();
    if ($stmt->rowCount() == 1) {
        echo "Error! " . $stmt->error;
        exit(0);
    } else {
        header("location: clientes.php");
        exit(0);
    }
}
```

Fonte: Autoral (2020).

O sistema foi desenvolvido utilizando a linguagem de programação PHP o que na prática possibilitou um vasto material de apoio ao desenvolvimento, além de permitir que o sistema seja web, o que expande as possibilidade de design, utilizando o *Bootstrap 4* aliado ao *JavaScript* para uma melhor experiencia de usuário.

Assim como o PHP que além de escolhido por material de apoio também foi escolhido por possuir código aberto, o MySQL foi escolhido da mesma maneira.

Os requisitos do sistema foram definidos como módulos do sistema, logo o sistema utiliza desses módulos para executar as funções definidas pelo levantamento de requisitos e diagramação. Atualmente no sistema existem os seguintes módulos:

Administrativo: Módulo responsável por gerenciar as informações dos funcionários e suas respectivas cargas horarias de trabalho.

Atendimento: Módulo destinado ao gerenciamento de atendimentos, responsável por definir os serviços prestados em um animal que está em atendimento ou também sendo possível exibir um histórico de atendimentos finalizados.

Clientes: O módulo de clientes é responsável por gerenciar clientes e seus respectivos animais.

Compras: Módulo criado com o objetivo de registrar compras realizadas.

Financeiro: O módulo financeiro apresenta ao usuário do sistema uma tabela com todas as transações realizadas podendo ser elas em Atendimentos, Compras ou Vendas.

Fornecedores: O módulo de fornecedores consiste na possibilidade de gerenciar fornecedores de produtos e seus representantes de vendas.

Produtos: Módulo dividido em três seções que se relacionam entre si, sendo elas Categorias de Produtos, Produtos e Estoque.

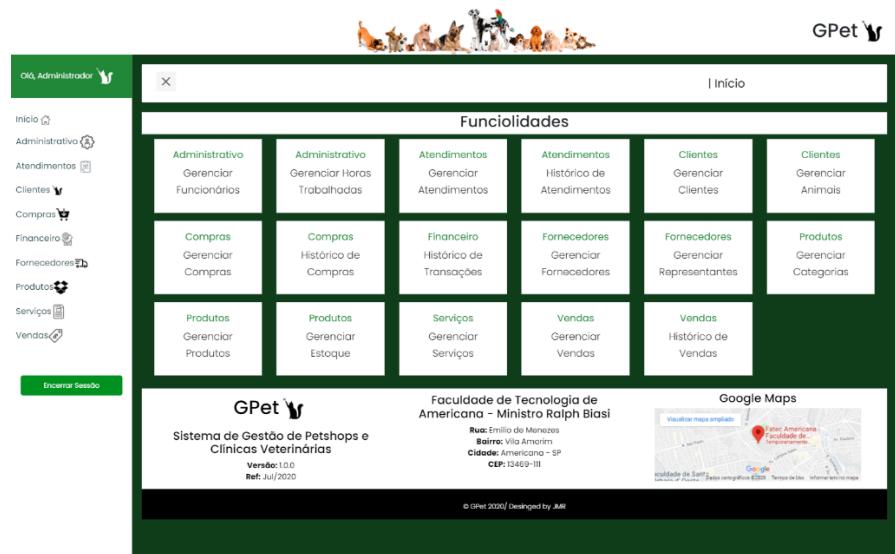
Serviços: Módulo destinado ao gerenciamento de serviços que são prestados em Atendimentos.

Vendas: O módulo de vendas é responsável por gerenciar as vendas realizadas no Petshop, criando um carrinho de compras para cada compra além de exibir o histórico de vendas finalizadas.

5.1 Layout do Genérico Sistema

As cores principais do sistemas são em diferentes tonalidades de verde alinhados as cores brancas e preta que dão ao sistema certo contraste além de utilizar o verde que segundo a psicologia das cores remete características como equilíbrio, saúde, tranquilidade, entre outros.

Figura 41 – Layout do Sistema



Fonte: Autoral (2020).

Figura 42 – Logotipo do Sistema



Fonte: Autoral (2020).

Figura 43 – Cabeçalho do Sistema



Fonte: Autoral (2020).

A barra de orientação permite retornar a níveis superiores do sistema de forma direta.

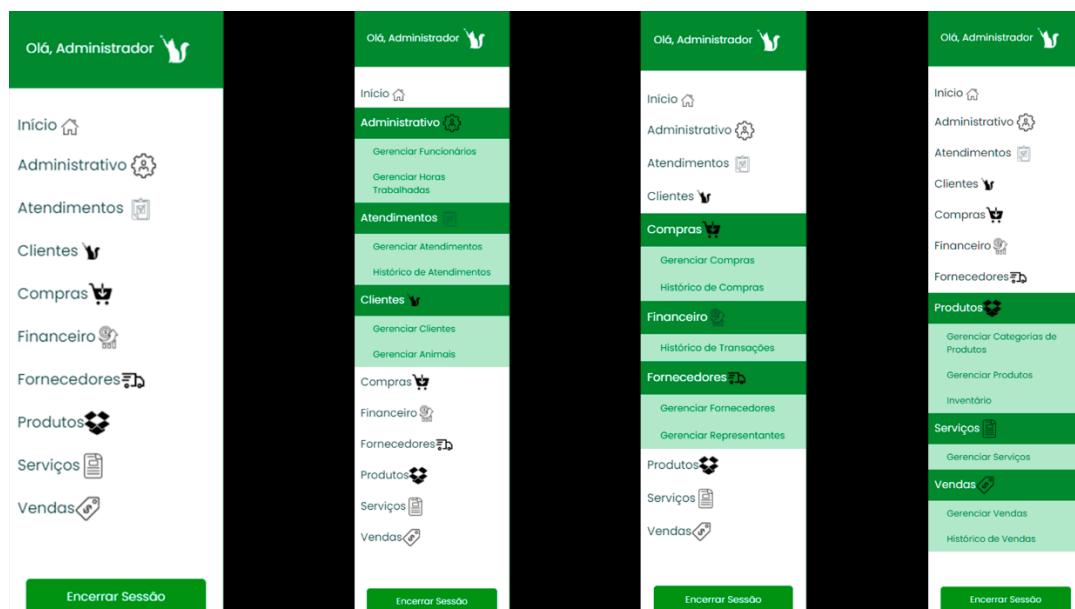
Figura 44 – Barra de Orientação



Fonte: Autoral (2020).

Abaixo estão representados o Menu Lateral e os módulos do sistemas.

Figura 45 – Menu Lateral



Fonte: Autoral (2020).

Como mencionado, o sistema possui padronização de telas. Abaixo é possível visualizar os layouts de conteúdo disponíveis no sistema.

Figura 46 – Área de conteúdo

Fonte: Autoral (2020).

Figura 47 – Informações do Sistema e Footer

Fonte: Autoral (2020).

