

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ADRIANO GEBERT GOMES

**Sistema de Agendamento de Clínica
Veterinária: desenvolvimento de uma
aplicação Web utilizando *framework* Yii 2.0**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em Ciência
da Computação

Orientador: Prof. Dra. Renata de Matos Galante

Porto Alegre
2023

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões

Vice-Reitora: Prof^ª. Patricia Pranke

Pró-Reitora de Graduação: Prof^ª. Cíntia Inês Boll

Diretora do Instituto de Informática: Prof^ª. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Ciência de Computação: Prof. Marcelo Walter

Bibliotecário-chefe do Instituto de Informática: Alexsander Borges Ribeiro

*“Eu acredito que às vezes são as pessoas que ninguém espera nada
que fazem as coisas que ninguém consegue imaginar.”*

— ALAN TURING

AGRADECIMENTOS

Agradeço primeiramente a Deus pelo dom da vida e por estar presente sempre em todas as ocasiões. Aos meus pais e familiares pela paciência e apoio durante meu curso de graduação, principalmente no período em que morei em Alegrete durante meus estudos na Universidade Federal do Pampa.

Agradeço aos professores que possibilitaram o aprendizado durante a minha trajetória acadêmica, em especial aos docentes da Universidade Federal do Pampa e da Universidade Federal do Rio Grande do Sul, ambas instituições em que cursei o bacharelado em Ciência da Computação.

Fica meu agradecimento em especial a minha orientadora, Professora Renata, pela paciência, dedicação e incentivo que possibilitaram de desenvolver esse trabalho de graduação.

Também agradeço aos professores da Escola Técnica Estadual Marechal Mascarenhas de Moraes que ensinaram e contribuíram para minha formação técnica, em especial aos professores Sandro, José Ricardo e Rafaela, cujo ensino de suas áreas possibilitaram o desenvolvimento do sistema de agendamento em versão *desktop* no curso Técnico em Informática.

Ao Centro de Processamento de Dados da UFRGS pela experiência e aprendizado no desenvolvimento de sistemas Web no período como bolsista e servidor, em especial ao meu chefe, Daniel Del Sent Soares, a quem teve a paciência em auxiliar no meu desenvolvimento profissional e aos colegas de trabalho que conheci nesse período.

Aos participantes do experimento de usabilidade que possibilitaram a validação da aplicação desenvolvida e as considerações deste experimento.

Fica meu agradecimento aos amigos e colegas que conheci durante a minha trajetória acadêmica, em especial as minhas amigas e colegas Izadora Dourado Berti e Letícia dos Santos pelo incentivo e motivação durante a minha graduação na UFRGS e aos amigos Gabriel Bronzatti Moro e Tauana Ohland dos Santos pela parceria durante esse período.

RESUMO

Atualmente os sistemas de informações permitiram grande avanços nas organizações, desde a automatizar processos antes manuais, bem como a facilitar a vida das pessoas e organizações. O presente trabalho é derivado de um projeto de conclusão de um curso técnico em informática que consiste em uma aplicação *desktop* de agendamento de consultas veterinárias. O principal objetivo da aplicação é facilitar o agendamento que não precisa mais ser feito via telefone ou até mesmo presencialmente, além de evitar a manutenção de fichas e cadastros manuscritos de tutores e animais. O trabalho proposto nessa monografia tem como objetivo realizar a implementação de uma aplicação Web que possibilite o agendamento de consultas veterinárias através da *internet*, mantendo os cadastros de tutores, médicos e animais de estimação em um sistema de banco de dados. Além disso, com os dados obtidos há a possibilidade de acesso ao histórico de saúde dos animais e a geração de relatórios dos atendimentos realizados. A implementação desta aplicação de agendamento é realizada em linguagem PHP, utilizando o *framework* Yii que adota o padrão arquitetural MVC. Para a persistência dos dados é utilizado um banco de dados relacional MySQL. Após a implementação desta aplicação de agendamento, são realizados testes com os usuários para validação das funcionalidades e avaliação da usabilidade. Constata-se que a aplicação pode agilizar o agendamento e acompanhamento de consultas veterinárias entre tutores e médicos veterinários, além de demonstrar a importância e relevância da tecnologia da informação e dos sistemas informatizados no atendimento de saúde animal.

Palavras-chave: Migração de software. Desenvolvimento de aplicação Web. Agendamento de consultas. Clínica Veterinária.

Veterinary Clinic Scheduling System: development of a web application using framework Yii 2.0

ABSTRACT

Information systems have allowed great advances in organizations, from automating previously manual processes to making life easier for people and organizations. The present work is derived from a final project for a technical course in informatics that consists of a desktop application for scheduling veterinary appointments. The main objective of the application is to facilitate scheduling that no longer needs to be done over the telephone or even in person, in addition to avoiding the maintenance of handwritten forms and records of tutors and animals. The work proposed in this monograph aims to implement a Web application that allows the scheduling of veterinary appointments through the Internet, keeping the records of tutors, doctors, and pets in a database system. Furthermore, with the data obtained, there is the possibility of accessing the health history of the animals and the generation of reports of the services performed. The implementation of this scheduling application is realized in PHP language, using the framework Yii which adopts the architectural pattern MVC. For data persistence, a MySQL relational database is used. After the implementation of this scheduling application, tests are carried out with the users to validate the functionalities and evaluate usability. The application can streamline the scheduling and monitoring of veterinary appointments between tutors and veterinaries, demonstrating the importance and relevance of information technology and computerized systems in animal health care.

Keywords: Software migration. Web application development. Appointment scheduling. Veterinary clinic.

LISTA DE FIGURAS

Figura 2.1	Figura ilustrando a diferença entre o processo cascata (acima) e o processo ágil (abaixo).....	16
Figura 2.2	Representação MVC.....	17
Figura 2.3	Exemplo de página HTML com PHP embutido.....	19
Figura 2.4	Estrutura geral de uma aplicação Yii.....	20
Figura 2.5	Tratamento de requisições no Yii.....	21
Figura 2.6	Painel de controle do XAMPP.....	23
Figura 2.7	Tela Visual Studio.....	24
Figura 2.8	Tela DBeaver.....	25
Figura 3.1	Tela da aplicação VetSoft.....	27
Figura 3.2	Tela da aplicação VetWork.....	28
Figura 3.3	Tela da aplicação VetBase.....	29
Figura 4.1	Tela de Agendamento da Aplicação <i>Desktop</i>	32
Figura 4.2	Diagrama do modelo entidade relacionamento.....	33
Figura 5.1	Diagrama do modelo entidade relacionamento.....	40
Figura 5.2	Layout padrão utilizado nas telas.....	43
Figura 5.3	Layout padrão utilizado nas telas.....	44
Figura 5.4	Tela com os horários em versão responsiva.....	45
Figura 5.5	Tela com campos de Animal para computador.....	45
Figura 5.6	Tela com campos de Animal em versão responsiva.....	46
Figura 5.7	Tela com os horários em versão responsiva.....	46
Figura 5.8	Tela com os horários agendados - versão normal.....	47
Figura 5.9	Tela com os horários agendados - versão responsiva.....	47
Figura 5.10	Autocomplete de tutor no relatório.....	48
Figura 5.11	FiltroTutorAnimalWidget após a seleção de tutor.....	48
Figura 5.12	Trecho de código PHP - <code>actionAlterarDados()</code>	50
Figura 5.13	Trecho de código PHP - <code>obterParametro()</code>	51
Figura 5.14	Diagrama de classes - Modelos Active Record.....	52
Figura 5.15	Diagrama de classes - Modelos auxiliares.....	53
Figura 5.16	Diagrama de classes - Controladores CRUD.....	53
Figura 5.17	Diagrama de classes - Controladores principais.....	54
Figura 5.18	Fluxo de publicação.....	55
Figura 6.1	Tela inicial de login.....	57
Figura 6.2	Tela de cadastro de tutor.....	58
Figura 6.3	Tela de cadastro de tutor - botão Salvar.....	58
Figura 6.4	Tela recuperar senha.....	59
Figura 6.5	Tela recuperar senha - mensagem de envio.....	59
Figura 6.6	E-mail enviado pelo sistema com a nova senha.....	60
Figura 6.7	Tela da página inicial - Tutor.....	61
Figura 6.8	Tela da página inicial sem bloco agendamento - Tutor.....	62
Figura 6.9	Tela da página inicial - Funcionário.....	62
Figura 6.10	Tela da página inicial - Veterinário.....	63
Figura 6.11	Tela novo animal - Veterinário.....	64
Figura 6.12	Tela novo animal - Tutor.....	64
Figura 6.13	Tela novo agendamento.....	65

Figura 6.14	Tela novo agendamento com dados da consulta preenchidos.....	65
Figura 6.15	Tela novo agendamento com horários para agendamento	66
Figura 6.16	E-mail enviado confirmando o agendamento	66
Figura 6.17	Janela modal confirmando cancelamento	67
Figura 6.18	E-mail enviado confirmando o cancelamento.....	68
Figura 6.19	Tela Consultar agenda - Filtros.....	68
Figura 6.20	Tela Consultar agenda - Resultados.....	69
Figura 6.21	Tela finalizar atendimento - dados paciente.....	70
Figura 6.22	Tela finalizar atendimento - dados consulta.....	71
Figura 6.23	Tela finalizar atendimento - dados financeiro.....	71
Figura 6.24	Relatório de consultas - Filtro	72
Figura 6.25	Relatório de consultas - Resultados.....	72
Figura 6.26	Consultar tutor - Filtro	73
Figura 6.27	Consultar tutor - Dados.....	74
Figura 6.28	Consultar tutor - Histórico consultas	74
Figura 6.29	Consultar tutor - Filtro	75
Figura 6.30	Histórico do paciente - Dados	75
Figura 6.31	Histórico do paciente - Histórico de consultas	76
Figura 6.32	Janela de histórico de pesos	76
Figura 6.33	Janela de evolução do peso	77
Figura 6.34	Tela de visualizar atendimento	78
Figura 6.35	Tela de Editar atendimento	78
Figura 6.36	Tela de gerência de logins.....	79
Figura 6.37	Tela de gerência de parâmetros.....	80
Figura 7.1	Gráfico - Faixa etária	85
Figura 7.2	Gráfico - Gênero	86
Figura 7.3	Gráfico - Escolaridade	86
Figura 7.4	Gráfico - Área de formação	87
Figura 7.5	Gráfico - Conhecimentos em informática.....	87
Figura 7.6	Gráfico - Uso dispositivos móveis	88
Figura 7.7	Gráfico - Pessoas que são tutoras	88
Figura 7.8	Gráfico - Animais que pessoas tutoras possui	89
Figura 7.9	Gráfico - Atendimento veterinário.....	89
Figura 7.10	Gráfico - Dispositivo usado no experimento	90
Figura 7.11	Gráfico - Experiência das instruções	91
Figura 7.12	Gráfico - Responsividade.....	91
Figura 7.13	Gráfico - Facilidade cadastro de animal	92
Figura 7.14	Gráfico - Facilidade de agendamento da consulta	92
Figura 7.15	Gráfico - Facilidade de cancelamento da consulta	93
Figura 7.16	Gráfico - Facilidade de finalizar a consulta	93
Figura 7.17	Gráfico - Uso aplicação	94
Figura 7.18	Gráfico - Desempenho aplicação.....	94
Figura 7.19	Gráfico - Utilidade aplicação.....	95
Figura 7.20	Gráfico - Recomendação da aplicação.....	95
Figura E.1	Relatório gerado pela aplicação	127

LISTA DE TABELAS

Tabela 3.1 Comparação entre as aplicações VetSoft, VetWork e VetBase sob diferentes características	30
Tabela 5.1 Parâmetros da aplicação	50
Tabela 5.2 Configuração ambientes de trabalho	56
Tabela D.1 Dicionário de dados da tabela Pessoa.....	123
Tabela D.2 Dicionário de dados da tabela Pessoa.....	124
Tabela D.3 Dicionário de dados da tabela Espécie	124
Tabela D.4 Dicionário de dados da tabela Animal.....	125
Tabela D.5 Dicionário de dados da tabela Peso	125
Tabela D.6 Dicionário de dados da tabela Consulta	126
Tabela D.7 Dicionário de dados da tabela Peso	126

LISTA DE ABREVIATURAS

BD	Banco de Dados
CASE	Computed-Aided Software Engineering
CDN	<i>Content Delivery Network</i>
CRMV	Conselho Regional de Medicina Veterinária
CSS	<i>Cascading Style Sheets</i>
CRUD	<i>Create, Read, Update, Delete</i>
DER	Diagrama Entidade Relacionamento
ER	Entidade-Relacionamento
FK	<i>Foreign Key</i>
HTML	<i>Hipertext Markup Language</i>
JDK	<i>Java Development Kit</i>
JRE	<i>Java Runtime Environment</i>
MVC	<i>Model, View, Controller</i>
PK	<i>Primary Key</i>
PDF	<i>Portable Document Format</i>
PHP	<i>PHP: Hipertext Preprocessor</i>
RG	Registro Geral
SACV	<i>Sistema de Agendamento de Clínica Veterinária</i>
SGBD	Sistema de Gerência de Banco de Dados
TI	Tecnologia da Informação
UML	<i>Unified Modelling Language</i>
Unipampa	Universidade Federal do Pampa
UFRGS	Universidade Federal do Rio Grande do Sul
VCS	<i>Version Control System</i>

