



**UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DE CASTANHAL
FACULDADE DE COMPUTAÇÃO
BACHARELADO EM SISTEMAS DE INFORMAÇÕES**

HEITOR MONTEIRO DA SILVA SOBRAL

**SISTEMA INTEGRADO DO HOSPITAL VETERINÁRIO DA
UFPA CAMPUS CASTANHAL**

CASTANHAL — PA

2019



**UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DE CASTANHAL
FACULDADE DE COMPUTAÇÃO
BACHARELADO EM SISTEMAS DE INFORMAÇÕES**

HEITOR MONTEIRO DA SILVA SOBRAL

**SISTEMA INTEGRADO DO HOSPITAL VETERINÁRIO DA
UFPA CAMPUS CASTANHAL**

Trabalho de Conclusão de Curso apresentado
para obtenção do grau de Bacharel em Sistemas
de Informações.

Orientador: Prof. Dr. Marcos Cesar da Rocha
Seruffo

Coorientador: Prof. Dr. Tássio Costa de Carva-
lho

**CASTANHAL — PA
2019**

Sobral, Heitor Monteiro da Silva

SISTEMA INTEGRADO DO HOSPITAL VETERINÁRIO DA UFPA CAMPUS
CASTANHAL/ HEITOR MONTEIRO DA SILVA SOBRAL. – CASTANHAL — PA,
2019.

91 p. : il. color.

Orientador: Prof. Dr. Marcos Cesar da Rocha Seruffo

Coorientador: Prof. Dr. Tássio Costa de Carvalho

Monografia – UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DE CASTANHAL
BACHARELADO EM SISTEMAS DE INFORMAÇÕES, 2019.

1. Desenvolvimento. 2. Software Web. 3. Veterinária. I. Título.

HEITOR MONTEIRO DA SILVA SOBRAL

**SISTEMA INTEGRADO DO HOSPITAL
VETERINÁRIO DA UFPA CAMPUS CASTANHAL**

Trabalho de Conclusão de Curso apresentado
para obtenção do grau de Bacharel em Sistemas
de Informações.

Data da Defesa: 18 de Junho de 2019

Conceito: Excelente

Banca Examinadora

Prof. Dr. Marcos Cesar da Rocha Seruffo

Faculdade de Computação - UFPA

Orientador

Prof. Dr. Tássio Costa de Carvalho

Faculdade de Computação - UFPA

Coorientador

Prof. Dr. José Jailton Henrique Ferreira Junior

Faculdade de Computação - UFPA

Membro da Banca

Prof. Dr. Danilo Ferreira Rodrigues

Faculdade de Medicina Veterinária - UFPA

Membro da Banca

CASTANHAL — PA

2019

AGRADECIMENTOS

A Deus por ter me dado saúde, força e persistência para transpor desafios.

A Universidade Federal do Pará, pois, esse lugar me proporcionou diferentes experiências e saberes que levarei por toda vida.

A todos os professores da faculdade de computação que contribuirão de diferentes formas no meu aprendizado.

Ao professor Marcos Cesar da Rocha Seruffo que me encaminhou este tema, e propiciou recursos que me ajudaram de maneira substancial.

Ao pessoal do Hospital veterinário da UFPA de Castanhal, por ceder seu tempo e conhecimento para a construção deste trabalho, e oportunidade para vivenciar experiências que contribuirão substancialmente para minha carreira profissional.

A avó Heloísa Monteiro da Silva e mãe Hércia Helene Monteiro da Silva, ambas sempre atenciosas e carinhosas comigo, realizando tudo ao alcance para ajudar a percorrer minhas jornadas. Sou grato a elas por me ensinarem a escolher aquilo que é certo, e ignorar o errado, devo muito às duas.

A namorada Maiara Meireles Reis, por toda sua dedicação, apoio e incentivos para minha vida acadêmica, pessoal e profissional.

Aos colegas do Laboratório de Desenvolvimento de Sistemas, por partilhar comigo momentos de descontração e conhecimento.

*“Sozinhos podemos ver pouco do futuro,
porém o suficiente para darmos conta
de que há muito que se fazer.”
(Alan Turing)*

RESUMO

A realização desta monografia compreende a apresentação do processo de desenvolvimento adotado para construir uma proposta de *software-web* de uso direcionado ao Hospital Veterinário — HV — da Universidade Federal do Pará — UFPA — de Castanhal. Para isso, agregando na aplicação conhecimento técnico relativo ao cotidiano desempenhado na clínica veterinária. Naturalmente, a realização deste tipo trabalho demanda ao desenvolvedor de sistemas, adentrar em conceitos que pertencem a outra área do conhecimento. Com isso, de forma substancial demandando ao projeto aplicar a troca de experiências entre profissionais de medicina veterinária e da Tecnologia da Informação e Comunicação. Durante o projeto, a absorção de tais informações viabilizaram-se através de competências adquiridas durante o curso de Bacharelado em Sistemas de Informações oferecido pela UFPA de Castanhal. Dentre as capacitações utilizadas, citamos: programação de computadores; engenharia de *software*; interação humano e computador; banco de dados; entre outros. Ao final do trabalho é apresentado como resultado: as publicações provenientes deste projeto; telas de usuário desenvolvidas no sistema, e *link* de acesso ao código-fonte da aplicação construída.

Palavras-chave: Desenvolvimento; *software-web*; veterinária.

ABSTRACT

The accomplishment of this monograph includes the presentation of the development process adopted to construct a proposal of software-web of use directed to the Veterinary Hospital — HV — of the Federal University of Pará — UFPA — of Castanhal. For this, adding in the application technical knowledge related to the daily life performed in the veterinary clinic. Of course, the realization of this kind of work demands the systems developer to enter into concepts that belong to another area of knowledge. With this, in a substantial way demanding the project to apply the exchange of experiences between veterinary medicine professionals and Information and Communication Technology. During the project, the absorption of such information was made possible through the skills acquired during the Bachelor's Degree in Information Systems offered by UFPA de Castanhal. Among the capacities used, we mention: computer programming; Software Engineering"; human and computer interaction; database; among others. At the end of the work is presented as a result: the publications coming from this project; user screens developed in the system, and link access to the source code of the application built.

Keywords: Development; software-web; veterinary.

LISTA DE ILUSTRAÇÕES

Figura 1 – Ciclo de vida de requisições JSF	23
Figura 2 – Processo de execução multiplataforma de software java	27
Figura 3 – Utilização de <i>branches</i> para isolar módulos em desenvolvimento	34
Figura 4 – Dinâmica de trabalho com ferramenta <i>git desktop</i>	35
Figura 5 – Processo de desenvolvimento	38
Figura 6 – Diagrama de Entidade Relacionamento final	46
Figura 7 – DER responsável por armazenar dados de pessoa	47
Figura 8 – DER responsável por armazenar dados de usuários do sistema	47
Figura 9 – DER responsável por armazenar o endereço de pessoas	48
Figura 10 – DER responsável por armazenar dados de animais	49
Figura 11 – DER responsável por armazenar dados de agendamentos de consultas	49
Figura 12 – DER responsável por armazenar dados básicos de consultas	50
Figura 13 – DER responsável por armazenar dados de exames específicos	51
Figura 14 – DER responsável por armazenar dados de procedimentos do HV	51
Figura 15 – Caso de uso: usuários do sistema	52
Figura 16 – Caso de uso: clientes do HV	52
Figura 17 – Caso de uso: login e logout do sistema	53
Figura 18 – Caso de uso: funcionalidades de agenda	53
Figura 19 – Caso de uso: nova consulta	54
Figura 20 – Caso de uso: funcionalidades de exame físico específico	54
Figura 21 – Caso de uso: funcionalidades de exame por imagem	55
Figura 22 – Caso de uso: funcionalidades de procedimentos aplicados	55
Figura 23 – Caso de uso: funcionalidades de radiografias pendentes	56
Figura 24 – Caso de uso: funcionalidades para controle de fichas e entrada de caixa	56
Figura 25 – Rede cliente-servidor	57
Figura 26 – Padrão arquitetural em 3 camadas	58
Figura 27 – Diagrama de pacotes do sistema	58
Figura 28 – Tela de <i>login</i>	62
Figura 29 – Tela principal	62
Figura 30 – Telas para cadastrar usuários do sistema	63
Figura 31 – Telas para finalizar cadastro de usuário do sistema	64
Figura 32 – Telas para editar dados de usuários do sistema	65
Figura 33 – Tela para editar ou criar nova categoria de peso	66
Figura 34 – Tela para editar ou criar novos procedimentos	66
Figura 35 – Telas para agendar, confirmar e cancelar agendamentos de consultas	67
Figura 36 – Telas para cadastrar/identificar proprietário	68
Figura 37 – Telas para cadastrar ou selecionar animal pequeno	69

Figura 38 – Telas para cadastrar ou selecionar animal pequeno	70
Figura 39 – Telas para buscar agendamentos confirmados e cancelados	70
Figura 40 – Telas para buscar agendamentos confirmados e cancelados	71
Figura 41 – Telas de controle de fichas e entrada de caixa	71
Figura 42 – Telas para buscar e editar dados de clientes	72
Figura 43 – Tela para selecionar agendamento confirmado	73
Figura 44 – Tela com formulário de exame físico geral	74
Figura 45 – Exemplo de campo reativo a resposta	74
Figura 46 – Tela com formulário de anamnese	75
Figura 47 – Tela com formulários de exame físico específico	76
Figura 48 – Tela para solicitar exame por imagem	77
Figura 49 – Tela com formulário de diagnósticos diferenciais	77
Figura 50 – Tela para selecionar procedimentos aplicados na consulta	78
Figura 51 – Tela para finalizar consulta e confirmar identidade do veterinário	79
Figura 52 – Tela de nova radiografia	79
Figura 53 – Tela de nova radiografia	80
Figura 54 – Tela de erro 500 do sistema	80
Figura 55 – Desenvolvedores do projeto	81

LISTA DE QUADROS

Quadro 1 – Escopo das <i>Managed Beans</i>	25
Quadro 2 – Exemplo de anotações utilizadas em ORM	26
Quadro 3 – Valores para “hibernate.hbm2ddl.auto”	30
Quadro 4 – Procedimentos do “Hibernate Tools”	31
Quadro 5 – Funcionalidade de comandos SQL	33
Quadro 6 – Itens investigados durante o projeto	40
Quadro 7 – Requisitos de proprietário e usuário	41
Quadro 8 – Requisitos de animais	42
Quadro 9 – Requisitos para agendar consultas	42
Quadro 10 – Requisitos para consulta	43
Quadro 11 – Requisitos de exame por imagem	44
Quadro 12 – Requisitos para procedimentos do HV	44
Quadro 13 – Requisitos para controle de fichas e entrada de caixa	45
Quadro 14 – Publicações provenientes do SIHV	60
Quadro 15 – Principais Desafios e Recomendações da Interdisciplinaridade no Desenvolvimento de Sistemas	61

LISTA DE ABREVIATURAS E SIGLAS

A	Alta
Ajax	Asynchronous JavaScript and XML
API	Application Programming Interface
B	Baixa
BD	Banco de Dados
BSI	Bacharel em Sistemas de Informações
CRUD	Create, Read, Update, and Delete
CSS	Cascade Style Sheet
DAO	Data Access Object
DCL	Data Control Language
DDL	Data Definition Language
DER	Diagrama de Entidade e Relacionamento
DML	Data Manipulation Language
DNS	Domain Name System
EAR	Enterprise ARchive
EJB	Enterprise Java Beans
EL	Expression Language
F	Funcional
GP	Grande Porte
HQL	Hibernate Query Language
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
HV	Hospital Veterinário
IDE	Integrated Development Environment

IP	Internet Protocol
JAR	Java Archive
JCP	Java Community Process
JDBC	Java Database Connectivity
JDK	Java Development Kit
JEE	Java Enterprise Edition
JME	Java Micro Edition
JPA	Java Persistence API
JRE	Java Runtime Environment
JSE	Java Platform Standard Edition
JSF	Java Server Faces
JSP	Java Server Pages
JSR	Java Specification Requests
JVM	Java Virtual Machine
LADES	Laboratório de Desenvolvimento de Sistemas
Ltda	Limitada
M	Media
MP	Médio Porte
N.D.N.	Nada digno de nota
NF	Não Funcional
N:N	Muitos-para-muitos
ORM	Object Relational Mapping
PHP	acrônimo recursivo para: Hypertext Preprocessor
POM	Project Object Model
POO	Programação Orientada a Objeto
PP	Pequeno Porte

PRaP	Partial Rendering and Processing
SaaS	Software as a Service
SGBD	Sistema Gerenciador de Banco de Dados
SGHV	Sistema Gerenciador de Hospitais Veterinários
SIHV	Sistema Integrado do Hospital Veterinário
SQL	Structured Query Language
TIC	Tecnologia da Informação e Comunicação
TLS	Transport Layer Security
UFPA	Universidade Federal do Pará
UI	User Interface
URL	Uniform Resource Locator
WAR	Web Application Resource
XHTML	Extensible HyperText Markup Language
XML	eXtensible Markup Language

SUMÁRIO

1	INTRODUÇÃO	16
1.1	Objetivos	17
1.2	Justificativa	18
1.3	Estrutura do Trabalho	18
2	TRABALHOS CORRELATOS	19
2.1	Sistema gerenciador de hospitais veterinários	19
2.2	Aplicação de inspeção zootécnica	19
2.3	InfoPet	20
2.4	SimplesVet	20
3	TECNOLOGIAS E FERRAMENTAS	22
3.1	Java Enterprise Edition	22
3.1.1	Java Server Faces 2.2	22
3.1.2	Enterprise Java Beans	25
3.1.3	Java Persistence API	25
3.2	Java Development Kit 8	27
3.3	Maven para o controle de bibliotecas	28
3.4	Primefaces	29
3.5	Hibernate	30
3.6	Ferramentas de codificação	31
3.6.1	Netbeans IDE	31
3.6.2	MySQL	32
3.7	Gerência de configuração	33
3.7.1	GitHub	33
3.7.2	Gitkraken	35
3.8	Servidor de aplicações Glassfish	36
4	DESENVOLVIMENTO DA APLICAÇÃO	37
4.1	Etapas do desenvolvimento da aplicação	38
4.2	Itens investigados e requisitos elicitados	40
4.3	Modelagem do sistema	46
4.3.1	Banco de dados	46
4.3.2	Diagramas de caso de uso	52
4.4	Arquiteturas utilizadas	56
4.4.1	Cliente/Servidor	57
4.4.2	Modelo-Controlé-Visão	57
4.4.3	Organização interna de arquivos do SIHV	58
5	PUBLICAÇÕES	60
6	SISTEMA DESENVOLVIDO	62

6.1	Código-fonte da aplicação	81
7	CONSIDERAÇÕES FINAIS	82
	REFERÊNCIAS	84
	 ANEXOS	 88
	ANEXO A – FICHA DE CONSULTAS DO HV	89
	ANEXO B – CONTROLE DE FICHAS E ENTRADA DE CAIXA . . .	90
	ANEXO C – TABELA DE PROCEDIMENTOS DO HV	91

1 INTRODUÇÃO

A informatização com base em pensamentos de Melo e Leitão (2010), consiste no ato ou efeito de implementar recursos computacionais para otimizar a coleta, processamento e utilização de informações inserida em um dado contexto, e segundo reflexões sobre a obra de Sommerville (2011). O recurso se faz cada vez mais onipresente nos mais diferentes aspectos da realizada humana, apresentando difícil dissociação da mesma conforme visão do autor, e de acordo com dados obtidos em CNI (2018). Percebe-se no cenário da indústria brasileira, cerca de 73% das grandes empresas relatam já utilizar tais recursos, e 48% das instituições entrevistadas alegam planos para investir em inovação tecnológica, devido ao resultado proporcionado pela utilização de soluções antecessoras.

Outro exemplo da presença de meios tecnológicos é relatado por Bernardi et al. (2014), situando-se na agricultura de precisão, no qual se utiliza a análise estatística para melhor aplicar fatores que norteiam o cultivo de uma cultura, com isso, viabilizando seu aumento/sucesso de produtividade. Na medicina segundo Eveleth (2014), a telecirurgia promove o estreitamento geográfico entre médico e paciente, viabilizando a realização de procedimentos médicos, em locais que o profissional mais adequando para a tarefa, não pode estar presencialmente. Por fim, diferentes seguimentos de mercado, mídia de entretenimento e informação usufruem da inteligência de conteúdo segundo a fonte hawkeye (2018), para cada vez mais fornece um produto/serviço direcionado ao dia-a-dia do espectador/leitor.

Percebe-se então como fator em comum, o empenho de diferentes áreas do conhecimento buscando o aprimoramento proporcionado pela informatização. Inerente a esta ação, temos a utilização de recursos que englobam a Tecnologia da Informação e Comunicação — TIC, dentre os quais, emprega-se o *hardware* como primeiro item para executar ferramentas desenvolvidas em *softwares*. Este como segundo item, combinado ao desenvolvimento gerenciado, e somando sua característica intangível e sem restrição física quanto ao seu uso para moldar soluções/algoritmos, tona-se o meio flexível na satisfação de distintas necessidades (SOMMERVILLE, 2011).

A construção de sistemas para os mais diferentes aspectos de público-alvo, requisita aos seus desenvolvedores, a imersão em outras áreas do conhecimento. Ação que propicia a comunicação entre profissionais de distintos campos de atuação, ambos, trabalhando em conjunto para compreender as reais necessidades de usuários diretos e indiretos do sistema, também denominados *stakeholders*. O esforço aplicado tem a finalidade de projetar e dispor ferramentas que correspondam aos objetivos de tais indivíduos, em contrapartida, essa característica exige maior complexidade no processo de desenvolvimento de *softwares* (SOMMERVILLE, 2011; SOBRAL; SOUZA; SERUFFO, 2018).

Abranger outra área de estudo consiste não apenas na análise de processos e artefatos que norteiam as partes interessadas em implementar a informatização, mais sim a obtenção e entendimento de sua visão de negócio, para agregar ao processo de concepção do sistema sob me-

didada. No decorrer disso, aplicam-se técnicas para intermediar a troca e absorção de informações entre desenvolvedor e cliente (SOMMERVILLE, 2011; SOBRAL; SOUZA; SERUFFO, 2018).

Percebe-se então que profissionais de TIC transpõem diferentes cenários, atuando como otimizadores de processos, mediante utilização de capacitações adquiridas em seus campos de estudo, dos quais citamos: engenharia de *softwares*; teoria geral dos sistemas de informação; programação de computadores; interação humano-computador, entre outros (SOMMERVILLE, 2011; SOBRAL; SOUZA; SERUFFO, 2018).

Partindo das explanações supracitadas, este trabalho envolve investigações/observações *in loco* no Hospital Veterinário — HV da Universidade Federal do Pará — UFPA, campus 2 de Castanhal. Do qual elencar através de requisitos de *software*, processos essenciais ao cotidiano de atividades da clínica veterinária, e identificando problemáticas provenientes da ausência ou incompleta implementação de recurso informático em suas instalações.

Dentre o conjunto de questões encontradas no HV, o principal situa-se na manipulação manual de registros utilizados durante o expediente da clínica, tarefa que envolve diariamente atenção para: preencher corretamente; buscar e anexar laudas referentes ao mesmo cliente; guardá-los no arquivo e organizar tais documentos em série. Esse manuseio diário abre oportunidades para outras irregularidades ocorrerem. O exemplo disso seria a perda de informações, podendo variar entre algum dado do cliente durante o cadastro. Ausência de carimbo ou assinatura para atestar pagamento/consentimento de algum procedimento, ou em processo que demande o manejo do registro, viabilizando o sumiço temporário ou não de partes no documento, ou o histórico completo.

1.1 Objetivos

Tomando como base as problemáticas citadas acima, o objetivo geral abordado neste trabalho é expor o processo de desenvolvimento e implementação de uma aplicação *web* denominada Sistema Integrado do Hospital Veterinário — SIHV, direcionada ao gerenciamento e otimização de atividades realizadas no HV da UFPA de Castanhal. Utilizando para isso, competências adquiridas no curso de Bacharel em Sistemas de Informações — BSI, para construir uma proposta de sistema que disponha ferramentas para informatizar a busca, criação, organização e utilização de dados essenciais em processos do HV. Abaixo, é listado os objetivos específicos a serem alcançados com este projeto.

1. Abordar no trabalho o referencial teórico e prático necessário para a construção deste projeto, dentre os quais citamos: trabalhos/soluções correlatas; documentação referente a tecnologias utilizadas no desenvolvimento da aplicação *web*.
2. Gerar publicação de artigo científico utilizando a experiência e material adquirido no processo de desenvolvimento do projeto.

3. Desenvolver no sistema ferramentas de registro referente ao dados de: agenda de consultas; proprietários/tutores de animais; cães e gatos; anamnese; exame físico geral; diagnósticos diferenciais; exames específicos do sistema digestório e glândulas anexas, respiratório e cardiovascular, gênito-urinário e glândulas mamárias, tegumentar, neurológico, oftálmico e músculo-esquelético;
4. Disponibilizar em plataforma de hospedagem de código-fonte, os *scripts* desenvolvidos durante o processo de construção da aplicação veterinária.
5. Apresentar funcionalidades desenvolvidas no sistema, mediante demonstração de *interfaces* de usuário.

1.2 Justificativa

A construção deste trabalho busca atender uma demanda direcionada ao Laboratório de Desenvolvimento de Sistemas — LADES da UFPA campus 1 de Castanhal, para desenvolver um *software* veterinário que disponibilize mediante suas funcionalidades, ferramentas para otimizar e mitigar problemáticas provenientes da ausência de informatização no HV.

O desenvolvimento deste trabalho também é uma oportunidade para aplicar o conhecimento prático e teórico em ambiente norteado por necessidades reais de uma instituição, no que se refere a construção de sistemas, tornando-se uma experiência substancial na formação do profissional em BSI.

1.3 Estrutura do Trabalho

A estrutura deste trabalho é composta 7 capítulos, iniciando com a introdução, o segundo levanta a descrição de trabalhos correlatos, e apresenta características de *softwares* com similaridade na problemática a ser resolvida neste projeto, ou tecnologia utilizada na construção de tais sistemas. O capítulo 3 aborda peculiaridades de recursos utilizados no desenvolvimento prático do projeto, para isso, consultando documentações e autores que expliquem informações referentes a ferramentas e tecnologias implementadas durante o trabalho.

O capítulo 4 inicia explicando a metodologia de desenvolvimento de *software* empregada no SIHV, e aborda as etapas do processo de construção, mediante descrição de suas atividades. Junto a isso se apresenta artefatos que: auxiliaram na concepção do sistema fornecendo informações para o rastreo e delimitação de objetivos; descrevem a estrutura interna da aplicação, e ilustram seu funcionamento e organização. A seguir no capítulo 6, apresenta-se resultados obtidos no decorrer do projeto, dentre os quais citamos publicações, *interfaces* do sistema e *link* para o código-fonte da aplicação. Por fim, finalizando com a exposição da conclusão do projeto.

2 TRABALHOS CORRELATOS

O capítulo descreve características relevantes de quatro *softwares*, sendo eles provenientes de monografias ou de uso comercial, variando entre as plataformas *web* ou *desktop*, cada um deles apresentando similaridade com a proposta deste trabalho.

2.1 Sistema gerenciador de hospitais veterinários

Consistindo num trabalho de conclusão de curso, do qual aplica-se o desenvolvimento de um *software web* para o gerenciamento de processos em hospitais de medicina veterinária, realizando o: controle de consultas efetuadas; cadastro de pacientes (animais), proprietários e funcionários; e relatório das atividades desempenhadas. Objetivando fornecer uma correta ergonomia e apresentação de informações relevantes aos utilizadores do software (AGUIAR, 2015).

A construção do Sistema Gerenciador de Hospitais Veterinários — SGHV embasa-se no paradigma de Orientação a Objetos, sendo este conceito mantido durante: o levantamento e análise de requisitos; modelagem e desenvolvimento da aplicação; e implementação do *software*. O projeto utiliza a linguagem de programação Java, empregando o *framework* Java Server Faces 2.2 — JSF, recurso que possibilita utilizar o Java em projetos na plataforma *web*, somado a isto, utiliza-se a biblioteca de componentes pré-programados PrimeFaces 4.0 (AGUIAR, 2015).

A modelagem e gerenciamento do Banco de Dados — BD foi realizada no *software* PostgreSQL 9.1, e utiliza uma *interface* de programação de aplicações (API, do inglês *Application Programming Interface*) denominada Java Persistence API para realizar a persistência de informações no BD (AGUIAR, 2015).

2.2 Aplicação de inspeção zootécnica

Tratando-se também de um *software* construído com a tecnologia Java, porém, utilizando a plataforma Java Micro Edition — JME que disponibiliza recursos/ferramentas que auxiliem no desenvolvimento de aplicações para dispositivos com menor poder de processamento disponível no *hardware*, do qual podemos citar: celulares; *smartphone*; e tablets (KLINGER, 2014).

A aplicação móvel tem como finalidade dar suporte na execução em inspeções zootécnica para uma associação de criadores da raça bovina Crioula Lageana, procedimento realizado durante o cadastro de novos animais para então registrar/consultar *in loco* o histórico de cruzamentos e a árvore genealógica do animal (KLINGER, 2014).

O *software* tem como propósito garantir a integridades das informações cadastradas/consultadas através da sistematização/padronização de processos que anteriormente realizavam o controle por

planilhas em manuscrito, o que possibilita a inserção/compreensão erradas de dados utilizados durante e posteriormente em novos cadastros (KLINGER, 2014).

O aplicativo consiste na continuidade de um trabalho anterior, atuando como módulo adicional para um *software desktop*, visando aplicar a sincronização de informações presentes na base de dados compartilhada entre versão móvel e *desktop* do *software*, com isso, mantendo a integridades de informações relevantes aos *stakeholders* (KLINGER, 2014).

2.3 InfoPet

Consiste num *software* comercial desenvolvido pela empresa Infologika com seguimento na gestão de clínicas veterinárias, do qual dispõem uma licença de aquisição com valor único, que possibilita realizar: o cadastro completo de clientes e animais com foto; o registro de exames e laudos efetuados; o agendamento de consultas, vacinações e serviços de tosa e banho em animais domésticos (INFOLOGIKA, 2017; PINTO, 2011).

O *software* foi somente desenvolvido para o sistema operacional Windows e trabalhando sobre a arquitetura de duas camadas intitulada cliente gordo, apresentando duas modalidades de instalação: na primeira temos uma aplicação servidora responsável por administrar e compartilhar o BD em uma rede de computadores com seus terminais (clientes gordos) conectados; o segundo modo concentra nos terminais maior parte das funcionalidade e lógica de negócio do *software* (INFOLOGIKA, 2017; GRANATYR, 2007).

Devido às características da arquitetura utilizada pelo InfoPet, o *software* mantém o armazenamento de informações do BD sobre a responsabilidade do cliente que contratou a licença de uso do programa, por padrão realiza *backups* em dispositivos externos de armazenamento, e opcionalmente o *uploads* de arquivos em nuvem (INFOLOGIKA, 2017).

2.4 SimplesVet

Desenvolvido pela SimplesVet Tecnologia Ltda (Limitada), consiste numa *web-applications* direcionada para a gestão e suporte a lógica de negócio em petshops e clínicas veterinárias, do qual fornece cinco licenças de custo mensal para sua utilização, variando o valor e quantidade de usuários conectados ao sistema, porém, independente do plano escolhido, todas as funcionalidades do sistema vem disponíveis por padrão (SIMPLESVET, 2016).

Devido sua arquitetura, a aplicação abstrai a responsabilidade do usuário/contratante em manter a infraestrutura necessária para deixar a aplicação *web* operante, do qual podemos citar o custo com armazenamento de dados e manutenção do *hardware*, disponibilizando funcionalidade do sistema mediante um *browser*, acesso que independe de instalação, plataforma e assumindo o papel de software como serviço (SaaS, do inglês *software as a service*) (SIMPLESVET, 2016; SOMMERVILLE, 2011).

A aplicação permite a seus utilizadores mantêm o controle de tarefas que demandem periodização, como o histórico de vacinas e agendamento de consultas para clientes, o *software* também otimizando a dinâmica e fluxo de trabalho concentrando na *interface* informações relevantes para os *stakeholders*, favorecendo suporte em análises de caixa e projeções de estratégias que auxiliem na tomada de decisão (SIMPLESVET, 2016).

3 TECNOLOGIAS E FERRAMENTAS

3.1 Java Enterprise Edition

A plataforma Java Enterprise Edition — JEE 7 consiste num conjunto de especificações direcionadas para construção de aplicações corporativas utilizando o Java, contendo implementações para auxiliar desenvolvedores na incorporação simplificada de requisitos não-funcionais necessários à estrutura do sistema, do qual destacamos o gerenciamento de: sessões web; cache de objetos; conexões de Protocolo de Transferência de Hipertexto (HTTP, do inglês *Hyper-Text Transfer Protocol*); *threads*; persistência de dados; balanceamento de carga; entre outros (CAELUM, 2017a; KUROSE JAMES F.; ROSS, 2013).

Aplicações desenvolvidas com JEE podem ou não atuarem como serviços intermediados por tecnologias *web*, então, o padrão também permite a distribuição do *software* através de componentes modulares executados em servidores de aplicações, característica que permite a escalabilidade e manutenibilidade da *web-application*, disposta numa estrutura multi-camada de *software* centrado no servidor, promovendo assim, a adição de novos componente independente da implementação, aspecto que evita *vendor lock-in* (expressão inglesa referente ao aprisionamento tecnológico) (ANDRADE, 2015a; CAELUM, 2017a; ORACLE, 2013a).

A elaboração de novas especificações JSR (Java *Specification Requests*) do JEE conta constantemente com a participação da comunidade Java, sendo este *feedback* facilitado pela Java Community Process — JCP, conjunto de procedimentos formalizados para auxiliar a colaboração das partes interessadas (organizações, fornecedores e indivíduos) no desenvolvimento de novas JSRs para o grupo de especialistas da plataforma (LUCKOW; MELO, 2010; ORACLE, 2013a; ORACLE, 2017b; ORACLE, 2017c).

Dentre as tecnologias/especificações que compõem o acervo de recursos do JEE 7, destacamos o JSF para confecção de páginas web com Java; Enterprise Java Beans — EJB para construção de componente de aplicação servidora; Java Persistence API — JPA para armazenar informações no BD (ANDRADE, 2015a; ORACLE, 2017d).