5.1.1 Layout Tela de Cadastro

Figura 48 – Tela de Cadastro

GPet

| Início > Funcionários > Cadastrar Funcionário

Adicionar Funcionário

Cadastro de Funcionário

Campo Obrigatório(*)

Nome*

Sobrenome*

Email:*

seuemail@gmail.com

CPF:*

000.000.000-00

Logradouro*

Rua:XXXXXXXXXXXXXX, NºXXXX

Complemento

Bloco:XXXXXXXXXX, Apto:XXXX

CEP:*

XXXXX-XXX

Bairro*

Cidade

Americana

Estado*

São Paulo - SP

Telefone:

(xx)XXXX-XXXX

Celular:*

(xx)XXXX-XXXX

Login:*

Senha:*

Função*

Administrador

Cadastrar **Voltar**

GPet

Sistema de Gestão de Petshops e Clínicas Veterinárias

Versão: 1.0.0
Ref: Jul/2020

Faculdade de Tecnologia de Americana -
Ministro Ralph Biasi

Rua: Emílio de Menezes
Bairro: Vila Amorim
Cidade: Americana - SP
CEP: 13469-111

Google Maps

© GPet 2020/ Designed by JMR

Fonte: Autoral (2020).

5.1.2 Layout Tela de Visualização

Figura 49 – Tela de Visualização

The screenshot shows a web-based application for managing petshops and veterinary clinics. At the top, there's a decorative banner with various dog breeds. The header includes the GPet logo and navigation links: Início > Funcionários > Visualizar Funcionário. Below this, the main title is "Visualizar Cadastro". The central content area is titled "Informações do Funcionário" and displays the following information:

- Código de Identificação**: 10
- Nome**: Administrador
- Sobrenome**: Logradouro
- Email**: admin@admin.com
- CPF**: 000.000.000-00
- Logradouro**: Administrador
- Complemento**: CEP 00000-000
- Bairro**: Administrador
- Cidade**: Americana
- Estado**: São Paulo - SP
- Telefone**: (00)0000-0000
- Celular**: (00)00000-0000
- Função**: Administrador
- Data de Registro**: 2020-07-11 17:58:41
- Última Atualização**: 2020-07-11 17:58:41

A red "Voltar" (Back) button is located at the bottom of this section. At the very bottom of the page, there's footer information for "Faculdade de Tecnologia de Americana - Ministro Ralph Biasi", a Google Maps link showing the location in Americana, and copyright information: © GPet 2020 | Designed by JMR.

Fonte: Autoral (2020).

5.1.3 Layout Tela de Alteração

Figura 50 – Tela de Alteração

Início > Funcionários > Alterar Funcionário

Alterar Funcionário

Alterar
Preencha os novos dados e clique em atualizar para finalizar a edição.

Campo Obrigatório(*)

Código de Identificação*

Nome*

Sobrenome*

Email:*

admin@admin.com

CPF:*

000.000.000-00

Logradouro*

Administrador

Complemento

Bloco:XXXXXX, Apto:XXXX

CEP:*

00000-000

Bairro*

Administrador

Cidade

Americana

Estado*

São Paulo - SP

Telefone:

(00)0000-0000

Celular:*

(00)00000-0000

Login:*

admin

Senha:*

Função*

Administrador

Data de Registro*

2020-07-11 17:58:41

Última Atualização*

2020-07-11 17:58:41

Atualizar **Voltar**

GPet

Sistema de Gestão de Petshops e Clínicas Veterinárias

Versão: 1.0.0
Ref: Jul/2020

Faculdade de Tecnologia de Americana –
Ministro Ralph Biasi

Rua: Emílio de Menezes
Bairro: Vila Amorim
Cidade: Americana - SP
CEP: 13469-111

Google Maps

Visualizar mapa ampliado

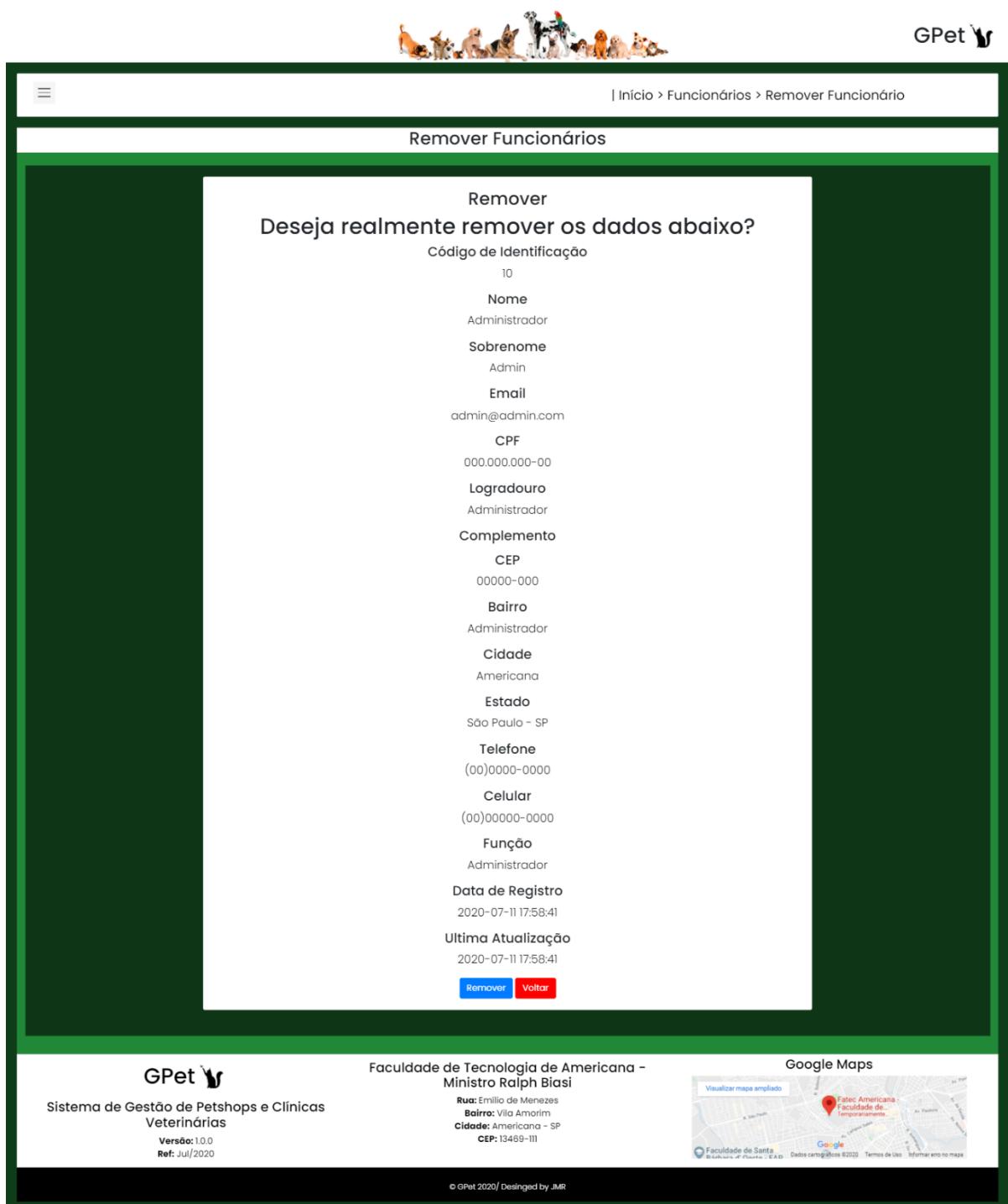
Faculdade de Santa

Dados cartográficos ©2020 Google. Termos de Uso. Informar erro no mapa.