SUMÁRIO

1 INTRODUÇÃO	13
2 CONCEITOS	15
2.1 Processo de desenvolvimento de software	15
2.1.1 Processo ágil	15
2.2 História de usuários	16
2.3 Arquitetura MVC	17
2.4 Tecnologias usadas	17
2.4.1 JavaScript	18
2.4.2 Highcharts	18
2.4.3 PHP	18
2.4.4 Composer	19
2.4.5 Yii	19
2.4.5.1 Estrutura aplicação Yii	20
2.4.6 HTML	22
2.4.7 CSS	22
2.4.8 Bootstrap	22
2.4.9 MySQL	23
2.4.10 XAMPP	23
2.4.11 Visual Studio	24
2.4.12 DBeaver	25
2.4.13 GitHub	25
3 TRABALHOS RELACIONADOS	26
3.1 VetSoft	26
3.2 VetWork	28
3.3 VetBase	28
3.4 Comparação	29
4 TRABALHO PRÉVIO	31
4.1 Visão Geral do Sistema de Agendamento de Clínica Veterinária	31
4.2 Tecnologias usadas	32
4.3 Modelo Entidade Relacionamento	33
4.4 Limitações	34
5 PROJETO E DESENVOLVIMENTO	35
5.1 Levantamento de requisitos	35
5.1.1 Histórias do usuário	36
5.1.2 Requisitos não-funcionais	38
5.2 Arquitetura	38
5.3 Modelo Entidade Relacionamento	39
5.4 Desenvolvimento da aplicação	40
5.4.1 Ambiente de desenvolvimento	40
5.4.2 Projeto	41
5.4.3 Frontend	42
5.4.3.1 Layout	42
5.4.3.2 Responsividade	44
5.4.3.3 Widgets	48
5.4.4 Backend	49
5.4.4.1 Rotas	49
5.4.4.2 Processamento de formulários	49
5.4.4.3 Parâmetros de sistema	50

5.4.5	Implementação.....	51
5.4.5.1	Componentes.....	54
5.4.6	Ambiente de produção.....	55
5.4.7	Protocolo de publicação.....	55
6	DEMONSTRAÇÃO.....	57
6.1	Tela de entrada.....	57
6.2	Criar conta.....	57
6.3	Recuperar senha.....	58
6.4	Página inicial.....	61
6.4.1	Tutor.....	61
6.4.2	Funcionário.....	62
6.4.3	Médico Veterinário.....	63
6.5	Novo animal.....	63
6.6	Novo agendamento.....	64
6.7	Cancelar consulta.....	67
6.8	Consultar agenda.....	68
6.9	Finalizar atendimento.....	70
6.10	Relatórios.....	71
6.10.1	Consultas.....	72
6.10.2	Consultar tutor.....	73
6.10.3	Histórico do paciente.....	75
6.10.4	Visualizar atendimento.....	77
6.10.5	Editar atendimento.....	78
6.10.6	Gerência.....	79
7	EXPERIMENTO DE USABILIDADE.....	81
7.1	Metodologia adotada.....	81
7.2	Etapas do experimento.....	82
7.3	Teste de usabilidade.....	82
7.3.1	Lista de ações do tutor.....	82
7.3.2	Lista de ações do médico veterinário.....	83
7.4	Resultados obtidos.....	85
7.4.1	Perfil dos participantes.....	85
7.4.2	Avaliação da usabilidade.....	90
7.5	Considerações do experimento.....	96
8	CONCLUSÃO.....	97
	REFERÊNCIAS.....	99
	APÊNDICE A — SCRIPT PARA GERAÇÃO DO ESQUEMA DE DADOS.....	100
	APÊNDICE B — SCRIPT PARA INSERÇÃO DE DADOS NO BANCO.....	105
	APÊNDICE C — FORMULÁRIO TESTE DE USABILIDADE.....	108
	APÊNDICE D — DICIONÁRIO DE DADOS.....	123
D.1	Pessoa.....	123
D.2	Login.....	124
D.3	Especie.....	124
D.4	Animal.....	124
D.5	Peso.....	125
D.6	Consulta.....	126
D.7	Parametro.....	126
	APÊNDICE E — RELATÓRIO GERADO PELA APLICAÇÃO.....	127

1 INTRODUÇÃO

Os sistemas de informações são amplamente utilizados em diversos setores da sociedade, que possibilitaram melhorar a interação das pessoas, agilidade e eficiência nas organizações, como a automatização de processos manuais e burocráticos e armazenamento em meio digital de cadastros físicos, possibilitando assim o uso estratégico das informações digitalizadas (OLIVEIRA, 2003).

Em algumas clínicas veterinárias o atendimento é realizado por ordem de chegada, sem controle adequado do horário para atendimento. Em outras clínicas o controle é feito de maneira mais rudimentar, utilizando cadernos para o agendamento e fichas para cadastros e consultas. A quantidade de documento físico para manutenção não é eficiente, pois demanda um tempo considerável e pode haver dados inconsistentes, além de ocupar um certo espaço físico e papéis que estão sujeitos a extravio. Para possibilitar um atendimento rápido e de qualidade se faz necessário o uso de um sistema informatizado que concentre essas informações e possibilite novas funcionalidades (*e.g.* acesso ao histórico de consultas) que não seria viável com o uso de fichas e cadernos.

Tendo como base o contexto relatado, isso serviu de inspiração para o desenvolvimento de uma aplicação *desktop* de agendamento de clínica veterinária desenvolvida pelo autor nas disciplinas de Projeto 1 e Projeto 2 do seu curso Técnico em Informática, e que possibilitou a manutenção dos cadastros das personas envolvidas, **médicos veterinários**, **clientes** (tutores) e **pacientes** (animais), como também o agendamento e registro da consulta prestada. Com o registro dessas consultas em um sistema de informação foi possível registrar alguns dados relevantes (*e.g.* prognóstico e diagnóstico) que possibilitaram ter um prontuário eletrônico do paciente e a geração de relatórios dos atendimentos prestados. No entanto, essa aplicação *desktop* possui algumas limitações, tais como: somente funcionários da clínica podem utilizar a aplicação, a usabilidade da aplicação não é muito intuitiva devido a múltiplas janelas e a cada atualização da aplicação deve ser realizada uma nova instalação em cada máquina local.

Considerando essa temática, a proposta deste trabalho de graduação será o desenvolvimento de uma aplicação Web para o agendamento de consultas veterinárias, permitindo que tutores de animais realizem o agendamento e o cancelamento de consultas pela *internet*, mantendo os registros em um sistema de banco de dados. Além disso, com esses registros os veterinários terão acesso ao histórico do paciente, gráfico de evolução do peso e geração de relatórios. Ao longo deste trabalho será mostrado o desenvolvimento de

uma aplicação contemplando todas as etapas de fabricação de um *software*: engenharia de requisitos, modelagem do banco de dados, arquitetura do sistema, implementação e validação através de testes de usabilidade com usuários. A implementação deste sistema de agendamento será realizada no *framework Yii*, em linguagem PHP, utilizando o Bootstrap como *framework* de *frontend* permitindo que a aplicação Web tenha uma boa usabilidade e seja responsiva a diferentes telas. Será adotado o banco de dados relacional MySQL para persistência dos dados e outros componentes serão abordados ao longo do trabalho.

Este trabalho de graduação é descrito ao longo de 8 capítulos. No Capítulo 2 são abordados conceitos importantes para o entendimento deste trabalho e as tecnologias e as ferramentas usadas para o desenvolvimento do sistema.

No Capítulo 3, é apresentada uma revisão do estado da arte, que nesse trabalho é a descrição sucinta de três aplicações voltadas para uso em clínicas veterinárias, sendo realizada uma breve comparação e que servem de inspiração para o desenvolvimento deste trabalho.

No Capítulo 4, tem-se uma breve apresentação da aplicação *desktop* que serviu de base para a aplicação Web descrita neste trabalho. Também são relatadas as principais limitações que justificam a implementação da aplicação Web proposta nesse trabalho.

Na sequência, o Capítulo 5 irá tratar da modelagem e implementação do novo sistema objeto deste trabalho, cobrindo a arquitetura de *software* usada, o modelo de banco de dados adotado, os requisitos levantados e detalhes da implementação.

No Capítulo 6, é apresentada a demonstração da aplicação Web, descrita no estilo de manual do usuário, onde são apresentadas as telas e funcionalidades da nova aplicação.

No Capítulo 7, é abordada a aplicação do teste de usabilidade da aplicação com usuários, com o objetivo de obter dados sobre a percepção de uso da aplicação e apresentação dos resultados obtidos nesse experimento.

Finalmente, no Capítulo 8 é apresentada a conclusão com base no desenvolvimento da nova aplicação e da validação do teste de usabilidade.

2 CONCEITOS

Neste capítulo serão vistos os principais conceitos e referenciais teóricos usados nesse trabalho, relacionados ao processo de desenvolvimento, como também de algumas ferramentas e tecnologias utilizadas para o desenvolvimento da aplicação Web.

2.1 Processo de desenvolvimento de software

A construção de um *software* não é muito diferente de uma construção de um carro, tendo uma série de atividades, métodos e etapas para serem seguidas em comum no desenvolvimento. A essa série de eventos denominamos como processo de desenvolvimento de *software*. Os processos de desenvolvimento são divididos em dois grandes tipos: processo cascata (ou tradicional) e o processo ágil (VALENTE, 2020).

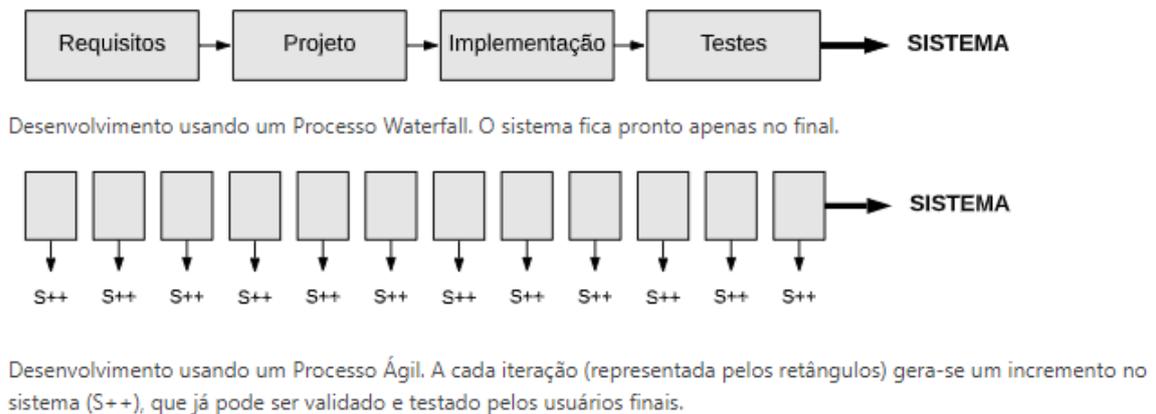
2.1.1 Processo ágil

Os processos ágeis surgem então como uma proposta de evitar os problemas do processo cascata tendo em vista a natureza de desenvolvimento de *softwares*. Diferentemente de outros produtos, a construção de um *software* é difícil mensurar o tamanho, complexidade e prazos. Além disso, muitas vezes os requisitos não estão definidos ou o cliente não sabe o quer com a aplicação (VALENTE, 2020).

Considerando isso, o processo ágil possui como principal proposta a realização de entregas incrementais e iterativas, possibilitando assim que validações com o cliente sejam mais frequentes e a mudança de requisitos não seja traumática ou impacte severamente no desenvolvimento do *software* (VALENTE, 2020).

Temos na Figura 2.1 a comparação entre os processos cascatas e processos ágeis, onde no processo cascata o *software* só fica pronto no final e no processo ágil a cada iteração temos um produto entregável e que traz valor ao usuário (VALENTE, 2020).

Figura 2.1 – Figura ilustrando a diferença entre o processo cascata (acima) e o processo ágil (abaixo)



Fonte: (VALENTE, 2020)

2.2 História de usuários

Histórias de usuários segundo Valente (2020) é uma técnica ágil sugerida por Ron Jeffries para resolver a questão de certas dificuldades que podem ocorrer num processo cascata. O processo cascata tende a gerar uma documentação extensa, não sendo adequado para situações em que existem mudanças de requisitos, como também durante a fase de elicitação de requisitos a fala humana possui muitas ambiguidades que podem levar as funcionalidades do produto final não serem as mesmas que o cliente esperava (VALENTE, 2020).

A técnica de história de usuário consiste dos "três Cs": cartão, conversas e confirmação. Na fase **Cartão** é solicitado aos usuários que escrevam com poucas palavras em um cartão a nova funcionalidade, após isso na fase **Conversas** são realizadas conversas entre desenvolvedores e clientes sobre a nova funcionalidade para detalhamento e explicações e finalmente na fase **Confirmação** é realizado um teste de aceitação da história com os usuários em linguagem de alto-nível, não confundir com teste de *software*, onde as validações das histórias são com base de cenários e casos de exemplos (VALENTE, 2020).

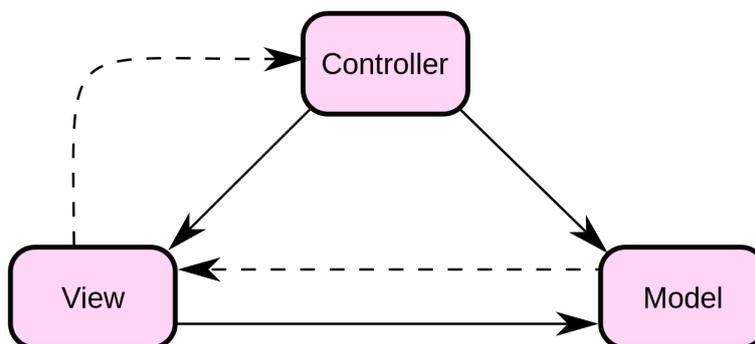
Seria uma boa prática fazer o levantamento inicial dos papéis de usuários que usaram o sistema para definir as histórias de usuários. A história de usuário é geralmente escrita no seguinte formato, sugerido pelo autor Valente: "*Como [papel do usuário], eu gostaria de [realizar algo com o sistema]*" (VALENTE, 2020).