3.1.1 Java Server Faces 2.2

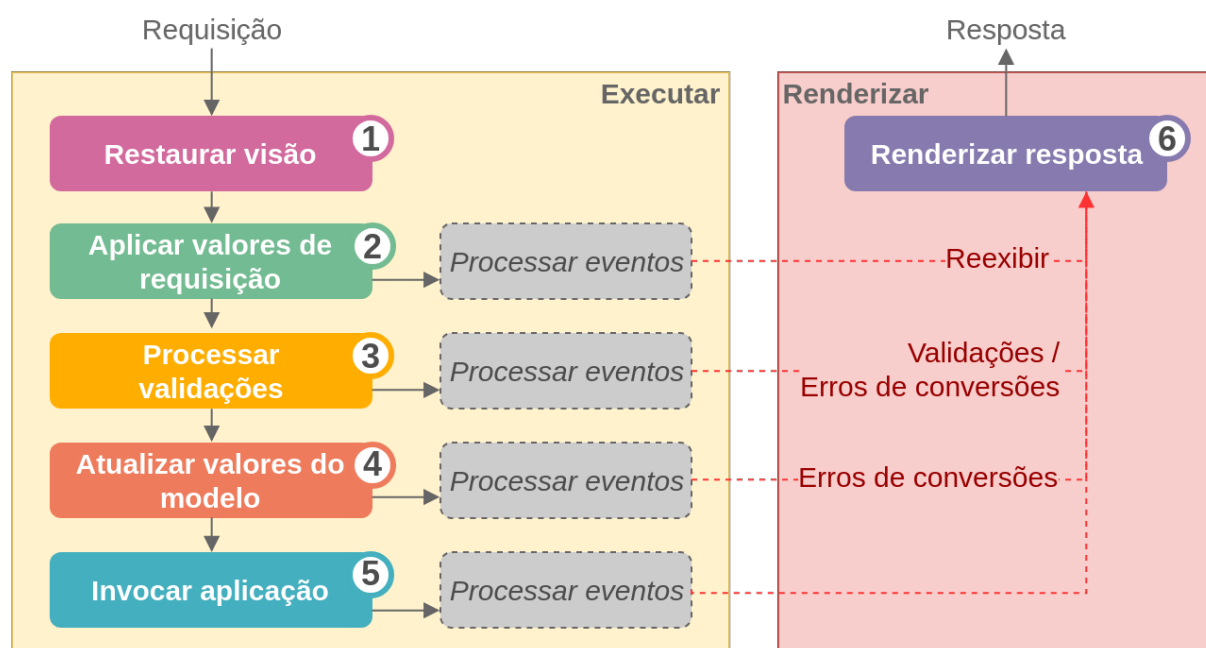
O JSF 2.2 é um *framework* também definido pela JCP com a especificação JSR 344, sendo uma tecnologia de *Interface* de Usuário (UI, do inglês *User Interface*) que dispõe de componentes reutilizáveis para o desenvolvimento de telas em aplicações *web* construídas com Java, a ferramenta demanda para utilizá-lo, uma implementação que disponibilize suas funcionalidades aos programadores (LUCKOW; MELO, 2010; ANDRADE, 2015a; ORACLE, 2017d).

Dentre as implementações disponíveis no mercado, destacamos a biblioteca Mojarra JSF, projeto *open source* mantido pela Oracle e compatível com a especificação JSR 344, recurso que viabiliza a interação do contêiner Java na concepção/conversação de páginas XHTML (*Extensible HyperText Markup Language*), aplicando a separação da linguagem Java de tags HTML (*HyperText Markup Language*), beneficiando a legibilidade e manutenção do código-fonte, característica oposta das Java Server Pages — JSP, tecnologia anterior ao JSF que não aplica essa delimitação (LUCKOW; MELO, 2010; CAELUM, 2017a; ORACLE, 2017e; W3SCHOOLS, 2017a; W3SCHOOLS, 2017b).

Outro benefício consiste no Processamento e Renderização Parcial (PRaP, do inglês *Partial Rendering and Processing*) de componentes disponíveis em tela, isto possibilita direcionar as tarefas processar/executar e renderizar para itens dos quais o utilizador esteja interagindo, dispensando a submissão desnecessária de partes não requisitadas na *interface* ao ciclo de vida JSF, recurso viabilizado pela tecnologia Ajax (*Asynchronous JavaScript and XML*) incorporada ao *framework* desde a versão 2, reconhecendo o(s) componente(s) acionado pela identificação do contêiner que o esteja englobando (ANDRADE, 2015a; ORACLE, 2013f; ÇIVICI, 2018).

O ciclo de vida JSF pode ser visualizado em duas partes, a primeira consiste na execução, compreendendo etapas que objetivam tratar os dados contidos na requisição, a fim de identificar parâmetros que possam comprometer o resultados da pesquisa, a segunda é a renderização, responsável pela entrega da resposta proveniente da execução ao *host* requisitante, todo o procedimento está contido num processo de 6 etapas demonstrado na Figura 1 (LUCKOW; MELO, 2010; ANDRADE, 2015a; ORACLE, 2013f).

Figura 1 – Ciclo de vida de requisições JSF



Fonte: Produzido pelo autor utilizando as fontes:
(ORACLE, 2013f; JGRAPH, 2019)

A primeira etapa chama-se restaurar visão (do inglês *Restore View*), responsável por executar ações *postback*, método presente em páginas dinâmicas para evitar a perda de informações contidas em formulários emitidos ao servidor, caso ocorra questionamentos com validação ou regras de negócio. O recurso recupera a árvore de objetos caso esteja em *cache*, retornando ao cliente com informações sinalizadas no formulário referente ao erro (LUCKOW; MELO, 2010; ANDRADE, 2015a; ORACLE, 2013f).

Caso ocorra o preenchimento de formulários, a segunda etapa — aplicar valores de requisição (do inglês *Apply Request Values*) — tem como responsabilidade coletar os valores presentes na requisição URL (*Uniform Resource Locator*), para então inclui-los na árvore de componentes, preparando-os para uma posterior associação com a classe *Bean* responsável pela página, no entanto, caso o campo respondido apresente o atributo “*immediate=true*”, as informações são diretamente vinculadas a classe *Bean* (LUCKOW; MELO, 2010; ANDRADE, 2015a; MOZILLA, 2017; ORACLE, 2013f).

Posteriormente a terceira etapa — processar validações (do inglês *Process Validation*) — utiliza as regras de negócio inseridas pelo desenvolvedor para validar os valores anexados na requisição, não ocorrendo questionamentos, o fluxo prossegue para quarta etapa — atualizar valores do modelo (do inglês *Update Model Values*), recurso que associa os atributos informados para sua respectiva classe *Bean*, também realizando a conversão de valores para tipos básicos do Java ou personalizados pelo desenvolvedor (LUCKOW; MELO, 2010; ANDRADE, 2015a; ORACLE, 2013f).

Em seguida na quinta etapa — invocar aplicação (do inglês *Invoke Application*) — executa-se possíveis ações solicitadas na requisição URL, elas correspondem a métodos localizados na classe *Bean*, sendo este processo aplicado após a atribuição de valores sobre a mesma. Por fim, a sexta etapa — renderizar resposta (do inglês *Render Responde*) — retornar para o host requisitante a página JSF renderizada com o novo resultado (LUCKOW; MELO, 2010; ANDRADE, 2015a; ORACLE, 2013f).

Observe que o ciclo de vida JSF apresenta um procedimento denominado “processar eventos” (do inglês *Process Events*), método existente desde a etapa 2 a 5, tendo como objetivo redirecionar para a sexta etapa, questionamentos identificados e interromper a sequência de processos para informar o usuário sobre o ocorrido, a mensagem acompanha a classificação padrão do problema ou inserida pelo desenvolvedor (LUCKOW; MELO, 2010; ORACLE, 2013f).

Tratando-se de uma especificação JCP, grandes empresas demonstram interesse na criação de novos componentes personalizado/adicionais para incorporar no desenvolvimento *web* com JSF, dentre os quais podemos citar as bibliotecas: RichFaces da JBoss; ICEFaces da ICESoft; ADF Faces Rich Client da Oracle; e PrimeFaces da PrimeTek (LUCKOW; MELO, 2010; RICHFACES, 2016; ICESOFTE, 2017; ORACLE, 2017g; PRIMETEK, 2017).

3.1.2 Enterprise Java Beans

EJB 3.2 consiste na especificação JSR 345 presente no JEE 7, sendo uma arquitetura de componentes multiplataforma implantada no lado servidor, o recurso compreender a lógica de negócio da organização e objetiva: facilitar o trabalho de desenvolvedores abstraindo aspectos relacionados a infraestrutura da empresa; viabilizar a construção em multi-camadas de aplicações orientadas a objetos; e permitir a componentização distribuída da aplicação, característica que possibilita promover a escalabilidade e segurança transacional das informações (LUCKOW; MELO, 2010; ANDRADE, 2015a; ORACLE, 2017d; MEDEIROS, 2014).

A tecnologia EJB apresenta diferentes modalidades de funcionamento, cada uma especializada para um objetivo específico, neste trabalho destacamos as *Managed Beans*. O recurso é declarado na classe Java por meio da *annotation* “@ManagedBean”, responsável por interligar através de Linguagem de Expressão (EL, do inglês *Expression Language*). A comunicação e o comportamento entre telas de uma aplicação web (*front-end*), com a lógica de negócio empregada no código-fonte para o acesso e manipulação de informações do BD (*back-end*), com isso, retornando para UI o resultado do tratado de informações/ações emitidas pelo usuário. As *Managed Beans* possuem em seu escopo um tempo de vida também definido por *annotation*, o recurso delimita a execução e instanciação de uma classe *Bean* nas seguintes regras destacadas no Quadro 1 (LUCKOW; MELO, 2010; ANDRADE, 2015a; ORACLE, 2013h; CORDEIRO, 2014).

Quadro 1 – Escopo das *Managed Beans*

Nome	Descrição
@NoneScoped	Instância a classe <i>Bean</i> sempre que página <i>web</i> for referenciada.
@RequestScoped	O funcionamento persiste durante a requisição HTTP, finalizando após entrega de pacotes ao cliente.
@ViewScoped	Mantém a instância bean viva enquanto o usuário acessar páginas que a utilizem.
@SessionScoped	Preserva o objeto bean durante a sessão do utilizador no sistema.
@ApplicationScoped	A <i>Managed Bean</i> permanece vida durante a execução do sistema no servidor de aplicações, conseguindo também compartilhar informações entre usuários do sistema.

Fonte: (LUCKOW; MELO, 2010; ANDRADE, 2015a; CORDEIRO, 2014)

3.1.3 Java Persistence API

O JPA 2.1 é uma padronização de *frameworks* que apliquem a técnica de programação intitulada de Mapeamento Objeto Relacional (ORM, do inglês *Object Relational Mapping*), a especificação idealizada em 2006 pela JCP, define uma *interface* comum para ferramentas de persistência de dados com Java, estabelecendo normas de nomenclaturas para métodos,

variáveis, classes e boas práticas de design de software (ANDRADE, 2015a; ORACLE, 2017d; CORDEIRO, 2014).

A técnica ORM consiste em adequar a interação da Programação Orientada a Objeto — POO com BDs, objetivando abstrair a presença do paradigma relacional no código Java e suprimir a Impedância Objeto-Relacional (*Impedance Mismatch*), condição que representa os fatores divergentes entre as tecnologias (ANDRADE, 2015a; CORDEIRO, 2014).

Aplicando o método ORM objetiva-se construir aplicações utilizando somente a POO, procedimento viabilizado pelo mapeamento de classes, que utiliza *annotation* para arquitetar o *design* de uma aplicação Java, de modo similar a estrutura de uma base de dados. Exemplo disto, seria a associação entre: classes com tabelas; métodos de retorno com *datatypes* de colunas; e objetos com registros de uma *query* (ANDRADE, 2015a; CORDEIRO, 2014).

Quadro 2 – Exemplo de anotações utilizadas em ORM

Annotations	Obrigatório	Descrição
@Entity	Sim	Anotação define a classe como uma entidade liga a unidade de persistência.
@Id	Sim	Define/identifica a variável java como chave primária de uma entidade.
@GeneratedValue	Não	Gera automaticamente um valor para a variável definida com <i>primary-key</i> .
@Column	Não	Usado para descrever a propriedade da coluna exemplo: valor não pode ser nulo; nome da tabela que possui a coluna; tamanho de caracteres ou dígitos do campo; entre outros.

Fonte: (ANDRADE, 2015a; DEMICHIEL, 2013; CORDEIRO, 2014)

Ao se trabalhar com persistência de informações em Java, é comum a utilização do padrão de classe denominado Objeto de Acesso a Dados (DAO, do inglês *Data Access Object*), o mesmo dispõe uma *interface* que determina quais operações de criação, leitura, atualização e destruição de dados (CRUD, do inglês *Create, Read, Update, and Delete*) podem ser realizadas sobre o BD, utilizando determinada instância de classes mapeada, desta forma, essa característica promove a separação da lógica de negócio, e infraestrutura responsável pela persistência de dados (ANDRADE, 2015a; CORDEIRO, 2014; MAK; GURUZU; OTTINGER, 2015).

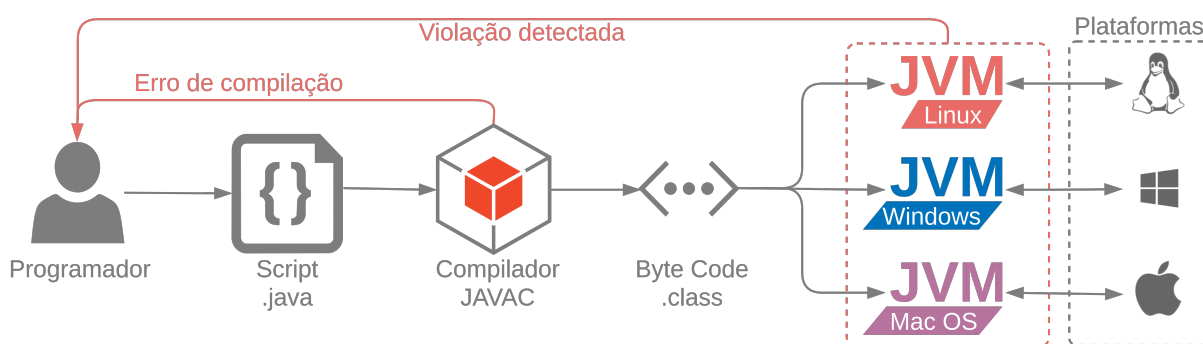
No JPA, os parâmetros necessários para estabelecer a comunicação com a base de dados, são descritos no arquivo “persistence.xml”, nele é adicionado a propriedade “persistence-unit”, que circunda informações como: credenciais de acesso e URL de conexão com a camada de persistência; bibliotecas que dispõem o dialeto SQL (*Structured Query Language*) e *driver* JDBC (*Java Database Connectivity*) apropriado para cada *software* de BD (ANDRADE, 2015a; DEMICHIEL, 2013; CORDEIRO, 2014).

O JPA presente no JEE 7 encontra-se na versão 2.1, e consiste na especificação JSR 338, para utilizá-lo é necessário empregar *frameworks* ORM que o implementem. Dentre as opções de mercado disponíveis destaca-se: EclipseLink; OpenJPA; e Hibernate sendo a biblioteca mais difundida no contexto ORM, as características desta ferramenta serão abordadas posteriormente (DEMICHIEL, 2013; CORDEIRO, 2014).

3.2 Java Development Kit 8

O JDK (*Java Development Kit*) é uma tecnologia presente no JSE (*Java Platform Standard Edition*) disponibilizado pela Oracle Corporation, consistindo numa biblioteca que fornece um ambiente de desenvolvimento/execução de softwares e *web-applications* utilizando a linguagem de programação Java, o JDK está disponível nas plataformas Windows, Linux e Mac OS X, trabalhando sobre as arquiteturas de 32 e 64 bits dos sistemas operacionais citados (DEITEL, 2017; ORACLE, 2017i; ORACLE, 2017j).

Figura 2 – Processo de execução multiplataforma de software java



Fonte: Produzido pelo autor utilizando as fontes:

(DEITEL, 2017; ORACLE, 2017i; ORACLE, 2017j; FONTAWESOME, 2018; JGRAPH, 2019)

A Figura 2 ilustra o processo de execução de aplicações Java, do qual primeiramente é utilizado os “scripts.java”, código escrito e compreendido pelo programador, submetido ao compilador “javac” que o interpreta e converte em “byte-codes”, arquivos na extensão “.class” executado na Máquina Virtual Java (JVM, do inglês *Java Virtual Machine*) apropriada ao sistema operacional subjacente, sempre respeitando suas características e limitações (DEITEL, 2017; DIONISIO, 2013).

A JVM é um componente pertencente ao Java Runtime Environment — JRE, podendo ser adquirido separadamente apenas para execução dos softwares java, ou seja, sem a necessidade de instalar o JDK. A mesma também permite a execução em multiplataformas de aplicações java, porém, o sistema operacional necessita ter uma JVM implementada especificamente para a plataforma (LUCKOW; MELO, 2010; DEITEL, 2017; DIONISIO, 2013).

Observe na Figura 2 que ambos os arquivos “.java” e “byte-code” viabilizam a portabilidade de código-fonte, no entanto, os script redigidos na notação java devem ser submetidos

novamente a análise de sintaxe realizada pelo compilador “javac”, para verificação de erros e criação de byte-codes, tarefa que demanda tempo de processamento, ao contrário do “.class”, com tudo, por se tratar de um arquivo editável é suscetível a violação da integridade do código, mesmo sendo de difícil compreensão aos programadores, as JVMs realizam durante a execução do software java, verificações para encontra inconsistências no “.class” (DEITEL, 2017).

O acervo de ferramentas do JDK além de incluir bibliotecas/APIs necessárias ao desenvolvimento de aplicações java, engloba o compilador “javac” e recursos do JRE como a JVM, este trabalho utiliza a 8ª versão do kit, trazendo consigo a adição e aprimoramento de funcionalidades presente em versões anteriores (DEITEL, 2017).

3.3 Maven para o controle de bibliotecas

O Maven é uma ferramenta mantida pela *Apache Software Foundation* e desenvolvida em Java, podendo ser executado em sistemas que implementem a JVM, a aplicação não demanda robustez de hardware nas estações de trabalho dos desenvolvedores. O programa funciona sobre uma estrutura de execução de “plugins” e destina-se ao gerenciamento de artefatos/repositórios utilizados durante o desenvolvimento de aplicações Java (ANDRADE, 2015a; SIRIWARDENA, 2014; VARANASI; BELIDA, 2014).

O Maven apresenta duas categorias de repositórios, *localhost* e remoto, localmente destina-se ao armazenamento de bibliotecas utilizadas tanto para o funcionamento do Maven, quanto para a construção do projeto de software, tal espaço de alocação é denominado *heap size*, podendo ser personalizado conforme as necessidades dos utilizadores, modificando a variável “MAVEN_OPTS” por linha de comando (SIRIWARDENA, 2014; VARANASI; BELIDA, 2014).

A ferramenta dispõe um arquivo na extensão XML (*eXtensible Markup Language*) denominado *Project Object Model* — POM, usado para administrar dependências e *plugins* utilizados no repositório local, através de fragmentos de código inseridos no XML pelo programador, com isso, permitindo ao Maven realizar na estação de trabalho o *downloads/upgrade* de bibliotecas essenciais ao projeto, artefatos estes disponíveis em centrais de repositório remoto (ANDRADE, 2015a; SIRIWARDENA, 2014; VARANASI; BELIDA, 2014; QUIN, 2016).

A infraestrutura da plataforma Maven consiste no espelhamento e dispersão geográfica de repositórios em nuvem, tal característica somada a ferramenta local (repositório *localhost*), que por padrão designa a melhor *mirrors* com base na localização de seu utilizador, porém, essa configuração pode ser personalizada no arquivo “settings.xml” localizado no diretório “.m2” destinado ao Maven, objetivando melhor atender as necessidades do desenvolvedor, como exemplo disto destaca-se a centralização de todos os artefatos necessários ao projeto numa intranet, seja para atender políticas de restrição de uso da internet implementados por empresas, ou para otimizar a demanda por tráfego de dados e processamento/compilação de *builds* (SIRIWARDENA, 2014; VARANASI; BELIDA, 2014).

Uma *builds* é o resultado da compilação do projeto de *software* utilizando Maven, ela pode representar versões de teste, finais ou parciais do programa, o tempo de processamento utilizado para gerá-la está diretamente relacionado com a complexidade do repositório, ou seja, quantidade de artefatos/*plugins* utilizados para produzir e compor a aplicação, tarefa que pode exigir monitoramento dos desenvolvedores em processos que demandem algumas horas para sua conclusão (ANDRADE, 2015a; SIRIWARDENA, 2014; VARANASI; BELIDA, 2014).

O processo de execução/compilação Maven é composto por fases, cada uma possui um conjunto de *plugins* específicos para satisfazer determinados objetivos, onde o encapsulamento/empacotamento especificado no POM, ou seja, arquivos na extensão JAR (Java Archive), WAR (Web Application Resource) e EAR (Enterprise ARchive), determina quais conjunto de *plugins* serão utilizados para concluir a tarefa, para entregar uma versão da aplicação (LUCKOW; MELO, 2010; SIRIWARDENA, 2014; VARANASI; BELIDA, 2014; CAELUM, 2017b).

3.4 Primefaces

O PrimeFaces é um *framework* de código aberto baseado no JSF, idealizado por Çağatay Çivici, a biblioteca disponibiliza aos desenvolvedores Java *web* um conjunto aprimorado de componentes UI, objetivando melhorar a produtividade no processo de desenvolvimento, mediante ao fornecimento de ferramentas pré-programadas que dispensem estilização CSS (*Cascade Style Sheet*), e que possam abranger diferentes necessidades para representar e organizar informações em *interfaces* de *software*, por meio da variedade de componentes disponíveis (ANDRADE, 2015a; PRIMETEK, 2017; ANDRADE, 2016b; ÇIVICI, 2018).

Além de disponibilizar documentações de uso do *framework*, o página oficial do PrimeFaces dispõem a *Showcases*, área dedicada à experimentação *on-line* de componentes de *interface*, demonstrando ao desenvolvedor como discorre o funcionamento das ferramentas, acompanhado ao código-fonte utilizado no exemplo (LUCKOW; MELO, 2010; ANDRADE, 2015a; PRIMETEK, 2017; ÇIVICI, 2018).

O pacote JAR do *framework* pode ser adquirido via *download* no site PrimeFaces ou como dependência Maven, destaca-se que a biblioteca não substitui a implementação padrão do JSF, e sim agindo de maneira complementar para a mesma, sendo compatível ao JSF desde a versão 2.0 até a mais atual (2.3) (PRIMETEK, 2017; ÇIVICI, 2018).

As funcionalidades do PrimeFaces não requerem a implementação adicional de recurso no lado cliente, isso em razão da biblioteca já incluir os artefatos necessários para o funcionamento, dentre eles destaca-se os componentes de *interface* baseados no *framework* javascript jQuery, biblioteca amplamente compatível com os *browser* mais populares do mercado, e o dinamismo de páginas proveniente do AJAX (ANDRADE, 2015a; PRIMETEK, 2017; ÇIVICI, 2018).

Para este projeto foi utilizado a biblioteca 6.2 do PrimeFaces, disponibilizada gratuitamente em pacotes mantidos pela comunidade de desenvolvedores, no entanto, não abarcando os serviços pagos fornecidos pela PrimeTek, dentre os quais destaca-se as *dashboards* (recurso visual para mostrar em métricas os objetivos/dados alcançados) com *layout* mais sofisticado, e o serviço de suporte ao desenvolvedor fornecido pela empresa (PRIMETEK, 2017; ÇIVICI, 2018; ORLANDO; SUNINDYO, 2017).

3.5 Hibernate

O Hibernate é um *framework* que implementa especificações como o JPA no sentido de padronizar a aplicabilidade da técnica ORM, o mesmo também administra aspectos referentes a infraestrutura necessária na comunicação entre aplicações Java e BD. A ferramenta desenvolvida em Java, consiste num projeto *open-source* iniciado pela empresa JBoss, atualmente a corporação Red Hat adquiriu e mantém ambos empreendimentos (CAELUM, 2017a; REDHAT, 2017).

O hibernate realiza a abstração de aspectos inerente aos *softwares* de BD, dentre os quais destaca-se o dialeto SQL utilizado por cada ferramenta, característica que inviabiliza o portabilidades dos *scripts* entre as opções de mercado, então, a ferramenta intercede a comunicação entre aplicação Java e BD, com a Linguagem de Consulta do Hibernate (HQL, do inglês *Hibernate Query Language*), gerando operações de CRUD apropriadas para cada BD, tal recurso favorece a portabilidade do BD, demandado apenas mudança do drive de conexão com o BD na aplicação Java (ANDRADE, 2015a; CAELUM, 2017a; MAK; GURUZU; OTTINGER, 2015; REDHAT, 2017).

O Hibernate possui diferentes estratégias de manipulação de tabelas em BD, tais formas de interação com o esquema relacional, são definidos na propriedade “hibernate.hbm2ddl.auto” inserida no arquivo “persistence.xml”, abaixo, o Quadro 3 lista os possíveis valores.

Quadro 3 – Valores para “hibernate.hbm2ddl.auto”

Nome	Descrição
<i>Create</i>	Aplicação sempre trabalhar com uma base de dados vazia, excluindo e criando as tabelas ao inicia-lá.
<i>Create-drop</i>	Conforme especificado pelo programador, essa propriedade exclui opcionalmente o BD, após a finalização da aplicação.
<i>Update</i>	Utiliza-se este valor em aplicações de produção, que deseja-se manter e atualizar as informações persistidas no BD, bem como sua estrutura.
<i>Validate</i>	A escolha deste valor não realiza alterações sobre o BD, apenas verifica se o esquema relacional condiz ao mapeamento de classes, caso contrário, será lançado uma exceção.

Fonte: (CORDEIRO, 2014)

O Hibernate também possui ferramentas que permitem a criação do mapeamento de classes a partir de um BD existente, tal característica faz parte do conjunto de procedimentos denominado “Hibernate Tools”. A utilização dessa estratégia de desenvolvimento demanda arquivos de configuração específicos do *framework*, ocasionando na mudança e adição de procedimentos descritos no Quadro 4 (REDHAT, 2017).

Quadro 4 – Procedimentos do “Hibernate Tools”

Nome	Descrição
“hibernate.cfg.xml”	Substitui o “persistence.xml” usando pelo JPA, usando os mesmos parâmetros necessários para estabelecer conexão com BD, porém, informando todas as tabelas identificadas no mapeamento reverso.
“hibernate.reveng.xml”	Conforme a especificado pelo desenvolvedor, o arquivo informa quais tabelas que serão submetidas ao processo de mapeamento reverso.
“*.hbm.xml”	Consiste em arquivos de mapeamento utilizado pelo Hibernate, com eles descreve-se características estruturais de cada tabelas do esquema relacional, por meio deles, o <i>framework</i> gera automaticamente as classes mapeadas.
“hibernate.hbm2ddl.auto”	A estratégia de mapeamento reverso não utiliza a propriedade, uma vez que o mapeamento de classes é realizadas a partir no BD.

Fonte: (MAK; GURUZU; OTTINGER, 2015; REDHAT, 2017)

A página oficial do hibernate dispõe duas formas para adquirir a ferramenta: a primeira consiste em baixar o pacote pelo SourceForge (repositório em nuvem de códigos fontes), tarefa que demanda instalação manual do *framework*; a segunda ocorre como dependência Maven. Este trabalho utiliza a implementação 4.3, compatível ao JPA 2.1 (REDHAT, 2017; SOURCEFORGE, 2017).

3.6 Ferramentas de codificação

A sessão tem como objetivo explicar características relevantes de duas ferramentas utilizadas na codificação do projeto, ressaltando itens como: usabilidade da UI; funcionalidades disponíveis no *software*; para qual sistema operacional apresenta suporte; dentre outros.

3.6.1 Netbeans IDE

O NetBeans é um Ambiente de Desenvolvimento Integrado (IDE, do inglês *Integrated Development Environment*) modular, o qual permite atualizar, excluir e adicionar novos complementos utilizando seu gerenciador de *plugins*, e suas ferramentas são disponibilizadas em uma

UI customizável, viabilizando adaptar as guias de trabalho, conforme a necessidade particular de cada usuário (NETBEANS.ORG, 2017a; NETBEANS.ORG, 2017b).

O IDE é um projeto de código aberto mantido e disponibilizado gratuitamente pela Oracle, incluindo no programa o suporte ao ecossistema de desenvolvimento JEE mediante a utilização do JDK, e também o uso de tecnologias externas como: Maven; PrimeFaces; Hibernate e servidores de aplicação para executar e manipular *web software* durante sua construção (NETBEANS.ORG, 2017a).

O NetBeans também dispõe recursos que auxiliem na edição de códigos fonte. Destacando-se o editor multilíngue que fornece reconhecimento sintático/semântico para linguagens como: Java; PHP (acrônimo recursivo para: *Hypertext Preprocessor*); C/C++; HTML; JavaScript; entre outros. Além disso, dispendo também de ferramentas para indentação e associação automática de palavras, depuração de scripts, e modelos de projetos para tecnologias suportadas (NETBEANS.ORG, 2017a; NETBEANS.ORG, 2017b; PHPGROUP, 2017).

O programa é disponibilizado nas plataforma Windows, Linux e Mac OS X, este projeto utiliza a versão 8.2 do *software*, agregando características que promovam um espaço completo de criação/manipulação de aplicações *webs*, *desktops* e dispositivos móveis (KLINGER, 2014)(NETBEANS.ORG, 2017a; NETBEANS.ORG, 2017b).

3.6.2 MySQL

O MySQL é um Sistema Gerenciador de Banco de Dados — SGBD, direcionado ao armazenamento organizado de informações em estruturas tabulares constituídas por linhas, campos e colunas que podem ser configuradas em diferentes “datatypes” (exemplo: VARCHAR, INT, DOUBLE, entre outros) (CARVALHO, 2015).

As tabelas do SGBD mantêm o sentido lógica das informações mediante relacionamentos, caracterizados pela exportação do atributo identificador denominado *primary key* para outra tabela, passando a intitular-se chave estrangeira (*foreign key*), aspecto que configura o paradigma relacional usando pelos SGBDs (CARVALHO, 2015).

Os programas SGBDs assim como o MySQL, realizam operações de CRUD utilizando seu próprio dialeto SQL, para isso, o software permite aplicar linguagens de manipulação, definição e controle de Dados. Os procedimentos são descritos no Quadro 5.