Fonte: Autoral (2020).

5.1.4 Layout Tela de Remoção

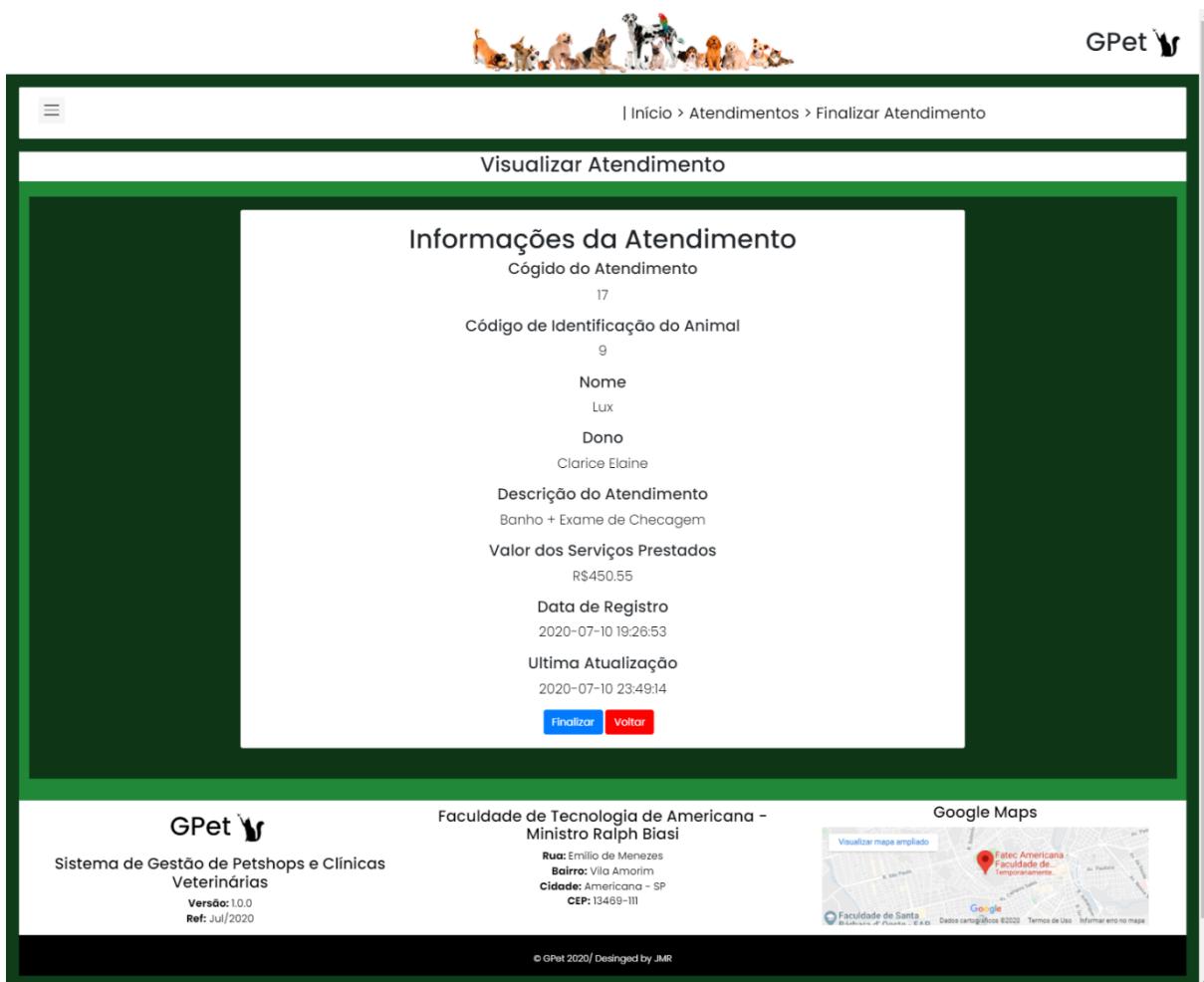
Figura 51 – Tela de Remoção



Fonte: Autoral (2020).

5.1.5 Layout Tela de Finalização

Figura 52 – Tela de Finalização

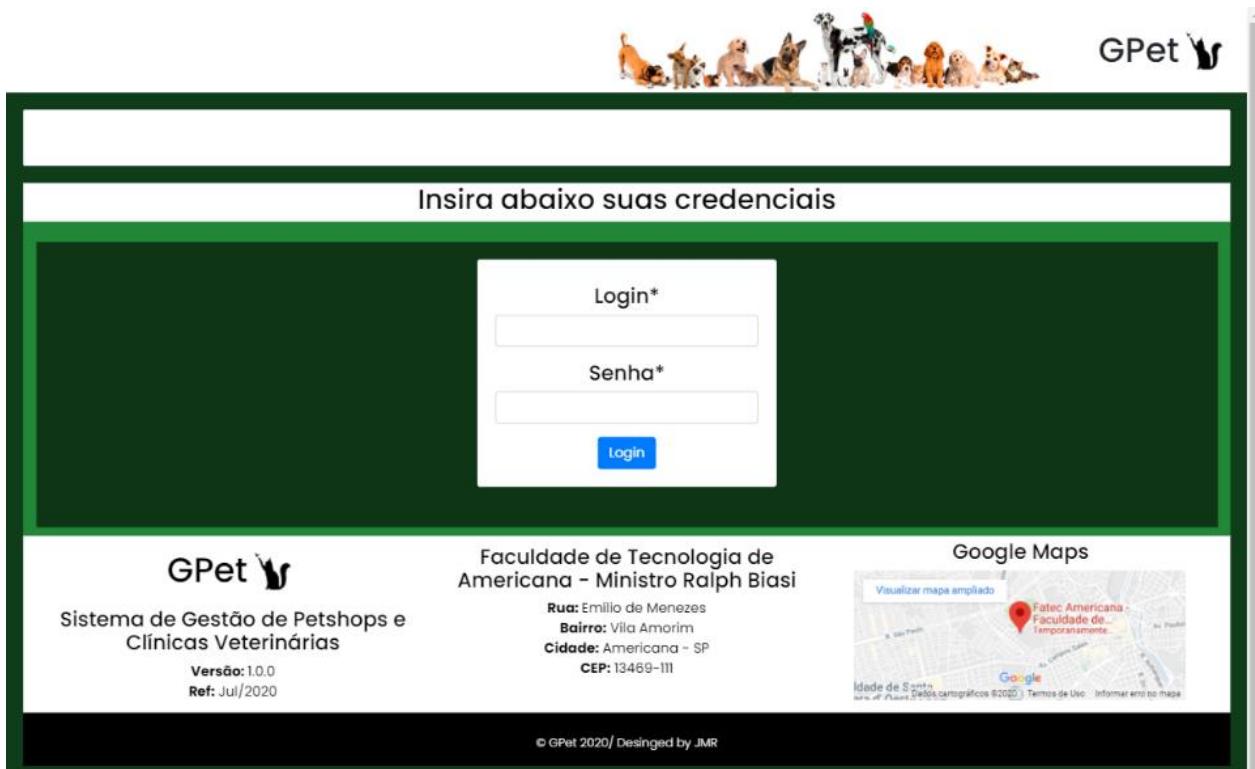


Fonte: Autoral (2020).