2.3 Arquitetura MVC

O MVC, em inglês *Model, View, Controller*, é um padrão arquitetural de separação das responsabilidades de uma aplicação em modelo, visão e controlador, onde cada grupo tem as seguintes responsabilidades: o **modelo** com as classes relacionadas a regra de negócio, domínio do problema e persistência de dados, a **visão** com as classes relacionadas a interface gráfica e elementos de tela e finalmente o **controlador** com as classes responsáveis por tratar as chamadas, entradas e saídas entre as classes do modelo e da visão (VALENTE, 2020).

Podemos conferir a relação das classes desse padrão arquitetural na Figura 2.2. As associações diretas estão representadas com as linhas sólidas e as associações indiretas estão representadas com as linhas tracejadas (WIKIPÉDIA, 2023).

Figura 2.2 – Representação MVC



Fonte: (WIKIPÉDIA, 2023)

2.4 Tecnologias usadas

Nesta seção são apresentadas as tecnologias e ferramentas utilizadas para o desenvolvimento da aplicação Web proposta nesse trabalho. As tecnologias apresentadas envolvem linguagens de programação e bibliotecas para implementação do *frontend* e do *backend* da aplicação, e as ferramentas usadas para criar um servidor local, codificação e acesso ao banco de dados.

2.4.1 JavaScript

Javascript é uma linguagem multiparadigma no formato de *script* amplamente usada em páginas Web. Ela é uma linguagem do tipo interpretada, com tipagem fraca, utilizada no lado do cliente, ou seja, é executada pelo navegador Web que possibilita que as páginas Web sejam mais dinâmicas e interativas para o usuário (DOCS, 2023).

2.4.2 Highcharts

Highcharts é uma biblioteca escrita em Javascript que possibilita a geração de gráficos em formato SVG, tendo uma vasta documentação e exemplos de gráficos, sendo responsiva, customizável e possibilitando a interação do usuário com os gráficos gerados, além de possuir recursos de exportação dos gráficos em formatos de imagens e arquivo PDF (HIGHCHARTS, 2023).

Para uso do Highcharts na aplicação deve ser feito o *download* da biblioteca ou por meio da inclusão de *links* de CDN. Para organizações sem fins lucrativos, projetos pessoais e escolares não é necessário adquirir uma licença para uso (HIGHCHARTS, 2023).

2.4.3 PHP

O PHP, *PHP: Hypertext Preprocessor*, é uma linguagem de programação amplamente usada para o desenvolvimento de aplicações Web, tendo uma sintaxe semelhante à linguagem C e Java. É uma linguagem interpretada no servidor Web onde é gerado código HTML a ser renderizado no navegador, possibilitando assim a criação de páginas dinâmicas. Possui uma tipagem dinâmica, sendo multi-paradigma, tendo recursos de programação orientada a objetos e programação funcional. Uma das características marcantes do PHP é que o código-fonte pode ser embutido dentro no HTML com o uso das *tags* `<?php` e `?>`, conforme pode ser conferido na Figura 2.3 (PHP, 2023).

Figura 2.3 – Exemplo de página HTML com PHP embutido

```
1 <html>
2   <head>
3     <title>Página Web</title>
4   </head>
5   <body>
6     <h1>Página Web Dinâmica</h1>
7
8     <?php
9       echo "Olá mundo!";
10    ?>
11  </body>
12 </html>
```

Fonte: próprio autor

2.4.4 Composer

O Composer é um gerenciador de dependências para projetos desenvolvidos em PHP, lembrando um pouco a ideia dos gerenciadores de pacotes de distribuições Linux. Os projetos PHP que utilizam o Composer possuem uma pasta *vendor* que contém as bibliotecas PHP que o projeto vai precisar utilizar, isso facilita para a implantação em outros ambientes, na atualização dos pacotes e organização e na reutilização de código-fonte (COMPOSER, 2023).

2.4.5 Yii

O Yii é um *framework* da linguagem PHP que possibilita o desenvolvimento de uma ampla variedade de aplicações Web, sendo um dos seus diferenciais o alto desempenho e o uso de *caching*. O nome Yii é um acrônimo da língua chinesa para "simples e evolutivo" que é filosofia que o Yii se propõe de que a codificação deve ser simples, no entanto, elegante (YII, 2023).

A exemplo de outros *frameworks* da linguagem PHP, o Yii adota como padrão arquitetural e de organização o MVC (*Model, View, Controller*). Além disso, o Yii possui funcionalidades como construtor de consultas, *Active Records*, conexão a bancos de dados relacionais e não relacionais, além de diversas extensões para o Yii que podem ser incluídas no projeto (YII, 2023).

Existem duas versões disponíveis deste *framework*: a Yii 1.1 que está em modo de manutenção e a Yii 2.0 que é a versão estável mais recente. A versão 2.0 requer o uso do PHP 5.4 ou superior. Existe previsão de lançamento da versão Yii 3.0, mas ela ainda não é estável para uso em produção (YII, 2023). Para a implementação da aplicação Web proposta nessa monografia será usada a versão Yii 2.0.

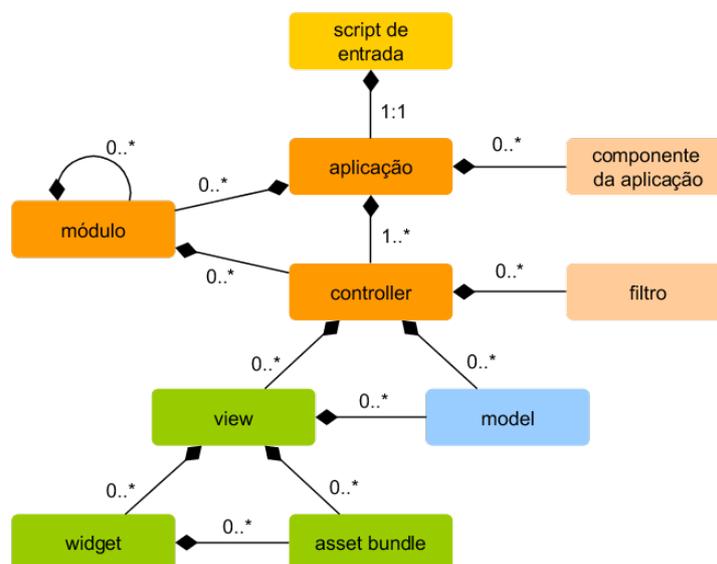
2.4.5.1 Estrutura aplicação Yii

Uma aplicação Yii adota o padrão arquitetural MVC, composto de controladores, modelos e visão. Além disto, uma aplicação Yii é composta de outras entidades (YII, 2023). Conforme a literatura da documentação do Yii (2023), podemos descrever as principais como:

- **Módulo:** um módulo pode ser considerado uma subaplicação coesa com uma estrutura MVC completa, sendo que podemos ter diversos módulos no mesmo projeto;
- **Filtros:** são componentes utilizados para executar algum código antes ou depois de uma chamada de ação;
- **Widgets:** são componentes reutilizáveis de interface usadas na visão da aplicação, tendo um controlador independente.

Podemos ver a relação entre as entidades em uma aplicação Yii na Figura 2.4.

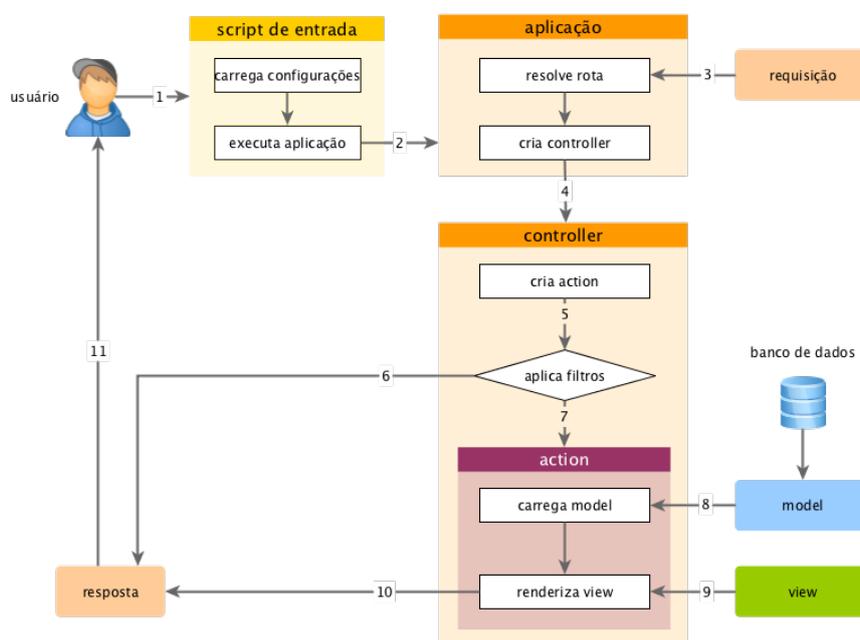
Figura 2.4 – Estrutura geral de uma aplicação Yii



Fonte: (YII, 2023)

Temos na Figura 2.5 a sequência de eventos relacionadas ao ciclo de vida de uma requisição em Yii. Toda a requisição tem início quando usuário acessa um *script* de entrada, que pode ser uma página */web/index.php* por exemplo. Após isso são aplicadas as configurações da aplicação, feita uma nova instância dessa aplicação e realizada a resolução da rota. Após isso é feita uma nova instância do controlador, executa a ação definida na rota, são aplicados os filtros, feita a instância dos modelos e finalmente a renderização da visão no navegador (YII, 2023).

Figura 2.5 – Tratamento de requisições no Yii



Fonte: (YII, 2023)

2.4.6 HTML

O HTML (Linguagem de Marcação de Hipertexto) é uma linguagem utilizada para estruturar e escrever as páginas estáticas da Web. O conceito de Hipertexto é referente aos *links* que possibilitam interligar a outras páginas Web. A escrita de um arquivo HTML é utilizando as *tags* HTML, que consistem no formado `<tag>` para abrir e `</tag>` para fechar (DOCS, 2023).

2.4.7 CSS

O CSS, em inglês *Cascading Style Sheets*, é uma linguagem utilizada para definir estilo em HTML ou XML, determinando as características como tamanho, cores, margens e entre outros elementos visuais de páginas Web (DOCS, 2023). Uma das grandes vantagens do CSS é a possibilidade de reuso e de possibilitar que todas as páginas Web adotem o mesmo padrão visual (DOCS, 2023).

2.4.8 Bootstrap

O Bootstrap é um *framework front-end* de código aberto amplamente usado em diversas aplicações. Foi criado inicialmente para o Twitter nos anos 2010 como um guia de estilo e foi posteriormente tornado público. É de fácil instalação, sendo disponível por meio de *links* CDN ou como pacote do *composer* (BOOTSTRAP, 2023).

O Bootstrap adota o *mobile first*, que é pensar primeiramente no *design* responsivo, aplicando o mínimo de estilos para as telas ficarem adequadamente ajustadas para o menor *breakpoint*, cabendo ao desenvolvedor fazer apenas ajustes pontuais para adequar o *layout* para dispositivos maiores (BOOTSTRAP, 2023). Os *breakpoints* que são pontos de interrupção disponíveis no Bootstrap, que juntamente com o CSS *media queries*, possibilitam controlar o *layout* a partir de um determinado tamanho de tela (BOOTSTRAP, 2023).

Para uso do estilo e componentes do Bootstrap basta apenas utilizar as classes CSS em *tags* HTML, sendo seu uso bastante fácil e universal para diversas aplicações (BOOTSTRAP, 2023).

2.4.9 MySQL

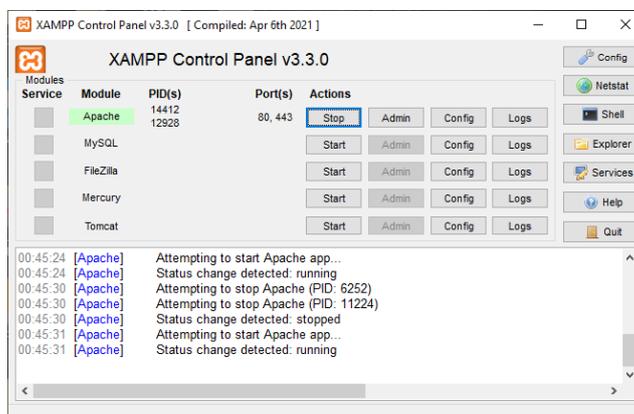
O MySQL é um dos mais conhecidos sistema gerenciador de banco de dados (SGBD), sendo desenvolvido pela Oracle. Suas características principais são: escrito em C e C++, trabalha em diferentes plataformas, suporte a *mutithreaded* e usa tabelas de discos em *B-tree*. O MySQL é um banco de dados relacional, que utiliza a linguagem SQL para a realização dos acessos e consultas ao banco de dados (MYSQL, 2023).

2.4.10 XAMPP

O XAMPP é uma distribuição que realiza a instalação e a configuração automática de diversas aplicações que são necessárias para criar um servidor Web para possibilitar a execução de aplicações Web, e consequentemente possibilitar o desenvolvimento destas aplicações Web. A distribuição XAMPP consiste dos seguintes serviços e ferramentas: Servidor Apache, MariaDB, PHP, phpMyAdmin, OpenSSL, XAMPP Control Panel, Webalizer, Mercury Mail Transport System, FileZilla FTP Server, Tomcat e Perl (XAMPP, 2023).

O XAMPP inclui um painel de controle, conforme a Figura 2.6, que possibilita ativar/desativar os serviços, acesso às configurações e *logs*. Esse painel possibilita o controle dos serviços de maneira fácil, auxiliando no aumento da produtividade.