O MySQL dispõe diferentes modos de instalação para satisfazer finalidades específicas, sejam na implementação de servidores de dados utilizando “mysql-server-only”, *software* que engloba funcionalidades DML, DDL e DCL para administrar e disponibilizar a base de dados em rede de computadores. A instalação “mysql-client-only” auxilia no acesso remoto/localhost mediante console aos recursos do BD, utilizando senha, porta de acesso ao serviço e o Protocolo de Internet (IP, do inglês *Internet Protocol*) do servidor de dados (KUROSE JAMES F.; ROSS, 2013; CARVALHO, 2015).

Quadro 5 – Funcionalidade de comandos SQL

Nomenclatura	Descrição
Linguagens de Manipulação (DML, do inglês <i>Data Manipulation Language</i>)	Engloba operações para acessar ou modificar o estado de informações armazenadas no BD, dentre os quais destaca-se comandos de: INSERT; DELETE; UPDATE; SELECT; e LOCK.
Linguagem de Definição de Dados (DDL, do inglês <i>Data Definition Language</i>)	Seus comandos são responsáveis por modificar a estrutura tabular do BD utilizando as operações de: CREATE TABLE; CREATE INDEX; ALTER TABLE; DROP TABLE; DROP VIEW; e DROP INDEX
Linguagem de Controle de Dados (DCL, do inglês <i>Data Control Language</i>)	Comandos DCL são responsáveis por delimitar prioridades de acesso a informações contidas no BD, para isso dispendo operações de: COMMIT; ROLLBACK; GRANT; e REVOKE

Fonte:(CARVALHO, 2015)

O MySQL também disponibiliza opcionalmente uma IDE denominada “Workbench”, permitindo modelar graficamente o Diagrama de Entidade e Relacionamento — DER correspondente ao modelo físico do BD. O *software* também aplica funcionalidades dos pacotes “mysql-server-only” e “mysql-client-only”, atuando como ferramenta de desenvolvimento e administração de aspectos inerentes (backups, gerenciar usuários, configurar servidor de dados, etc) aos BD (CARVALHO, 2015; ORACLE, 2017k).

3.7 Gerência de configuração

A gerência de configuração é uma sub-área da engenharia de *software* que dispõem metodologias de apoio ao processo de inclusão/modificação de artefatos e requisitos de sistemas, aprimorando o método de desenvolvimento mediante o rastreamento e documentação de alterações realizadas no projeto. Com essa ideia, a sessão apresenta/explica duas ferramenta utilizadas no trabalho, ambas direcionadas ao controle de versão de *software* e embasadas no “Git”, programa criado por Linus Torvalds em 2005 para desenvolvimento do “kernel” Linux (SOMMERVILLE, 2011; AQUILES A.; FERREIRA, 2014).

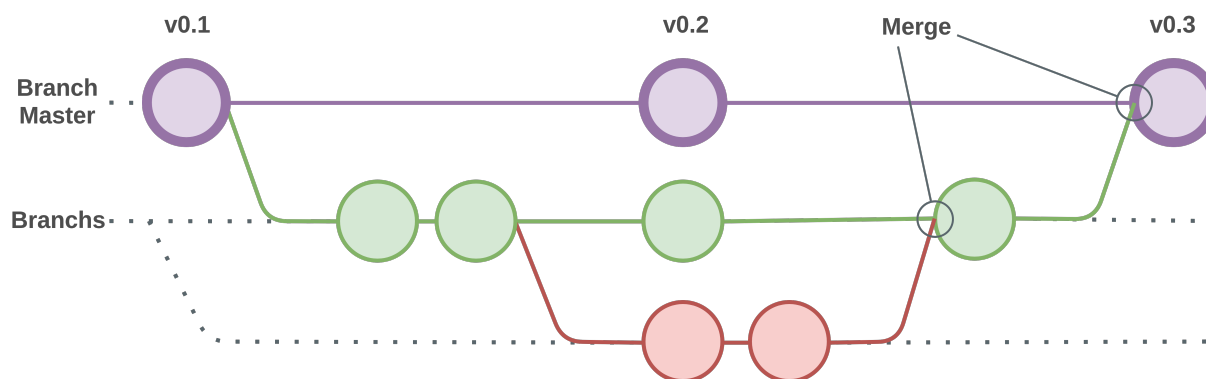
3.7.1 GitHub

O GitHub é uma aplicação *web* que dispõe gratuitamente um conjunto de ferramentas direcionados ao controle de versão de *software* e arquivos, permitindo gerenciar e documentar modificações realizadas em projetos hospedados na plataforma. Viabilizando também, o trabalho colaborativo de outros usuários em repositórios privados e/ou *Open Source* (AQUILES A.; FERREIRA, 2014).

Em aplicações Git com GitHub, a consolidação de arquivos/modificações no repositório

é realizada mediante envio/*push* de *commits*, com eles, é permitido rastrear alterações utilizando: seu atributo identificador único; e título atribuído ao *commit*. Sendo um *web-software on-line*, as alterações são envidas utilizando o protocolo de Segura na Camada de Transporte (TLS, do inglês *Transport Layer Security*) (KUROSE JAMES F.; ROSS, 2013; AQUILES A.; FERREIRA, 2014).

Figura 3 – Utilização de *branchs* para isolar módulos em desenvolvimento



Fonte: Produzido pelo autor utilizando as fontes:
(SOMMERVILLE, 2011; AQUILES A.; FERREIRA, 2014; JGRAPH, 2019)

Toda alteração enviada ao repositório central, são organizadas em ramificações (*branches*) conforme ilustrado na Figura 3, elas consistem em linhas de desenvolvimento independentes que permitem realizar modificações em arquivos, sem comprometer a versão estável do projeto. Geralmente contido na ramificação principal denominada *branch-master*, o qual não pode ser deletada, e posteriormente, as ramificações filhas podem ser mescladas (*merge*) ao ramo *master* (SOMMERVILLE, 2011; AQUILES A.; FERREIRA, 2014).

A plataforma permite ser utilizada de diferentes formas, primariamente mediante sua *Dashboard on-line*, porem, não disponibilizando o rastreio de alterações, exclusões e criações de arquivos em *off-line* na estação de trabalho de seus utilizadores. A segunda consiste com ferramentas *desktops* conectadas ao GitHub, do qual destacamento: a utilização do pacote Git mediante console de sistemas operativos; e *softwares* com UI como o GitHub-Desktop, SmartGit, e GitKraken (AQUILES A.; FERREIRA, 2014; AXOSOFT, 2017; GITHUB, 2017b; SYNTEVO, 2017).

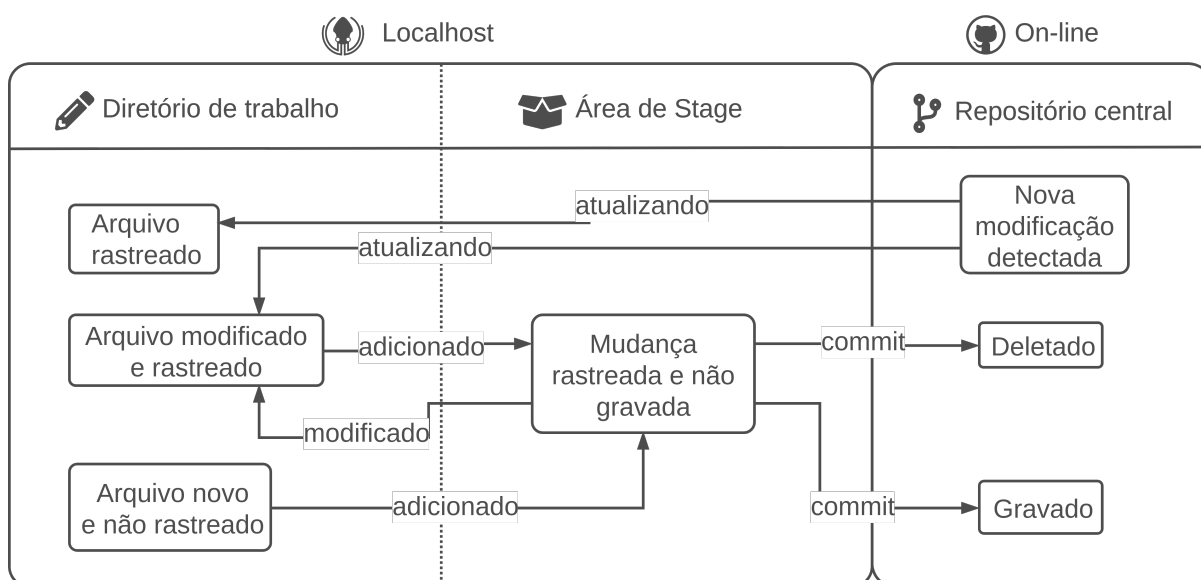
O GitHub foi implementado em 2008 por Tom Preston-Werner, Chris Wanstrath e PJ Hyett, e vendida à Microsoft em 2018, e segundo fontes obtidas em CIRIACO (2018) , o GitHub ainda “ [...] vai operar independentemente para oferecer uma plataforma aberta para todos os desenvolvedores de todas as indústrias”. Além do GitHub existe outro serviços de hospedagem de projetos baseados no Git como o Bitbucket. (AQUILES A.; FERREIRA, 2014; CIRIACO, 2018; MICROSOFT, 2018).

3.7.2 Gitkraken

O Gitkraken é uma ferramenta *Git desktop* construída com a API de desenvolvimento “Electron”, que utiliza tecnologias da Internet (JavaScript, HTML e CSS) para desenvolver aplicações nativas. O *software* foi idealizado e mantido pela empresa norte americana Axosoft, e demanda a instalação de um único pacote para sua utilização, do qual é disponibilizado para os sistemas Windows, Linux e Mac (AXOSOFT, 2017; GITHUB, 2017a).

O cliente Git possui uma UI que dispensa a utilização do console para administrar e efetuar tarefas relacionadas aos serviços de hospedagem Git, tais como: *commits*; *branches*; *merge*; *push*, entre outros. O *software* fornece suporte as plataformas Bitbucket e GitHub, e ajuda administrar o fluxo de trabalho de seus usuários conforme ilustrado na Figura 4 (AXOSOFT, 2017).

Figura 4 – Dinâmica de trabalho com ferramenta *git desktop*



Fonte: Produzido pelo autor utilizando as fontes:

(AQUILES A.; FERREIRA, 2014; AXOSOFT, 2017; FONTAWESOME, 2018; JGRAPH, 2019)

O diretório de trabalho consiste na clonagem de um repositório para uma estação *localhost*, sobre ele é desenvolvido todas as possíveis contribuições de um colaborador para o projeto. Ocorrendo a manipulação e distinção de arquivos novos e/ou rastreados pelo GitHub ou área de *stage* (AQUILES A.; FERREIRA, 2014).

A área de *stage* consiste no agrupamento de alterações estáveis adicionadas pelo colaborador do projeto, a partir delas, será preparado novos *commits* que serão enviados ao repositório central para efetivar o rastreamento no GitHub. Na *stage*, arquivos novos e ainda não encaminhados pelo *push*, serão somente rastreados em *localhost* pelo cliente Git (AQUILES A.; FERREIRA, 2014; AXOSOFT, 2017).

A Figura 4 também exemplifica arquivos atualizados conforme a detecção de novas modificações realizadas no repositório on-line, então, o GitKraken também auxilia seus utilizadores na resolução de alterações conflitantes entre colaborações, exibindo as diferenças entre as versões do mesmo arquivo, para adicionar, manter ou descartar modificações. (AXOSOFT, 2017).

3.8 Servidor de aplicações Glassfish

O GlassFish consiste num servidor de aplicações *open source* desenvolvido em Java e mantido pela Oracle Corporation, sua utilização demanda a instalação da JVM no sistema operacional adjacente, e fornecendo suporte a todas as especificações do JEE, com isso, sendo capaz de disponibilizar em rede Cliente-Servidor através da porta 8080, *web-sofware*s/componentes encapsulados nos formatos JAR, WAR ou EAR (ORACLE, 2017).

O *software* através da porta administrativa (4848), disponibiliza na rede uma UI que auxilia seus administradores a configurar o programa, por meio dela, o GlassFish dispõe um conjunto de ferramentas voltadas para a administração e otimização da execução de *web-applications*, dentre os quais destacamos: o Sistema de Nomes de Domínio (DNS, do inglês *Domain Name System*); a instalação de certificados para implementar o protocolo HTTPS (*HyperText Transfer Protocol Secure*); gerenciar as aplicações inseridas no servidor; entre outros (KUROSE JAMES F.; ROSS, 2013; ORACLE, 2017).

A nível de segurança, o GlassFish permite personalizar dados sensíveis utilizados pelo programa, dentre os quais citamos: sua porta administrativa 4848; senha e login de acesso a sua *Dashboard*. O mesmo encontra-se na sua versão 5.0, porém este trabalho implementa a opção 4.1.2 do *software* (ORACLE, 2017).

4 DESENVOLVIMENTO DA APLICAÇÃO

O capítulo 4 descreve características inerentes ao desenvolvimento do *web-software* veterinário SIHV, do qual é apresentado: o processo de construção utilizado no projeto; artefatos que demonstram a modelagem de características arquiteturais do *software*.

A metodologia aplicada no processo de construção do sistema, utiliza o meio-termo entre duas abordagens de engenharia de *software* que se complementam. A primeira consiste no desenvolvimento dirigido a planos, do qual contempla processos que investiguem aspectos inseridos no contexto dos *stakeholders*, para delinear os objetivos a serem alcançados no projeto mediante documentações, esta abordagem é marcada pela produção de uma documentação vasta e complexa (SOMMERVILLE, 2011).

A segunda abordagem tem como foco o desenvolvimento incremental de funcionalidades do sistema, onde a construção de módulos é priorizada conforme o critério de necessidade estabelecido por *stakeholders*. A metodologia também enfatiza a participação destes indivíduos, nas etapas de especificação, desenvolvimento e implementação do *software*, porém, a construção de funcionalidades não baseia-se em uma documentação elaborada e bem descrita, ou seja, o foco situa-se no desenvolvimento (SOMMERVILLE, 2011).

O autor Sommerville (2011, p.22) ainda destaca a seguinte problemática do desenvolvimento incremental: “*A estrutura do sistema tende a se degradar com a adição dos novos incrementos. [...] as constantes mudanças tendem a corromper sua estrutura. Incorporar futuras mudanças do software torna-se cada vez mais difícil e oneroso*”.

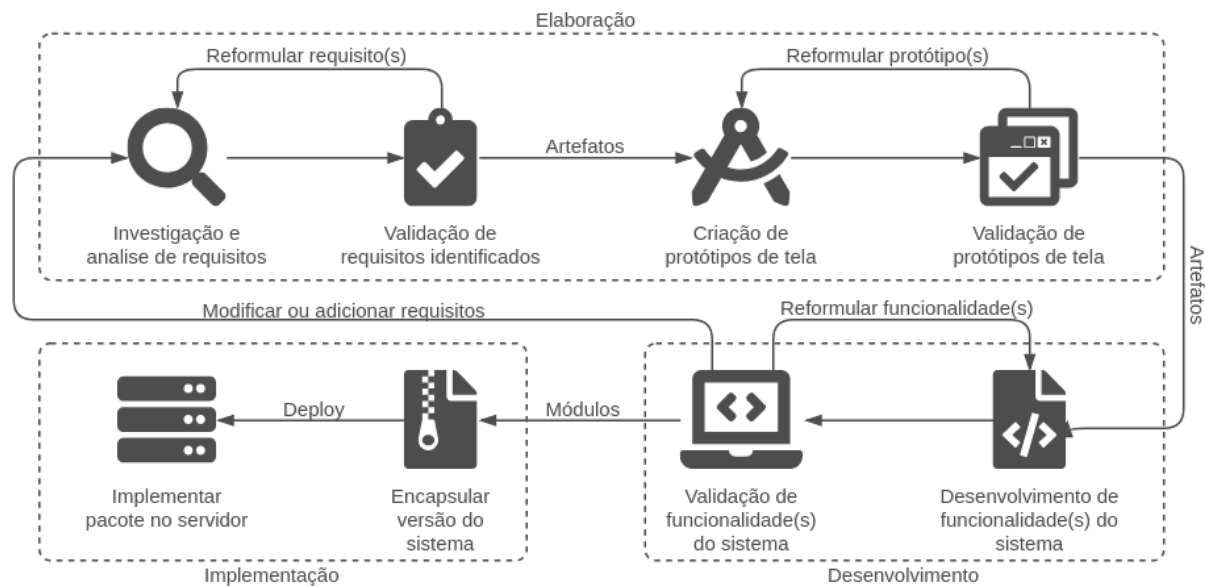
Seguindo o pensamento de Sommerville (2011), este trabalho mescla as metodologias de desenvolvimento citadas acima da seguinte forma. Aplicar a postura mais atenta na investigação e análise de requisitos proporcionada pela abordagem dirigida a planos, porém, produzindo artefatos com menor teor de detalhes. Em seguida, direcioná-las ao desenvolvimento incremental, aprimorando a construção de funcionalidades devido à utilização de especificações/documentações mais estudadas, com isso, objetivando amenizar a degradação da estrutura do *software*.

O trabalho também aplica conforme a metodologia utilizada e ideias do autor Sommerville (2011), a necessária aproximação de utilizadores do sistema, no processo de concepção da aplicação, para definir e manter o gerenciamento sobre artefatos que expressem os objetivos do *software*, para atender as expectativas e solucionar problemáticas dos *stakeholders* (SOMMERVILLE, 2011).

4.1 Etapas do desenvolvimento da aplicação

A Figura 5 ilustra o processo utilizado na construção do sistema veterinário, do qual dividi-se em três momentos: elaboração; desenvolvimento; e implementação do *web-software*. Cada um possuindo subprocessos com objetivos estabelecidos, e gerando artefatos específicos para posterior utilização em etapas seguintes.

Figura 5 – Processo de desenvolvimento



Fonte: Produzido pelo autor utilizando as fontes:
(FONTAWESOME, 2018; JGRAPH, 2019)

A elaboração engloba a fase conceitual de módulos do projeto, nela compreendei-se técnicas contempladas pela engenharia de *software*, dentre os quais destacamos na etapa inicial a investigação e análise de requisitos, objetivando especificar as características do sistema mediante entrevistas semi-estruturadas realizadas com *stakeholders* diretos e indiretos, bem como também a observação de artefatos e processos utilizados pelos mesmos (SOMMERVILLE, 2011).

A etapa seguinte consiste na validação de requisitos, ela demanda a presença dos *stakeholders* para verificar e autenticar o entendimento do engenheiro de *software*, transcrito em documentações parciais como listagem de requisitos e/ou fluxogramas. Por meio deles, representando a dinâmica de trabalho do cliente e suas necessidades, não havendo o aceite, o fluxo retorna para a investigação e análise de requisitos (SOMMERVILLE, 2011).

O autor Sommerville (2011, p.25) também ressalta que:

As atividades no processo de requisitos não são feitas em apenas uma sequência. A análise de requisitos continua durante a definição e especificação, e novos requisitos emergem durante o processo. Portanto, as atividades de análise, definição e especificação são intercaladas.

Apos a validação de requisitos, o processo resulta em especificações iniciais do *software*. Na terceira etapa utilizam-se os artefatos aprovados pelo cliente para iniciar o esboço de características importantes do sistema, para isso, aplicando a prototipação de telas, do qual desenvolve-se de maneira rápida e simples imagens ou animações que representem as funcionalidades da aplicação, conforme o grau de importância definido pelos *stakeholders* (SOMMERVILLE, 2011).

Finalizando a elaboração ocorre a validação de protótipos de tela, novamente requerindo a opinião dos *stakeholders*, é apresentado uma prévia da dinâmica de trabalho proposta na tela de usuário, com isso, objetivando aprimorar a usabilidade e amigabilidades das UI mediante *feedbacks* obtidos (SOMMERVILLE, 2011).

Para Sommerville (2011) esse processo não se restringe apenas ao aprimoramento do *design* telas, também contribui na prevenção de mudanças no projeto, devido à experimentação e noção prévia dos utilizadores, sobre partes do sistema. O autor ainda destaca os seguintes benefícios para o empreendimento:

Protótipos do sistema permitem aos usuários ver quão bem o sistema dá suporte a seu trabalho. Eles podem obter novas ideias para requisitos e encontrar pontos fortes e fracos do software; podem, então, propor novos requisitos do sistema. Além disso, o desenvolvimento do protótipo pode revelar erros e omissões nos requisitos propostos (SOMMERVILLE, 2011, p.30).

Após definir na elaboração a especificação de um ou mais módulos do sistema, iniciam-se etapas que compreendem o desenvolvimento do *software*, a primeira consiste em utilizar tecnologias e ferramentas apresentadas no capítulo 3 na produção de funcionalidades da aplicação. Utilizando *frameworks* e especificações do JEE para atender requisito funcionais ou não, e ferramentas que auxiliem na construção e administração de mudanças no projeto, do qual destacado o GitHub, Gitkraken, Maven, Netbeans, entre outros. A segunda etapa do desenvolvimento valida módulos funcionais da aplicação, apresentando seu funcionamento aos *stakeholders*, e tendo o seguinte objetivo:

[...] mostrar que um software se adequa a suas especificações ao mesmo tempo que satisfaz as especificações do cliente do sistema. Teste de programa, em que o sistema é executado com dados de testes simulados, é a principal técnica de validação. A validação também pode envolver processos de verificação, como inspeções e revisões, em cada estágio do processo de software, desde a definição dos requisitos de usuários até o desenvolvimento do programa [...] (SOMMERVILLE, 2011, p.27).

Observa-se também o seguinte aspecto adotado na construção do projeto:

O sistema é desenvolvido em uma série de versões. Os usuários finais e outros stakeholders do sistema são envolvidos na especificação e avaliação de cada versão. Eles podem propor alterações ao software e novos requisitos que devem ser implementados em uma versão posterior do sistema (SOMMERVILLE, 2011, p.39).

A etapa de implementação marca a finalização do ciclo de construção de uma ou varias funcionalidades do *software*. Após a validação de um conjunto de módulos do sistema, os mesmos são encapsulados no formato WAR para compor a nova versão do programa, para então ser implementada na plataforma que executa a aplicação.

O ambiente de implantação é formado por *softwares* responsáveis em gerenciar e fornecer recursos necessários ao funcionamento da aplicação, dentre os quais se destaca: o sistema operacional; BD; bibliotecas/*frameworks*; e o servidor de aplicações apresentado no tópico 3.8, utilizado para publicar (*deploy*) a *web-application* (SOMMERVILLE, 2011).

4.2 Itens investigados e requisitos elicitados

O Quadro 6 demonstra os itens investigados e requisitados durante o processo de elaboração de características sistema. A pesquisa e análise é realizada quando deseja-se definir novos módulos que iram integrar a aplicação veterinária.

Quadro 6 – Itens investigados durante o projeto

Item	Descrição
<i>stakeholders</i>	Realizou-se entrevistas semi-estruturadas para coletar a opinião de 16 indivíduos, em sua grande maioria, profissionais que serão beneficiados diretamente pelo uso do sistema. Cada um apresentando o seguinte perfil profissional: docente do curso de veterinária; médico veterinário residente ou técnico do HV; radiologista; e recepcionista.
Artefatos	Foi analisado os seguintes documentos: cadastro de proprietários e animais pequenos; formulários de anamnese, exame físico geral e diagnósticos diferenciais; ficha para requisitar e preencher dados da radiografia; exames específicos do sistema digestório e glândulas anexas, respiratório e cardiovascular, gênito-urinário e glândulas mamárias, tegumentar, neurológico, oftálmico e músculo-esquelético; tabela de procedimentos do HV; relatório de controle de fichas e entrada de caixa; agenda de consultas.
Estágio no local de trabalho dos <i>stakeholders</i>	Aplicaram-se observações em setores do HV, de modo a compreender a visão de trabalho do cliente, e com isso, objetivando encontrar problemáticas, processos e requisitos ocultos que possam ser solucionados através do sistema. As observações se concentraram na clinica de cães e gatos, e superficialmente no setor de silvestres.

Fonte: hospital veterinário da UFPA Castanhal

Abaixo, a partir do Quadro 7 ao 13, é apresentado os requisitos elicitados e validados junto aos *stakeholders*, os dados estão organizados da seguinte forma: numeração de cada item coletado; é informado se o requisito é **funcional (F)** ou **não funcional (NF)**, e somado a **descrição** do requisito, percebe-se as capacidades e restrições que o sistema veterinário deve

atender, com isso direcionado o objetivo a ser alcançado na modelagem e construção do *software*; e por fim, destaca-se a necessidade dos *stakeholders* em relação aos requisitos, do qual é categorizado como **alta (A)**, **media (M)** ou **baixa (B)**.

Quadro 7 – Requisitos de proprietário e usuário

N ^a	F/NF	A/M/B	Descrição
R1	F	A	O sistema possui cadastro de usuários.
R2	NF	A	Usuário do sistema é somente pessoa física.
R3	F	A	Usuário do sistema possui cargo e função no HV.
R4	F	A	Conforme o cargo e função no HV, o usuário adquire permissões para utilizar funcionalidades do sistema.
R5	NF	A	Ao cadastrar usuário do sistema, é obrigatório RG e CPF.
R6	NF	A	Ao cadastrar usuário do sistema, é obrigatório: nome de usuário; senha e e-mail.
R7	F	A	Para entrar no sistema, usuário deve fornecer a senha de acesso, e informar nome de usuário ou e-mail.
R8	F	A	O sistema possui cadastro de proprietários.
R9	NF	A	Proprietário pode ser pessoa física.
R10	NF	B	Proprietário pode ser pessoa jurídica.
R11	F	A	Ao cadastrar proprietário (pessoa física), é necessário fornecer RG ou CPF, ou ambos.
R12	NF	A	Ao cadastrar pessoa física, seja proprietário ou usuário do sistema, é obrigatório os seguintes dados: nome; gênero; UF; cidade; bairro; logradouro; número do imóvel; e telefone.
R13	FN	B	Ao cadastrar pessoa jurídica, é obrigatório os seguintes dados: nome; UF; cidade; bairro; logradouro; número do imóvel; telefone e CNPJ.
R14	F	M	Ao cadastrar proprietário ou usuário do sistema, o complemento para endereço é opcional.
R15	F	A	Proprietários e usuários do sistema podem possuir até 3 telefone no registro.
R16	NF	A	É obrigatório proprietário e usuário do sistema possuir 1 telefone no registro.
R17	F	M	Usuário do sistema também pode ser proprietário.
R18	F	A	Proprietário possui 1 ou vários animais associados ao seu registro.
R19	NF	A	No setor de silvestres, o "responsável" é equivalente ao "proprietário".

Fonte: Produzido pelo autor

Quadro 8 – Requisitos de animais

N^a	F/NF	A/M/B	Descrição
R1	F	A	O sistema possui cadastro de animais pequenos.
R2	NF	A	Somente é considerado animal pequeno cão ou gato.
R3	F	M	O sistema registra o historio de pesos do animal pequeno.
R4	NF	A	Ao cadastrar animal pequeno, é obrigatório informar: nome do animal; especie; raça; idade escrita por extenso; e gênero do animal.
R5	F	A	Caso a raça do cão ou gato não seja identificada, deve ser informado S.R.D (Sem Raça Definida).
R6	F	M	Ao informar a raça do animal pequeno, o sistema disponibiliza uma listagem de raças para selecionar uma.
R7	F	A	Ao informar a raça do animal pequeno, o sistema permite registrar uma nova raça.
R8	F	A	O sistema registra classe, ordem, gênero, especie e raças de animais silvestres.
R9	F	A	Um animal silvestre pode ser associados a no mínimo 1 ou vários responsáveis.

Fonte: Produzido pelo autor

Quadro 9 – Requisitos para agendar consultas

N^a	F/NF	A/M/B	Descrição
R1	F	A	O sistema registra o agendamento de consultas.
R2	NF	A	Ao agendar consulta o tipo deve ser informado, sendo “nova consulta” ou “retorno”.
R3	F	A	O sistema permite confirmar consultas agendadas.
R4	F	A	O sistema permite buscar agendamentos confirmados.
R5	NF	A	Proprietário e animal pequeno são registrados durante a confirmação de consulta agendada.
R6	F	A	O sistema permite cancelar consultas agendadas.
R7	F	A	O sistema permite buscar agendamentos cancelados.
R8	F	A	O sistema permite editar consultas agendadas.
R9	NF	M	Somente o recepcionista pode registrar, confirmar, cancelar ou editar agendamentos.
R10	NF	M	Somente o recepcionista pode buscar agendamentos confirmados ou cancelados.

Fonte: Produzido pelo autor

Quadro 10 – Requisitos para consulta

N^a	F/NF	A/M/B	Descrição
R1	F	A	O sistema registra dados de consultas.
R2	F	A	O médico veterinário deve selecionar um agendamento confirmado para iniciar uma nova consulta.
R3	F	A	O sistema registra o formulário de anamnese.
R4	F	A	O sistema registra o formulário de exame físico.
R5	F	A	O sistema registra formulários de exame físico específico.
R6	F	A	São formulários de exame físico específico são: digestório e glândulas anexas; respiratório e cardiovascular; gênito-urinário e glândulas mamárias; tegumentar; neurológico; oftálmico e músculo-esquelético.
R7	F	A	O sistema registra o formulário para solicitar radiografia.
R8	F	A	O sistema registra o formulário para solicitar ultrassonografia.
R9	F	M	O sistema registra o formulário de diagnósticos diferenciais.
R10	F	A	O sistema possui formulário para selecionar procedimentos realizados no cliente.
R11	F	A	Ao selecionar procedimentos, é disponibilizado botão para solicitar desconto.
R12	NF	A	Somente médico veterinário do tipo técnico ou professor pode conceder o valor do desconto.
R13	NF	A	Consultas do tipo "retorno", possuem isenção somente no valor do atendimento.
R14	NF	A	A consulta é composta pelos seguintes formulários: selecionar agendamento confirmado; anamnese; exame físico; exame físico específico; solicitar radiografia; solicitar ultrassonografia; diagnósticos diferenciais; selecionar procedimentos a serem aplicados.
R15	NF	A	Durante a consulta é obrigatório o preenchimento dos seguintes formulários: anamnese; exame físico; diagnósticos diferenciais.
R16	NF	A	Durante a consulta é opcional o preenchimento dos seguintes formulários: exame físico específico; solicitar radiografia; solicitar ultrassonografia; selecionar procedimentos a serem aplicados.
R17	F	A	Para finalizar a consulta, o médico veterinário informa sua senha de acesso ao sistema e CRMV.
R18	NF	A	Somente o médico veterinário pode registrar consultas.