5.2 Características da Tela de Login

Abaixo é possível visualizar a tela de login do sistema

Figura 53 – Tela de Login

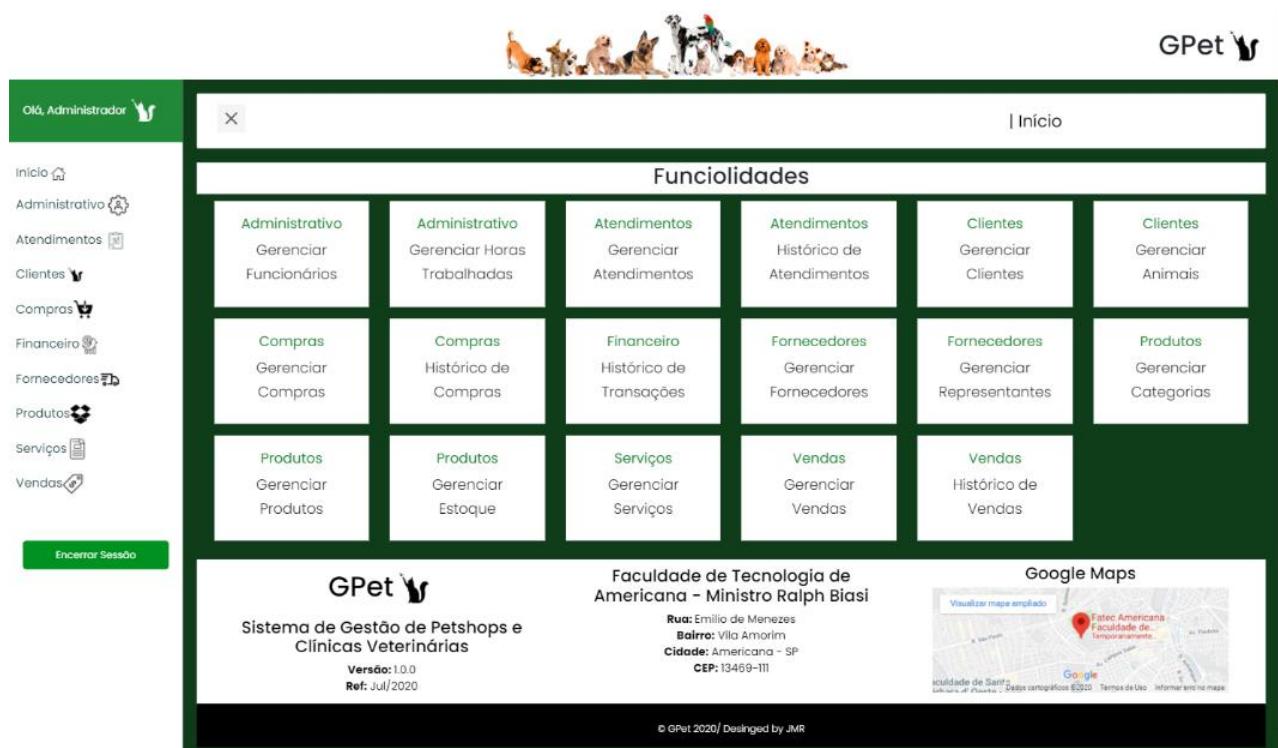


Fonte: Autoral (2020).

5.3 Características da Tela Principal

A tela principal do sistema permite a visualização das funcionalidades do sistema. Vale ressaltar que é o ponto de origem de todas as telas na barra de orientação, com exceção de Serviços Prestados e Carrinho que possuem como origem respectivamente Atendimentos e Vendas. A tela inicial foi representada na seção 5.1.2.

Figura 54 – Tela de Inicial



Fonte: Autoral (2020).

5.4 Características da Tela Gerenciamento de Funcionários

A tela gerenciamento de funcionários encontra-se em Administrativo>Gerenciar Funcionários.

É possível visualizar uma prévia das informações dos usuários cadastrados além de realizar pesquisas por Código de Identificação, Nome e Função.

Clicando em Adicionar Novo Usuário, será carregada a página de cadastro como representado na seção 5.1.1, assim como clicando em Visualizar, Editar e Remover serão carregadas as páginas como representado nas seções 5.1.2, 5.1.3 e 5.1.4.

Figura 55 – Gerenciamento de Funcionários

#	Nome	Sobrenome	Função	Action
10	Administrador	Admin	Administrador	Visualizar Editar Remover

Fonte: Autoral (2020).

5.5 Características da Tela Gerenciamento de Horas Trabalhadas

A tela gerenciamento de horas trabalhadas encontra-se em Administrativo>Gerenciar de Horas Trabalhadas.

É possível visualizar uma prévia das informações dos usuários cadastrados além de realizar pesquisas por Código de Identificação, Nome, Horas Trabalhadas e o Mês de Referência.

Clicando em Adicionar Nova Referencia, será carregada a página de cadastro que segue um layout genérico como representado na seção 5.1.1, assim como clicando em Visualizar, representado nas seções 5.1.2.

Figura 56 – Gerenciamento de Horas Trabalhadas

#	Nome	Quantidade de Horas	Referência	Action
16	Administrador Admin	10	Janeiro 2020	Visualizar Adicionar Horas

Fonte: Autoral (2020).

Ao clicar em Adicionar Horas o usuário é levado para uma tela que permite a ele adicionar horas na referência selecionada do funcionário. Após clicar em Atualizar o sistema gera automaticamente o valor a ser pago ao funcionário no campo Valor a Receber. A Figura 57 representa a tela em questão.

Figura 57 – Holerite

The screenshot shows a web-based application for managing payroll (Holerite) in the GPet system. The main title is "Alterar Holerite". The form is titled "Ficha 16 Referência Janeiro 2020" and contains the following fields:

- Campo Obrigatório(*)
- Número de Identificação*
- Nome*
- Referência*
- Valor por Hora Trabalhada em Reais(R\$)*
- Hora Trabalhadas(R\$)*
- Valor a Receber em Reais(R\$)*
- Data de Registro*
- Última Atualização*

At the bottom of the form are two buttons: "Atualizar" (Update) and "Voltar" (Back).