Figura 2.6 – Painel de controle do XAMPP



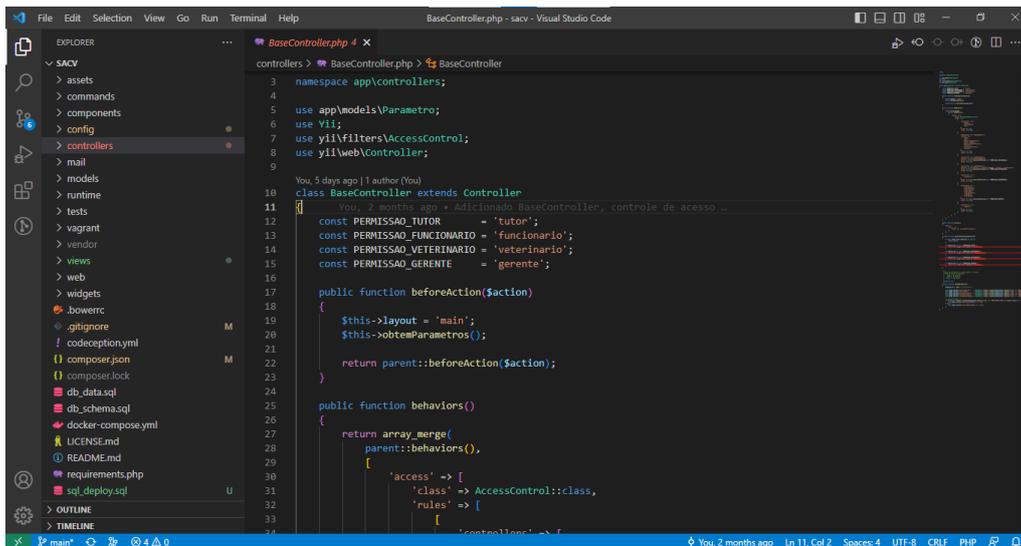
Fonte: próprio autor

2.4.11 Visual Studio

O Visual Studio é um editor de código-fonte riquíssimo desenvolvido pela Microsoft e está disponível gratuitamente para os sistemas Windows, Linux e MacOS. Entre as características marcantes são o uso de extensões que adicionam funcionalidades e suporte a outras linguagens de programação. Podemos citar como recursos interessantes a integração com o Git, depuração de código, formatação de código-fonte, troca de tema, paleta de comandos e integração com o console (MICROSOFT, 2023).

Para o desenvolvimento da aplicação Web deste trabalho foi utilizada essa ferramenta devido à facilidade e produtividade. Temos uma imagem desta ferramenta na Figura 2.7.

Figura 2.7 – Tela Visual Studio

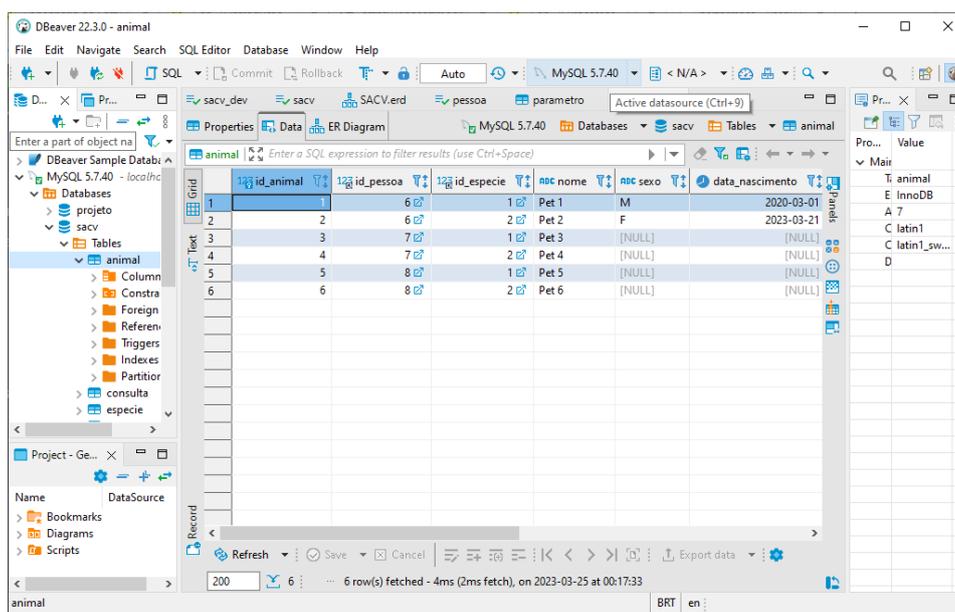


Fonte: próprio autor

2.4.12 DBeaver

O DBeaver é uma ferramenta de banco de dados, com suporte para diversos SGBDs, tais como: MySQL, PostgreSQL, SQLite, Firebird, MS SQL, entre outros (DBE-AVER, 2023). A ferramenta possui uma interface gráfica fácil de usar, conforme a Figura 2.8.

Figura 2.8 – Tela DBeaver



Fonte: próprio autor

2.4.13 GitHub

O Git é um sistema de controle de versão (VCS) descentralizado. Um sistema de controle de versões consiste em um controle das alterações realizadas em um código-fonte, permitindo ter um histórico das alterações, reverter o código-fonte para um estado anterior e possibilitar o trabalho no mesmo código por várias pessoas. Entre as vantagens do Git comparado a outros VCS é que ele é descentralizado, onde cada máquina possui uma cópia completa do repositório, possibilitando a codificação do projeto mesmo estando *offline* e os recursos avançados para realização de *branches* (GITHUB, 2023).

O GitHub oferece a hospedagem de repositórios que utilizam o Git, possibilitando uma melhor colaboração com as pessoas e compartilhamento de código. O serviço do GitHub possui planos gratuito e pagos (GITHUB, 2023).

3 TRABALHOS RELACIONADOS

Neste capítulo serão apresentados três sistemas para gestão de clínicas veterinárias como estado da arte e feita uma comparação e análise dos requisitos entre eles.

Para o trabalho do curso técnico foram estudadas duas ferramentas, sendo que nesse trabalho de graduação será revista a aplicação VetSoft e incluídos para estudo duas novas aplicações: VetWork e VetBase. A escolha destas ferramentas foi devida a relevância em site de buscas para aplicações voltadas a clínicas veterinárias.

3.1 VetSoft

O VetSoft é um *software* proprietário de gestão de *pet shops*, clínicas e hospitais veterinários, possuindo uma variedade de funcionalidades relacionadas a gestão da clínica e a saúde animal (VETSOFT, 2023b).

O VetSoft é disponibilizado nas versões *desktop* e *Web* (nuvem), porém uma aplicação não se comunica com a outra, sendo tratados como produtos distintos. Ambas versões, tem planos pagos conforme as funcionalidades oferecidas, sendo que versão *Desktop* possui um plano de testes limitado e a versão *Web* possui um plano gratuito que permite apenas um usuário e trinta animais (VETSOFT, 2023b). Existe a possibilidade de migração da base de dados com outros sistemas de clínica veterinária conhecidos no mercado (VETSOFT, 2023b).

Conforme o site desta aplicação temos como as principais funcionalidades: ficha tutor; ficha animal; histórico clínico; gestão de estoques; financeiro; emissão fiscal; banho e tosa; farmácia/materiais e imunizações (VETSOFT, 2023b). Na Figura 3.1 temos uma tela dessa aplicação.

Figura 3.1 – Tela da aplicação VetSoft

The screenshot displays the 'Ficha do Cliente' (Client Card) for Daniela Fátima Joana Araújo. The interface is in Portuguese and shows a summary of the account balance and a list of transactions. A red box highlights the 'Conta' (Account) tab in the navigation menu.

Summary of Account Balance:

- Total da conta: 299,00
- Crédito/Parcial: 100,00
- Saldo devedor: 199,00

Table of Transactions:

Situação	Data	Hora	Usuário	Animal	Produto/Serviço	Grupo	VI Unitário	Qtd	Desconto	VI Total
✓	20/10/20	13:29	master	Robin	Vacina V10	Clinica/Vacinas	100,00	1,000	1% / 1,00	99,00
✓	20/10/20	13:18	master	Robin	Vacina V10	Clinica/Vacinas	100,00	1,000	—	100,00
✓	20/10/20	13:18	master	Robin	Vacina V70	—	100,00	1,000	—	100,00
Total										299,00

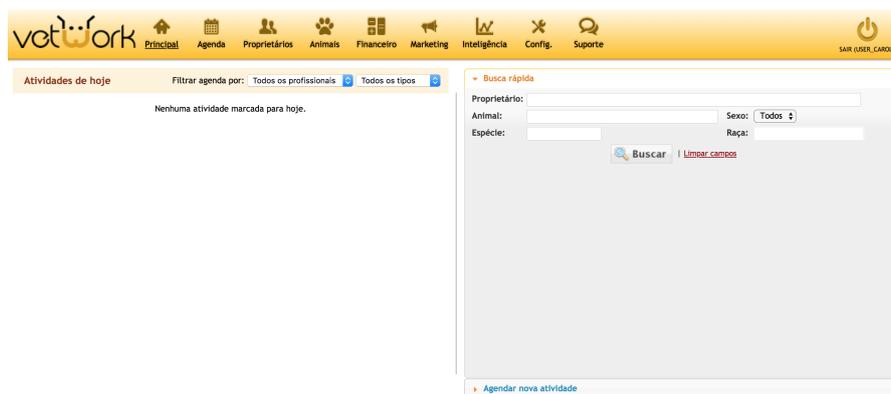
Fonte: (VETSOFT, 2023a)

3.2 VetWork

O VetWork é um sistema Web em nuvem destinado para *pet shops* e clínicas veterinárias, abrangendo diversas funcionalidades desde a gestão financeira a prontuário eletrônico. O diferencial dessa ferramenta é que muitas ações são automatizadas como a geração de indicadores e controle de estoque, além da possibilidade de envio de arquivos e imagens no prontuário eletrônico dos animais. Para uso desta plataforma é necessário aderir a algum dos planos pagos oferecidos. Existe a possibilidade de um período gratuito para testes durante 7 dias (VETWORK, 2023b).

Dentre as funcionalidades e recursos oferecidas pela desenvolvedora do sistema, podemos destacar: agenda vinculada ao caixa; gestão clínica; gestão financeira; gestão de documentos; NFC-e; calendário; inteligência de negócios; possibilidade de migração de dados e treinamento (VETWORK, 2023b). Na Figura 3.2 temos uma tela dessa aplicação.

Figura 3.2 – Tela da aplicação VetWork



Fonte: (VETWORK, 2023a)

3.3 VetBase

O VetBase é um sistema online voltado para *pet shops* e veterinária, abrangendo recursos relacionados ao cadastro de tutores e animais, saúde animal, controle financeiro e lançamento de vendas. Existe a possibilidade de utilizar por um período de teste gratuito, porém após isso deve-se escolher entre duas possibilidades de planos pagos com recorrência mensal (VETBASE, 2023).

Dentre as funcionalidades do sistema VetBase, destacam: cadastro de tutores e

animais; histórico clínico; exames e vacinas aplicadas; agenda dos procedimentos realizados; controle financeiro e consulta aos cadastros. Na Figura 3.3 temos uma tela dessa aplicação.

Figura 3.3 – Tela da aplicação VetBase

AGENDA	CLÍNICA	ESTÉTICA	INTERNAÇÃO	FILA							
0	1	2	3	4	5	6	7	8	9	10	11
29	30	31									
Jan 2020	1	2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31				
08:00			Ther (Beata)								
08:30	RETORNO/REVISÃO MAY (Vicente)		Está comengando a ser movimentar, não totalmente sem tosse, afirmação de...	Matheo (Melissa)		Pratisha (Hebea)		Shiff (Augusto)		Luigo (Alicia)	
09:00	ORTO Sara (Marcela)		Está muito bem! Nada de alteração n...	Smacking Pumpkins (Marcelo)		REAVALIAÇÃO Lara (Erika)		CONSULTA CÃO/CATADO Roxley (Marcelo)		Minimiro (Gustavo)	
09:30	CONSULTA CÃO/CATADO Nicolau (Marcelo)		Tem de calmar, vai em outra clinic...	Prerito (Carla)		CONSULTA CÃO/CATADO Diplo El Shaded Yeh (Melissa)		CONSULTA CÃO/CATADO Aira (Samuel)		Rube (Gustavo)	
10:00	RETORNO/REVISÃO Lindora (Marcelo)		Deu papo 5 dias, tá comendo só na co...	CONSULTA CÃO/CATADO Zora (Massimiliano)		CONSULTA CÃO/CATADO Rico (Marcelo)		CONSULTA CÃO/CATADO Sonia (Camilla)		CONSULTA CÃO/CATADO Mariano (Vicente)	

Fonte: (VETBASE, 2023)

3.4 Comparação

Os sistemas apresentados apresentam muitas características em comum, se destacando que todos possuem versão Web, mantém muitas funcionalidades que não ficam restritos somente ao uso de clínica veterinária, tais como: controle financeiro, controle de estoque e venda de produtos.

Tanto os sistemas VetSoft e VetWork divulgam funcionalidades de agendamento para banho e tosa, apesar que no VetBase temos um módulo referente ao controle estético, que parece estar também relacionado a banho e tosa. Aparentemente o VetBase possui um foco maior a saúde animal, apesar de também contar com as funcionalidades de venda e estoque. O VetWork e o VetBase indicam a possibilidade de anexar fotos e arquivos relacionados a saúde do animal.

Apenas o VetSoft oferece uma versão *desktop* para clínicas, apesar de não se comunicar com a versão Web. Todos os sistemas compartilham em comum que são sistemas proprietários.

Em relação à navegação responsiva, apenas o sistema da VetSoft em sua versão Web indicou isso em seu site como um diferencial.

Aparentemente as ferramentas citadas são utilizadas pelos usuários da clínica ve-

terinária, isto é, não parece ter envolvimento direto do cliente no uso destas aplicações, tais como acesso ao próprio cadastro ou agendamento de consultas veterinárias.

Conforme a Tabela 3.1, temos o comparativo de recursos entre as aplicações estudadas como estado da arte.