Fonte: Produzido pelo autor

Quadro 11 – Requisitos de exame por imagem

N^a	F/NF	A/M/B	Descrição
R1	F	A	O sistema permite pesquisar radiografias pendentes.
R2	F	A	O sistema registra o formulário de radiografia.
R3	F	A	O sistema permite pesquisar ultrassonografias pendentes.
R4	F	A	O sistema registra o formulário de ultrassonografia.
R5	F	A	Ao preencher o formulário de radiografia ou ultrassonografia, deve ser apresentado as seguintes informações do animal pequeno: RGHV; nome; espécie; raça; peso; gênero; e idade.
R6	F	M	Ao preencher o formulário de radiografia ou ultrassonografia, deve ser apresentado as seguintes informações: nome do residente que solicitou o exame por imagem; suspeita clínica; e região.
R7	NF	A	Somente o radiologista pode registrar radiografias.
R8	NF	A	Somente o técnico em ultrassom pode registrar ultrassonografias.

Fonte: Produzido pelo autor

Quadro 12 – Requisitos para procedimentos do HV

N^a	F/NF	A/M/B	Descrição
R1	F	A	O sistema permite cadastrar novos procedimentos do HV.
R2	NF	A	Um procedimento representa um serviço ou conjunto de insumos e processos necessários para desempenhar uma tarefa do HV.
R3	F	A	O sistema permite listar procedimentos.
R4	F	A	O sistema permite editar procedimentos.
R5	F	A	O sistema permite habilitar ou desabilitar procedimentos.
R6	NF	A	Cada procedimento possui no mínimo 1 ou vários preços.
R7	NF	A	Cada preço possui uma ou nenhuma categoria de peso.
R8	NF	A	São categorias de peso: pequeno porte (PP); médio porte (MP); grande porte (GP).
R9	F	B	O sistema permite registrar novas categorias de peso.
R10	F	B	O sistema permite listar categorias de peso.
R11	F	B	O sistema permite habilitar ou desabilitar categorias de peso.
R12	F	B	O sistema permite editar categorias de peso.

Fonte: Produzido pelo autor

Quadro 13 – Requisitos para controle de fichas e entrada de caixa

N^a	F/NF	A/M/B	Descrição
R1	F	A	O sistema permite listar consultas efetivadas.
R2	F	A	Para listar consultas efetivadas, o usuário informa um intervalo de datas.
R3	F	A	Ao listar consultas efetivadas, é disponibilizado filtros para auxiliar a busca por um dado item.
R4	F	A	Ao listar consultas efetivadas, é informado o custo total de cada consulta.
R5	F	A	Ao listar consultas efetivadas, é informado o status de pagamento de cada consulta.
R6	F	A	Ao listar consultas efetivadas, o sistema inclui o desconto ao custo total de cada consulta, caso tenha sido concedido.
R7	F	A	Ao listar consultas efetivadas, o sistema informa o total pendente a ser pago por clientes.
R8	F	A	Ao listar consultas efetivadas, o sistema informa o total pago por clientes.
R9	F	A	Ao clicar no custo total de um item da lista de consultas efetivadas, é listado todos os procedimentos realizados na consulta.
R10	F	A	Ao listar procedimentos realizados na consulta efetivada, é permitido alterar o status de pagamento de cada procedimento.
R11	NF	M	Somente o recepcionista pode utilizar a funcionalidade "Controle de fichas e entrada de caixa".

Fonte: Produzido pelo autor

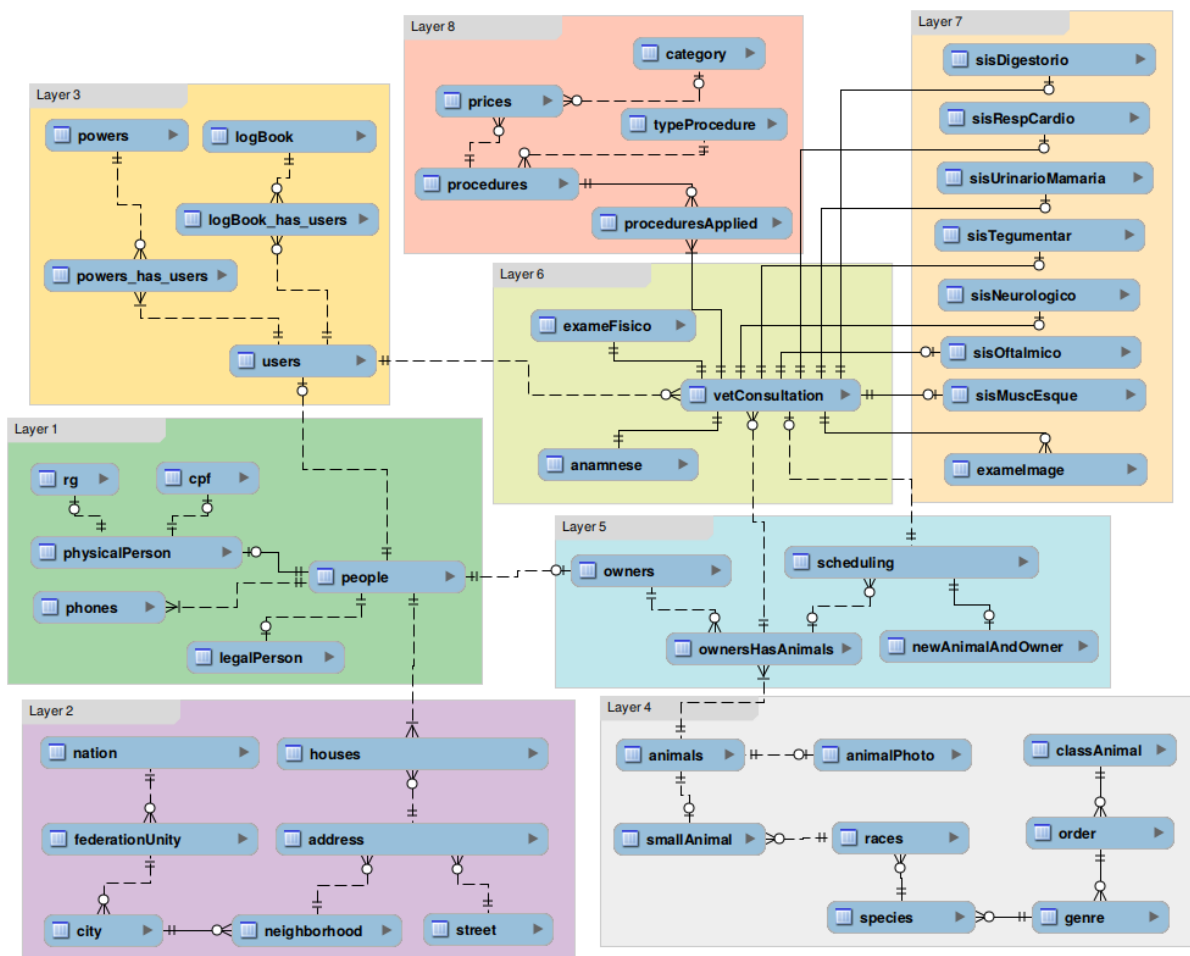
4.3 Modelagem do sistema

O sessão 4.3 apresenta artefatos produzidos e utilizados para descreverem características arquiteturais do SIHV, para isso, utilizam-se diagramações em: DER para descrever o BD; e a Linguagem de Modelagem Unificada (UML, do inglês *Unified Modeling Language*) para ilustrar a organização e funcionamento da camada de aplicação do sistema (SOMMERVILLE, 2011).

4.3.1 Banco de dados

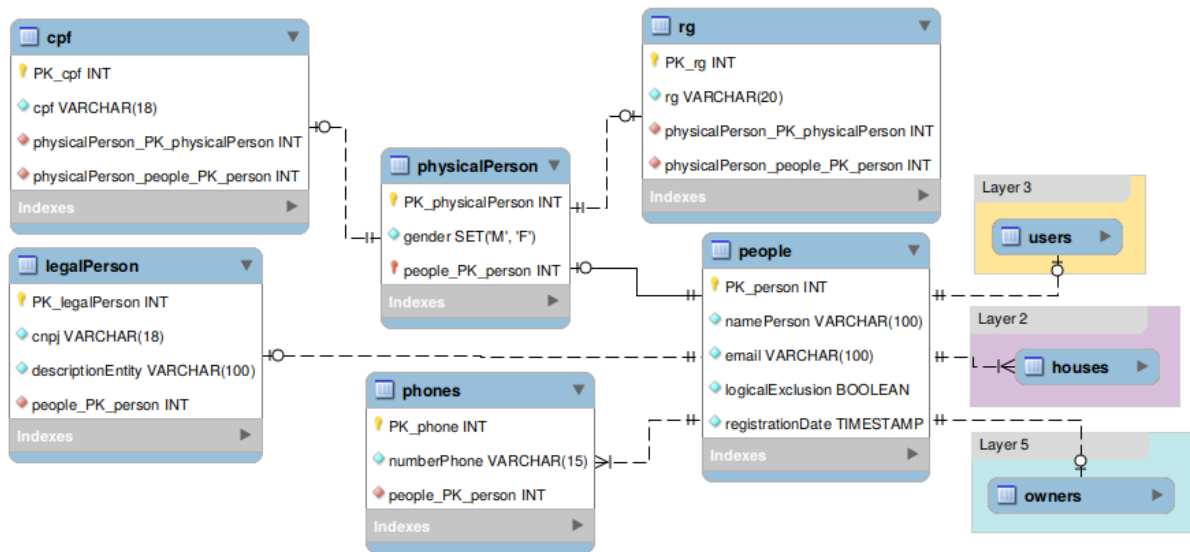
A Figura 6 apresenta em modo compactado, a visão geral do DER correspondente ao BD utilizado no SIHV, do qual se aplica na modelagem, técnicas para aprimorar o armazenamento estruturado de informações, de modo que satisfaça da melhor forma as necessidades dos *stakeholders* do HV, para isso, utilizando conhecimentos adquirido no curso de Sistemas de Informações da UFPA e referencial teórico como Sommerville (2011), Carvalho (2015) e Elmasri et al. (2005). A seguir, a partir da Figura 7 à 14, é apresentado a fragmentação da Figura 6.

Figura 6 – Diagrama de Entidade Relacionamento final



Fonte: produzido pelo autor

Figura 7 – DER responsável por armazenar dados de pessoa

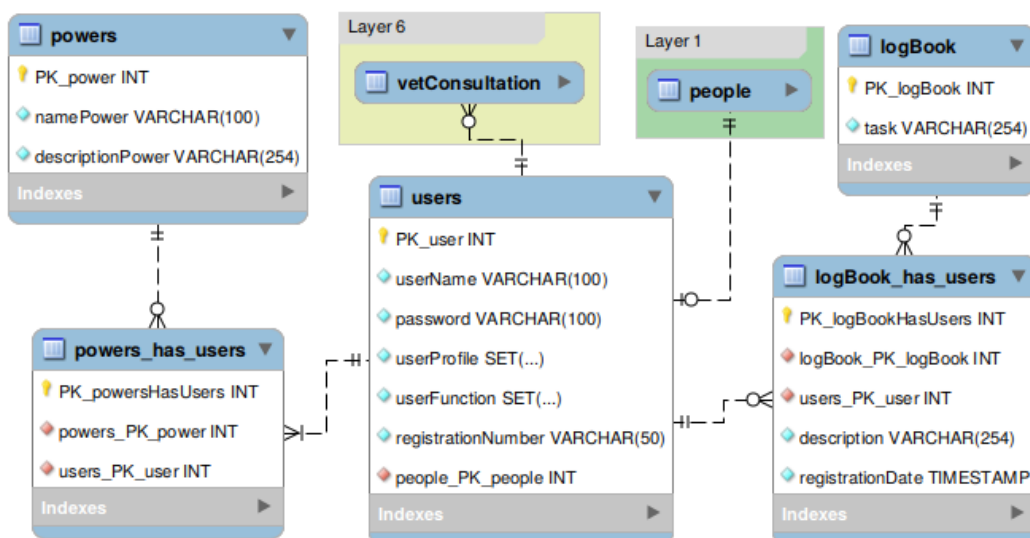


Fonte: produzido pelo autor

A Figura 7 demonstra associações da entidade *people* (pessoas no português). O qual é retratado os relacionamentos com *physicalPerson* para identificar pessoa física, e *legalPerson* como jurídica, *owners* delimita o indivíduo como proprietário e *users* como utilizador do sistema. A entidade *phones* (telefones no português) armazena números de contato com o indivíduo, e *houses* dados básicos de seu endereço.

Ainda observando o DER ilustrado na Figura 7, percebe-se que sua normalização evita campos opcionais. Exemplo disto, seria a não obrigatoriedade de persistência das entidades *cpf* ou *rg*, característica em conformidade com o 7ª Quadro de requisitos, o qual descreve proprietários que não possuem algum destes documentos.

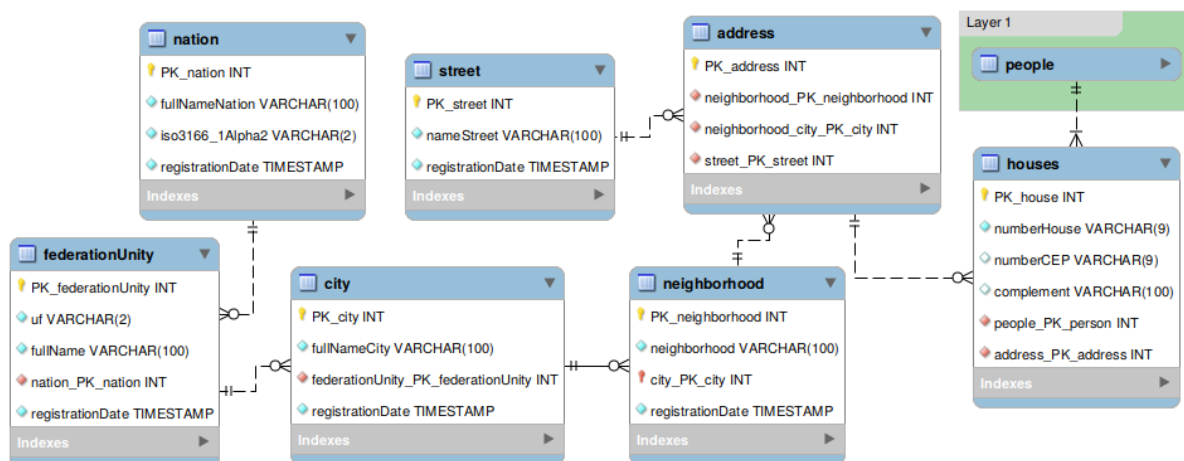
Figura 8 – DER responsável por armazenar dados de usuários do sistema



Fonte: produzido pelo autor

Utilizando a entidade *users* (usuários no português), o DER exemplificado na Figura 8 armazena perfis de usuários do sistema. As permissões de acesso são delimitadas mediante o relacionamento “muitos-para-muitos” (condição representada por N:N) com a tabela *powers*, e suas atividades são rastreadas usando a associação “N:N” entre *users* e *logBook*.

Figura 9 – DER responsável por armazenar o endereço de pessoas



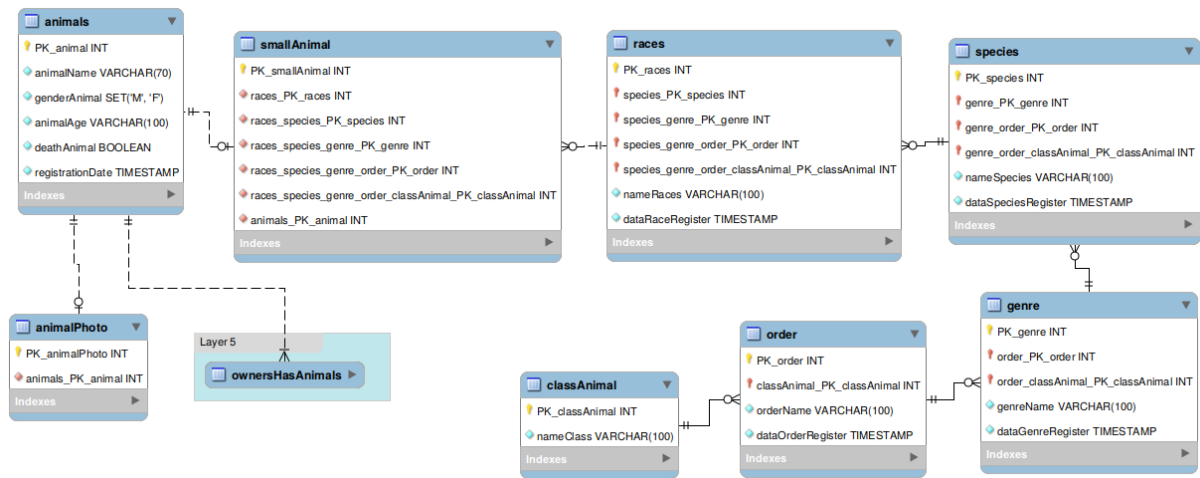
Fonte: produzido pelo autor

O conjunto de tabelas ilustradas na Figura 9, tem como responsabilidade persistir dados relacionados ao endereço de pessoas (entidade *people*), seja usuário do sistema ou cliente do HV. A normalização aplicada no DER, permite reaproveitar dados de federação (*federationUnity*), cidade (*city*), bairro (*neighborhood*) e logradouro (*street*) em cadastros posteriores. Pois, mediante observação em cima de artefatos utilizados por recepcionistas do HV, percebe-se a recorrência de nomes para os campos citados.

O relacionamento “N:N” entre as tabelas *neighborhood* e *street*, permite reutilizar endereços em diferentes registros, através da entidade associativa *address*, pois, como exemplo dois indivíduos podem residir na mesma rua e bairro, de uma determinada cidade. Porém, a distinção entre eles ocorrem na tabela *houses* (casas no português), do qual se relaciona com *people* e *address*.

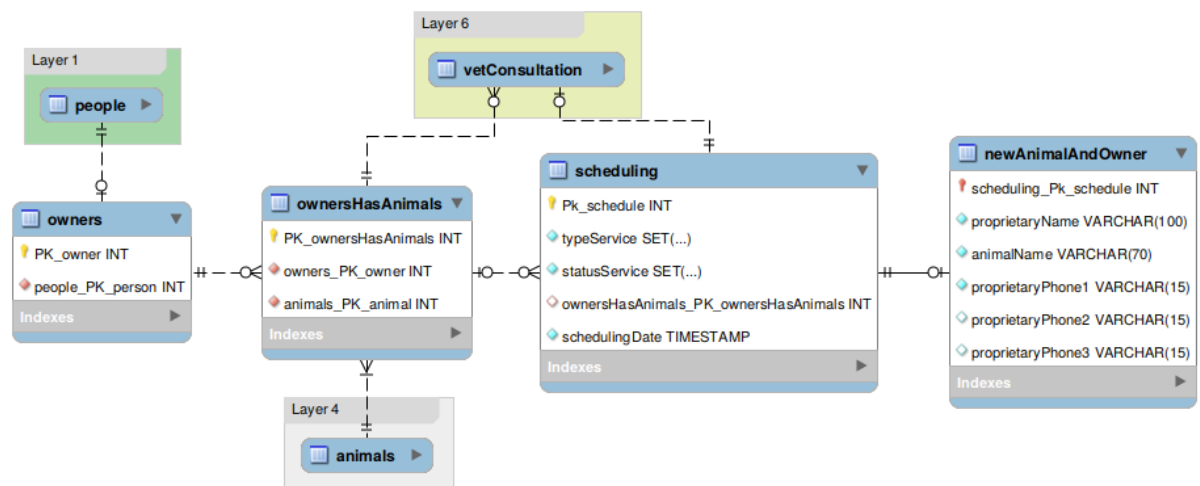
A 10ª Figura agrupa entidades referentes aos animais clientes do HV, e conforme ao 8ª Quadro de requisitos. Armazenam-se os seguintes dados: informações básicas do animal na tabela *animals*; animais atendidos pelo setor de pequenos, são identificados mediante relacionamento com entidade *smallAnimal*; a associação modelada em tabelas para catalogar raças (*races*), espécies (*species*), gêneros (*genre*), ordens (*order*) e classes (*classAnimal*), permite reutilizar tais informações em registros posteriores.

Figura 10 – DER responsável por armazenar dados de animais



Fonte: produzido pelo autor

Figura 11 – DER responsável por armazenar dados de agendamentos de consultas

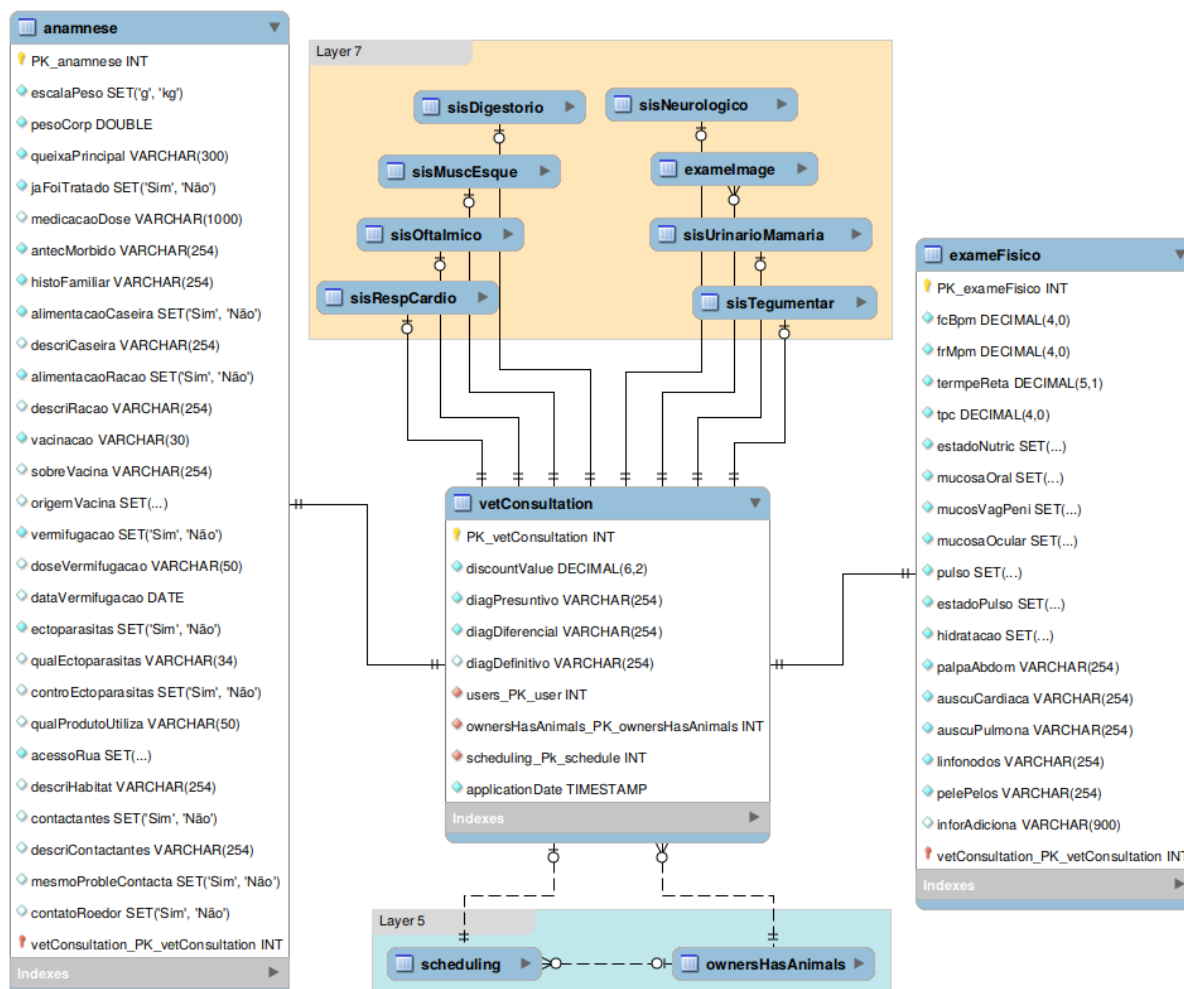


Fonte: produzido pelo autor

Acima, o DER ilustrado na Figura 11 promove dois aspectos, o primeiro refere-se ao agendamento de consultas, do qual a entidade *scheduling* (agendamento no português) armazena: o *status* de um serviço (coluna: *statusService*) e seu tipo (coluna: *typeService*); data e hora da consulta (coluna: *schedulingDate*); associação com cliente mediante chave estrangeira (coluna: *ownersHasAnimals_PK_ownersHasAnimals*); e dados temporários do proprietário em tabela “*newAnimalAndOwner*”, pois seu registro ou confirmação somente ocorre ao comparecer no dia da consulta.

O segundo aspecto tange ao cliente, quando um proprietário (tabela: *owners*) e seu animal (tabela: *animals*) tem o registro efetuado no sistema, um relacionamento “N:N” entre eles é criado. Caso já exista, a chave primária da entidade associativa é utilizada em agendamentos (tabela: *scheduling*), e posteriormente em novas consulta (tabela: *vetConsultation*).

Figura 12 – DER responsável por armazenar dados básicos de consultas



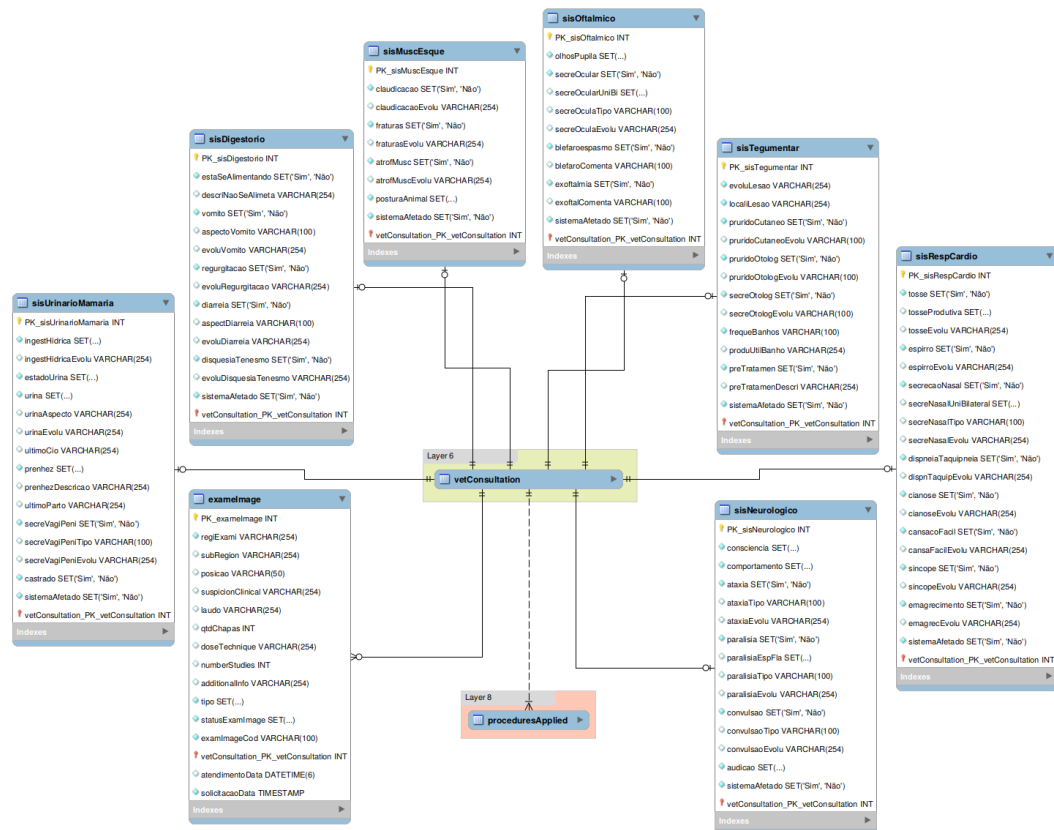
Fonte: produzido pelo autor

Conforme artefatos e opinião consultada de *stakeholders* participantes do projeto, o DER ilustrado na Figura 12, destaca entidades referentes ao exame clínico veterinário. Sendo este, constituído por tabelas de anamnese e exame físico geral (intitulado: exameFisico). Tais entidades objetivam padronizar o formulário de perguntas aplicadas em consultas no HV.

Durante o exame, a consulta é identificada pela entidade “*vetConsultation*”, a partir dela associa-se tabelas que identificam para qual agendamento (*scheduling*) e cliente (*ownersHasAnimals*) destina-se o serviço. Bem como quais categorias de exames foram aplicados, sendo eles do tipo clínico ou físico específico (representado pelo conjunto de tabelas compactadas: *sisDigestorio*; *sisMuscEsque*; *sisOftalmico*; *sisRespCardio*; *sisNeurologico*; *sisUrinarioMamaria*; *sisTegumentar* e *exameImage*).

A 13ª Figura apresenta entidades que compõem a categorias de exames físicos específicos. O preenchimento delas é opcional, pois somente ocorre quando o veterinário atesta que um sistema orgânico (exemplo: sistema digestivo; neurológico; entre outros) do animal examinado apresenta alteração.

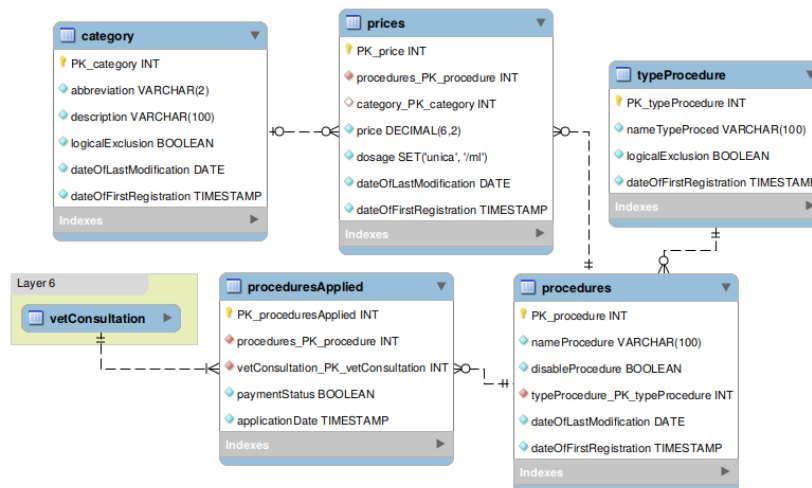
Figura 13 – DER responsável por armazenar dados de exames específicos



Fonte: produzido pelo autor

A última modelagem em DER ilustrada na Figura 14, compreende entidades responsáveis por armazenar aspectos relacionados aos procedimentos aplicados no HV. O qual se destaca a associação “N:N” entre “*vetConsultation*” e *procedures* (procedimentos no português), para rastrear quais serviços foram aplicados em uma consulta.