The footer of the page includes the GPet logo, system information (Sistema de Gestão de Petshops e Clínicas Veterinárias, Versão 1.0, Ref: Jul/2020), contact details for Faculdade de Tecnologia de Americana - Ministro Ralph Biasi, and a Google Maps link.

Fonte: Autoral (2020).

5.6 Características da Tela Gerenciamento de Atendimentos

A tela gerenciamento de Atendimentos encontra-se em Atendimentos>Gerenciar Atendimentos.

É possível visualizar uma prévia das informações dos atendimentos em andamento além de realizar pesquisas por Código de Identificação, Código do Animal, Nome, Dono e Valor.

Clicando em Novo Atendimento, será carregada a página de cadastro como representado na seção 5.1.1, assim como clicando em Visualizar seção 5.1.2 ou em Finalizar Atendimento seção 5.1.5.

Figura 58 – Gerenciamento de Atendimentos

#	Código do Animal	Nome	Dono	Valor	Action
17	9	Lux	Andreia Lúcia	450.55	Visualizar Serviços Prestados Finalizar Atendimento

Fonte: Autoral (2020).

Ao clicar em Serviços Prestados o usuário é levado para uma tela que permite a visualização de todos os serviços prestados no atendimento em questão, podendo também adicionar, visualizar e remover serviços prestados.

Figura 59 – Gerenciamento Serviços Prestados

#	Código Servico	Descrição	Preço	Data	Action
62	4	Checagem de Rotina	100.00	2020-07-10 20:00:26	Visualizar Remover
65	6	Castração do órgão reprodutor feminino felino	260.66	2020-07-10 23:49:08	Visualizar Remover
66	7	Castração do órgão reprodutor felino masculino	100.00	2020-07-10 23:49:14	Visualizar Remover

Serviços de Prestados no Atendimento

Novo Serviço Prestado

GPet

Sistema de Gestão de Petshops e Clínicas Veterinárias

Versão: 1.0.0
Ref: Jul/2020

Faculdade de Tecnologia de Americana - Ministro Ralph Biasi

Rua: Emílio de Menezes
Bairro: Vila Amorim
Cidade: Americana - SP
CEP: 13469-111

Google Maps

Visualizar mapa ampliado

Fatec Americana Faculdade de...
Faculdade de Santa Bárbara do Piauí - EAD

© GPet 2020/ Designed by JMR

Fonte: Autoral (2020).

5.7 Características da Tela de Histórico de Atendimentos

A tela gerenciamento de Atendimentos encontra-se em Atendimentos>Histórico de Atendimentos.

É possível visualizar uma prévia das informações dos atendimentos finalizados além de realizar pesquisas por Código de Identificação, Código do Animal, Nome, Dono e Valor.

Clicando em Visualizar, será carregada a página de visualização como representado na seção 5.1.2.

Figura 60 – Histórico de Atendimentos

#	Código do Animal	Nome	Dono	Valor	Action
13	7	Tody	Clarice Elaine	225.00	Visualizar
16	7	Tody	Clarice Elaine	125.00	Visualizar

Fonte: Autoral (2020).

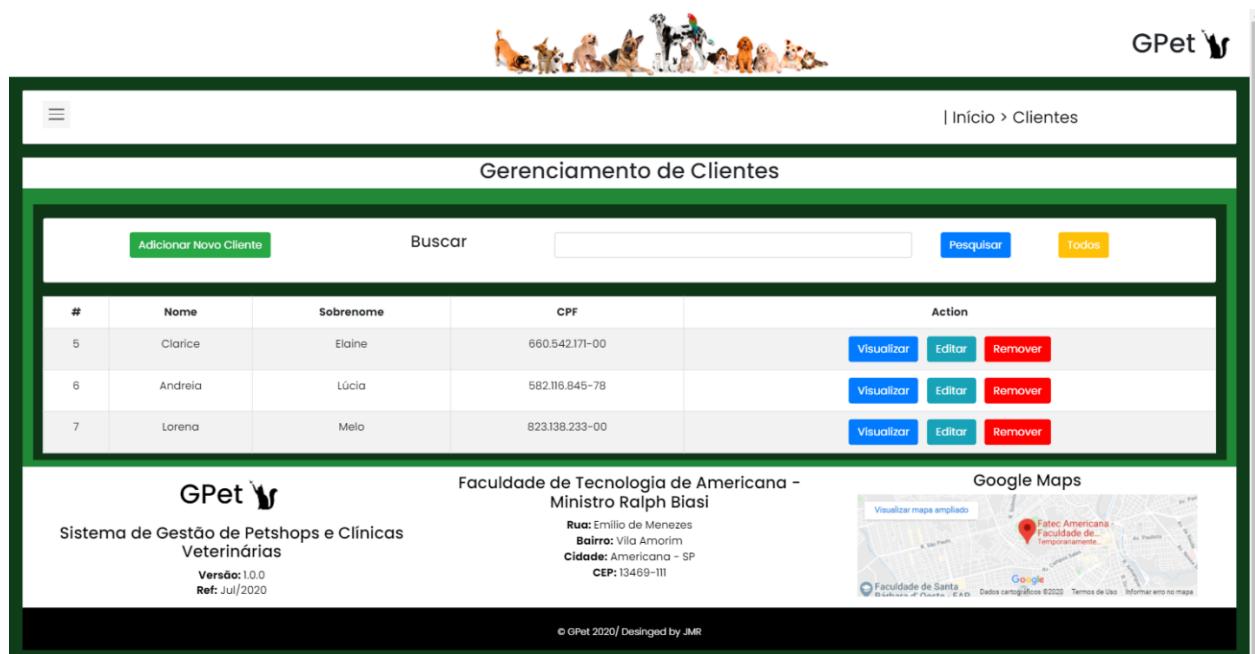
5.8 Características da Tela Gerenciamento de Clientes

A tela gerenciamento de clientes encontra-se em Clientes>Gerenciar Clientes.

É possível visualizar uma prévia das informações dos Clientes cadastrados além de realizar pesquisas por Código de Identificação, Nome e CPF.

Clicando em Adicionar Novo Cliente, será carregada a página de cadastro como representado na seção 5.1.1, assim como clicando em Visualizar, Editar e Remover serão carregadas as páginas como representado nas seções 5.1.2, 5.1.3 e 5.1.4.

Figura 61 – Gerenciamento de Clientes



The screenshot shows the 'Gerenciamento de Clientes' (Client Management) page of the GPet system. At the top, there's a decorative banner featuring various dogs and cats. The header includes the GPet logo and a navigation bar with 'Início > Clientes'. Below the header is a search bar with 'Gerenciamento de Clientes' placeholder text, and buttons for 'Pesquisar' (Search) and 'Todos' (All). A table lists three clients with columns for '#', 'Nome' (Name), 'Sobrenome' (Last Name), 'CPF', and 'Action'. Each client row has three buttons: 'Visualizar' (View), 'Editar' (Edit), and 'Remover' (Delete). At the bottom of the page, there are footer sections for 'GPet' (with a logo), 'Faculdade de Tecnologia de Americana - Ministro Ralph Biasi' (with address details), and 'Google Maps' (showing a map of the university location).