Tabela 3.1 – Comparação entre as aplicações VetSoft, VetWork e VetBase sob diferentes características

<i>Característica</i>	<i>VetSoft</i>	<i>VetWork</i>	<i>VetBase</i>
Versão <i>desktop</i>	Sim	Não	Não
Versão Web	Sim	Sim	Sim
Gratuito	Sim, limitada	Não	Não
Possui período de testes	Sim	Sim	Sim
Possui agendamento pelo tutor	Não	Não	Não
Possui funcionalidades banho e tosa	Sim	Sim	Talvez
Possui controle de estoques	Sim	Sim	Sim
Possui histórico de saúde	Sim	Sim	Sim

Fonte: próprio autor

Sendo assim, poderia ser desenvolvida uma aplicação que contemple a possibilidade do tutor poder realizar seu próprio cadastro e de seus animais, bem como realizar o agendamento e cancelamento de consultas veterinárias. Funcionalidades estas que não estão contempladas no comparativo das ferramentas citadas.

4 TRABALHO PRÉVIO

Neste capítulo são abordados introdução, funcionalidades, telas principais e limitações de uma aplicação *desktop* desenvolvida como trabalho de conclusão do seu curso Técnico em Informática do autor desta monografia. A revisão desse trabalho prévio serviu como inspiração para o desenvolvimento da aplicação Web proposta nessa monografia.

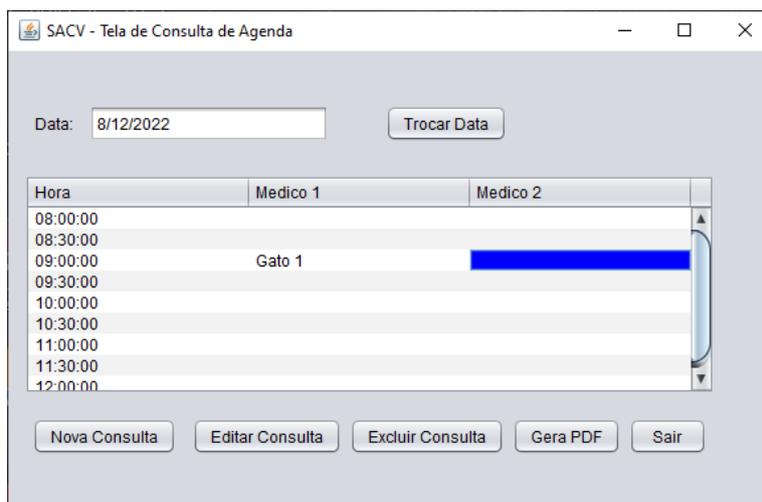
4.1 Visão Geral do Sistema de Agendamento de Clínica Veterinária

A aplicação *desktop* prévia possui como objetivos principais a informatização dos cadastros de clientes e animais, a realização de agendamento das consultas via aplicação e a geração de relatórios em formato PDF. Essa aplicação foi desenvolvida em 2014 como projeto de conclusão nas disciplinas de Projeto 1 e 2 do curso Técnico em Informática da Escola Técnica Estadual Marechal Mascarenhas de Moraes, em Cachoeirinha-RS. O orientador deste trabalho prévio foi o Professor José Ricardo Borba.

Para a aplicação de agendamento *desktop* podemos elencar como as principais funcionalidades:

- Cadastro de cliente, caixa, veterinário e animal;
- Agendamento de consultas;
- Consulta a agenda;
- Histórico dos atendimentos, com acesso ao prontuário do animal;
- Emissão de relatórios em formato PDF;
- Senha criptografada no banco de dados.

Na Figura 4.1 é apresentada a tela de agendamento de consultas da aplicação *desktop*. Nessa tela é possível acompanhar as consultas agendadas (com o nome do animal) ou horários disponíveis para agendamento. A seleção é feita via clique do *mouse* e deve-se clicar nos botões disponíveis em tela para executar uma determinada funcionalidade.

Figura 4.1 – Tela de Agendamento da Aplicação *Desktop*

Fonte: próprio autor

4.2 Tecnologias usadas

O desenvolvimento dessa aplicação *desktop* de agendamento foi na linguagem Java, utilizando o paradigma de programação orientado a objetos, sendo o JDK 1.8 utilizado. Foi adotado o padrão arquitetural MVC para a construção dessa aplicação, separando as classes com alguma lógica de negócio e comunicação entre dados e interfaces em **controle**, as classes de mapeamento objeto-relacional em **modelo** e as interfaces em **visão**. Para a construção das interfaces gráficas (GUI) foi utilizada a biblioteca Swing nativa do Java. A geração dos relatórios PDF foi possível com o uso da biblioteca iText 4.2.0.

Para a persistência dos dados foi utilizado um banco de dados relacional MySQL 5.7. Para a conexão do banco de dados e persistência dos dados no sistema desenvolvido em Java, foi utilizado o *framework* Hibernate que possibilitou o mapeamento objeto-relacional entre as classes do modelo da aplicação e as tabelas do banco de dados relacional.

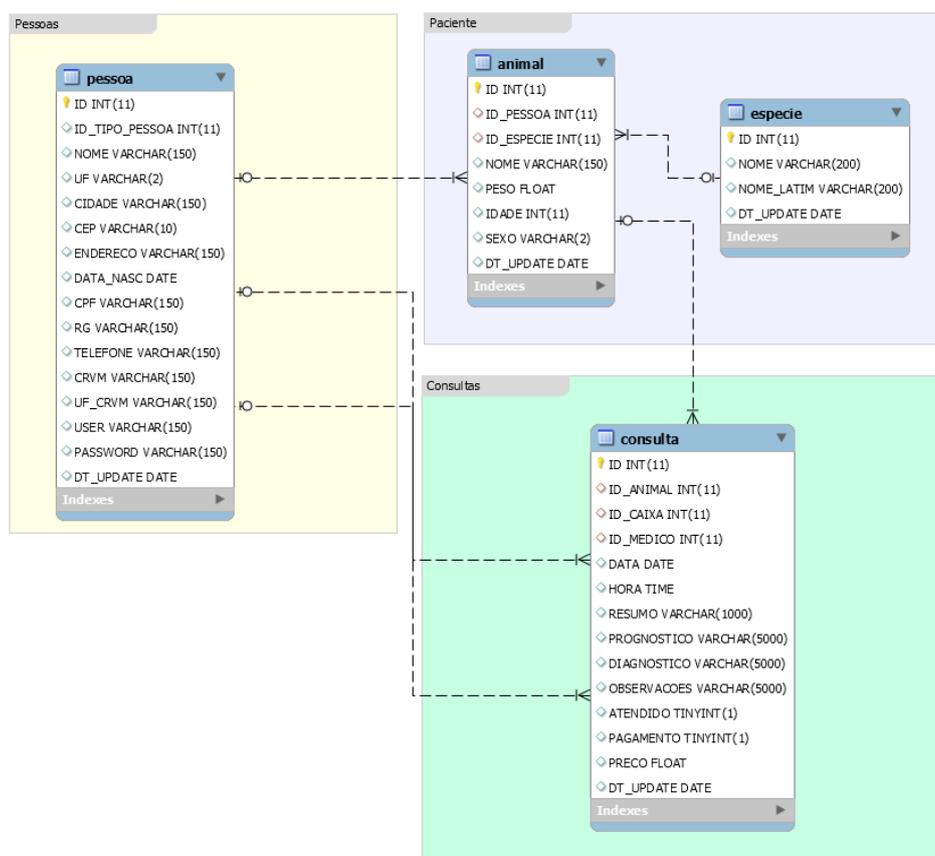
Para o desenvolvimento, depuração e projeto das telas gráficas foi usada a ferramenta IDE NetBeans e para a geração do diagrama do modelo ER foi usado a ferramenta MySQL Workbench através do método de engenharia reversa. A elaboração dos diagramas de caso de uso e diagramas de atividade foi utilizada como ferramenta CASE o Astah Community.

4.3 Modelo Entidade Relacionamento

Para a aplicação *desktop* desenvolvida foi feito o modelo conceitual do banco de dados, conforme a Figura 4.2, consistindo de quatro entidades: Pessoa, Animal, Especie e Consulta.

Na tabela **Pessoa** são mantidos os registros de Médicos Veterinários, Clientes, Caixas e Gerente de TI, também possui os atributos relacionados ao endereço e específicos da especialização (e.g. CRMV do Médico Veterinário). Na tabela **Especie** são armazenados os registros que representam o tipo de animal (e.g. Gato, Cachorro), tendo relacionamento com a tabela **Animal**. Na tabela **Consulta** é onde temos o registro de cada consulta agendada, persistindo também os dados referentes ao diagnóstico, prognóstico e observações da consulta. Com base nessa última tabela serão gerados os relatórios e prontuários do paciente (animal).

Figura 4.2 – Diagrama do modelo entidade relacionamento



Fonte: (GOMES, 2014)

4.4 Limitações

Após a revisão da aplicação *desktop* percebeu-se que para execução das ações são geradas muitas telas e são necessários vários acionamentos de botões que podem dificultar a usabilidade do usuário. Sendo necessária uma revisão das interfaces gráficas e uma reformulação do *frontend* com vistas a melhoria de usabilidade e produtividade.

Em relação à implantação de uma aplicação Java desenvolvida em 2014, deve-se ter uma atenção especial no uso das versões mais recentes do JDK do Java, devido à compatibilidade do Hibernate e também ao uso da versão 5.7 do MySQL. Isso foi um fator de dificuldade para o autor deste trabalho ao tentar implantar esse sistema novamente para revisão.

Na revisão do modelo de dados foram identificadas limitações como não ter o histórico de peso do animal, a idade ser um valor variável e impreciso, não tendo atributos para identificar se o animal veio a óbito. Como possíveis melhorias poderia ser expandido o modelo ER para contemplar uma tabela apenas com os atributos de *login* e uma tabela para manter parâmetros e configurações do sistema. Não temos o histórico e evolução da medição de peso do animal.

Também foram encontrados alguns *bugs* que precisariam ser corrigidos, como também comportamentos que precisariam de certas melhorias, como exemplo, retorno das validações dos cadastros efetuados. Um outra limitação de uma aplicação *desktop* é que a cada necessidade de atualização da aplicação é necessário realizar uma nova instalação em cada máquina local da organização, além do uso da aplicação estar restrito apenas uma *intranet*, sem muitas possibilidades para uma possível evolução para permitir o acesso remoto do cliente a aplicação, por exemplo.

Em relação ao papel das personas existe um grande protagonismo do **Caixa** ou do **Médico Veterinário** no uso desta aplicação *desktop*, onde o cliente não possui nenhum acesso a esta aplicação e fica dependente de outra pessoa para uma simples atualização de seus dados, fazer um agendamento ou emitir o histórico de saúde do animal que está sob seus cuidados.

5 PROJETO E DESENVOLVIMENTO

Neste capítulo são abordadas as etapas para a produção da nova implementação do Sistema de Agendamento de Clínica Veterinária como uma aplicação Web, tendo como inspiração as funcionalidades e modelo de dados da aplicação *desktop* anteriormente mencionada. A aplicação Web proposta foi refeita sem aproveitamento direto de código-fonte da aplicação anterior, pela diferença entre as linguagens de programação e plataformas usadas (*desktop* e Web).

Portanto, é descrito nesse capítulo sobre o projeto, ambiente de desenvolvimento, a revisão dos atores (personas) envolvidos, dos requisitos funcionais e não funcionais e do modelo de dados, não sendo o objetivo deste trabalho de evolução do software manter a retrocompatibilidade da versão *desktop* da aplicação de agendamento. A implementação dessa nova aplicação foi realizada utilizando o *framework* Yii 2.0 em linguagem PHP, que possui diversos componentes e classes que possibilitarão que o desenvolvimento seja realizado com maior produtividade.

5.1 Levantamento de requisitos

Foi visto no Capítulo 4 que o sistema *desktop* tinha um contexto de ser utilizado apenas por funcionários da clínica veterinária e isso implicava em uma série de limitações, tais como: o tutor ter que ligar ou ir presencialmente para agendar ou cancelar a consulta, precisar agendar ou cancelar uma consulta em horário comercial, usabilidade prejudicada do sistema pela necessidade de vários cliques e a necessidade de instalação de uma nova versão do sistema a cada atualização.

Sendo assim, com a revisão do sistema de agendamento e a implementação de uma aplicação Web, optou-se em mudar o objetivo da aplicação quanto ao usuário, ou seja, agora o protagonista da aplicação será o próprio tutor, tendo a possibilidade dele mesmo realizar o seu próprio cadastro, do seu animal e realizar agendamentos ou cancelamentos de consultas.

Para que isso fosse possível, foram incluídos novos requisitos e revisados os anteriores para contemplar o novo cenário, principalmente em relação a criação de conta, alteração de dados, alteração de senhas, envio de e-mails e acompanhamento da evolução do peso do animal. Além disso, foram realizados novos ajustes nas telas e devido ao CPF se o documento padrão para identificação, foi removido o campo relacionado ao RG.

Em relação à segurança a senha, na aplicação *desktop* era usado o sha1 para criptografar a senha digitada pelo usuário no momento da gravação do banco. No entanto, o sha1 é considerado inseguro para isso, sendo adequado outro tipo. No Yii 2 existem métodos auxiliares que possibilitam a geração da senha, utilizando a função *crypt*, segundo o guia do *framework* Yii (2023). Segundo a documentação do *framework*, a criptografia usada é a *bcrypt* que possui uma segurança bem alta em relação a ataques de força bruta (YII, 2023).

5.1.1 Histórias do usuário

Devido à evolução da aplicação *desktop* para uma nova aplicação Web e a mudança dos papéis de usuários, onde temos o protagonismo do tutor na utilização do sistema, se faz necessário uma nova análise e engenharia de requisitos.