Figura 14 – DER responsável por armazenar dados de procedimentos do HV

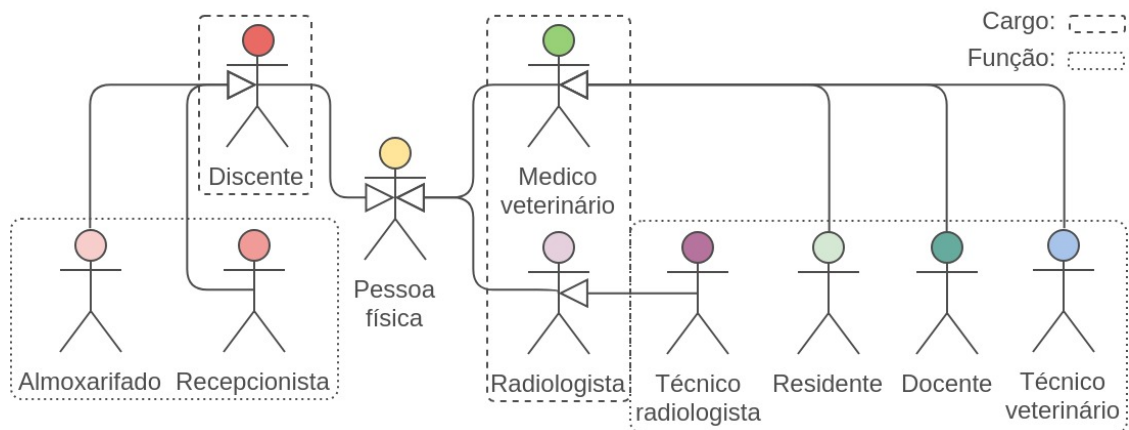


Fonte: produzido pelo autor

4.3.2 Diagramas de caso de uso

Neste tópico apresenta-se em diagramas de caso de uso, a descrição de características funcionais e de interação com a aplicação SIHV, demonstrando capacidades e limitações de utilização do sistema, através das ilustrações a seguir.

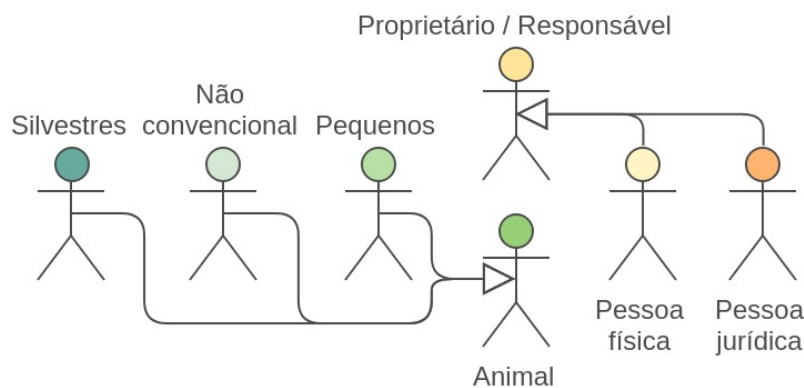
Figura 15 – Caso de uso: usuários do sistema



Fonte: Produzido pelo autor utilizando as fontes: (SOMMERVILLE, 2011; JGRAPH, 2019)

O primeiro caso de uso ilustrado na Figura 15, apresenta os *stakeholders* diretos do sistema, do qual se demonstra a generalização e especialização de utilizadores da aplicação veterinária, e conforme o requisito “R4” apresentado no 7ª Quadro, essa divisão equivale ao cargo (retângulo tracejado) e função (retângulo pontilhado) do usuário.

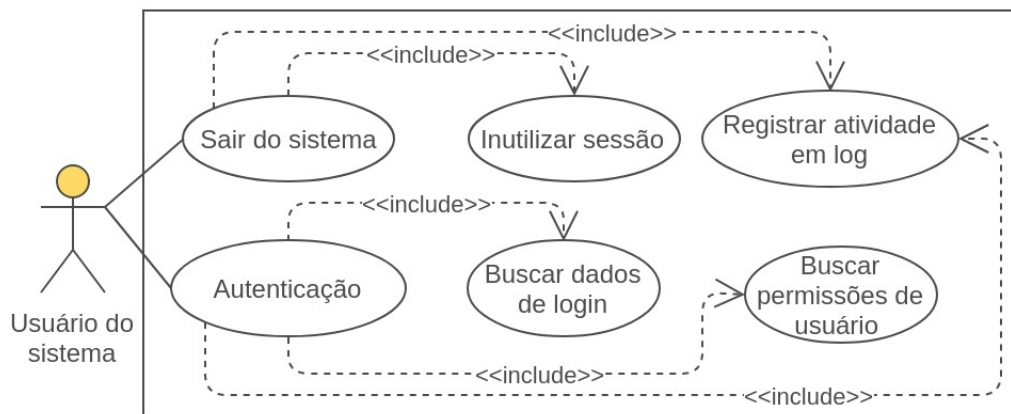
Figura 16 – Caso de uso: clientes do HV



Fonte: Produzido pelo autor utilizando as fontes: (SOMMERVILLE, 2011; JGRAPH, 2019)

A Figura 16 apresenta os *stakeholders* indiretos do sistema, esse grupo de indivíduos é composto por clientes que usufruem de serviços prestados pelo HV. Os atores que compõem a imagem acima são: o proprietário ou responsável pelo animal, podendo ser pessoa física ou jurídica; e o animal, possuindo as especializações de pequenos, não convencionais e silvestres.

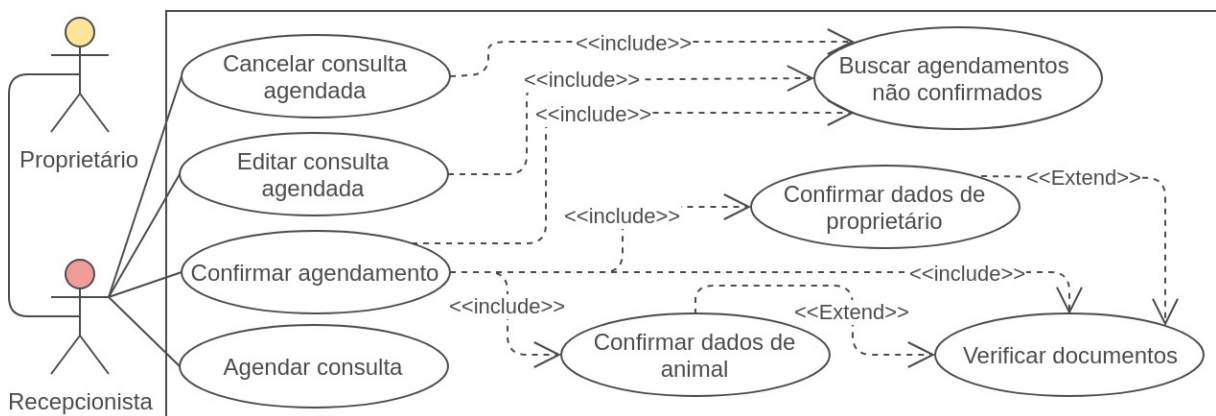
Figura 17 – Caso de uso: login e logout do sistema



Fonte: Produzido pelo autor utilizando as fontes:(SOMMERVILLE, 2011; JGRAPH, 2019)

O caso de uso exemplificado na Figura 17, apresenta a modelagem de funcionalidades realizadas quando o usuário efetuar *login* e *logout* no sistema. Primeiramente, ao autenticar-se/entrar na aplicação, verificam-se credenciais de acesso. Caso *login* e senha se confirmem, é carregado as permissões do usuário, e sua entrada é registrada em *log* de atividades do utilizador. Ao sair do sistema, a sessão será inutilizada, liberando recurso no servidor.

Figura 18 – Caso de uso: funcionalidades de agenda

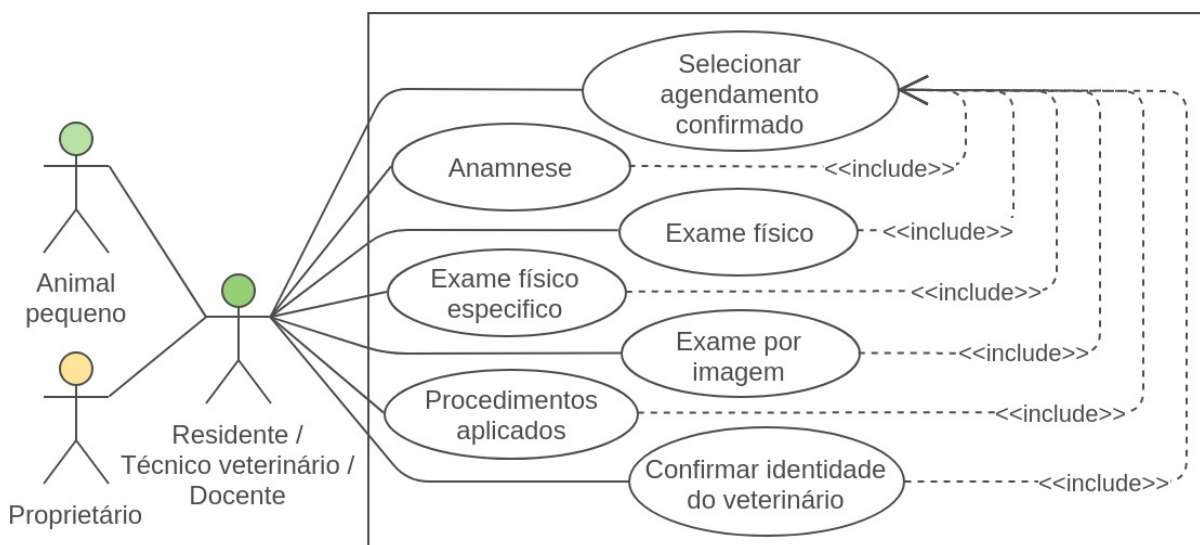


Fonte: Produzido pelo autor utilizando as fontes:(SOMMERVILLE, 2011; JGRAPH, 2019)

O caso de uso da Figura 18, representa funcionalidades para controlar o agendamento de consultas, tais funções direcionam-se ao uso do *stakeholder* recepcionista. A modelagem estabelece ao usuário sendo capaz de agendar consulta, editar ou cancelar agendamentos com *status* de não confirmado.

Ainda na Figura 18, ao confirmar agendamento para um cliente, o proprietário deve fornecer ao recepcionista documento de identificação (exemplo, rg e cpf), com finalidade de: verificar a validade do CPF; carregar dados do cliente no formulário, caso RG e CPF já existam no registro. Do contrário, libera-se tela para cadastrar proprietário e animal para o qual se destina a consulta.

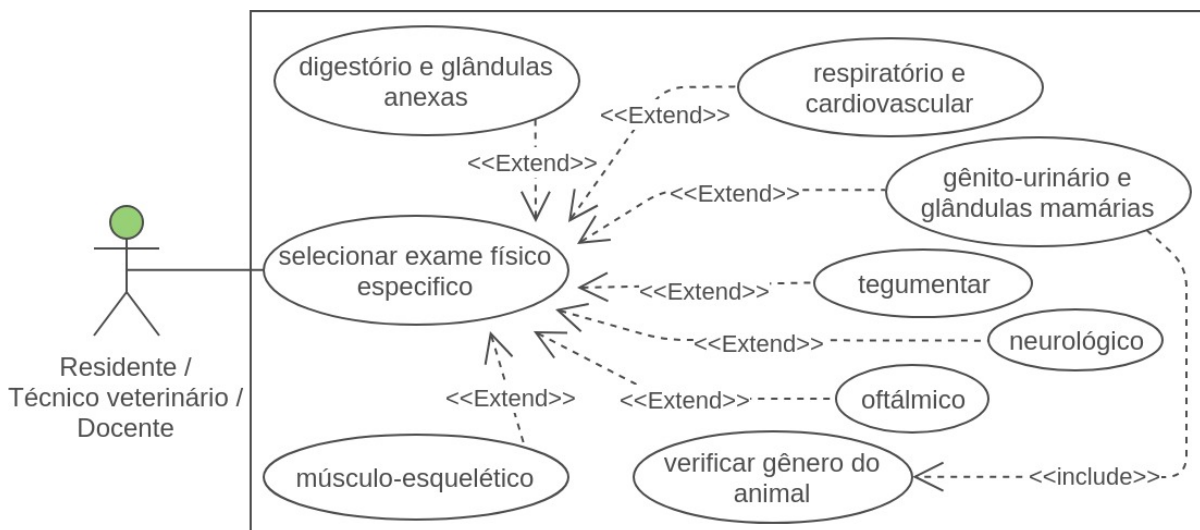
Figura 19 – Caso de uso: nova consulta



Fonte: Produzido pelo autor utilizando as fontes:(SOMMERVILLE, 2011; JGRAPH, 2019)

A Figura 19 compreende etapas que uma consulta pode aplicar através de funcionalidades dispostas no caso de uso, e para utilizá-las, é necessário selecionar um agendamento confirmado pelo cliente. O uso de tais funções direcionam-se ao ator médico veterinário com uma das seguintes funções profissionais: residente; técnico ou docente. Abaixo, a partir da Figura 20 ao 22, ocorre a desfragmentação da Figura 19 em sub casos de uso.

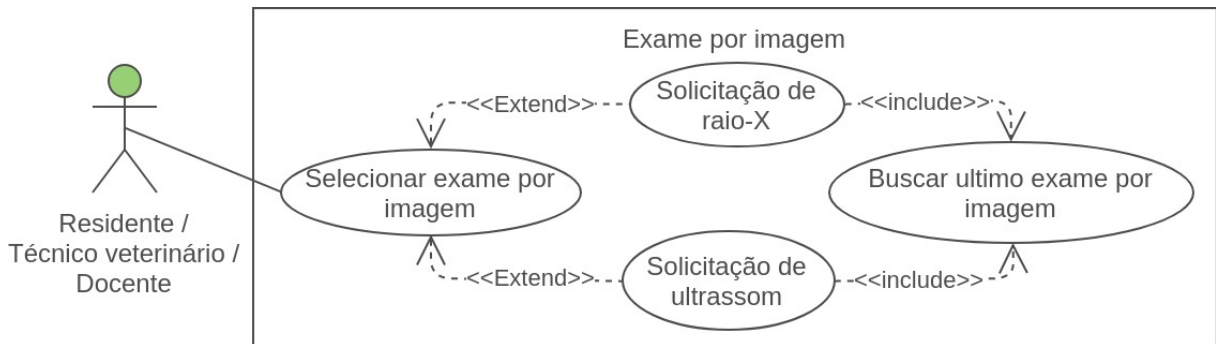
Figura 20 – Caso de uso: funcionalidades de exame físico específico



Fonte: Produzido pelo autor utilizando as fontes:(SOMMERVILLE, 2011; JGRAPH, 2019)

A 20ª Figura compreende casos de uso do “exame físico específico”, e sua modelagem permite ao veterinário selecionar quais formulários necessita preencher, conforme avaliação clínica realizada sobre o animal atendido. Observa-se também na imagem, que o caso de uso referente ao sistema específico “gênito-urinário e glândulas mamárias”, verifica o gênero do animal selecionado para a consulta, e adapta o conjunto de respostas disponíveis no exame.

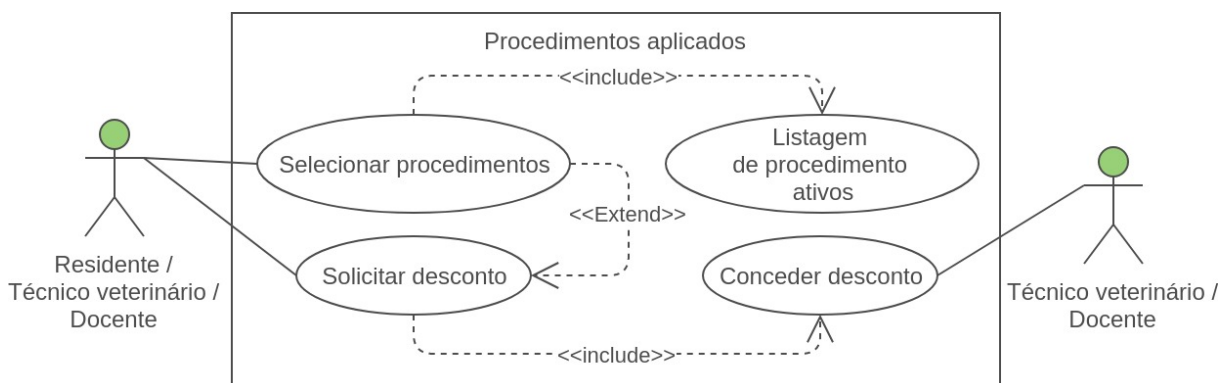
Figura 21 – Caso de uso: funcionalidades de exame por imagem



Fonte: Produzido pelo autor utilizando as fontes:(SOMMERVILLE, 2011; JGRAPH, 2019)

O caso de uso ilustrado na Figura 21, exemplifica a etapa de solicitação de exames por imagem, dentre eles o “raio-x” e ultrassom, cabendo ao veterinário quando aplicá-los. Ao solicitar “raio-x”, ultrassom ou ambos, o sistema contabiliza o valor máximo de exames por imagem registrados, e apresenta em tela o código identificado do próximo exame.

Figura 22 – Caso de uso: funcionalidades de procedimentos aplicados

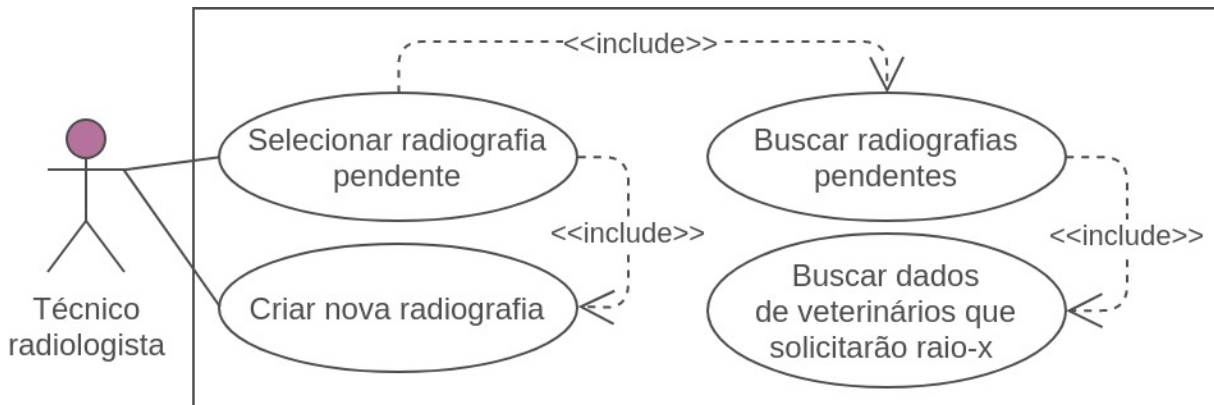


Fonte: Produzido pelo autor utilizando as fontes:(SOMMERVILLE, 2011; JGRAPH, 2019)

A Figura 22 apresenta aspectos relacionados a escolha de procedimentos aplicados pelo veterinário durante a consulta. O caso de uso também expressa que somente é selecionado serviços com *status* ativo, e opcionalmente podendo ser solicitado desconto sobre os custos advindo do mesmo, cabendo aos preceptores do HV (medico veterinário com função de técnico ou docente) conceder a dedução informando o valor descontado.

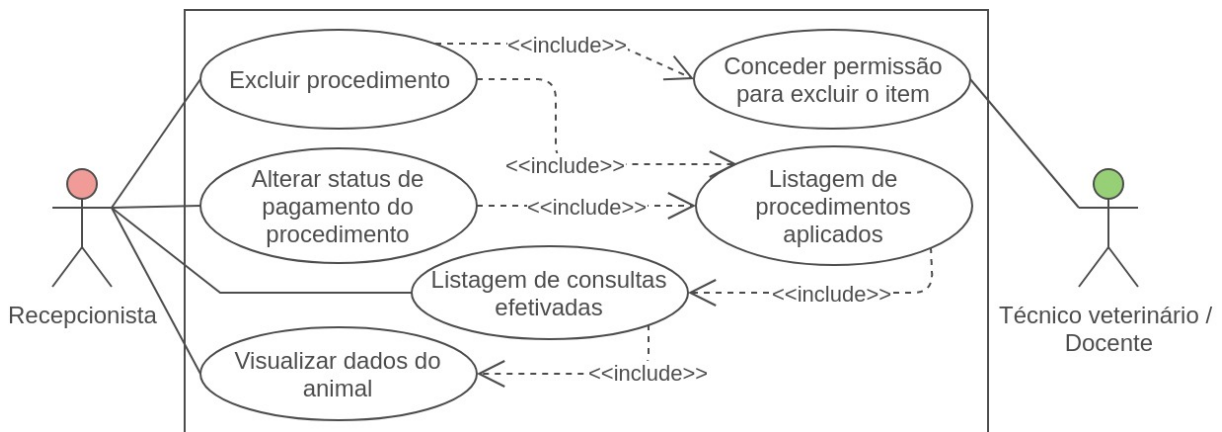
O caso de uso apresentado na Figura 23, compreende funcionalidades direcionadas ao *stakeholder* radiologista. O qual permite ao profissional buscar no registro do sistema, solicitações de radiografias emitidas por médicos veterinários durante a realização de consultas. Ao selecionar uma requisição de “raio-x”, o ator radiologista visualiza dados/descrições fornecidas pelo veterinário, que o orientaram na condução de seu trabalho, e preenchimento do formulário referente a nova radiografia.

Figura 23 – Caso de uso: funcionalidades de radiografias pendentes



Fonte: Produzido pelo autor utilizando as fontes:(SOMMERVILLE, 2011; JGRAPH, 2019)

Figura 24 – Caso de uso: funcionalidades para controle de fichas e entrada de caixa



Fonte: Produzido pelo autor utilizando as fontes:(SOMMERVILLE, 2011; JGRAPH, 2019)

O caso de uso ilustrado na Figura 24, retrata a modelagem de funcionalidades para administrar o *status* de pagamento de serviços realizados em consultas. A utilização de tais funções direcionam-se ao ator recepcionista, por meio delas pode: listar consultas efetivadas, bem como procedimentos aplicados em cada uma; confirmar ou negar o pagamento sobre cada serviço empregado em uma consulta; visualizar dados do cliente. Por fim, excluir a aplicação de procedimentos numa consulta para casos de desistência ou ressarcimentos, porem, a ação demanda confirmação de preceptores do HV.

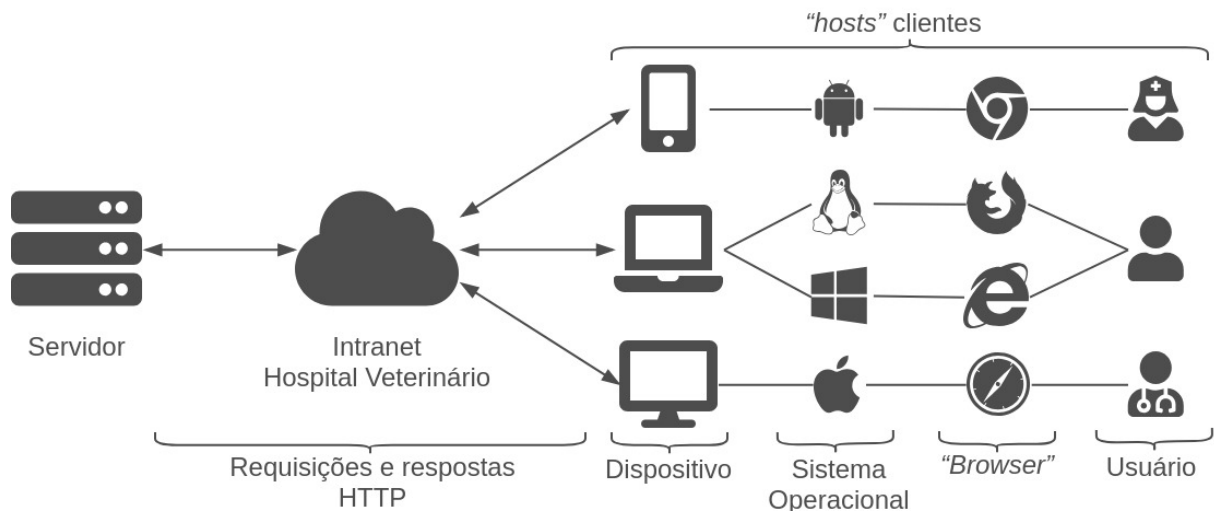
4.4 Arquiteturas utilizadas

A sessão 4.4 apresenta arquiteturas empregadas no projeto, no qual a primeira consiste no modelo de comunicação utilizado pelo *software*, para disseminar suas ferramentas no ambiente de trabalho de seus usuários. A segunda direciona-se para a estrutura organizacional de arquivos que compõem o programa, assim, objetivando auxiliar desenvolvedores a compreender interações internas do sistema com camadas.

4.4.1 Cliente/Servidor

O *software* veterinário utiliza a arquitetura de comunicação intitulada cliente-servidor, escolha necessária devido a tecnologias de desenvolvimento e implementação aplicadas no projeto, do qual citamos no capítulo 3: JSF; EJB; PrimeFaces e GlassFish. Recursos que direcionam o escopo do programa ao ambiente *web* (ANDRADE, 2015a; KUROSE JAMES F.; ROSS, 2013).

Figura 25 – Rede cliente-servidor



Fonte: Produzido pelo autor utilizando as fontes:

(ANDRADE, 2015a; KUROSE JAMES F.; ROSS, 2013; FONTAWESOME, 2018; JGRAPH, 2019)

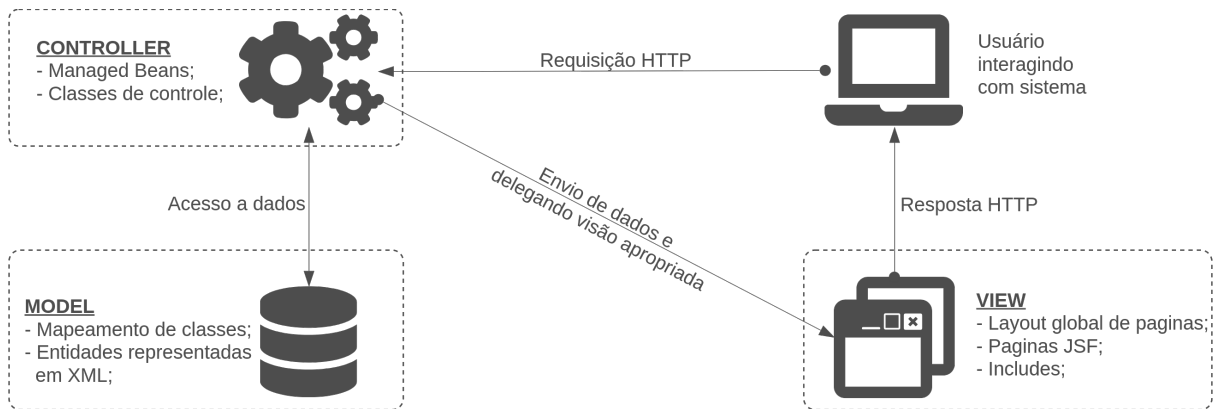
O SIHV é um *web-software* fornecido em rede com auxílio do ambiente de execução. Espaço formado por um conjunto de programas instalados no hospedeiro denominado servidor, que conforme a Figura 25, configuram o meio para disponibilizar mediante requisições e respostas HTTP, funcionalidades da aplicação veterinária aos *stakeholders* diretos, ou seja, clientes da arquitetura cliente-servidor (ANDRADE, 2015a; KUROSE JAMES F.; ROSS, 2013).

A aplicação por utilizar tecnologias da internet, independe de instalação, plataforma do dispositivo ou sistema operacional. Necessitando apenas do *browser* como via de acesso para suas ferramentas, o que caracteriza o SIHV com SaaS, somado a isto, temos a responsividade de componentes do PrimeFaces. Permitindo a adaptação das UI a diferentes dimensões de tela, em dispositivos que acesse o *software* por comunicação *web* conforme ilustrado na Figura 25 (SOMMERVILLE, 2011; ANDRADE, 2015a; CORDEIRO, 2014).

4.4.2 Modelo-Controle-Visão

A aplicação utiliza o padrão arquitetural de *software* intitulado Modelo Controle e Visão (MVC, do inglês *Model View Controller*), técnica que busca simplificar o desenvolvimento de programas construídos com POO, através da divisão de responsabilidades em três camadas, conforme ilustrado na Figura 26 (LUCKOW; MELO, 2010; ANDRADE, 2015a).

Figura 26 – Padrão arquitetural em 3 camadas



Produzido pelo autor utilizando as fontes:

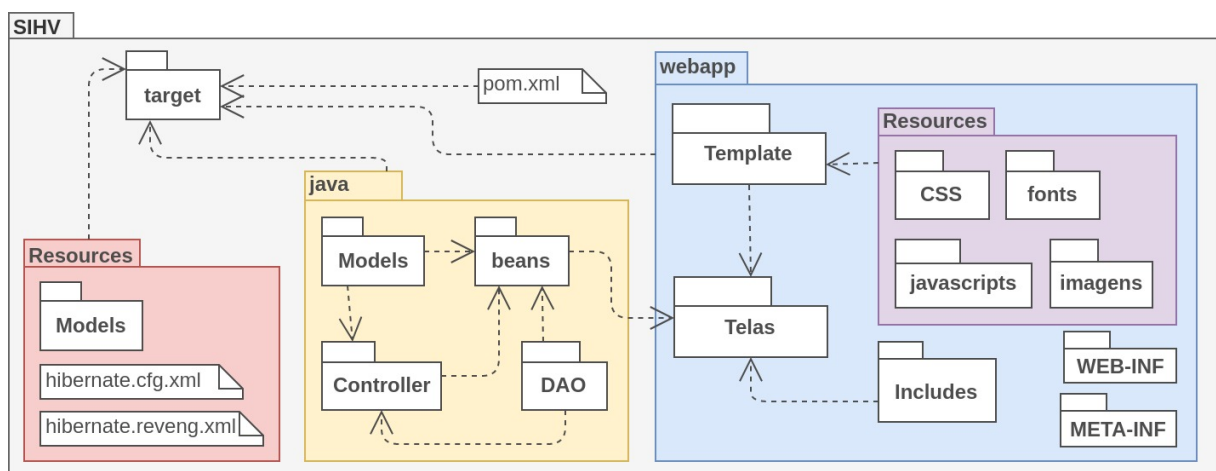
(LUCKOW; MELO, 2010; ANDRADE, 2015a; FONTAWESOME, 2018; JGRAPH, 2019)

A camada *Model* fornece o acesso a dados persistidos, disponibilizando e gerenciando objetos que representem a lógica de negócio em BD; a camada *View* determina a forma de apresentação das informações mediante UI, sendo este nível responsável por agrupar as telas de usuário da aplicação; e por fim, *Controller* mantém a comunicação das camadas Modelo e Visão, interpretando as requisições dos usuários, verificando/realizando mudanças em dados salvos na *Model* e retornando a requisição com uma renderização de *interface* adequada ao pedido (LUCKOW; MELO, 2010; ANDRADE, 2015a).