#	Nome	Sobrenome	CPF	Action
5	Clarice	Elaine	660.542.171-00	Visualizar Editar Remover
6	Andreia	Lúcia	582.116.845-78	Visualizar Editar Remover
7	Lorena	Melo	823.138.233-00	Visualizar Editar Remover

Fonte: Autoral (2020).

5.9 Características da Tela Gerenciamento de Animais

A tela gerenciamento de animais encontra-se em Clientes>Gerenciar Animais.

É possível visualizar uma prévia das informações dos Animais cadastrados além de realizar pesquisas por Código de Identificação, Nome e Dono.

Clicando em Adicionar Novo Animal, será carregada a página de cadastro como representado na seção 5.1.1, assim como clicando em Visualizar, Editar e Remover serão carregadas as páginas como representado nas seções 5.1.2, 5.1.3 e 5.1.4.

Figura 62 – Gerenciamento de Animais

#	Nome	Dono	Action
7	Tody	Clarice Elaine	Visualizar Editar Remover
9	Lux	Andreia Lúcia	Visualizar Editar Remover

Sistema de Gestão de Petshops e Clínicas Veterinárias
Versão: 1.0.0
Ref: Jul/2020

Faculdade de Tecnologia de Americana – Ministro Ralph Biasi
Rua: Emílio de Menezes
Bairro: Vila Amorim
Cidade: Americana - SP
CEP: 13469-111

Google Maps
Visualizar mapa ampliado

Fonte: Autoral (2020).

5.10 Características da Tela Gerenciamento de Compras

A tela gerenciamento de compras encontra-se em Compras>Gerenciar Compras.

É possível visualizar uma prévia das informações das Compras de fornecedores realizadas, além de realizar pesquisas por Código de Identificação, Razão Social do Fornecedor, Detalhes e Valor.

Clicando em Nova Compra, será carregada a página de cadastro como representado na seção 5.1.1, assim como clicando em Visualizar e Remover serão carregadas as páginas como representado nas seções 5.1.2 e 5.1.4.

Figura 63 – Gerenciamento de Compras

#	Fornecedor	Detalhes da Compra	Valor	Action
9	Vicente e Fátima Telas Ltda	Vacina Anti-Rábica	1000	Visualizar Remover

Fonte: Autoral (2020).

5.11 Características da Tela Histórico de Compras

A tela Histórico de Compras encontra-se em Compras>Histórico de Compras.

É possível visualizar uma prévia das informações das Compras de fornecedores realizadas, além de realizar pesquisas por Código de Identificação, Razão Social do Fornecedor, Detalhes e Valor.

Clicando em Visualizar será carregada a página de Visualização como representado na seção 5.1.2.

Figura 64 – Histórico de Compras

#	Fornecedor	Detalhes da Compra	Valor	Action
9	Vicente e Fátima Telas Ltda	Vacina Anti-Rábica	1000	Visualizar

GPet Sistema de Gestão de Petshops e Clínicas Veterinárias Versão: 1.0.0 Ref: Jul/2020

Faculdade de Tecnologia de Americana – Ministro Ralph Biasi
Rua: Emílio de Menezes
Bairro: Vila Amorim
Cidade: Americana – SP
CEP: 13469-111

Google Maps

Fonte: Autoral (2020).

5.12 Características da Tela Histórico de Transações

A tela Histórico de Transações encontra-se em Financeiro>Histórico de Transações.

É possível visualizar uma prévia das informações das Compras de fornecedores realizadas, além de realizar pesquisas por Código de Identificação, Descrição, Valor e Data de Registro.

Clicando em Visualizar será carregada a página de Visualização como representado na seção 5.1.2.

Figura 65 – Histórico de Transações

#	Descrição	Valor	Data	Action
27	Atendimento realizado no Animal: 7	225.00	2020-07-09 17:03:38	Visualizar
28	Atendimento realizado no Animal: 8	125.00	2020-07-09 17:04:21	Visualizar
31	Atendimento realizado no Animal: 7	125.00	2020-07-10 15:29:00	Visualizar
32	Compra de produtos do fornecedor: 2	1000	2020-07-10 16:11:07	Visualizar
33	Venda para o Cliente: 5	82	2020-07-10 16:40:22	Visualizar
34	Atendimento realizado no Animal: 9	450.55	2020-07-10 19:26:53	Visualizar
37	Venda para o Cliente: 7	116.00	2020-07-10 23:09:49	Visualizar

GPet Sistema de Gestão de Petshops e Clínicas Veterinárias Versão 1.0.0 Ref: Jul/2020

Faculdade de Tecnologia de Americana - Ministro Ralph Biasi Rua: Emílio de Menezes Bairro: Vila Amorim Cidade: Americana - SP CEP: 13469-111

Google Maps

Fonte: Autoral (2020).

5.13 Características da Tela Gerenciamento de Fornecedores

A tela gerenciamento de fornecedores encontra-se em Fornecedores>Gerenciar Fornecedores.

É possível visualizar uma prévia das informações dos Fornecedores cadastrados além de realizar pesquisas por Código de Identificação, Razão Social e CNPJ.

Clicando em Adicionar Novo Fornecedor, será carregada a página de cadastro como representado na seção 5.1.1, assim como clicando em Visualizar e Editar serão carregadas as páginas como representado nas seções 5.1.2 e 5.1.3.

Figura 66 – Gerenciamento de Fornecedores

#	Razão Social	CNPJ	Action
2	Vicente e Fótima Telas Ltda	24.142.282/0001-20	Visualizar Editar
3	Eliane e Paulo Entregas Expressas ME	69.720.225/0001-19	Visualizar Editar
4	Comercial Brasil Ltda	07.567.426/0001-70	Visualizar Editar

GPet
Sistema de Gestão de Petshops e Clínicas Veterinárias
Versão: 1.0.0
Ref: Jul/2020

Faculdade de Tecnologia de Americana -
Ministro Ralph Biasi
Rua: Emílio de Menezes
Bairro: Vila Amorim
Cidade: Americana - SP
CEP: 13469-111

Google Maps
Visualizar mapa ampliado | FATEC Americana - Faculdade de... temporaneamente... | Faculdade da Santa | Dados cartográficos 03/2020 | Termos de Uso | Informar erro no mapa

Fonte: Autoral (2020).

5.14 Características da Tela Gerenciamento de Representantes

A tela gerenciamento de representantes encontra-se em Fornecedores>Gerenciar Representantes. É possível visualizar uma prévia das informações dos Representantes de Venda cadastrados além de realizar pesquisas por Código de Identificação, Nome e Razão Social do Fornecedor.