Com a inspiração da aplicação *desktop*, se optou por uma abordagem ágil para o levantamento dos requisitos através das histórias do usuário, que consiste em desejos de um determinado papel de usuário, que nesse caso foram: usuário, tutor, funcionário, veterinário e gerente de TI.

A definição das histórias do usuário foram realizadas pelo próprio autor desta monografia. No trabalho anterior as funcionalidades também foram definidas pelo próprio autor da monografia em modelo cascata, porém com pedidos do orientador para inclusão da emissão de relatórios e geração de PDFs.

O papel do usuário consiste das funcionalidades mais básicas da aplicação e é comum a todos os utilizadores da aplicação Web. A seguir, são listadas as histórias de usuários para esta aplicação.

- **Usuário**

- Como usuário, eu gostaria de entrar na aplicação.
- Como usuário, eu gostaria de criar minha conta.
- Como usuário, eu gostaria de poder recuperar minha senha.
- Como usuário, eu gostaria de alterar meus dados.
- Como usuário, eu gostaria de alterar minha senha.

- **Tutor**

- Como tutor, eu gostaria de ver meus agendamentos.

- Como tutor, eu gostaria de fazer um novo agendamento.
 - Como tutor, eu gostaria de cancelar uma consulta agendada.
 - Como tutor, eu gostaria de receber um e-mail de confirmação do agendamento.
 - Como tutor, eu gostaria de receber um e-mail de cancelamento da consulta.
 - Como tutor, eu gostaria de adicionar meu animal.
 - Como tutor, eu gostaria de editar os dados do meu animal.
 - Como tutor, eu gostaria de ver os animais cadastrados.
- **Funcionário**
 - Como funcionário, eu gostaria de ver as consultas agendadas do dia.
 - Como funcionário, eu gostaria de fazer um novo agendamento para um cliente.
 - Como funcionário, eu gostaria de cancelar as consultas agendadas de um cliente.
 - Como funcionário, eu gostaria de adicionar um animal de um cliente.
 - Como funcionário, eu gostaria de consultar a agenda das consultas marcadas.
- **Veterinário**
 - Como veterinário, eu gostaria de ver as consultas agendadas para mim.
 - Como veterinário, eu gostaria de fazer um novo agendamento para um cliente.
 - Como veterinário, eu gostaria de cancelar as consultas agendadas de um cliente.
 - Como veterinário, eu gostaria de adicionar um animal de um cliente.
 - Como veterinário, eu gostaria de editar os dados de um animal de um cliente.
 - Como veterinário, eu gostaria de consultar a agenda das consultas marcadas.
 - Como veterinário, eu gostaria de gerar relatórios dos atendimentos realizados.
 - Como veterinário, eu gostaria de gerar relatórios em PDF dos atendimentos realizados.
 - Como veterinário, eu gostaria de acessar os dados de um tutor.
 - Como veterinário, eu gostaria de acessar os dados de um animal.
 - Como veterinário, eu gostaria de ver o histórico de consultas dos animais de um tutor.
 - Como veterinário, eu gostaria de ver o histórico de consulta de um animal.
 - Como veterinário, eu gostaria de ver os dados de uma consulta.

- Como veterinário, eu gostaria de editar os dados de uma consulta.
- Como veterinário, eu gostaria de ver o histórico de peso de um animal.
- Como veterinário, eu gostaria de ver a evolução do peso de um animal em um gráfico.
- **Gerente de TI**
 - Como gerente de TI, eu gostaria de ter os mesmos acessos de um veterinário.
 - Como gerente de TI, eu gostaria de gerenciar os CRUDs das tabelas do banco pela aplicação.
 - Como gerente de TI, eu gostaria de gerenciar os parâmetros de configuração da aplicação.

5.1.2 Requisitos não-funcionais

Além das histórias de usuários, também deve-se considerar os requisitos não funcionais desta aplicação Web, sendo eles:

- Servidor Web com PHP 5.4 ou superior.
- Conexão ao banco de dados MySQL 5.7.
- Senha do usuário da aplicação criptografada no banco de dados.
- Responsividade para dispositivos móveis.

5.2 Arquitetura

As aplicações desenvolvidas em *framework* Yii 2.0 possuem como padrão arquitetural MVC (*Model, View, Controller*) que determina a separação das responsabilidades das classes em modelo para regra de negócio e persistência da base de dados, as visões com a parte do *frontend* e o controlador servindo como um elo entre eles.

5.3 Modelo Entidade Relacionamento

Foi realizada a revisão do esquema do banco de dados da aplicação *desktop*, dentre as principais mudanças estão a inclusão de novos atributos e uma separação melhor das entidades e relacionamentos. Sendo assim, para a versão Web da aplicação o esquema teve a inclusão de três novas entidades, Login, Peso e Parametro, totalizando sete tabelas, conforme a Figura 5.1. O dicionário de dados detalhado está documentado no Apêndice D.

Foi incluída uma nova entidade **Login** para armazenar os registros das contas de acesso da aplicação, tendo como atributos principais o nome de usuário, senha e indicadores dos níveis de permissão na aplicação. Os atributos da entidade Login poderiam estar incluídos na entidade Pessoa, no entanto, por questões de segurança e permissões se optou por criar uma entidade independente. Sendo assim, existe um autorrelacionamento entre as entidades Pessoa e Login, tendo uma cardinalidade entre as tabelas de 1:1 (um para um).

Na tabela **Animal** havia sido identificado um atributo *idade e peso*, que não estavam normalizados, com isso se optou por remover esses atributos e incluir duas novas colunas: *data_nascimento* e *data_obito*, e para o atributo *peso* se optou pela criação de uma nova tabela **Peso** que manterá a evolução das medidas de peso.

Foram revistos também os atributos nas demais tabelas, e em todos estão previstos atributos de controle do usuário, data e IP para registrar os dados da última atualização no registro.

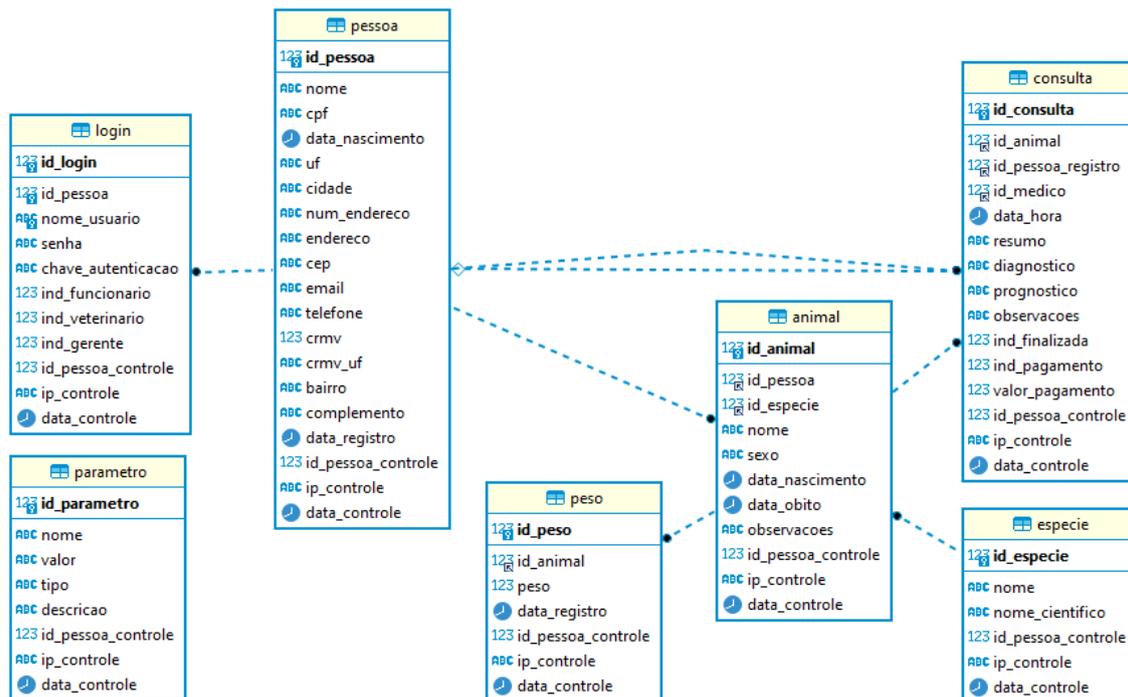
O esquema completo do banco de dados ficou definido nas seguintes setes tabelas:

- **Pessoa** : onde ficam mantidos os registros do Tutor, Médicos Veterinários, Funcionários e Gerente de TI, também possui os atributos relacionados ao endereço e específicos da especialização (e.g. CRMV do Médico Veterinário).
- **Login** : ficam mantidos os campos relativos a nome de usuário, senha e permissões do usuário.
- **Espécie** : registros do tipo de espécie do animal, tais como Gato e Cachorro.
- **Animal** : registros dos animais que são pacientes na clínica veterinária.
- **Peso** : registros das medições de peso do animal.
- **Consulta** : registros de cada consulta agendada, persistindo também os dados referentes ao atendimento prestado. Essa tabela é de fundamental importância no

sistema, pois além dos agendamentos é mantido todo o histórico do paciente (animal) e possuem dados relevantes aos relatórios.

- **Parâmetro** : registros com parâmetros e valores de configuração da aplicação Web administrados pelo gerente de TI.

Figura 5.1 – Diagrama do modelo entidade relacionamento



Fonte: próprio autor

5.4 Desenvolvimento da aplicação

Nessa seção serão abordados sobre os detalhes do desenvolvimento da aplicação, projeto, *frontend* e *backend*.

5.4.1 Ambiente de desenvolvimento

Para realizar o desenvolvimento de uma aplicação Web, é necessário o uso de algumas ferramentas e servidor Web, sendo necessário a instalação dos seguintes programas:

- Firefox Developer Edition: navegador usado para executar e desenvolver a aplicação Web. este navegador possui alguns recursos que auxiliam o desenvolvedor, tais como: CSS Grid, console, *debug*, simulador do modo responsivo, entre outros.
- DBeaver: ferramenta que possibilita acessar e gerenciar banco de dados.
- MySQL: é um sistema de gerenciador de banco de dados que é usado para armazenar o esquema e dados da aplicação.
- Git: é o sistema de versionamento utilizado para o controle e compartilhamento de codificação do sistema.
- Visual Studio Code: é uma IDE que possibilita a edição e depuração do código-fonte, tendo diversas extensões que facilitam o desenvolvimento como o autocomplete de métodos, integração com o git, linkar as *views* ao *controllers*, identifica variáveis sem uso, entre outros.
- XAMPP: é uma distribuição de programas que possibilita o ambiente necessário para o desenvolvimento e execução de aplicações Web. Consistindo do servidor Web Apache, do banco de dados Maria DB, interpretador PHP 8.

5.4.2 Projeto

Para esta aplicação Web se adotou o *framework* Yii 2.0 em linguagem PHP para o desenvolvimento, sendo a motivação dessa escolha os diversos recursos que esse *framework* possibilita como mapeamento objeto-relacional, através do *Active Record*, conexão com o banco, validação de formulários e uso dos pacotes do PHP pelo Composer. Também existe a motivação do autor desse trabalho possuir certa familiaridade e experiência com esse *framework*.

No site do Yii 2.0 existem duas possibilidades de *templates* de projeto, a *basic* na estrutura MVC padrão e o *advanced* que possui uma estrutura diferente com *common*, *frontend* e *backend* e dentro de cada uma delas se usa o padrão MVC. Para uso deste trabalho se optou pelo padrão *basic*.

Para começar uma nova aplicação em Yii 2, pode-se realizar o *download* de um projeto ou executar um comando no Composer para geração de um novo projeto. Inicialmente o autor começou pelo *download*, mas utilizou o Composer para atualização de dependências e inclusão de novas funcionalidades durante o desenvolvimento.

5.4.3 Frontend

Neste projeto, utilizou-se o framework Bootstrap 5 que vem incluído por padrão no projeto *basic* do Yii 2.0 nas suas dependências, não havendo a necessidade de grandes alterações no layout padrão ou do estilo CSS.

O Bootstrap possui uma abordagem chamada de *mobile first*, onde os elementos e suas classes são desenvolvidos primeiramente para adequar as telas de dispositivos móveis, com isso a adaptação dos elementos para uma tela maior (como a de computadores) não se torna tão trabalhosa.

Com isso, optou-se por manter o desenvolvimento utilizando os recursos oferecidos por este *framework* e que conseqüentemente tornaram o sistema de agendamento responsivo, mesmo não sendo inicialmente o objetivo principal deste trabalho de conclusão.

Além disso, estão incluídas no Yii 2 as bibliotecas jQuery e jQuery UI que são desenvolvidas em JavaScript e possibilitam uma melhor experiência nos elementos, citando como exemplo os campos de *autocomplete* que retornam registros conforme o termo digitado pelo usuário.

Para a funcionalidade de evolução de peso se utilizou a biblioteca Highcharts, em JavaScript, para geração de gráficos.

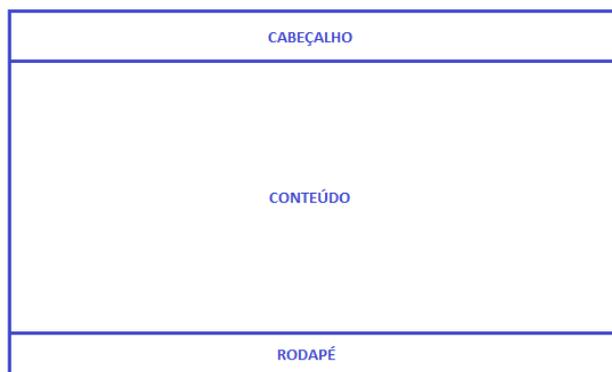
5.4.3.1 Layout

Para esta aplicação Web foi adotado o *framework* Bootstrap 5 que está incluído no projeto *basic* do *framework* Yii 2. O uso do Bootstrap facilita o desenvolvimento, pois existe uma série de padrões e componentes visuais definidos, bastando algumas modificações pontuais. Nesta aplicação, usou-se a seguinte estrutura de *layout*, conforme a Figura 5.2, onde no **CABEÇALHO** temos: o nome da aplicação; menu para as ações do sistema; no **CONTEÚDO** onde é renderizado as visões das telas e o **RODAPÉ** temos o nome do autor, data e que foi desenvolvido em Yii.