4.4.3 Organização interna de arquivos do SIHV

A Figura 27 ilustra o diagrama de pacotes da aplicação desenvolvida, sendo uma técnica de diagramação estrutural prevista na UML, que meio de recursos visuais, apresenta a disposição/organização de artefatos internos do *software* que atuam no funcionamento do programa (SILVA; VIDEIRA, 2001; GUEDES, 2011).

Figura 27 – Diagrama de pacotes do sistema



Fonte: Produzido pelo autor usando fontes:(JGRAPH, 2019; SILVA; VIDEIRA, 2001; GUEDES, 2011)

O diagrama de pacotes também propicia visualizar o agrupamento lógico de mecanismos presentes no *software*, cujo funcionamento dependam de outras partes ou subsistemas, e observando a Figura 27, temos setas direcionais que apontam para diretórios que englobam algoritmos/*scripts* que utilizam componentes externos para realizar determinada função, exemplo disso, a pasta *target* agrupa todos os artefatos para gerar o arquivo WAR utilizado para executar o sistema (VARANASI; BELIDA, 2014; SILVA; VIDEIRA, 2001; GUEDES, 2011).

A organização desses artefatos é aplicada através do programador, ou por configuração pré-estabelecida em ferramentas de desenvolvimento utilizadas no projeto. Ambas objetivam aplicar duas práticas, a primeira consiste em viabilizar o reuso de objetos e funções do sistema direcionados em uma única função, e junto a isso também promovendo a segunda, que se fundamenta na divisão de responsabilidades, técnica voltada para a especialização de artefatos. A exemplo disto, temos o diretório DAO contendo classes Java que disponibilizam as operações de CRUD utilizada no sistema (SOMMERVILLE, 2011; CAELUM, 2017b).

5 PUBLICAÇÕES

O desenvolvimento do projeto culminou na produção e publicação dos seguintes artigos científicos elencados no Quadro 14 abaixo.

Quadro 14 – Publicações provenientes do SIHV

Nº	Publicação
1 ^a	VII Simpósio de Iniciação Científica e Extensão - SIEX: SOBRAL, H. M. S.; SOUZA, K. E. S.; SERUFFO, M. C. R. . Sistema Integrado do Hospital Veterinário (SIHV). Realizado em: 14 a 16 de setembro de 2016.
2 ^a	I Congresso Interinstitucional de Ensino e Extensão - CIENEX: SOBRAL, H. M. S.; SOUZA, K. E. S.; SERUFFO, M. C. R. . Sistema Integrado do Hospital Veterinário (SIHV). Realizado: Pará, 2017.
3 ^a	XVI Escola Regional de Informática Norte 2 - ERIN: SOBRAL, H. M. S.; SOUZA, K. E. S.; SERUFFO, M. C. R. . Uma aplicação Java EE para o suporte a lógica de negócio de hospitais veterinários. realizado em: Belém-PA: Sbc, 26 a 27 de Setembro de 2017.
4 ^a	Revista Interdisciplinar Científica Aplicada - RICA: SOBRAL, H. M. S.; SOUZA, K. E. S.; SERUFFO, M. C. R. .Desafios da interdisciplinaridade no desenvolvimento de sistemas: um estudo de caso. 2018.

Fonte: Produzido com a colaboração do autor

Dentre os artigos listados acima, destacamos a quarta e ultima publicação proveniente do projeto SIHV, o qual aborda um estudo de caso em cima do processo de desenvolvimento aplicado no sistema veterinário. O trabalho identifica perspectivas interdisciplinares aplicadas na construção do *software*, sendo esta condição caracterizada pelo intercambio de conhecimento proporcionado pela maior aproximação entre desenvolvedores do sistema e *stakeholders* do HV (SOBRAL; SOUZA; SERUFFO, 2018).

A quarta publicação relata um apanhado de experiências da equipe de desenvolvimento, o qual destaca condutas com maior efetividade no empreendimento, e sintetiza as explicações em recomendações para desenvolvedores de sistemas *web*, com abordagens que mesclam ações interdisciplinares ao projeto. Abaixo no Quadro 15, é disponibilizado a listagem de tais recomendações descritas no artigo (SOBRAL; SOUZA; SERUFFO, 2018).

Quadro 15 – Principais Desafios e Recomendações da Interdisciplinaridade no Desenvolvimento de Sistemas

Desafios	Recomendações
Compreensão de termos técnicos	Entender as terminologias e conhecimento técnico utilizados pelos profissionais, buscando como fontes de informações literaturas relacionadas às disciplinas diversas, documentação utilizada pelos <i>stakeholders</i> e entrevistas presenciais.
Transcrição das regras de negócio	Compreender a lógica de negócio utilizada pelo cliente através da análise documental e presencial, a partir de visita no ambiente de trabalho e entrevistas com os <i>stakeholders</i> .
Desenvolvimento de <i>interfaces</i> amigáveis	Prototipar, avaliar, otimizar e usar técnicas de IHC nos elementos gráficos que serão utilizados pelos usuários. Teste e validação contínua, para melhor UX.
Níveis de acesso	Definir privilégios de acesso a partir da implementação no sistema de um controle de acesso por nível de função
Envolvimento das pessoas no projeto	Incentivar a participação com a equipe de desenvolvimento do projeto de <i>stakeholders</i> (min. 3 e máx. 5) estratégicos dos diferentes níveis da empresa.
Experiência da equipe de desenvolvimento	Buscar o nivelamento do conhecimento da equipe que está envolvida no projeto, não levando em consideração apenas aspectos técnicos, mas características como: capacidade de aprender coisas novas, interação em grupo, pró-atividade, empatia e disposição para estudos que fogem da zona de conforto.
Documentação	Documentar todas as etapas do projeto. A documentação facilita a apropriação do projeto por novos integrantes, além de auxiliar na entrega do produto, checando o que foi prometido e o que foi desenvolvido.

Fonte: (SOBRAL; SOUZA; SERUFFO, 2018)

6 SISTEMA DESENVOLVIDO

O capítulo apresenta como resultado a proposta de aplicação desenvolvida, para isso, expondo mediante ilustrações as UI disponibilizadas no sistema veterinário, agrupando/organizando as figuras conforme o processo de utilização de funcionalidades do *web-software*.

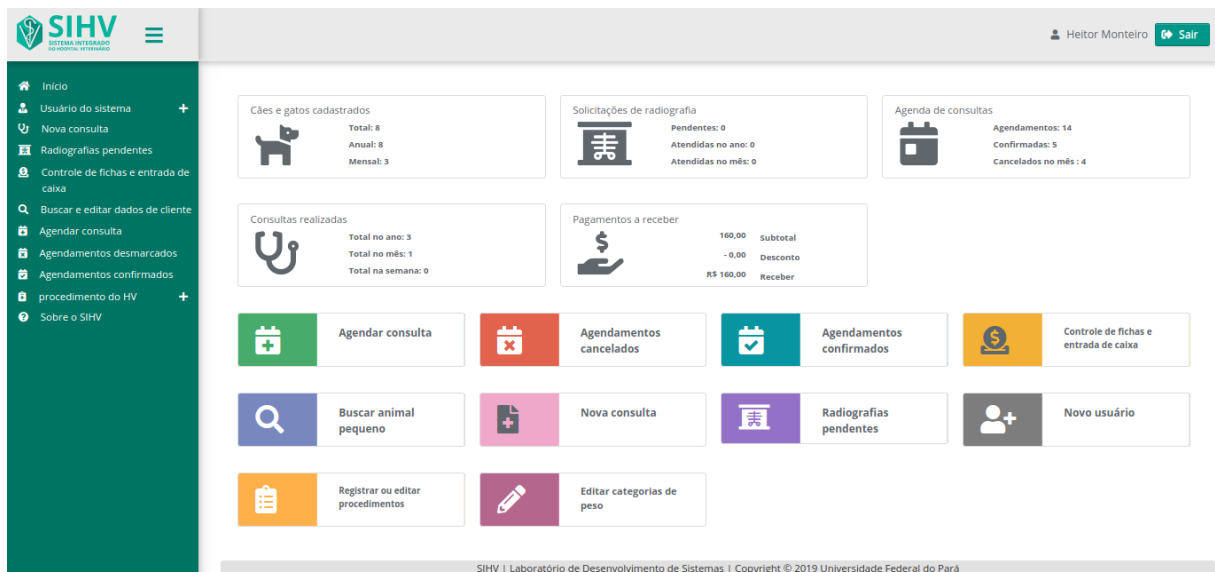
Figura 28 – Tela de *login*



Fonte: Produzido pelo autor

A tela de entrada do sistema retratado na Figura 28, demanda o preenchimento de *login* (nome de usuário ou e-mail) e senha para acessar o SIHV. Observa-se que a entrada de novos usuários somente é viabilizada, através de usuário já registrado no sistema, e com privilégio para registrar outros utilizadores.

Figura 29 – Tela principal



Fonte: Produzido pelo autor

A tela principal do sistema retratado na Figura 29, consiste no ponto de partida para seus usuários acessarem funcionalidades do *software*, e conforme seu cargo e função, ou poderes atribuídos ao mesmo. A aplicação disponibiliza funções condizentes com atribuições do indivíduo no HV.

As telas ilustradas a partir da Figura 30 até 34 são ferramentas administrativas, seu uso é direcionado aos preceptores do HV, indivíduos com cargo de médico veterinário, e com função de técnico ou docente.

Figura 30 – Telas para cadastrar usuários do sistema

Novo usuário do sistema Heitor Monteiro Sair

Dados do usuário Perfil no sistema Confirmação

CPF: Informe o campo CPF!

RG:

UF: selecione uma opção UF!

Cidade:

Bairro:

Rua:

CEP:

Numero da casa:

Complemento:

Nome: Informe o campo nome!

Sexo: Informe o campo sexo!

Endereço de email: Endereço de email: Erro de validação: o valor é necessário.

1º Celular: 1º Celular Informe o campo 1º Celular!

2º Celular: 2º Celular Informe o campo 2º Celular!

3º Celular: 3º Celular Informe o campo 3º Celular!

[próximo](#)

(A) Formulário de dados do usuário

Novo usuário do sistema Heitor Monteiro Sair

Dados do usuário **Perfil no sistema** Confirmação

Cargo profissional:

Função:

Matrícula:

Login de usuário:

Senha de usuário:

Digite novamente a senha:

[voltar](#) [próximo](#)

(B) Formulário para escolher perfil no sistema

Fonte das imagens: Produzido pelo autor

Figura 31 – Telas para finalizar cadastro de usuário do sistema

Novo usuário do sistema

Heitor Monteiro Sair

Dados do usuário Perfil no sistema **Confirmação**

Dados do usuário

CPF:	088.817.713-57	RG:	4343434
Nome:	Baltrano Silva	Sexo:	M
UF:	PA	Cidade:	Castanhal
Bairro:	Nova Olinda	Rua:	Alameda Itauba
CEP:	32342-434	Numero da casa:	22
Endereço de email:	beltrano@gmail.com	1° Celular:	(91) 96735-7435
2° Celular:	(91) 96357-4375	3° Celular:	(91) 93765-7576

Perfil no sistema

Cargo profissional:	discente	Função:	repcionista
Matricula	322342324	Login de usuário:	bel
Senha de usuário:	*****		

Poderes atribuídos à este perfil

cadastar proprietário
cadastar animal pequeno
manipular agendamentos de consultas

Salvar usuário

voltar

(C) Finalizar cadastro

Fonte das imagens: Produzido pelo autor

A Figura 30 apresenta UI para registrar um novo usuário. Iniciando com a coleta de dados cadastrais em 30 (A), em 30 (B) definem-se informações de acesso ao sistema, e conforme cargo e função escolhidos para o indivíduo. A aplicação determina quais funcionalidades o mesmo tem acesso, por fim, é disponibilizado a visão geral das informações preenchidas em 31 (C), bem como suas permissões e botão para confirmar o registro.

Na Figura 32 é disponibilizado a listagem de usuários do SIHV, o qual é permitido selecionar um para editar seus dados cadastrais. Como o perfil de acesso no sistema, podendo alterar cargo e função do mesmo, para redefinir suas permissões na aplicação, bem como modificar as credenciais de entrada *login* e senha, por fim em “poderes atribuídos a este usuário”, permite-se personalizar tais atribuições, independente do cargo e função do indivíduo sendo editado.

Figura 32 – Telas para editar dados de usuários do sistema

The screenshot shows the 'Editar dados de usuário' page in the SIHV system. The left sidebar contains navigation options like 'Início', 'Usuário do sistema', 'Nova consulta', etc. The main content area is titled 'Editar dados de usuário' and shows the user 'Shirley' selected. The 'Dados cadastrais' section includes the following fields:

- CPF: * (016.)
- RG: * (66)
- UF: * (PA - Pará)
- Cidade: * (Castanhal1)
- Bairro: *
- Rua: *
- CEP: * (68 ...)
- Numero da casa: *
- Complemento:
- Nome: * (Shirley)
- Sexo: * (M, F)
- Endereço de email: * (@gmail.com)
- 1º Celular: * ((91) ...)
- 2º Celular: (checkbox)

A 'Salvar alterações' button is located at the bottom right of the form.

(A) Formulário para editar dados do usuário

The screenshot shows the 'Editar dados de usuário' page for the user 'Shirley'. The '1º Celular' field is visible at the top. Below it is the 'Perfil no sistema' section with the following fields:

- Cargo profissional: * (Discente)
- Função: * (Recepcionista)
- Matrícula: *
- Login de usuário: * (shirley)
- Editar senha: (checkboxes for Sim and Não)

At the bottom is the 'Poderes atribuídos à este usuário' section, which contains a grid of checkboxes for various permissions:

- cadastrar usuário do sistema
- cadastrar proprietário
- cadastrar animal pequeno
- realizar consultas
- manipular agendamentos de consultas
- buscar logs do sistema
- cadastrar e editar procedimentos do HOVET
- declarar óbito do animal pequeno
- Atender radiografias pendentes
- Pode conceder desconto

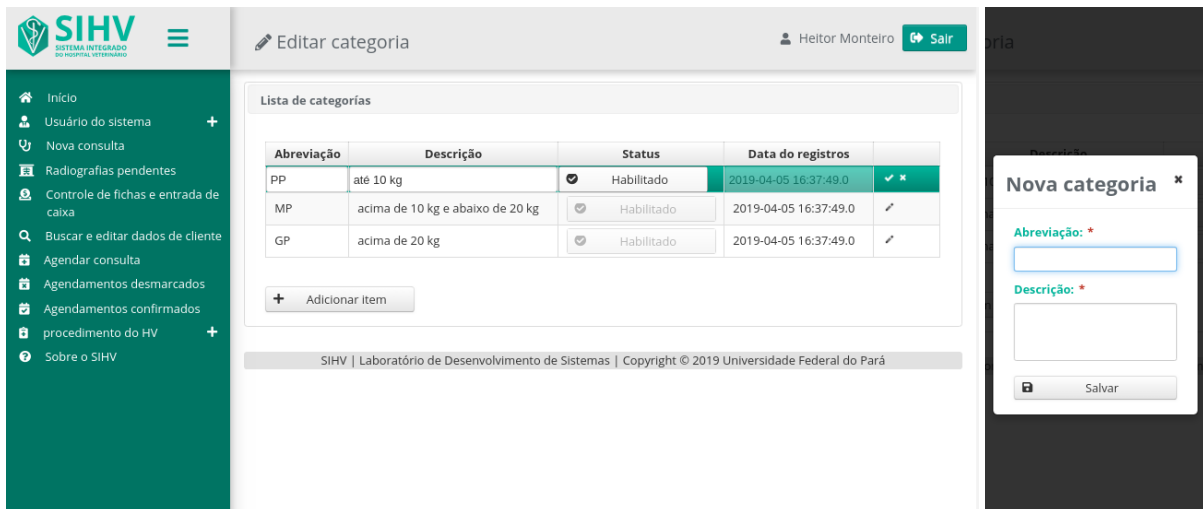
A 'Salvar alterações' button is located at the bottom right of the form.

(B) Formulário para editar perfil e poderes de usuário

Fonte das imagens: Produzido pelo autor

As telas exemplificadas nas Figuras 33 e 34, correspondem ao 12ª Quadro de requisitos apresentado na sessão 4.2. A primeira — Figura 33, permite editar ou criar novas categorias de peso, para posteriormente associar em procedimentos que demandem a distinção de preço através do peso.

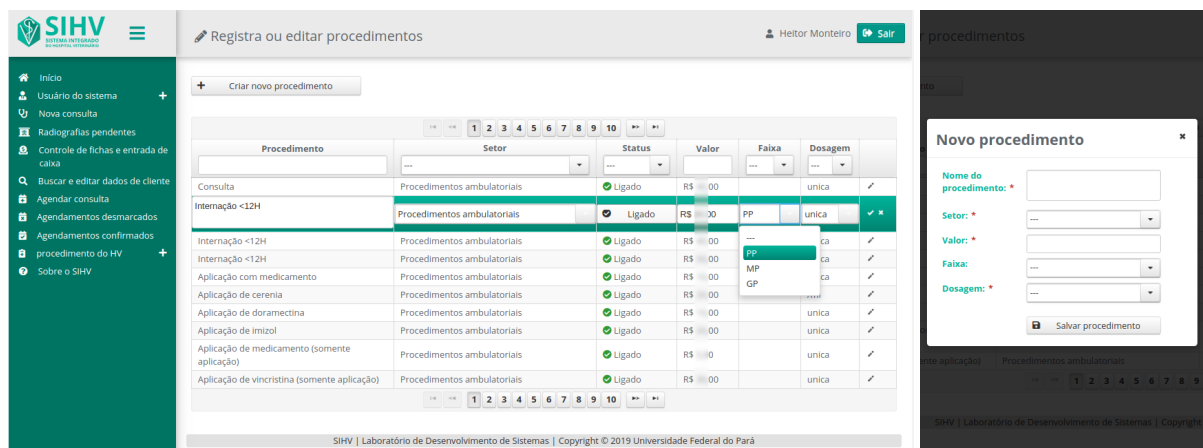
Figura 33 – Tela para editar ou criar nova categoria de peso



Fonte: Produzido pelo autor

Na Figura 34, é viabilizado aos preceptores do HV editar características de procedimentos realizados na clínica veterinária, como: atualizar a faixa de preços; desativar um serviço que se encontra temporariamente indisponível no HV, ou registrar uma nova modalidade de procedimento. A interface também disponibiliza ao usuário, filtros que o auxiliam na busca de um determinado item.

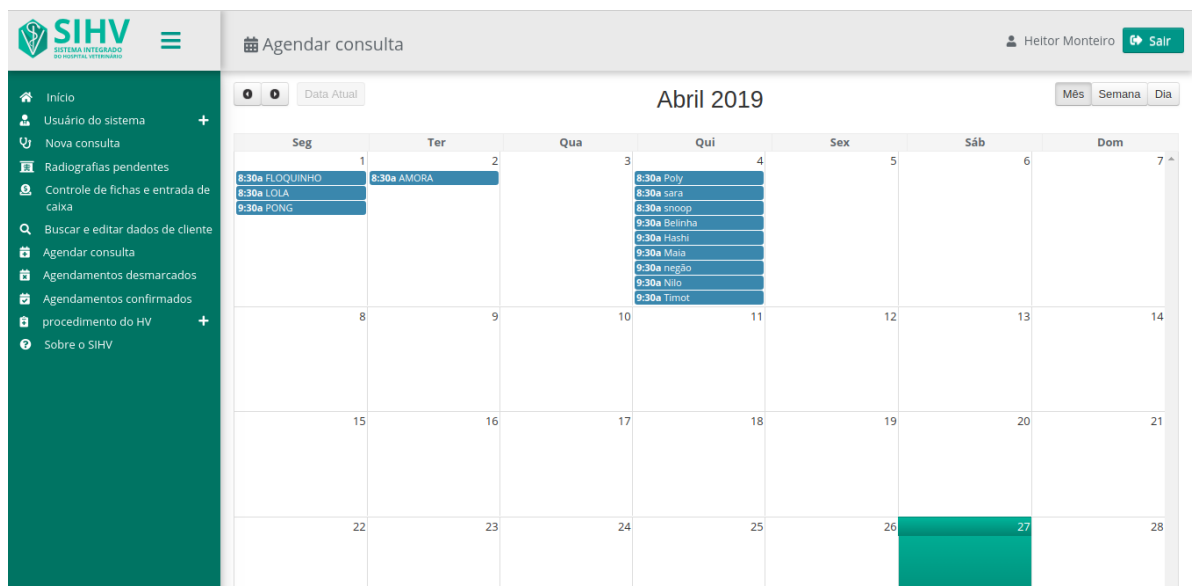
Figura 34 – Tela para editar ou criar novos procedimentos



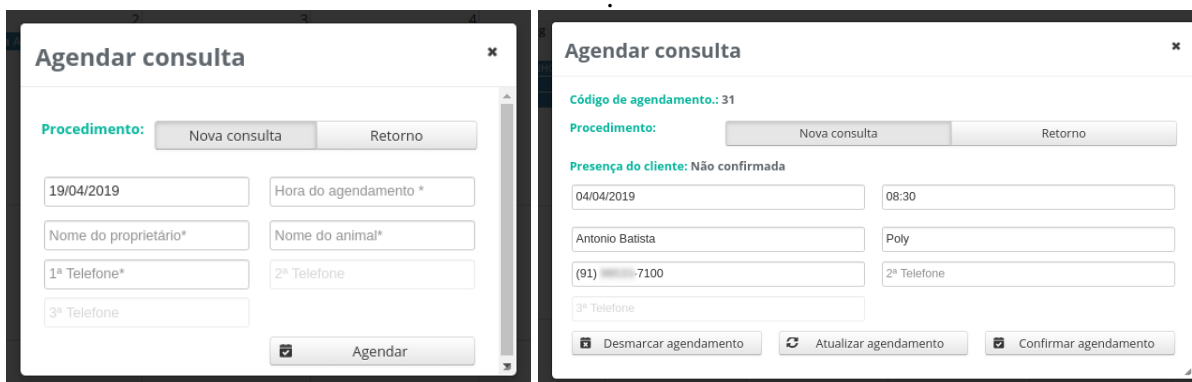
Fonte: Produzido pelo autor

As Figuras dispostas a partir da 35 ao 42, ilustram telas de usuário com ferramentas de uso direcionado ao *stakeholder* recepcionista. O qual as funcionalidades permitem realizar: administrar o agendamento de consultas; registrar e edição de dados cadastrais de clientes, e controle de caixa para gerir o pagamento de serviços.

Figura 35 – Telas para agendar, confirmar e cancelar agendamentos de consultas



(A) Tela para visualizar e receber agendamentos de consultas



(B) Tela para agendar consulta

(C) Tela para confirmar, editar ou cancelar agendamento

Fonte das imagens: Produzido pelo autor

A Figura 35 demonstra ferramentas para o agendamento de consultas no SIHV. Dentre os quais, o espaço apresentado na Figura 35 (A), disponibiliza o calendário para visualizar ou receber agendamentos. Do qual, clicando sobre uma data, o usuário é conduzido ao formulário para registrar um novo agendamento — Figura 35 (B), para isso é fornecido: a hora da consulta; nome do proprietário e seu animal; no máximo três telefones para contato, e a data é preenchida ao acionar a funcionalidade com o clique.

As ações: editar; cancelar ou confirmar um agendamento. São iniciadas clicando sobre o nome de um cliente, destacado em azul no calendário — Figura 35 (A). Conduzido o usuário para a *interface* retratada na Figura 35 (C), composta por formulário preenchido com dados do

agendamento selecionado, e disponibilizando as funcionalidades para desmarcar, atualizar/editar e confirmar um agendamento.

A função “confirmar agendamento” apresentada na Figura 35 (C), direciona o recepcionista para um conjunto de telas retratadas nas Figura 36 ao 38. A primeira etapa — Figura 36 (A), compreende a confirmação de RG e CPF do tutor do animal, caso o sistema não apresente questionamento. O usuário é direcionado para o formulário de cadastro do proprietário, retratado na Figura 36 (B).

Figura 36 – Telas para cadastrar/identificar proprietário

SIHV SISTEMA INTEGRADO DO HOSPITAL VETERINÁRIO

Confirmar presença do proprietário e animal Heitor Monteiro Sair

Dados iniciais do proprietário

Pessoa física/jurídica: * CPF/RG

CPF: * 013. - - - - - Obrigatório

UF: * PA - Pará

RG: * 65 - - - - - Obrigatório

próximo

SIHV | Laboratório de Desenvolvimento de Sistemas | Copyright © 2019 Universidade Federal do Pará

(A) Dados iniciais para identificar proprietário

SIHV SISTEMA INTEGRADO DO HOSPITAL VETERINÁRIO

Confirmar presença do proprietário e animal Heitor Monteiro Sair

Cadastrar proprietário

CPF: 013. - - - - -

RG: 65 - - - - -

Nome: * Everton - - - - -

Sexo: * M F

UF: * PA - Pará

Cidade: * Castanhal1

Bairro: * Imperador

Logradouro: * - - - - -

CEP: - - - - -

Numero da casa: * - - - - -

Complemento: - - - - -

Endereço de email: - - - - -

1ª Celular: * (91) 9 - - - - - 2ª Celular

voltar próximo

(B) Formulário de dados do proprietário

Fonte das imagens: Produzido pelo autor

A Figura 36 (A) também compreende os seguintes tratamentos para os dados RG e CPF: verifica se CPF é válido; caso RG ou CPF já constem no BD, o registro do proprietário é carregado no formulário, e seus animais são listados na Figura 37 (D) ; conforme requisito apresentado no 7ª Quadro, permite-se a opcionalidade de um documento, seja RG ou CPF.

As UI ilustradas na Figura 37 e 38, consistem na segunda etapa da funcionalidade “confirmar agendamento”. Nela é disponibilizado formulário de registro do animal — Figura 37 (C), ou listagem de *pets* vinculados ao proprietário — Figura 37 (D), caso o mesmo já possua registro no sistema. Por fim, na Figura 38 (E) apresenta-se uma visão geral dos dados informado e botão para confirmar consulta.

Figura 37 – Telas para cadastrar ou selecionar animal pequeno

SIHV SISTEMA INTEGRADO DO HOSPITAL VETERINÁRIO

Confirmar presença do proprietário e animal | Heitor Monteiro | Sair

Dados do animal

Novo animal: Sim

RGHV: 9P2019

RGHV antiga: *

Nome do animal: * Poly

Sexo: * M F

Idade do animal: *

Espécie: * Canino Felino

Raça: * Seleccione uma raça

← voltar | S.R.D. | → próximo

(C) Formulário para preencher dados de novo animal

SIHV SISTEMA INTEGRADO DO HOSPITAL VETERINÁRIO

Confirmar presença do proprietário e animal | Heitor Monteiro | Sair

Dados do animal

Novo animal: Não

Animais encontrados

RGHV	Animal	Espécie	Raça
5P2019	Mel - 4831	Canino	Labrador

← voltar | → próximo

(D) Tela para selecionar animal cadastrado

Fonte das imagens: Produzido pelo autor

Figura 38 – Telas para cadastrar ou selecionar animal pequeno

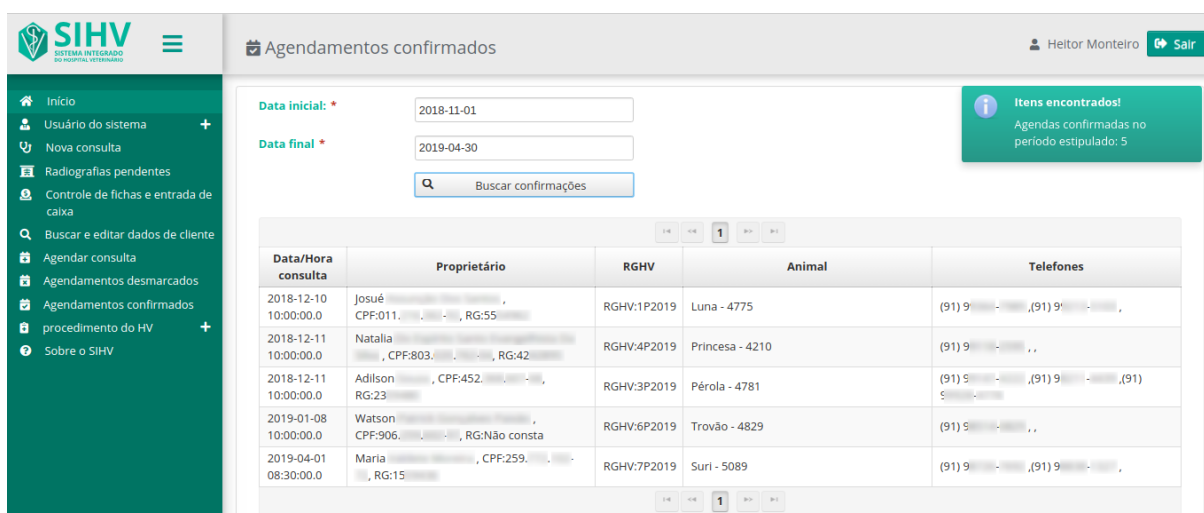


(E) Finalizando com a confirmação da consulta do animal

Fonte das imagens: Produzido pelo autor

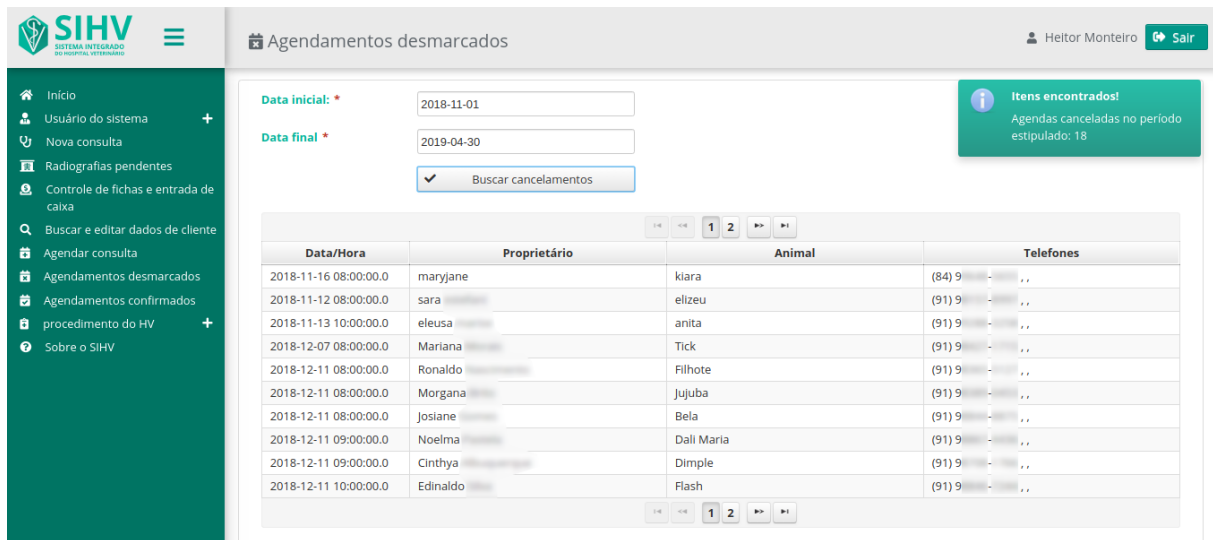
As telas apresentadas nas Figuras 39 e 40, tem como finalidade informa ao recepcionista o quantitativo de agendamentos confirmados — Figura 39, ou cancelados — Figura 40 — num período estipulado pelo mesmo.