Clicando em Adicionar Novo Representante, será carregada a página de cadastro como representado na seção 5.1.1, assim como clicando em Visualizar, Editar e Remover serão carregadas as páginas como representado nas seções 5.1.2, 5.1.3 e 5.1.4.

Figura 67 – Gerenciamento de Representantes

#	Nome	Razão Social do Fornecedor	Action
10	Julia Alves	Eliane e Paulo Entregas Expressos ME	Visualizar Editar Remover

GPet
Sistema de Gestão de Petshops e Clínicas Veterinárias
Versão: 1.0.0
Ref: Jul/2020

Faculdade de Tecnologia de Americana -
Ministro Ralph Biasi
Rua: Emílio de Menezes
Bairro: Vila Amorim
Cidade: Americana - SP
CEP: 13469-111

Google Maps

Fonte: Autoral (2020).

5.15 Características da Tela Gerenciamento de Categorias

A tela gerenciamento de categorias encontra-se em Produtos>Gerenciar Categorias. É possível visualizar uma prévia das informações das Categorias de Produtos cadastradas além de realizar pesquisas por Código de Identificação, Descrição e Razão Social do Fornecedor.

Clicando em Adicionar Nova Categoria, será carregada a página de cadastro correspondente , a seção 5.1.1 exibe o layout de cadastro genérico utilizado, assim como clicando em Visualizar, Editar e Remover serão carregadas as páginas como representado nas seções 5.1.2, 5.1.3 e 5.1.4.

Figura 68 – Gerenciamento de Categorias

#	Descrição	Fornecedor	Action
10	Shampoo Neutro	Eliane e Paulo Entregas Expressas ME	Visualizar Editar Remover
11	Tela de Proteção pra Gatos	Vicente e Fátima Telas Ltda	Visualizar Editar Remover
13	Laço de Orelha	Eliane e Paulo Entregas Expressas ME	Visualizar Editar Remover
15	Teste	Eliane e Paulo Entregas Expressas ME	Visualizar Editar Remover
16	Roupinha Dourada Tamanho P	Eliane e Paulo Entregas Expressas ME	Visualizar Editar Remover

GPet
Sistema de Gestão de Petshops e Clínicas Veterinárias
Versão: 1.0.0
Ref: Jul/2020

Faculdade de Tecnologia de Americana - Ministro Ralph Biasi
Rua: Emílio de Menezes
Bairro: Vila Amorim
Cidade: Americana - SP
CEP: 13469-111

Google Maps

Fonte: Autoral (2020).

5.16 Características da Tela Gerenciamento de Produtos

A tela gerenciamento de produtos encontra-se em Produtos>Gerenciar Produtos. É possível visualizar uma prévia das informações de Produtos cadastrados além de realizar pesquisas por Código de Identificação e Detalhes do Produto.

Clicando em Adicionar Novo Produto, será carregada a página de cadastro correspondente, vale ressaltar que para cadastrar um novo produto basta informar o código de identificação da categoria, a seção 5.1.1 exibe o layout de cadastro genérico utilizado, assim como clicando em Visualizar será carregada a página como representado na seção 5.1.2.

Figura 69 – Gerenciamento de Representantes

The screenshot shows the GPet software interface for managing products. At the top, there's a decorative banner featuring various small dog breeds. The main header has the GPet logo on the right and navigation links like 'Início > Produtos'. Below the header is a dark green navigation bar with the word 'Produtos'. Underneath is a white search and filter section with a 'Cadastrar Novo' button, a search input field, a 'Pesquisar' button, and a 'Todos' button. The main content area is a table with columns for '#', 'Detalhes do Produto', and 'Action'. One row is visible, showing product number 36 and the name 'Roupinha Dourada Tamanho P', with a 'Visualizar' button in the Action column. In the bottom left corner, there's a sidebar with the GPet logo, system version information ('Versão 1.0.0 Ref: Jul/2020'), and a 'Faculdade de Tecnologia de Americana - Ministro Ralph Biasi' section with address details. On the right side, there's a 'Google Maps' integration showing a location pin for 'Fatec Americana Faculdade de... Americana - SP'.

Fonte: Autoral (2020).

5.17 Características da Tela de Estoque

A tela de estoque encontra-se em Produtos>Inventario. É possível visualizar uma prévia das informações de Estoque cadastrados além de realizar pesquisas por Detalhes do Produto, Quantidade e Última atualização.

Clicando em Visualizar será carregada a página como representado na seção 5.1.2.

Figura 70 – Estoque

#	Detalhes do Produto	Quantidade	Última atualização	Action
8	Shampoo Neutro	2	2020-07-10 16:40:35	Visualizar
9	Tela de Proteção pro Gatos	0	2020-07-10 23:48:18	Visualizar
11	Laço de Orelha	1	2020-07-10 23:48:07	Visualizar
13	Teste	0	2020-07-10 23:30:41	Visualizar
14	Roupinha Dourada Tamanho P	1	2020-07-11 13:28:52	Visualizar

GPet
Sistema de Gestão de Petshops e Clínicas Veterinárias
Versão: 1.0.0
Ref: Jul/2020

Faculdade de Tecnologia de Americana -
Ministra Ralph Biasi
Rua: Emílio de Menezes
Bairro: Vila Amorim
Cidade: Americana - SP
CEP: 13469-III

Google Maps
Visualizar mapa ampliado
Fatec Americana - Faculdade de Temporaneamente
Faculdade de Santa
© GPet 2020 / Desinged by JMR

Fonte: Autoral (2020).

5.18 Características da Tela Gerenciamento de Serviços

A tela gerenciamento de Serviços encontra-se em Serviços>Gerenciar Serviços. É possível visualizar uma prévia das informações dos Serviços cadastradas além de realizar pesquisas por Código de Identificação, Nome e Departamento.

Clicando em Adicionar Novo Serviço, será carregada a página de cadastro correspondente , a seção 5.1.1 exibe o layout de cadastro genérico utilizado, assim como clicando em Visualizar, Editar e Remover serão carregadas as páginas como representado nas seções 5.1.2, 5.1.3 e 5.1.4.

Figura 71 – Gerenciamento de Serviços

#	Nome	Departamento	Action
3	Banho	Estética - Banho	Visualizar Editar
4	Checagem de Rotina	Veterinária - Clínico Geral	Visualizar Editar
5	Tosa	Estética - Tosa	Visualizar Editar
6	Castração Gato (Fem)	Veterinária - Clínico Geral	Visualizar Editar
7	Castração Gato (Mas)	Estética - Banho	Visualizar Editar

GPet
Sistema de Gestão de Petshops e Clínicas Veterinárias
Versão: 1.0.0
Ref: Jul/2020

Faculdade de Tecnologia de Americana - Ministro Ralph Biasi
Rua: Emílio de Menezes
Bairro: Vila Amorim
Cidade: Americana - SP
CEP: 13469-III

Google Maps

Fonte: Autoral (2020).