Tendo em vista que uma aplicação Web deve prover uma melhor experiência ao usuário, é importante que a navegação entre as telas seja padronizada e que facilite a usabilidade. Para isso, foi adotado o seguinte padrão visual nas telas, excetuando as telas de login e de página inicial, conforme a Figura 5.3.

Sendo assim foram adotados os seguintes elementos:

Figura 5.2 – Layout padrão utilizado nas telas



Fonte: próprio autor

- **Nome de tela:** título em destaque que indica qual funcionalidade ou serviço do sistema (e.g. Novo animal, Finalizar atendimento).
- **Caixa de ajuda:** elemento com fundo azul com instruções para realização do preenchimento e ações para o usuário.
- **Fieldset:** elemento para indicar o agrupamento de campos em comum no formulário. Está inserido em um elemento *container* do Bootstrap
- **Cores dos botões:** foi adotado o seguinte padrão de cores para os botões e ações:
 - Verde: envio de formulário e salvar informações.
 - Azul: demais ações, principalmente as que vão para outra funcionalidade (tela) ou exibir uma janela modal.
 - Vermelho: usado exclusivamente para cancelamento de consultas.
 - Cinza: usado exclusivamente para limpar dados de campos.
- **Campos obrigatórios:** campos de preenchimentos obrigatórios são sinalizados com * e em vermelho-escuro para destaque.

Figura 5.3 – Layout padrão utilizado nas telas

SACV [Início](#) [Cadastro](#) [Agendamento](#) [Sobre](#) [Logout \(Tutor 1\)](#)

[Página Inicial](#) / Nome tela

Nome tela

Caixa de ajuda para as ações do usuário.
Campos com asterisco (*) são de preenchimento obrigatório.

Nome fieldset

Tutor *
Digite o nome do tutor

Animal *

Enviar

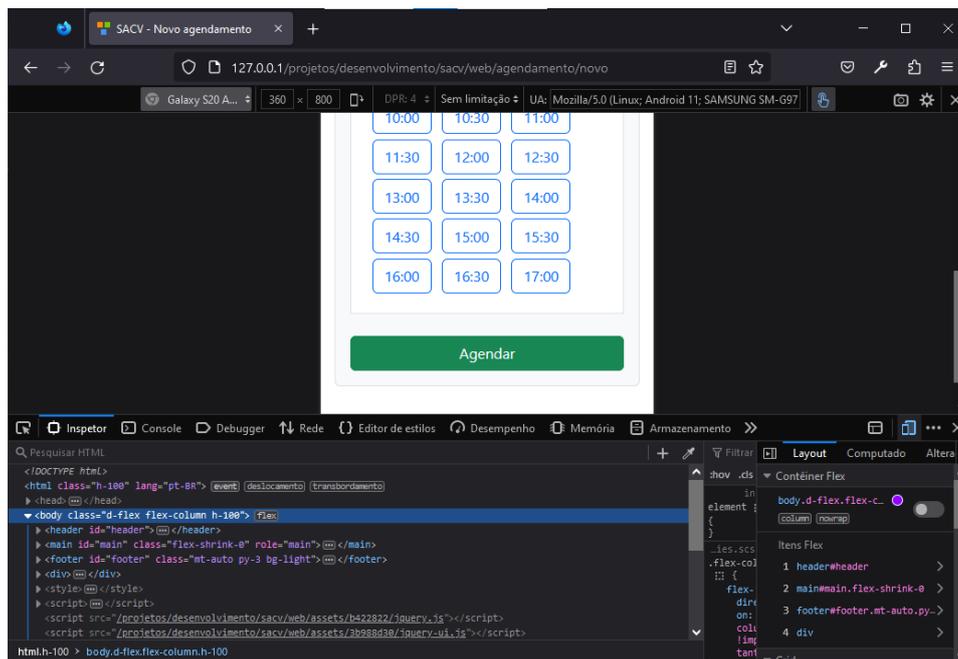
Fonte: próprio autor

5.4.3.2 Responsividade

Com o uso do Bootstrap 5 criar uma tela responsiva não é uma tarefa árdua, tendo em vista a abordagem *mobile first* deste *framework*. Para isso, temos classes CSS que possibilitam o *breakpoint* dos elementos, ou seja, a partir de um determinado tamanho de tela os elementos poderão ser quebrados ou realinhados. Utiliza-se o conceito de CSS Grids, que consiste em dividir a tela em colunas de tamanhos iguais e para cada elemento se indica na classe CSS quantas colunas serão ocupadas. Por padrão foi utilizada a divisão de 12 colunas. Ao desenvolver esta aplicação, também teve-se a preocupação em tornar as páginas da aplicação responsivas e utilizar dentro do possível os recursos oferecidos pelo Bootstrap.

Para verificar esse desenvolvimento considerando o responsivo é importante realizar testes de responsividade, que consiste em simular a tela de um dispositivo móvel para analisar a disposição e tamanho dos elementos da aplicação Web. Os navegadores atuais possuem ferramentas que possibilitam simular uma tela de dispositivo móvel, conforme a Figura 5.4.

Figura 5.4 – Tela com os horários em versão responsiva



Fonte: próprio autor

Abaixo, temos uma comparação dos campos do formulário da tela de cadastro de Animal, conforme a Figura 5.5, que se ajustaram adequadamente ao simular o modo responsivo do navegador, com a Figura 5.6. Para fins de testes, utilizou-se a simulação do dispositivo Galaxy S20 Android 11.

Figura 5.5 – Tela com campos de Animal para computador

Dados Animal

Nome do Animal *	Espécie *
<input type="text" value="Digite o nome do animal"/>	<input type="text"/>
Sexo *	
<input type="radio"/> Fêmea <input type="radio"/> Macho <input type="radio"/> Não Informado	
Data de Nascimento *	Peso
<input type="text" value="dd/mm/aaaa"/> <input type="button" value="📅"/>	<input type="text" value="Digite o peso em Kg (somente número)"/>
Observações	
<input style="width: 100%;" type="text" value="Informações relevantes sobre o animal (e.g. alergias, doenças crônicas, medicamento especial)"/>	
<input type="button" value="Salvar"/>	

Fonte: próprio autor

Na Figura 5.7 temos a disposição e o enquadramento dos elementos como campos, botões e *slots* de horários se adequando a tela de um dispositivo móvel.

Figura 5.6 – Tela com campos de Animal em versão responsiva



Dados do animal

Nome do Animal *

Espécie *

Sexo *

Fêmea Macho Não Informado

Data de Nascimento *

Peso

Observações

Salvar

Fonte: próprio autor

Figura 5.7 – Tela com os horários em versão responsiva



Médico Veterinário *

Data *

Verificar Horários

Horário da consulta *

08:30	09:00	09:30
10:00	10:30	11:00
11:30	12:00	12:30
13:00	13:30	14:00
14:30	15:00	15:30
16:00	16:30	17:00

Agendar

Fonte: próprio autor

Nas Figuras 5.8 e 5.9 temos a comparação entre a tela para monitores de conven-

cionais e a tela ajustada para larguras menores. A tabela se adapta em possibilitar uma navegação de cima para baixo para rolar a tela.

Figura 5.8 – Tela com os horários agendados - versão normal

SACV Início Cadastro Relatórios Agendamento Sobre Logout (Veterinário 1)

Minhas consultas

Novo agendamento

Exibindo 1-20 de 22 itens.

Tutor	Animal	Espécie	Data	Horário		
Tutor 2	Pet 3	Cachorro	20/03/2023	09:00	Cancelar	Finalizar
Tutor 2	Pet 4	Gato	20/03/2023	09:30	Cancelar	Finalizar
Tutor 3	Pet 5	Cachorro	20/03/2023	10:00	Cancelar	Finalizar
Tutor 3	Pet 6	Gato	20/03/2023	10:30	Cancelar	Finalizar
Tutor 1	Pet 1	Cachorro	21/03/2023	08:00	Cancelar	Finalizar
Tutor 1	Pet 2	Gato	21/03/2023	08:30	Cancelar	Finalizar
Tutor 2	Pet 3	Cachorro	21/03/2023	09:00	Cancelar	Finalizar

Fonte: próprio autor

Figura 5.9 – Tela com os horários agendados - versão responsiva

SACV

Minhas consultas

Novo agendamento

Exibindo 1-20 de 22 itens.

Tutor	Tutor 2
Animal	Pet 3
Espécie	Cachorro
Data	20/03/2023
Horário	09:00
Cancelar	
Finalizar	
Tutor	Tutor 2
Animal	Pet 4
Espécie	Gato

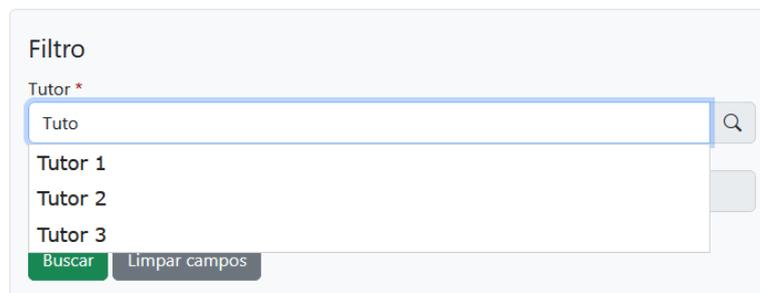
Fonte: próprio autor

5.4.3.3 Widgets

No Yii 2, os códigos de visão que podem ser reutilizados em outras telas podem ser implementados como um componente reutilizável chamado *Widgets*.

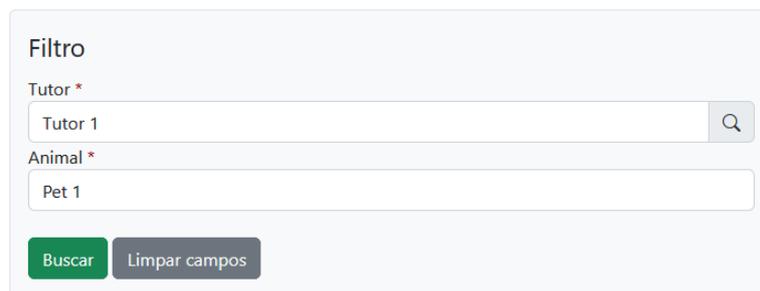
Na aplicação *desktop*, havia uma tela somente para buscar o tutor e uma nova tela para buscar o animal, que não faz muito sentido em termos de usabilidade em uma aplicação Web. Para isso se criou um componente chamado **FiltroTutorAnimalWidget**, conforme a Figura 5.10, que possibilita buscar o tutor através de um campo de *autocomplete*, que retorna alguns registros com base no termo digitado pelo usuário, e após essa seleção é realizada uma requisição AJAX que traz os animais que esse tutor possui na caixa de seleção *Animal*, conforme a Figura 5.11.

Figura 5.10 – Autocomplete de tutor no relatório



Fonte: próprio autor

Figura 5.11 – FiltroTutorAnimalWidget após a seleção de tutor



Fonte: próprio autor

5.4.4 Backend

Para o desenvolvimento *backend* foi utilizado o *framework* Yii 2.0 juntamente com o Composer para o gerenciamento de pacotes em PHP. A geração de PDFs foi utilizada a biblioteca mPDF através do pacote **kartik-pdf** do Composer.

5.4.4.1 Rotas

No Yii, temos as *controllers* que realizam a comunicação entre a visão e o modelo, sendo que temos métodos chamados *actions* que realizam uma determinada ação, geralmente renderizando uma tela do navegador.

Para executar uma determinada ação de um controlador, se utiliza o conceito de rotas na *url* do navegador. Nesta aplicação, adotou-se o padrão *idcontroller/idaction?param1=value1¶mn=valuen*, onde *id_controller* indica o nome do controlador e o *id_action* o nome da ação que será executada. Os parâmetros serão enviados através de uma requisição GET onde estão na própria *url* no formato *param_n* com o nome do parâmetro e o *value_n* o valor desse parâmetro, na quantidade de tantos que forem necessários.

Para a *url* <https://localhost/sacv/web/agendamento/finalizar?consulta=1> temos o controlador **agendamento**, a ação **finalizar** e um parâmetro enviado por GET que é consulta com valor 1.

5.4.4.2 Processamento de formulários

Para o acesso às telas e ações da aplicação Web serão realizadas através do endereço da *url* com a rota, sendo por padrão uma requisição GET.

No caso de telas com formulários, será realizada uma requisição POST. Porém para a mesma ação que gerou o formulário, pois em caso de erros de validação será renderizado novamente a mesma tela com os erros para usuário corrigir no preenchimento dos formulários. Em todos os casos o método irá executar a ação e renderizar a tela, no entanto em casos que for uma requisição POST, o formulário será processado e os dados inseridos serão salvos no banco de dados.

Ao utilizar o *Active Record* é possível utilizar a atribuição em massa do formulário, que ao invés de pegar campo a campo da variável global `$_POST` do PHP, basta apenas executar um comando `$modelo->attributes = $_["nomemodelo"]`, que irá obter

um vetor com o nome da classe do modelo. Para ficar melhor compreendido, temos o código da Figura 5.12 que exemplifica o caso de uma alteração de dados.