Figura 39 – Telas para buscar agendamentos confirmados e cancelados



Fonte das imagens: Produzido pelo autor

Figura 40 – Telas para buscar agendamentos confirmados e cancelados



Fonte das imagens: Produzido pelo autor

Figura 41 – Telas de controle de fichas e entrada de caixa



(A) Tela para listar pagamentos pendentes



(B) Alterar status de pagamento



(C) Visualizar dados do animal

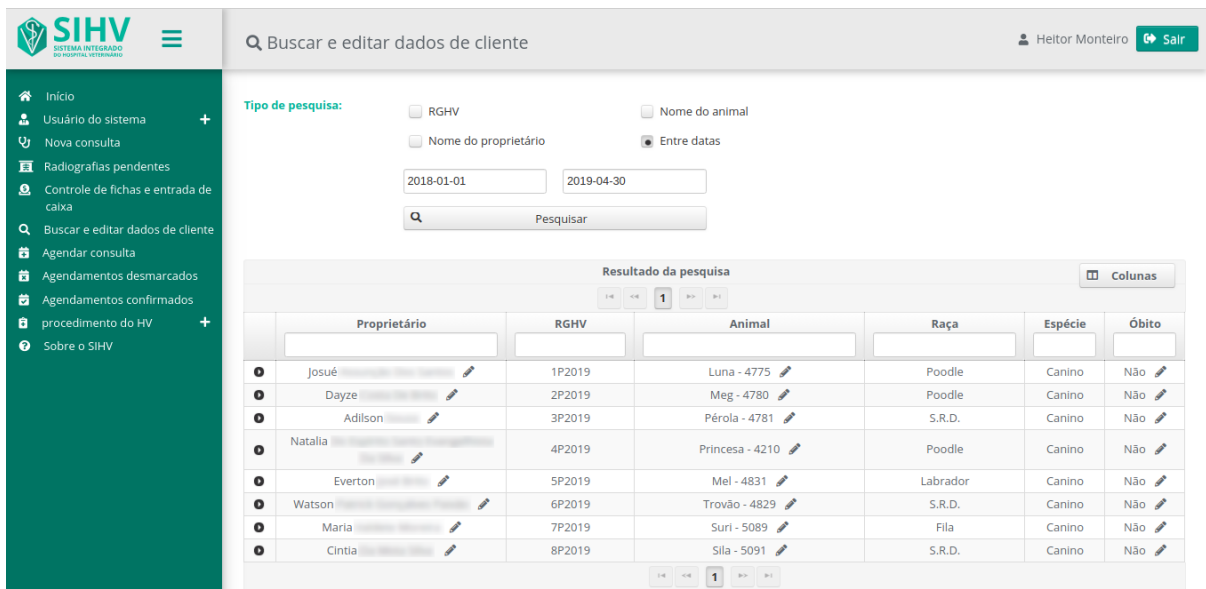
Fonte das imagens: Produzido pelo autor

A Figura 41 retrata funcionalidades para o controle de caixa, do qual permite listar num período estipulado — Figura 41 (A) — consultas efetivadas, acompanhadas do custo total proveniente da aplicação de procedimentos, e *status* de quitação.

A parte superior da *interface* retratado na Figura 41 (A), também demonstra os seguintes dados gerais sobre a listagem de consultas efetivadas: total de itens; arrecadação total desconsiderando o *status* de pagamento; total pago e pendente para o caixa.

Na Figura 41 (A) clicando sobre um valor presente na coluna “Subtotal”, é disponibilizado a Figura 41 (B). O qual permite alterar o *status* de pagamento de serviços realizado em uma consulta. A visualização de dados do animal — Figura 41 (C) — é viabilizada para auxiliar o recepcionista na identificação do cliente, e seu acionamento ocorre na coluna “Nome” presente na Figura 41 (A).

Figura 42 – Telas para buscar e editar dados de clientes



(A) Tela para listar animais e seus tutores

The 'Alterar dados do animal' form includes the following fields:

- Data do registro: 2019-02-14 14:11:15.0
- RGHV no sistema: 3P2019
- RGHV manual: 4781
- Nome do animal: Pérola
- Sexo: M (selected), F
- Idade do animal: 6 anos
- Espécie: Canino (selected), Felino
- Raça: S.R.D.
- Button: Salvar alterações

(B) Alterar dados de animal

The 'Alterar dados de proprietário' form includes the following fields:

- CPF: 803. - - - -
- RG: 42 - - - -
- UF: Pará
- Cidade: Belém
- Bairro: - - - -
- Rua: - - - -
- CEP: - - - -
- Complemento: - - - -
- Nome: Natalia
- Sexo: M, F (selected)
- Endereço de email: - - - -
- 1ª Celular: (91) - - - -
- 2ª Celular: - - - -
- Button: Salvar alterações

(C) Alterar dados de proprietário

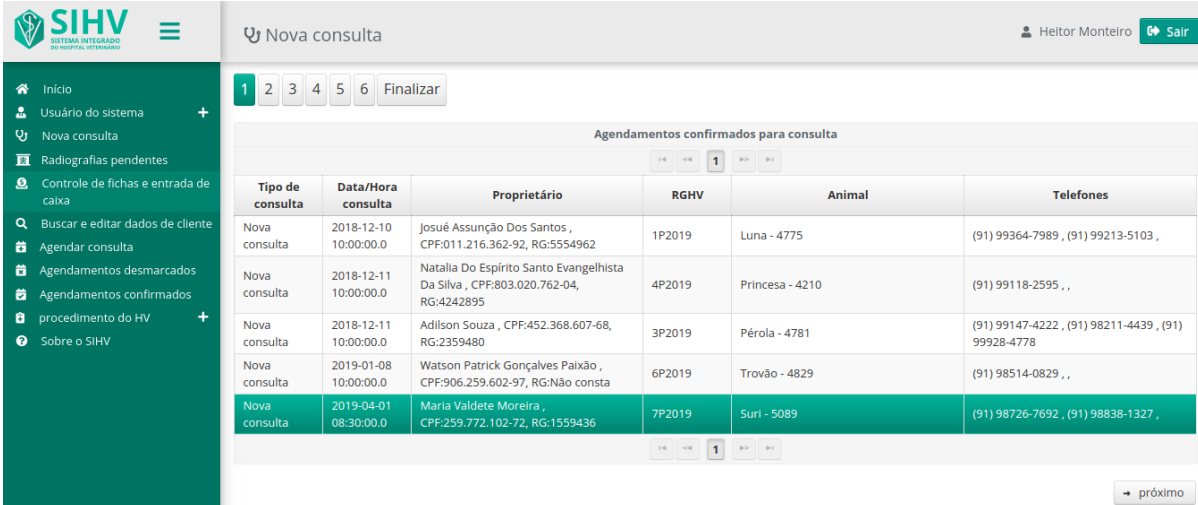
Fonte das imagens: Produzido pelo autor

O conjunto de *interfaces* retratado na Figura 42, permitem ao recepcionista buscar clientes através do RGHV, nome do proprietário ou animal, e entre duas datas. O acionamento das funcionalidades ocorre ao clicar no “ícone de lápis” apresentado em colunas da Figura 42 (A). Dentre as funções que podem ser acionadas, são formulários para editar dados cadastrais do animal — Figura 42 (B) — e proprietário — Figura 42 (C) —, ou declarar óbito do animal — Figura 42 (A).

O conjunto de telas elencadas a partir da Figura 43 ao 49, compreendem características descritas no 10^a Quadro de requisitos, e Figuras de casos de uso de número 19 ao 22. O qual retrata funcionalidades para registra/realizar uma nova consulta, e seu uso direciona-se ao *stakeholder* com cargo veterinário, e com a função de: residente; docente ou técnico veterinário.

A consulta é composta por sete etapas, e inicia-se na Figura 43 o qual efetua a listagem de agendamentos confirmados, ou seja, clientes aptos para submeterem-se ao exame do médico veterinário. Ao seleciona um cliente, o usuário é direcionado para a segunda etapa, formado por formulário de anamnese — Figura 46 ,e exame físico geral — Figura 44.

Figura 43 – Tela para selecionar agendamento confirmado



Tipo de consulta	Data/Hora consulta	Proprietário	RGHV	Animal	Telefones
Nova consulta	2018-12-10 10:00:00.0	Josué Assunção Dos Santos , CPF:011.216.362-92, RG:5554962	1P2019	Luna - 4775	(91) 99364-7989 , (91) 99213-5103 ,
Nova consulta	2018-12-11 10:00:00.0	Natalia Do Espírito Santo Evangelhista Da Silva , CPF:803.020.762-04, RG:4242895	4P2019	Princesa - 4210	(91) 99118-2595 , ,
Nova consulta	2018-12-11 10:00:00.0	Adilson Souza , CPF:452.368.607-68, RG:2359480	3P2019	Pérola - 4781	(91) 99147-4222 , (91) 98211-4439 , (91) 99928-4778
Nova consulta	2019-01-08 10:00:00.0	Watson Patrick Gonçalves Paixão , CPF:906.259.602-97, RG:Não consta	6P2019	Trovão - 4829	(91) 98514-0829 , ,
Nova consulta	2019-04-01 08:30:00.0	Maria Valdete Moreira , CPF:259.772.102-72, RG:1559436	7P2019	Suri - 5089	(91) 98726-7692 , (91) 98838-1327 ,

Fonte das imagens: Produzido pelo autor

Conforme artefatos e opiniões coletada de *stakeholder* do HV, e modelagem em DER apresentada nas Figuras 12 e 13. As *interfaces* inseridas na funcionalidade “nova consulta”, dispõem aos veterinários um conjunto padronizado de campos de preenchimento, e de acordo com progresso de respostas selecionadas, pode ser apresentado novo(s) campo(s) desencadeado(s) pela questão anterior. Na Figura 45 é exemplificado o campo “Pulso”, que ao responder “Ausente” — Figura 45 (B), descarta-se o preenchimento de “Estado do pulso” — Figura 45 (A).

Figura 44 – Tela com formulário de exame físico geral

Fonte das imagens: Produzido pelo autor

Figura 45 – Exemplo de campo reativo a resposta

(A) Resposta originou nova pergunta.

(B) Resposta não originou nova pergunta.

Fonte das imagens: Produzido pelo autor

Figura 46 – Tela com formulário de anamnese

The screenshot displays the 'Nova consulta' (New consultation) form in the SIHV system. The interface features a dark green sidebar with navigation options and a main content area with a light gray header. The header includes the user's name 'Heitor Monteiro' and a 'Sair' (Logout) button. The form is divided into two tabs: 'Anamnese' (Anamnesis) and 'Exame físico geral' (General physical exam). The 'Anamnese' tab is active, showing a series of questions with input fields and buttons. The questions are: 'Peso corporal do animal' (Animal body weight) with a unit dropdown set to 'kg' and a value of '0,000'; 'Queixa principal/Evolução' (Main complaint/Progression) with a text area; 'Já foi tratado' (Already treated) with 'Sim' and 'Não' buttons; 'Antecedentes mórbidos' (Morbid antecedents) with 'Sim' and 'N.D.N.' buttons; 'Histórico familiar' (Family history) with 'Sim' and 'N.D.N.' buttons; 'Alimentação (Dieta caseira)' (Feeding (Home diet)) with 'Sim' and 'Não' buttons; 'Alimentação (Ração)' (Feeding (Ration)) with 'Sim' and 'Não' buttons; 'Vacinação' (Vaccination) with checkboxes for 'Anti-rábica', 'Múltiplas', and 'Outras'; 'Vermifugação' (Deworming) with 'Sim' and 'Não' buttons; 'Ectoparasitas' (Ectoparasites) with 'Sim' and 'Não' buttons; 'Acesso à rua' (Access to street) with a dropdown menu set to 'Selecione'; 'Descrição do habitat' (Description of habitat) with a text area; 'Contactantes' (Contacts) with 'Sim' and 'Não' buttons; and 'Contato com roedores' (Contact with rodents) with 'Sim' and 'Não' buttons. At the bottom of the form, there are 'voltar' (Back) and 'próximo' (Next) buttons.

Fonte das imagens: Produzido pelo autor

As telas retratadas nas Figuras 47 e 48, consiste em formulários de preenchimento opcional, cabendo ao veterinário reponde-los conforme quadro clínico identificado no animal atendido. A interface ilustrada na Figura 47, permite sinalizar “Alterado” em sistemas orgânicos do animal que manifestem irregularidades. Com isso, liberando preenchimento de formulários do “exames físicos específicos”, e demarcando como N.D.N. (Nada digno de nota) itens os quais não houve alteração.

Na Figura 47, é exemplificado o exame físico específico denominado “Sistema digestório e glândulas anexas” sendo sinalizado como alterado, com isso, liberando formulário adequado para o caso.

Figura 47 – Tela com formulários de exame físico específico

SIHV SISTEMA INTEGRADO DO HOSPITAL VETERINÁRIO

Nova consulta Heitor Monteiro Sair

1 2 3 4 5 6 Finalizar

Exame físico específico

Sistema digestório e glândulas anexas: Alterado

Está se alimentando normalmente: * Sim Não

Animal não se alimenta, descreva: *

Vômito: * Sim Não

Regurgitação: * Sim Não

Diarreia: * Sim Não

Disquesia/Tenesmo: * Sim Não

Sistema respiratório e cardiovascular: N.D.N.

Sistema gênito-urinário e glândulas mamárias: N.D.N.

Sistema tegumentar: N.D.N.

Sistema neurológico: N.D.N.

Sistema oftálmico: N.D.N.

Sistema músculo-esquelético: N.D.N.

voltar próximo

Fonte das imagens: Produzido pelo autor

A quarta etapa da funcionalidade nova consulta ilustrada na Figura 48, compreende formulários para solicitar exames por imagem. O mesmo tem como objetivo coletar informações fornecidas pelo veterinário para direcionar o *stakeholder* radiologista ao procedimento que atende a solicitação.

Figura 48 – Tela para solicitar exame por imagem

Fonte das imagens: Produzido pelo autor

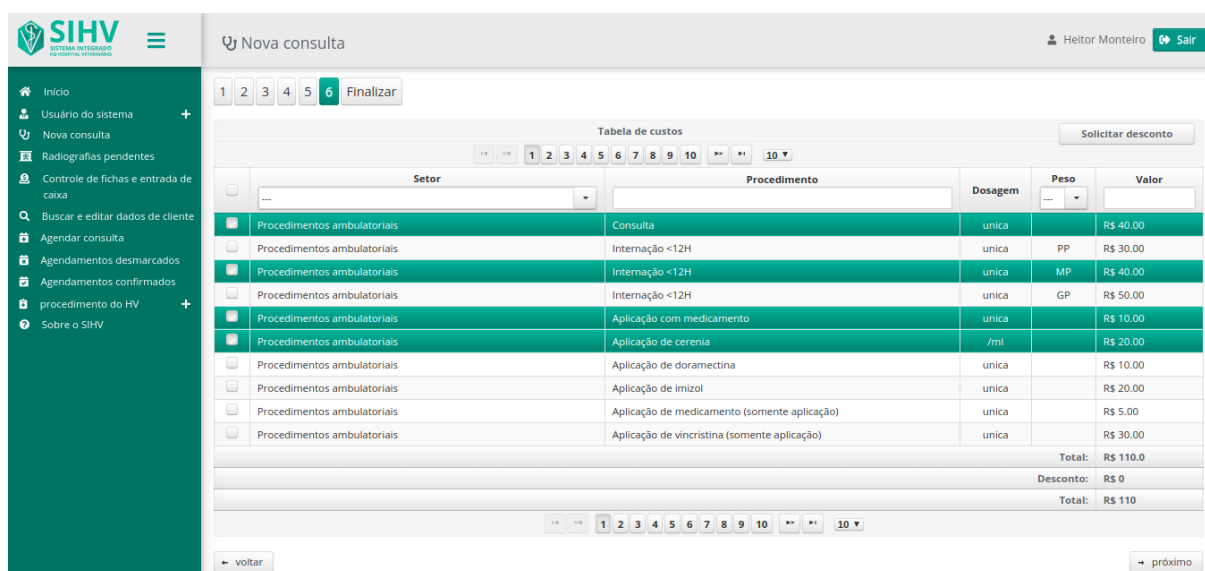
A próxima etapa demonstrada na Figura 49, contém perguntas relacionada ao “diagnósticos diferenciais”. O qual o veterinário informa neste formulário, possíveis enfermidade presentes no animal consultado, ou diagnóstico definitivo do quadro.

Figura 49 – Tela com formulário de diagnósticos diferenciais

Fonte das imagens: Produzido pelo autor

A interface ilustrada na Figura 50 (A), dispõem funcionalidades que permitem buscar e selecionar procedimentos realizados durante a consulta. Com isso, registrando todo o custo proveniente de serviços realizados pelo veterinário, e informando o total a ser custeado pelo cliente.

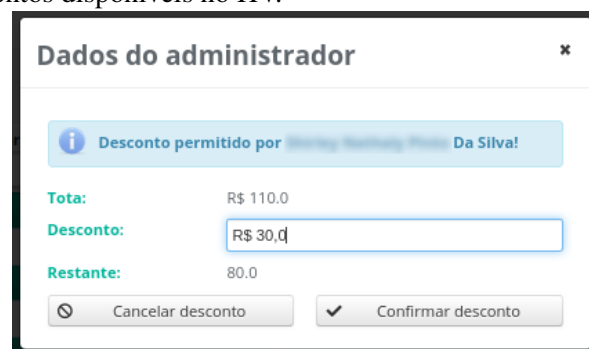
Figura 50 – Tela para selecionar procedimentos aplicados na consulta



(A) Tabela de procedimentos disponíveis no HV.



(B) Confirmação do preceptor.



(C) Informar valor do desconto.

Fonte das imagens: Produzido pelo autor

Na Figura 50 (A) também disponibiliza ao veterinário, mecanismo para solicitar desconto sobre a importância advinda dos serviços selecionados. Ao clicar no botão “Solicitar desconto”, é perguntado *login* e senha do administrador — Figura 50 (B). O qual deve ser preenchido por preceptor do HV, ou usuário com poder para conceder dedução. Obtendo confirmação de dados fornecidos, e liberado campo para informar o valor a ser debitado — Figura 50 (C).

A última etapa da funcionalidade nova consulta — Figura 51, apresenta ao veterinário uma visão geral de informações sobre a mesma, dentre os quais são: dados cadastrais do proprietário e animal atendido; procedimentos aplicados no exame; custo total e desconto caso tenha sido concedido. Conforme o 10^a Quadro de requisitos, a finalização da consulta ocorre quando o médico veterinário informa seu CRMV, e senha de acesso ao sistema. Para confirmar identidade do profissional, e registra o exame.

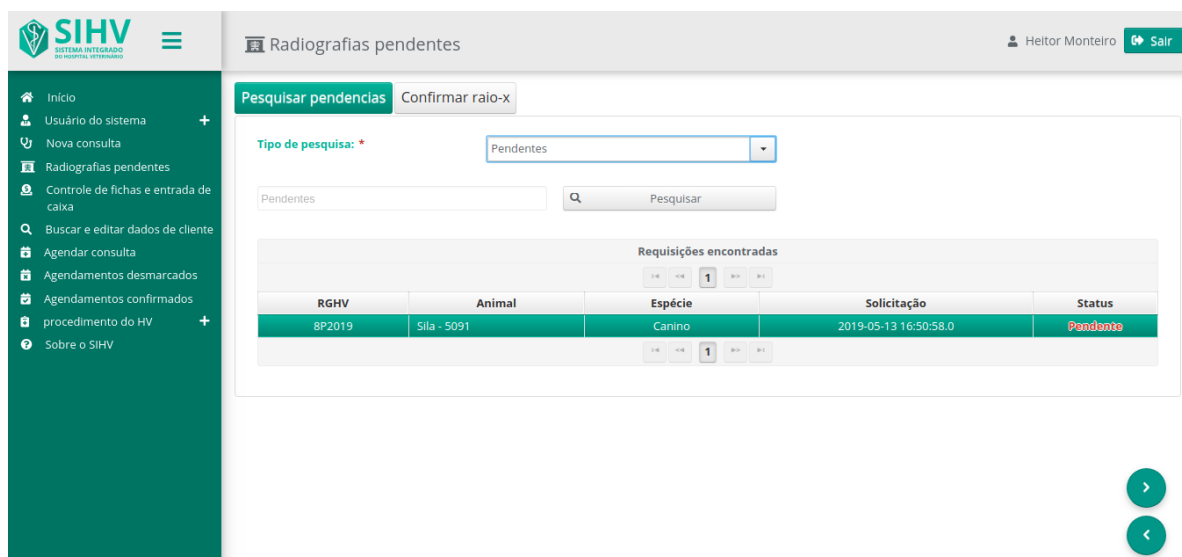
Figura 51 – Tela para finalizar consulta e confirmar identidade do veterinário



Fonte das imagens: Produzido pelo autor

As ilustrações retratadas nas Figuras 52 e 53, consiste em funções de uso direcionado ao usuário radiologista. Do qual em tela permite-se listar — Figura 52, e selecionar solicitações pendentes de raio-x. Com isso, conduzindo o utilizado para a interface retratada na Figura 53.

Figura 52 – Tela de nova radiografia



Fonte das imagens: Produzido pelo autor

A tela ilustrada na Figura 53 contem formulário para o radiologista preencher com dados técnicos referentes a projeções de imagens — raio-x — realizadas. Conforme orientações passadas pelo veterinário em “suspeita clínica”, e “região” a ser examinada.

Figura 53 – Tela de nova radiografia

SIHV SISTEMA INTEGRADO DO HOSPITAL VETERINÁRIO

Radiografias pendentes

Heitor Monteiro [Saír](#)

Pesquisar pendencias [Confirmar ralo-x](#)

Dados do animal

RGHV do animal:	8P2019	Gênero do animal:	F
Nome do animal:	Sila - 5091	Idade do animal:	7 anos
Espécie:	Canino	Raça do animal:	S.R.D.
Peso do animal:	kg 10.0		

Radiografia - Número: 2019-2

Residente que solicitou: Heitor Monteiro

Suspeita clínica: kujhkgdfkjhgk

Região: yëjhfsfdg

Sub Região: *

Posições: khrge

Quantidade de projeções: *

Técnica dose: *

Número de estudos: *

Informações adicionais:

[>](#) [<](#)

[Salvar radiografia](#)

Fonte das imagens: Produzido pelo autor

A tela retratada na Figura 54, tem como finalidade auxiliar o desenvolvedor do SIHV e *stakeholders* do HV durante erro interno no sistema. O qual dispõem aos seus usuários, botões para capturar informações do ocorrido para reportar ao desenvolvedor, e retorna para a tela inicial — Figura 29. Quanto ao programador, a tela informa classe e método que ocorreu o problema, e a natureza no erro. Observa-se que a funcionalidade “erro 500” rastreia 67 métodos do sistema, porem, esse numero ainda não compreendo todas as funcionalidades do SIHV.

Figura 54 – Tela de erro 500 do sistema

SIHV SISTEMA INTEGRADO DO HOSPITAL VETERINÁRIO

Bug identificado

Heitor Monteiro [Saír](#)

Erro 500

Desculpe, foi detectado uma falha interna no sistema, reporte as informações abaixo ao desenvolvedor.

Origem: MBScheduleConsulta > init

Informação coletada: java.lang.NullPointerException

[Voltar para tela inicial](#) [Capturar print da tela](#)

Fonte das imagens: Produzido pelo autor

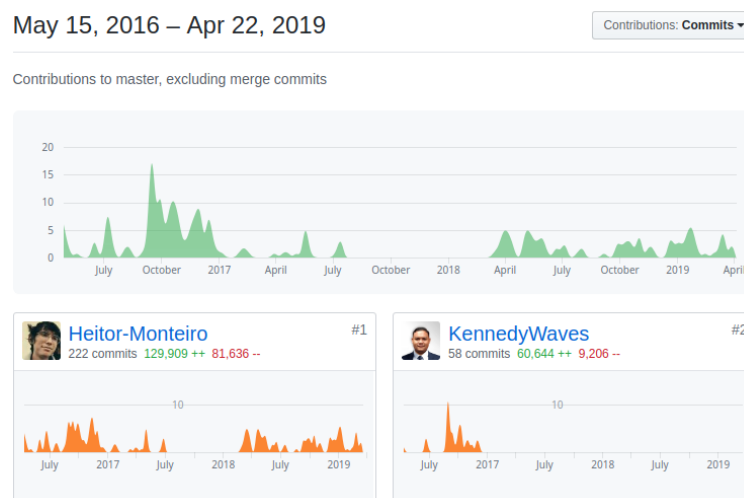
6.1 Código-fonte da aplicação

A sessão tem como objetivo disponibilizar os *scripts* da aplicação SIHV, mediante *link* de acesso ao repositório do projeto hospedado na plataforma GitHub. Com isso, dispondo para comunidade acadêmica, insumos que sirvam como base histórica em projetos que lidem com a mesma tecnologia ou problemática.

1. [Link para o repositório do projeto;](#)
2. [Link para as versões do software.](#)

Segundo dados obtidos em GitHub (2019c), o projeto possui um total de 600 *commits* realizados no repositório somando-se as ramificações existentes. Tais contribuições estão distribuídas entre os *branches master* e *Developing-functions*. A Figura 55 ilustra o intervalo de atividades no projeto, e quantitativo de colaborações por desenvolvedores envolvidos, o comparativo é realizado sobre a ramificação mestre, desconsiderando mesclagens e *commits* realizados em *Developing-functions*.

Figura 55 – Desenvolvedores do projeto



Fonte: GitHub

7 CONSIDERAÇÕES FINAIS

O desenvolvimento deste trabalho iniciou-se mediante demanda encaminhada ao LADES da UFPA de Castanhal, para construir um *software* veterinário que disponibiliza-se ferramentas com potencial para otimizar/mitigar de maneira gradativa, problemáticas provenientes da ausência de informatização no HV.

O projeto também foi uma oportunidade para aplicar conhecimento prático e teórico no que se refere a construção de sistemas de informações, e tomando como objetivo geral, transpor para o trabalho de conclusão de curso, a descrição do processo de desenvolvimento utilizado sobre a aplicação *web* denominada SIHV. Para isso, apresentando referencial teórico pesquisado sobre tecnologias implementadas no trabalho, e artefatos ilustrativos que expusesse características arquiteturais e funcionais do sistema.

O esforço empregado durante o tempo de atividades do projeto, também culminou no alcance dos seguintes objetivos específicos. O primeiro consiste no compartilhamento de código-fonte com a comunidade acadêmica, através da plataforma GitHub, este objetivo sempre foi presente desde o protótipo inicial do sistema, pois, como boa prática de desenvolvimento, aplicou-se o cuidado em manter o progresso das atividades a salvo no GitHub.

O segundo objetivo específico constitui-se na publicação de artigos científico que abordam diferentes fases do processo de desenvolvimento do SIHV. Dentre as publicações mais importantes, foi destacado no capítulo 5^a, a temática “desafios da interdisciplinaridade no desenvolvimento de sistemas: um estudo de caso”, publicado na revista interdisciplinar científica aplicada — RICA.

O terceiro e quarto objetivo tange ao desenvolvimento e apresentação de funcionalidades no 6^a capítulo, dentre os quais citamos as funções de registro para: agenda de consultas; proprietários/tutores de animais; cães e gatos; anamnese; exame físico geral; diagnósticos diferenciais; exames físico específicos; solicitação de exames por imagem; formulário do radiologista; controle de fichas e entrada de caixa; procedimentos realizados no HV.

Grande parte dos itens supracitados anteriormente, também possuem mecanismos de busca e edição de informações em BD, No entanto, citamos como objetivo específico não alcançado, a UI para buscar e editar formulários dos seguintes exames: anamnese; físico geral e físico específico. As funcionalidades desta UI ainda encontra-se em fase de desenvolvimento.

Durante todo o processo de construção do projeto SIHV, os *stakeholder* do HV demonstraram-se bastante receptivos para fornecer *feedback* sobre apresentação e execução de funcionalidades do *software*. Contribuindo com esclarecimento de dúvidas, e informando conhecimento técnico para agrega a aplicação SIHV. Porém, como dificuldades encontradas no decorrer do trabalho, o qual é destacado:

A instabilidade/corte no fornecimento de energia da clínica veterinária, somado a informatização de diferentes aspectos propostos pelo sistema, proporcionou insegurança aos seus *stakeholders* do HV no seguinte aspecto. Ocorrendo o sinistro citado, deve-se reagendar/desmarcar consultas com clientes utilizando dados telefônicos. No entanto, buscar tais informações nas devidas circunstâncias em infraestrutura alimentada pela rede elétrica, torna-se uma barreira. Uma contra medida acessível para isso, seria utilizar dispositivo com alimentação própria — um *notebook* — comportando versão *desktop* do sistema, o qual viabilizaria aplicar os reagendamentos necessários.