5.19 Características da Tela Gerenciamento de Vendas

A tela gerenciamento de Vendas encontra-se em Vendas>Gerenciar Vendas.

É possível visualizar uma prévia das informações dos Vendas em andamento além de realizar pesquisas por Código de Identificação, Código do Cliente, Nome e Valor.

Clicando em Nova Venda, será carregada a página de cadastro como representado na seção 5.1.1, assim como clicando em Visualizar seção 5.1.2 ou em Finalizar Atendimento seção 5.1.5.

Figura 72 – Gerenciamento de Vendas

#	Código do Cliente	Nome	Valor	Action
13	7	Lorena Melo	116.00	Visualizar Produtos Adquiridos Finalizar Venda

Fonte: Autoral (2020).

Ao clicar em Produtos Adquiridos o usuário é levado para uma tela que permite a visualização de todos os produtos adquiridos na Venda em questão, podendo também adicionar, visualizar e remover produtos ao carrinho.

Figura 73 – Gerenciamento Carrinho

#	Código do Produto	Descrição	Preço	Data	Action
81	33	Laço de Orelha	16.50	2020-07-10 23:17:36	Visualizar Remover
82	34	Laço de Orelha	16.50	2020-07-10 23:18:02	Visualizar Remover
85	29	Tela de Proteção pra Gatos	50	2020-07-10 23:48:18	Visualizar Remover
83	35	Laço de Orelha	16.50	2020-07-10 23:18:07	Visualizar Remover
84	35	Laço de Orelha	16.50	2020-07-10 23:48:07	Visualizar Remover

GPet
Sistema de Gestão de Petshops e Clínicas Veterinárias
Versão: 1.0.0
Ref: Jul/2020

Faculdade de Tecnologia de Americana –
Ministro Ralph Biasi
Rua: Emílio de Menezes
Bairro: Vila Amorim
Cidade: Americana – SP
CEP: 13469-111

Google Maps

Fonte: Autoral (2020).

5.20 Características da Tela Histórico de Vendas

A tela Histórico de Vendas encontra-se em Vendas >Histórico de Vendas.

É possível visualizar uma prévia das informações das Vendas finalizadas, além de realizar pesquisas por Código de Identificação, Descrição, Valor e Data de Registro.

Clicando em Visualizar será carregada a página de Visualização como representado na seção 5.1.2.

Figura 74 – Histórico de Transações

#	Descrição	Valor	Data	Action
27	Atendimento realizado no Animal: 7	225.00	2020-07-09 17:03:38	Visualizar
28	Atendimento realizado no Animal: 8	125.00	2020-07-09 17:04:21	Visualizar
31	Atendimento realizado no Animal: 7	125.00	2020-07-10 15:29:00	Visualizar
32	Compra de produtos do fornecedor: 2	1000	2020-07-10 16:11:07	Visualizar
33	Venda para o Cliente: 5	82	2020-07-10 16:40:22	Visualizar
34	Atendimento realizado no Animal: 9	450.55	2020-07-10 19:26:53	Visualizar
37	Venda para o Cliente: 7	116.00	2020-07-10 23:09:49	Visualizar

GPet Sistema de Gestão de Petshops e Clínicas Veterinárias Versão: 1.0.0 Ref: Jul/2020

Faculdade de Tecnologia de Americana - Ministro Ralph Biasi Rua: Emilio de Menezes Bairro: Vila Amorim Cidade: Americana - SP CEP: 13469-111

Google Maps

Fonte: Autoral (2020).

6. CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo a finalização da disciplina de Laboratório de Engenharia de Software ministrada pelo Prof. Dr. Ivan Menerval da Silva docente da Faculdade de Tecnologia de Americana (FATEC-AM).

A princípio o objetivo do trabalho era colocar em prática a teoria da Engenharia de Software e posteriormente desenvolver a prototipação(telas) do sistema, porém ao decorrer do desenvolvimento do projeto foi identificada a possibilidade de realizar o desenvolvimento completo do sistema, ou seja, torna-lo funcional.

No decorrer do desenvolvimento do sistema houveram algumas dificuldades pontuais mas que foram solucionadas mediante trabalho em equipe.

A interface do usuário foi construída a fim de manter a experiência do mesmo fácil e simples de se utilizar, seguindo os padrões de *bootstrap*.

O software cumpre o que promete, com base nos pontos levantados na etapa inicial do projeto mas também melhorando problemas lógicos e de usabilidade encontrados durante o desenvolvimento, o que posteriormente resultou em dois diagramas de caso de uso para que fosse possível identificar a diferença entre o projeto e o desenvolvimento.

Conclui-se portanto que mediante a solicitação feita durante as aulas, o projeto e sistema aqui apresentado aborda todos os requisitos e aspectos necessários além de poder servir como base de desenvolvimento para outros projetos mediante sua clara modularidade.

REFERÊNCIAS

BOOTSTRAP. **What is Bootstrap.** W3SCHOOLS Web Development What is Bootstrap, 2020. Disponível em : https://www.w3schools.com/whatis/whatis_bootstrap.asp Acesso em: 05 jul. 2020.

BOOTSTRAPIOUS. **Bootstrap Sidebar Tutorial.** Bootstrapious bootstrap sidebar, 2018. Disponível em : < <https://bootstrapious.com/p/bootstrap-sidebar> > Acesso em: 19 abr. 2020.

CSS. **O que é CSS?.** Tecmundo, 2009. Disponível em : < <https://www.tecmundo.com.br/programacao/2705-o-que-e-css-.htm> > Acesso em: 13 jul. 2020.

HEIDISQL. **What's this.** HeidSQL Home, 2020. Disponível em : < <https://www.heidisql.com/> > Acesso em: 27 jun. 2020.

HTML. **O básico: O que é HTML?.** Tableless o que é HTML, 2020. Disponível em : < <https://tableless.com.br/o-que-html-basico/> > Acesso em: 13 jul. 2020.

MDN. **Java Script.** MDN Web Docs Java Script, 2020. Disponível em : < <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript> > Acesso em: 27 jun. 2020.

MYSQL. **About MySQL.** About MySQL, 2020. Disponível em : < <https://www.mysql.com/about/> > Acesso em: 27 jun. 2020.

PHP. **O que é PHP?.** PHP Documentation, 2020. Disponível em : < https://www.php.net/manual/pt_BR/intro-whatis.php > Acesso em: 27 jun. 2020.

SOMMERVILLE, I. **Engenharia de Software.** 8. Ed. São Paulo: Pearson Addison-Wesley, 2007.

VISUAL STUDIO CODE. **Getting Started.** Visual Studio Code Docs, 2020. Disponível em : < <https://code.visualstudio.com/docs> > Acesso em: 27 jun. 2020.

XP. **Conceitos e Práticas sobre eXtreme Programming.** DevMedia, 2006. Disponível em : < <https://www.devmedia.com.br/extreme-programming-conceitos-e-praticas/1498/> > Acesso em: 17 jul. 2020.