O desenvolvimento do SIHV ainda se encontra em fase de execução em estágio no HV, e junto a isso, testes de utilização de partes/componeres do sistema são empregados, objetivado gradativamente aperfeiçoar funcionalidades, e prepara *stakeholders* a rotina de trabalho na aplicação *web*. O esforço para implementar o SIHV tem como estímulo, a visível necessidade evidenciada pela oportunidade de acompanhar a rotina de trabalho de profissionais do HV. Então, o trabalho contribuiu de maneira substancial para minha experiência acadêmica, e profissional como desenvolvedor de *software*, pois, permitiu experienciar situações que dificilmente são transmitidas em sala de aula.

REFERÊNCIAS

- AGUIAR, T. F. d. **SGHV: sistema gerenciador de hospitais veterinários**. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2015.
- ANDRADE, T. F. de. **Java EE 7 com JSF, PrimeFaces e CDI**. AlgaWorks Softwares, Treinamentos e Serviços Ltda, 2015a. Disponível em: <<https://cafe.algaworks.com/livro-java-ee-7-com-jsf-primefaces-e-cdi/>>.
- ANDRADE, T. F. de. **PrimeFaces: Uma visão geral da tecnologia e do mercado**. AlgaWorks Softwares, Treinamentos e Serviços Ltda, 2016b. Disponível em: <<http://blog.algaworks.com/tecnologia-e-mercado-do-primefaces/>>.
- AQUILES A.; FERREIRA, R. **Controlando versões com Git e Github**. Rua Vergueiro, 3185 - 8ª Andar 04101-300 – Vila Mariana – São Paulo – SP – Brasil: Editora Casa do Código, 2014. ISBN 9788566250534.
- AXOSOFT. **GitKraken**: See how our git client can make you a more productive git user! 2017. Disponível em: <<https://www.gitkraken.com/git-client>>.
- BERNARDI, A. d. C. et al. Agricultura de precisão: resultados de um novo olhar. **Embrapa Instrumentação-Livro técnico (INFOTECA-E)**, Brasília, DF: Embrapa, 2014. 596 p., 2014.
- CAELUM. **Apostila do curso FJ-21**: Java para desenvolvimento web. 2017a. Disponível em: <<https://www.caelum.com.br/apostila-java-web/>>.
- CAELUM. **Apostila do curso FJ-11**: Java e orientação a objetos. 2017b. Disponível em: <<https://www.caelum.com.br/apostila-java-orientacao-objetos/>>.
- CARVALHO, V. **MySQL**: Comece com o principal banco de dados open source do mercado. 18.6.27. ed. Rua Vergueiro, 3185 - 8ª Andar 04101-300 – Vila Mariana – São Paulo – SP – Brasil: Editora Casa do Código, 2015. ISBN 9788555190797.
- CIRIACO, D. **Microsoft oficializa a compra do GitHub por US 7,5 bilhões**. 2018. Disponível em: <<https://news.microsoft.com/2018/06/04/microsoft-to-acquire-github-for-7-5-billion/>>.
- CNI, C. N. da I. **Investimentos em Indústria 4.0**. 2018. Disponível em: <<http://www.portaldaindustria.com.br/estatisticas/pqt-investimentos-em-industria-40/>>.
- CORDEIRO, G. **Aplicações Java para a web com JSF e JPA**. 17.9.26. ed. Rua Vergueiro, 3185 - 8ª Andar 04101-300 – Vila Mariana – São Paulo – SP – Brasil: Editora Casa do Código, 2014. ISBN 9788566250015.
- DEITEL, H. M. **JAVA Como Programar**. 10ª. ed. São Paulo: Pearson Education do Brasil, 2017. ISBN 9788543019055.
- DEMICHIEL, L. **JSR 338: Java TM Persistence API**: Final release. [S.l.], 2013. 570 p. Disponível em: <http://download.oracle.com/otndocs/jcp/persistence-2_1-fr-spec/index.html>.
- DIONISIO, E. J. **Introdução ao Java JDK**. 2013. Disponível em: <<https://www.devmedia.com.br/introducao-ao-java-jdk/28896>>.
- ELMASRI, R. et al. **Sistemas de banco de dados**. [S.l.]: Pearson Addison Wesley São Paulo, 2005.

- EVELETH, R. **The surgeon who operates from 400km away**: Doctors are controlling scalpel-wielding robots in real operations from afar, finds rose eveleth. is this the future of surgery? 2014. Disponível em: <<http://www.bbc.com/future/story/20140516-i-operate-on-people-400km-away>>.
- FONTAWESOME. **All 1.500 Awesome Icons**. 2018. Disponível em: <<https://fontawesome.com/icons?d=listing&m=free>>.
- GITHUB. **Documentação do Electron**: Veja toda a documentação em uma só página ou verifique as perguntas frequente. 2017a. Disponível em: <<https://electronjs.org/docs>>.
- GITHUB. **The new native**. 2017b. Disponível em: <<https://desktop.github.com/>>.
- GITHUB. **LADES-Hospital-Veterinario-**. 2019c. Disponível em: <<https://github.com/Heitor-Monteiro/LADES-Hospital-Veterinario->>.
- GRANATYR, J. **Introdução ao modelo multicamadas**. 2007. Disponível em: <<http://www.devmedia.com.br/introducao-ao-modelo-multicamadas/5541>>.
- GUEDES, G. T. **UML 2, Uma Abordagem Prática**". 2. ed. [S.l.]: São Paulo, Novatec, 2011. ISBN 9788575222812.
- HAWKEYE. **What is content intelligence?**: How data-driven marketers will dominate. 2018. Disponível em: <<https://www.hawkeye.ai/what-is-content-intelligence/>>.
- ICESOFT. **ICEfaces Overview**. 2017. Disponível em: <<http://www.icesoft.org/java/projects/ICEfaces/overview.jsf>>.
- INFOLOGIKA. **InfoPet: gestão de petshops e clínica veterinária**. 2017. Disponível em: <<http://www.infopet.com.br/index.html#jcontato>>.
- JGRAPH. **Create and share diagrams**: Professional, open source diagramming. online or desktop, no registration. 2019. Disponível em: <<https://about.draw.io/>>.
- KLINGER, Í. desenvolvimento de uma aplicação móvel para auxiliar no processo de inspeção zootécnica. **repositório de relatórios-sistemas de Informação**, n. 2, 2014.
- KUROSE JAMES F.; ROSS, K. W. **Redes de computadores e a Internet: uma abordagem top-down**. 6ª edição. ed. São Paulo: Pearson Education do Brasil Ltda, 2013. ISBN 9788543014432.
- LUCKOW, D. H.; MELO, A. A. de. **Programação Java para a WEB**. [S.l.]: Novatec Editora, 2010.
- MAK, G.; GURUZU, S.; OTTINGER, J. **Hibernate Recipes: A problem-solution approach**. 2. ed. [S.l.]: Apress, 2015.
- MEDEIROS, H. **EJB: Introdução ao novo Enterprise JavaBeans 3.2**. 2014. Disponível em: <<https://www.devmedia.com.br/ejb-introducao-ao-novo-enterprise-javabeans-3-2/30807>>.
- MELO, H. dos S.; LEITÃO, L. C. **Dicionário Tecnologia e Inovação**. [S.l.]: Fortaleza: SEBRAE, 2010. ISBN 9788575222812.
- MICROSOFT. **Microsoft to acquire GitHub for 7.5 billion**. 2018. Disponível em: <<https://news.microsoft.com/2018/06/04/microsoft-to-acquire-github-for-7-5-billion/>>.

- MOZILLA. **URL**. 2017. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Glossario/URL>>.
- NETBEANS.ORG. **NetBeans IDE Features**. 2017a. Disponível em: <https://netbeans.org/features/index_pt_BR.html>.
- NETBEANS.ORG. **Base IDE**. 2017b. Disponível em: <<https://netbeans.org/features/ide/>>.
- ORACLE. **Introduction to Java Platform: Enterprise edition 7**. 2013a. Disponível em: <<http://www.oracle.com/technetwork/java/javase/javase7-whitepaper-1956203.pdf>>.
- ORACLE. **JavaServer™ Faces Specification: Jsr-344 javax.faces**. 2013f. Disponível em: <<https://jcp.org/aboutJava/communityprocess/final/jsr344/index.html>>.
- ORACLE. **JSR-000345 Enterprise JavaBeans™ 3.2 Final Release**. 2013h. Disponível em: <http://download.oracle.com/otndocs/jcp/ejb-3_2-fr-spec/index.html>.
- ORACLE. **JSRs: Java Specification Requests: Jsr overview**. 2017b. Disponível em: <<https://www.jcp.org/en/jsr/overview>>.
- ORACLE. **Welcome to the Java Community Process**. 2017c. Disponível em: <<https://www.jcp.org/en/home/index>>.
- ORACLE. **Java™ EE 7 Technologies**. 2017d. Disponível em: <<http://www.oracle.com/technetwork/java/javase/tech/index-jsp-142185.html>>.
- ORACLE. **Mojarra JavaServer Faces: Oracle's open source implementation of the jsf standard**. 2017e. Disponível em: <<https://javaserverfaces.github.io/>>.
- ORACLE. **Fusion Middleware Web User Interface Developer's Guide for Oracle Application Development Framework**. 2017g. Disponível em: <https://docs.oracle.com/cd/E23943_01/web.1111/b31973/gs_intro.htm#ADFUI101>.
- ORACLE. **README Java™ Platform, Standard Edition 8 Development Kit (JDK™ 8)**. 2017i. Disponível em: <<http://www.oracle.com/technetwork/java/javase/jdk-8-readme-2095712.html?ssSourceSiteId=otnpt#introduction>>.
- ORACLE. **Java documentation: Java platform, standard edition installation guide**. 2017j. Disponível em: <<https://docs.oracle.com/javase/8/docs/technotes/guides/install/index.html>>.
- ORACLE. **MySQL Workbench: Enhanced Data Migration**. 2017k. Disponível em: <<https://www.mysql.com/products/workbench/>>.
- ORACLE. **GlassFish: The open source java ee reference implementation**. 2017l. Disponível em: <<https://javaee.github.io/glassfish/documentation>>.
- ORLANDO, T. M.; SUNINDYO, W. D. Designing dashboard visualization for heterogeneous stakeholders (case study: Itb central library). In: IEEE. **Data and Software Engineering (ICoDSE), 2017 International Conference on**. [S.l.], 2017. p. 1–6.
- PHPGROUP. **História do PHP**. 2017. Disponível em: <https://secure.php.net/manual/pt_BR/history.php.php>.
- PINTO, P. **Alguns tipos de licenças de software**. 2011. Disponível em: <<https://pplware.sapo.pt/informacao/alguns-tipos-de-licencas-de-software/#comments>>.

PRIMETEK. **PrimeFaces Ultimate UI Framework for Java EE**. 2017. Disponível em: <<https://www.primefaces.org/>>.

QUIN, L. **Extensible Markup Language (XML)**. 2016. Disponível em: <<https://www.w3.org/XML/>>.

REDHAT. **Hibernate Documentation - 4.3**. 2017. Disponível em: <<http://hibernate.org/orm/documentation/4.3/>>.

RICHFACES. **RichFaces: The next-generation jsf component framework by jboss**. 2016. Disponível em: <<http://richfaces.jboss.org/>>.

SILVA, A.; VIDEIRA, C. **UML, Metodologias e Ferramentas CASE**. 1. ed. [S.l.]: Centro Atlântico,, 2001. ISBN 9728426364.

SIMPLESVET. **SIMPLESVET TECNOLOGIA LTDA**. 2016. Disponível em: <<https://simples.vet/conheca/clinica-petshop/>>.

SIRIWARDENA, P. **Mastering Apache Maven 3: Enhance developer productivity and address exact enterprise build requirements by extending maven**. Birmingham, UK: Packt Publishing Ltd, 2014. ISBN 9781783983865.

SOBRAL, H. d. S.; SOUZA, K. d.; SERUFFO, M. d. R. Desafios da interdisciplinaridade no desenvolvimento de sistemas: Um estudo de caso. **Revista Interdisciplinar Científica Aplicada**, v. 12, n. 1, p. 49–70, 2018.

SOMMERVILLE, I. **Engenharia de software**. 9. ed. [S.l.]: PEARSON BRASIL, 2011. ISBN 9788579361081.

SOURCEFORGE. **About SourceForge**. 2017. Disponível em: <<https://sourceforge.net/about>>.

SYNTEVO. **SmartGit: Get your commit done**. 2017. Disponível em: <<https://www.syntevo.com/smartgit/>>.

VARANASI, B.; BELIDA, S. **Introducing Maven**. [S.l.]: Apress, 2014. ISBN 9781484208410.

W3SCHOOLS. **HTML and XHTML: Xhtml is html written as xml**. 2017a. Disponível em: <https://www.w3schools.com/html/html_xhtml.asp>.

W3SCHOOLS. **HTML Introduction**. 2017b. Disponível em: <https://www.w3schools.com/html/html_intro.asp>.

ÇIVICI Çağatay. **PrimeFaces: User Guide 6.2**. [S.l.], 2018. Disponível em: <https://www.primefaces.org/docs/guide/primefaces_user_guide_6_2.pdf>.

Anexos

ANEXO A - FICHA DE CONSULTAS DO HV

Dois escritos
PCO + histopat

HOSPITAL VETERINÁRIO
UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DE CASTANHAL
FACULDADE DE MEDICINA VETERINÁRIA

VER FOLHA 56 DOE LISTAR

Data: 1/10/17 (Hora): _____
 RGHV: 110016 Espécie: _____ Raça: _____ Pelagem: _____
 Nome: _____ Idade: _____ Sexo: _____ Peso: _____
 Proprietário: _____
 Endereço: _____
 Cidade: _____ Estado: _____ CEP: _____
 Fone: _____ RG: _____ CPF: _____

DIAGNÓSTICO/PROCEDIMENTO: _____
 SISTEMA AFETADO: _____
 Médico Veterinário: _____ Assinatura: _____

QUEIXA PRINCIPAL/EVOLUÇÃO

Já foi tratado sim não medicação/dose _____

ANAMNESE
 antecedentes mórbidos _____ histórico familiar _____

Alimentação dieta caseira ração _____
 Vacinação anti-rábica VS outras _____ origem _____
 Vermifugação sim não qual dose _____ quando _____
 Ectoparasitas sim não qual controle _____ sim _____ não com
 que produto _____
 acesso à rua sim não sozinho acompanhado _____

SISTEMA NEUROLÓGICO
 consciência alerta depressão _____ esturpor _____ coma _____
 comportamento normal demência _____ agressividade _____
 ataxia sim não (tipo) _____
 Evolução _____
 Parêntese / paralisia sim não espástica _____ flácida (tipo) _____
 Evolução _____
 Convulsão sim não (tipo) _____
 Evolução _____
 Audição normal perda auditiva _____
 Visão normal miopia _____ hipermetropia _____ anisocoria _____ amaurose _____

SISTEMA OFTÁLMICO
 secreção ocular sim não unilateral _____ bilateral (tipo) _____
 Evolução _____
 Estrabismo sim não _____
 Exoftalmia sim não _____

SISTEMA MÚSCULO-ESQUELÉTICO
 Claudicação sim não membro afetado/ evolução _____
 Fraturas sim não membro afetado/ evolução _____
 atrofia muscular sim não membro afetado/ evolução _____

EXAME FÍSICO
 FC 120 bal/min FR _____ m/min TPC _____
 estado nutricional bom caquético _____ magro _____ gordo _____ obeso _____
 mucosa oral roses pálidas _____ peritécias _____ cianóticas _____ congestas _____ ictericas _____
 vaginal/peniana roses pálidas _____ peritécias _____ cianóticas _____ congestas _____ ictericas _____
 ocular roses pálidas _____ peritécias _____ cianóticas _____ congestas _____ ictericas _____

pulso fraco forte _____ ausente _____ regular _____ irregular _____
 hidratação hidratado 4-6% de desidratação _____ 6-8% de desidratação _____ 8-10% de desidratação _____ 10-12% de desidratação _____
 palpação abdominal _____
 auscultação cardíaca _____
 auscultação pulmonar _____
 linfonodos _____
 pele/pelos _____

SISTEMA DIGESTÓRIO E GLÂNDULAS ANEXAS
 Está se alimentando normalmente sim não _____
 Vômito sim não aspecto _____
 Evolução _____
 Regurgitação sim não evolução _____
 Diarreia sim não não aspecto _____
 Evolução _____
 Disqueia/tenesmo sim não _____
 Evolução _____

SISTEMA RESPIRATÓRIO E CARDIOVASCULAR
 Tosse sim não produtiva _____ não produtiva _____
 Freqüência/ Evolução _____
 Espirto sim não _____
 Evolução _____
 Secreção nasal sim não unilateral _____ bilateral (tipo) _____
 Evolução _____
 Dispneia / ortopnéia sim não _____
 Evolução _____
 Cianose sim não quando _____
 Evolução _____
 cansaço fácil sim não _____
 Evolução _____
 Síncope sim não _____
 Evolução _____
 Emagrecimento sim não _____
 Evolução _____

SISTEMA GÊNITO-URINÁRIO E GLÂNDULAS MAMÁRIAS
 ingestão hídrica normal hipodipsia _____
 evolução _____
 urina normal anúria _____ oligúria _____ poliúria _____
 iscoria _____ aspecto/odor _____
 incontinência _____
 Evolução _____
 último cio _____ último parto _____
 secreção vaginal ou peniana sim não (tipo) _____
 Evolução _____
 Cistite sim não Prênis sim não _____

SISTEMA Tegumentar
 Inicial/ evolução da lesão _____
 Localização e características das lesões _____
 Prurido sim não Evolução _____
 secreção otológica sim não Evolução _____
 Evolução _____
 Banhos: freqüência _____ produto _____

OBSERVAÇÕES COMPLEMENTARES

DIAGNÓSTICOS DIFERENCIAIS

1.	6.
2.	7.
3.	8.
4.	9.
5.	10.

EXAME POR IMAGEM
 RX: região a ser examinada/ suspeita abd - estom. hepática
 Posição US
 radiografia nº 01 quantidade 02 data: 10/02/17
 laudo _____

US: suspeita _____
 US nº _____ data: _____ responsável: _____
 laudo _____

TRATAMENTO


RESULTADOS EXAMES COMPLEMENTARES

RETORNO

hemograma
 FALT
 FA

EXAME POR IMAGEM
 1) hemograma
 2) exame físico
 3) exames complementares
 4) radiografia
 5) US
 6) tratamento

ANEXO B – CONTROLE DE FICHAS E ENTRADA DE CAIXA



HOSPITAL VETERINÁRIO
UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DE CASTANHAL
FACULDADE DE MEDICINA VETERINÁRIA

CONTROLE DE FICHAS E ENTRADA DE CAIXA


RSD - MILMA - LIS
DATA 15/01/19

Nº	NOME/RGHV	ESPÉCIE	PROPRIETÁRIO	VETERINÁRIO	PROCEDIMENTOS	VALOR
01	Valente-4841	Can	ODIVALDO NEGRÃO	CAIO	Retorno	-
02	Negônia-4833	Can	ARLEI ROBERTO	Lis	Retorno + 200,00	200,00
03	Doby-2876	Can	ADA CLAUDIO	Rafael	Consulta	40,00
04	NFE-4850	Can	CHERLENE PINTHEIRO	Milma	Consulta + Injeção	40,00
05	PER-4851	Can	PAULA MORAES	Marcos	Consulta	40,00
06	Ritinha-4852	Fel	LAINÉ SERAF	Lis	Consulta	40,00
07	Diel-3825	Fel	JANA CARINA	CAIO	Retorno	-
08	Martim-4853	Can	CAIO CARLOS	Jacse	Consulta	40,00
09	Neta-4854	Can	ROSILEIA GUILHERME	Rafael	Consulta	40,00
10	Robito-4830	Fel	FELICIA MATOS	Rafael	Retorno	70,00
11	01/2019					40,00
12	Arques-4835	Can	Katia Oliveira	Milma	Consulta	40,00
13	SUBMAR-2543	Can	CLAUDIO RIAN	Jacse	Retorno	-
14	Rick-4809	Can	MARIP DO SOCORRO	Milma	Retorno	-
15	Pirulito-4827	Can	ROSA MARIA	CAIO	Retorno	-

CIRURGIAS						
Nº	NOME/RGHV	ESPÉCIE	PROPRIETÁRIO	VETERINÁRIO	PROCEDIMENTOS	VALOR
01						
02						
03						
04						

Fonte: Hospital Veterinário da UFPA Castanhal

ANEXO C – TABELA DE PROCEDIMENTOS DO HV


 UNIVERSIDADE FEDERAL DO PARÁ
 CENTRO UNIVERSITÁRIO DE CASTANHAL
 INSTITUTO DE MEDICINA VETERINÁRIA
 HOSPITAL VETERINÁRIO - SETOR DE CIRURGIAS

PP= até 10kg MP= acima de 10 e abaixo de 20kg GP= ACIMA DE 20kg

TABELA DE PREÇOS DE PROCEDIMENTOS AMBULATORIAIS

Cód.	PROCEDIMENTO	PREÇO
1	EXAMINAÇÃO	PP=,00
2	INTERNAÇÃO <12H	MP=,00
3	INTERNAÇÃO 24H	MP=,00
4	APLICAÇÃO COM MEDICAMENTO	,00
5	APLICAÇÃO DE CORONA	,00
6	APLICAÇÃO DE IMÓDOL	,00
7	APLICAÇÃO DE MEDICAMENTO (SOMENTE APLICAÇÃO)	,00
8	APLICAÇÃO DE VINCISTINA (SOMENTE APLICAÇÃO)	,00
9	AVULSO DENTÍFICO	,00
10	CITOCENTESE	,00
11	COLÉTA DE SANGUIE SEM COAGULAÇÃO	,00
12	COLÉTA PARA BIÓPSIA	,00
13	COQUÍTEL ENDOVENOSO	,00
14	COITE DE BÊNJA	,00
15	CLIVATIVO SIMPLES	,00
16	CLIVATIVO ESTENDIDO	,00
17	CLIVATIVO COM DRENIO	,00
18	CLIVATIVO DE MÍASE / URINFEZA (RETRADA DE LARVAS)	,00
19	ENEMA	,00
20	KIT FLUOROTERAPIA E MEDICAMENTOS PARA MEIA INTERNAÇÃO	PP=,00
21	KIT FLUOROTERAPIA E MEDICAMENTOS PARA INTERNAÇÃO	MP=,00
21	IMOBILIZAÇÃO COM PENSO ESPARADRAPO	PP=,00
22	IMOBILIZAÇÃO COM PENSO ESPARADRAPO	MP=,00
23	TESTE RÁPIDO PARA CINDOMOSE	,00
24	TESTE RÁPIDO PARA PARVOVIRULOSE	,00
25	PASSAGEM DE SONDAS NASOFARÍNGEA	,00
26	PASSAGEM DE SONDAS NASOGÁSTRICA	,00
27	ESTANÁSIA	,00
28	DESORNSTIÇÃO URETRAL GATOS	,00
29	DESORNSTIÇÃO URETRAL CÃES	,00
30	TERMOCENTESE	,00
31	TRANSFUSÃO SANGÜÍNEA EXTERNA (ENCAMINHAMENTO)* Traber o doador	,00
32	TRANSFUSÃO SANGÜÍNEA * Traber o doador	,00

2. SETOR DE ANESTESIA

Cód.	PROCEDIMENTO	PREÇO
1	TAXA DE ORGEMO	PP=,00
2	EPIDURAL	MP=,00
3	ANESTESIA LOCAL	MP=,00
4	ANESTESIA GERAL DISSOCIATIVA	MP=,00
5	ANESTESIA GERAL INALATORIA (Curatário)	MP=,00
6	TRANQUILIZAÇÃO (SEDACÃO)	MP=,00

4. SETOR DE CIRURGIA

Cód.	PROCEDIMENTO	PREÇO
1	ABLAÇÃO DO CONDUITO AUDITIVO EXTERNO	PP=,00
2	AMPUTAÇÃO DE CAUDA (TERAPÉUTICA)	MP=,00
3	AMPUTAÇÃO DE MEMBRO	MP=,00
4	ANUSPLASTIA	MP=,00
5	BIÓPSIA HEPÁTICA	MP=,00
6	COELOCECTOMIA FEMURAL	MP=,00
7	CESARIANA + OSH	MP=,00
8	COSTO DERMÓIDE	MP=,00
9	CISTOTOMIA	MP=,00
10	DERIVAÇÃO ACETABULAR COM PECTINEOTOMIA	MP=,00
11	DEVIATOPLASTIA	MP=,00
12	DEURÓGRAFIA	MP=,00

13	DRENAGEM DE ABCESÃO	GP=,00
14	DRENAGEM DE OTOMEDIATÓMIA	MP=,00
15	ENTRECTOMIA	MP=,00
16	ENTROSTOMIA	MP=,00
17	ENTROPIO / ECTROPIO	MP=,00
18	ENUCLEAÇÃO	MP=,00
19	ESPLENECTOMIA	MP=,00
20	ESOFAGOTOMIA CERVICAL	MP=,00
21	ESOFAGOTOMIA TORÁCICA	MP=,00
22	EVENTRACÃO - correção	MP=,00
23	EXCIÃO DE GL. ADANAS	MP=,00
24	EXCIÃO DE GL. DA 3ª PÁLPEBRA	MP=,00
25	EXCIÃO DE NEPLASIA DE PELE COMPLEXA (cirurgia reconstructiva)	MP=,00
26	EXCIÃO DE NEPLASIA DE PELE SIMPLES	MP=,00
27	FARINGOSTOMIA OU ESOFAGOSTOMIA COM FIXAÇÃO DE SONDA	MP=,00
28	EXCIÃO DE CACDIA MAMÁRIA UNILATERAL (MASTEC PARCIAL)	MP=,00
29	EXCIÃO DE CACDIA MAMÁRIA BILATERAL (MASTEC TOTAL)	MP=,00
30	FLAP CONJUNTIVAL	MP=,00
31	GASTROTOMIA / GASTROPIEIA	MP=,00
32	HÉRNIA DIAPHRAGMÁTICA	MP=,00

33	HÉRNIA INGUINAL - correção	MP=,00
34	HÉRNIA PERINEAL	MP=,00
35	HÉRNIA SABBUCAL	MP=,00
36	LAPARATOMIA EXPLORATÓRIA	MP=,00
37	LOBECTOMIA PULMONAR	MP=,00
38	COIRAÇÃO DE LUXAÇÃO DE PATELA	MP=,00
39	ORCUECTOMIA - CANINO	MP=,00
40	ORCUECTOMIA - FELINO	MP=,00
41	OSH ELETIVA - CANINA	MP=,00
42	OSH ELETIVA - FELINA	MP=,00
43	OSTIOSSINTESE - ENVERTIA	MP=,00
44	OSTIOSSINTESE COM FIXADOR EXTERNO	MP=,00
45	OSTIOSSINTESE COM PINO INTRAMEDULAR E PARAFUSO DO CIRCLAGEM	MP=,00
46	OSTIOSSINTESE COM PLACA	MP=,00
47	COIRAÇÃO DE OTOMEDIATÓMIA	MP=,00
48	PECTINEOTOMIA	MP=,00
49	PEINECTOMIA PARCIAL	MP=,00
50	PREPARAMENTO DE RETNÁCLIO	MP=,00
51	REDUÇÃO DE PROLAPSO DE RETO COM COLOPLEXIA	MP=,00
52	REDUÇÃO DE PROLAPSO DE RETO SEM COLOPLEXIA	MP=,00
53	REDUÇÃO DE PROLAPSO DE VAGINA / ÚTERO	MP=,00

54	REDUÇÃO DE PROTRUSÃO DE GLOBO OCULAR	MP=,00
55	REPOSIÇÃO/REPLANTAMENTO DE GLÂNDULA 3ª PÁLPEBRA	MP=,00
56	TORACOTOMIA	MP=,00
57	TRANSPOSIÇÃO DA TUBEROSIDADE TIBIAL	MP=,00
58	TRICLOPLASTIA	MP=,00
59	TRICLOPLASTIA	MP=,00
60	URETOSTOMIA PERINEAL OU PRÉ ESCROTAL	MP=,00
61	VASICTOMIA	MP=,00

5. SETOR DE DIAGNÓSTICOS

Cód.	PROCEDIMENTO	PREÇO
01	ELETTROCARDIOGRAMA	,00

5.2 DIAGNÓSTICO POR IMAGEM

Cód.	PROCEDIMENTO	PREÇO
1	ULTRASSONOGRAFIA	,00
2	ENDOSCOPIA	,00
3	COLONOSCOPIA	,00
4	RX SIMPLES GRANDES ANIMAIS (1 incidência)	,00
5	RX SIMPLES PEQUENOS ANIMAIS (1 incidência)	MP=,00
6	EXAME CONTRASTADO	MP=,00
7	ENEMA BÁRIO	,00
8	ESOFAGOGRAFIA - TRÁNSITO GASTROINTESTINAL	,00
9	UROGRAFIA EXCRETÓRIA	,00
10	URETROCISTOGRAFIA	,00
11	MIELOGRAFIA	,00

5.3 PATOLOGIA VETERINÁRIA

Cód.	PROCEDIMENTO	PREÇO
01	OTILOGIA ASPIRATIVA	,00
02	HISTOPATOLOGIA (BIÓPSIA)	,00
03	COLÉTA PARA BIÓPSIA	,00
04	NECROPSIA ANIMAL DOADO PARA A AJIA	,00
05	NECROPSIA URGENTE	,00
06	NECROPSIA COM LAUDO (EXTERNO)	,00

5.4 PATOLOGIA CLÍNICA

Cód.	PROCEDIMENTO	PREÇO
01	HEMATÓRITO	,00
02	ERITROGRAMA	,00
03	HEMOGRAMA	,00
04	ALT	,00
05	AST	,00
06	TRIFLEBINA (TOTAL, DIRETA E INDIRETA)	,00
07	CRÁDIO	,00
08	OXALÉSTROLO TOTAL	,00
09	ORBITINA	,00
10	PROBIOGÊNIO PLASMÁTICO	,00
11	PROSTATASE CALULINA	,00
12	PROSPRO	,00
13	GNIA GT	,00
14	LACTATO	,00
15	PROTEÍNAS SÉRICAS (GASTROSTOTAL)	,00
16	TRICLOPÉDES	,00
17	UREIA	,00
18	URENULISE	,00