Figura 5.12 – Trecho de código PHP - actionAlterarDados()

```
public function actionAlterarDados()
{
    $pessoa = Pessoa::find()->with(['login'])->where(['id_pessoa' => Yii::$app->user->identity->id_pessoa])->one();

    if (Yii::$app->request->isPost) {
        $pessoa->attributes = $_POST['Pessoa'];

        if ($pessoa->save()) {
            Yii::$app->session->setFlash('success', 'Dados atualizados com sucesso.');
```

Fonte: próprio autor

Para acesso, edição e exclusão de registros existentes basta fazer uma busca com o método *find*, enquanto para criar um novo registro deve ser criada uma nova instância da classe com o comando *new*.

5.4.4.3 Parâmetros de sistema

Na aplicação Web foi adotada a persistência de alguns parâmetros em uma tabela do banco de dados para determinar algumas variáveis e configurações da aplicação que pudessem ser alteradas com mais facilidade pelo Gerente de TI depois que a aplicação estiver em produção. Os parâmetros usados nesta aplicação podem ser conferidos na Tabela 5.1

Tabela 5.1 – Parâmetros da aplicação

<i>Parâmetro</i>	<i>Tipo</i>	<i>Descrição</i>
PARAM_HORARIO_INICIO	string	Horário de início da consulta
PARAM_HORARIO_FIM	string	Horário de fim da consulta
PARAM_DURACAO_CONSULTA	inteiro	Tempo (minutos) de duração da consulta
PARAM_TOLERANCIA_CANC	inteiro	Tempo (minutos) de tolerância do cancelamento
PARAM_ENVIO_EMAIL	booleano	Ativa o envio de e-mail
PARAM_TESTE_USABILIDADE	booleano	Modo teste de usabilidade

Fonte: próprio autor

Para obter esses parâmetros, é realizada uma consulta e colocada numa variável como no código da Figura 5.13. Como esse método é chamado antes da execução de uma

action, isso iria causar um impacto de desempenho no banco de dados. Para evitar isso, utilizou-se uma cache de 20 minutos utilizando um recurso nativo de cache do *framework* ao realizar a consulta no banco de dados.

Figura 5.13 – Trecho de código PHP - obterParametro()

```
private function obterParametros()
{
    $tempoCache = 1200; /** 20 minutos */
    You, 4 days ago • Adicionado parametrização via banco ...
    Yii::$app->params['horarioInicio'] = Parametro::find()->cache($tempoCache)->where(['nome' => 'PARAM_HORARIO_INICIO'])->one()->valor;
    Yii::$app->params['horarioFin'] = Parametro::find()->cache($tempoCache)->where(['nome' => 'PARAM_HORARIO_FIM'])->one()->valor;
    Yii::$app->params['duracaoConsulta'] = Parametro::find()->cache($tempoCache)->where(['nome' => 'PARAM_DURACAO_CONSULTA'])->one()->valor;
    Yii::$app->params['toleranciaCanc'] = Parametro::find()->cache($tempoCache)->where(['nome' => 'PARAM_TOLERANCIA_CANC'])->one()->valor;
    Yii::$app->params['testeUsabilidade'] = Parametro::find()->cache($tempoCache)->where(['nome' => 'PARAM_TESTE_USABILIDADE'])->one()->valor;
}
```

Fonte: próprio autor

5.4.5 Implementação

Para iniciar a implementação após a criação do projeto *basic*, os primeiros passos consistem em configurar o arquivo `db.php` para possibilitar a conexão com o banco de dados e o arquivo `config.php` para ajuste das configurações da aplicação. Após isso, utilizou-se a ferramenta *Yii Gii* que gera código-fonte para facilitar o desenvolvimento desta aplicação.

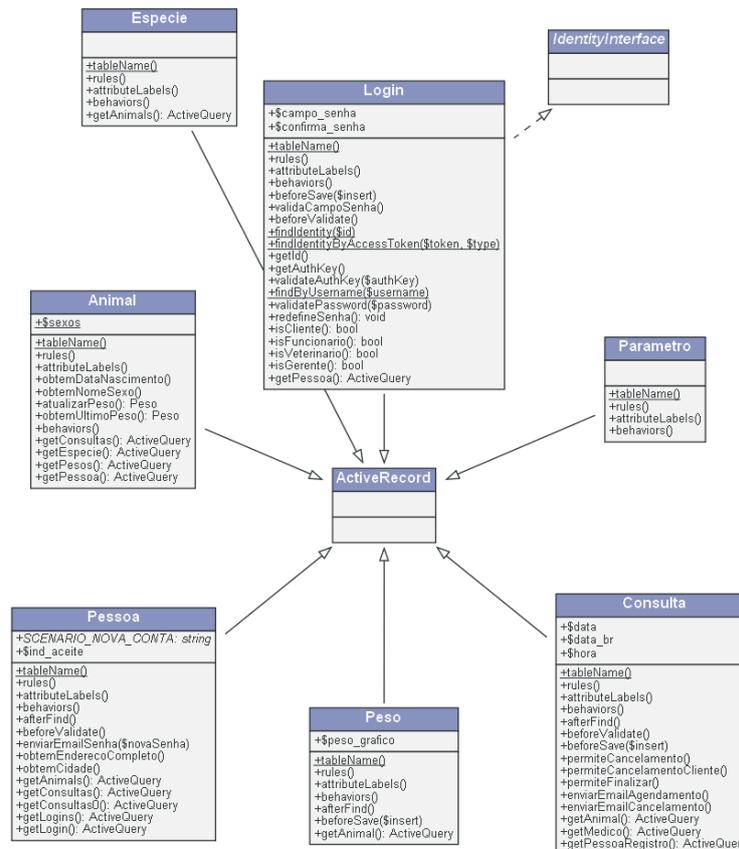
O *Yii* possui um recurso chamado de *Active Record* que possibilita o mapeamento objeto relacional das classes modelos com as tabelas do banco de dados, as classes possuem métodos para recuperar e salvar registros no banco sem a necessidade de escrever SQLs e possibilita realizar as validações com base das regras estabelecidas no modelo. Com o uso do *Yii Gii* foram criados os modelos que representam as tabelas do banco de dados através de engenharia reversa contendo as implementações de mapeamento, regras de validação de campos e as relações entre outras classes de *Active Record*. Podemos conferir as classes geradas no diagrama da Figura 5.14.

Note que no diagrama de classes, Figura 5.14, a classe **Login** implementa a interface **IdentityInterface** para possibilitar o *login* na aplicação *Yii*.

Foram usadas as classes **FiltroForm** e **LoginForm** para o processamento de formulários de filtro e da tela de login, respectivamente, que herdam a classe **Model** do *Yii*. Com a herança desta classe é adicionada a possibilidade de definição de regras para validações no envio de formulários.

No modelo **Relatorios** temos consultas SQL para construção dos relatórios da aplicação, no modelo **Horarios** para obter a disponibilidade de horários no banco de

Figura 5.14 – Diagrama de classes - Modelos Active Record



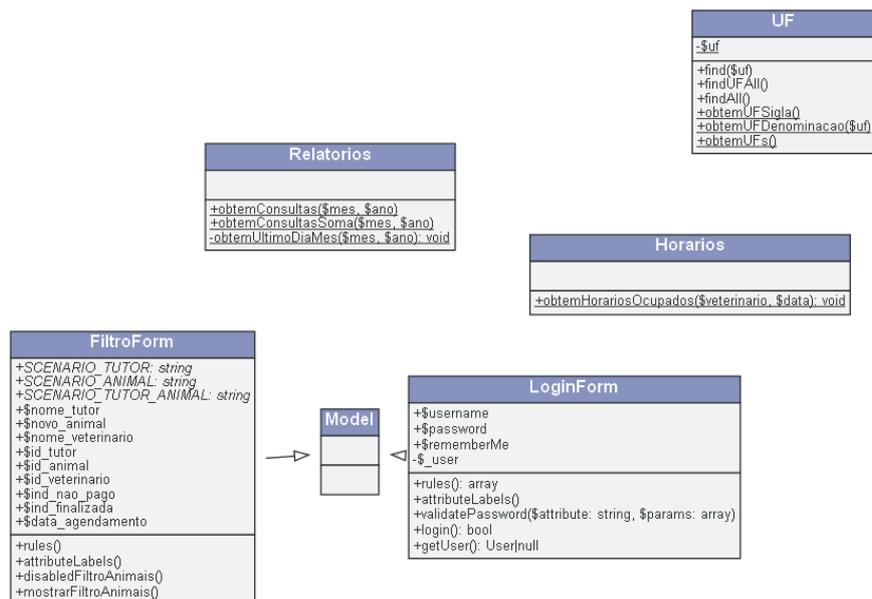
Fonte: próprio autor - gerado no phUML

dados e um modelo UF com as unidades federativas do Brasil para auxiliar nos campos de formulário. Temos o diagrama de classes desses modelos na Figura 5.15

Ainda foi usado o Yii Gii para criar os CRUDs para cada tabela do banco com o padrão de nomenclatura *CrudTabelaController*, gerando assim o código-fonte de métodos e visões. Além disso, foi criada uma **BaseController** para ser uma superclasse dos demais controladores, conforme podemos conferir no diagrama de classes da Figura 5.16.

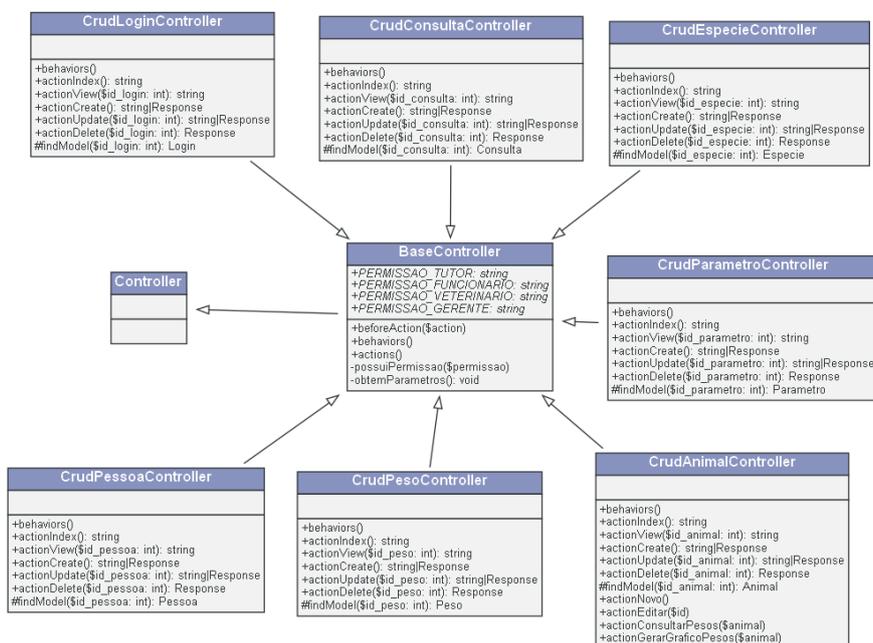
A classe **BaseController** foi criada com o propósito de centralizar alguns comportamentos na aplicação Web, tais como: o controle de acesso e permissões das demais *controllers* e *actions* com o uso da classe *AccessControl* e a implementação do método *possuiPermissao* que retorna o nível de permissão conforme o perfil de usuário logado; a obtenção de parâmetros de configuração através de uma verificação no método *beforeAction* que é executado antes do acionamento de uma ação do Yii; a definição de *layout* alternativos e ações de erro padrão do Yii. Foi definido como padrão de desenvolvimento que todas as demais *controllers* iriam de **BaseController**.

Figura 5.15 – Diagrama de classes - Modelos auxiliares



Fonte: próprio autor - gerado no phUML

Figura 5.16 – Diagrama de classes - Controladores CRUD

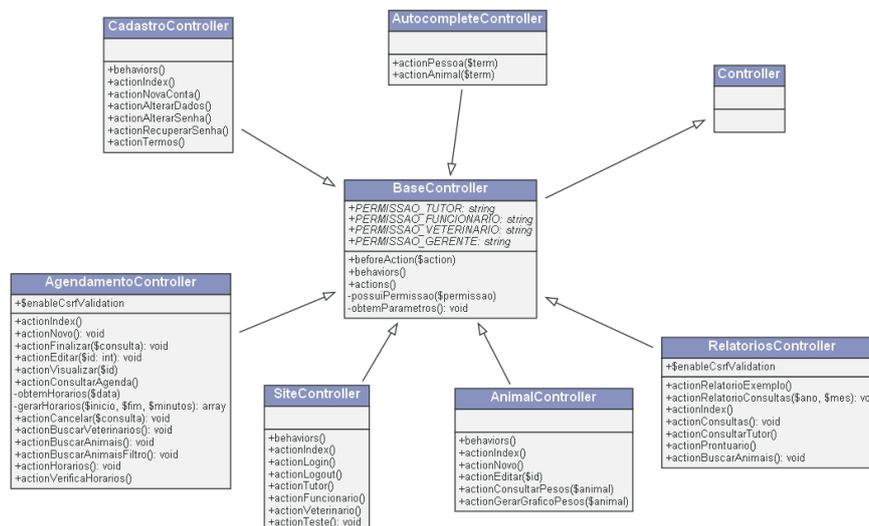


Fonte: próprio autor - gerado no phUML

Para o desenvolvimento das funcionalidades desta aplicação Web se optou por criar ou alterar outros controladores, porém utilizando quando necessário partes do código-fonte dos CRUDs gerados pelo Gii. As classes geradas pelo Gii foram mantidas para o uso do Gerente de TI.

No **SiteController** temos o controlador principal chamada pela rota index da aplicação, temos nela as ações da tela de login e da tela principal do usuário logado. No **AnimalController** temos as ações referente ao cadastro de animal e medições de peso, enquanto no **CadastroController** temos as ações referentes ao cadastro de pessoas, alteração de dados, alteração de senhas e termos de uso. A classe **AgendamentoController** possui as ações para telas de consultas e agendamento. A classe **RelatoriosController** contém as ações referente as telas de relatórios, histórico do paciente e geração de PDF. Foi utilizado um **AutocompleteController** como controlador auxiliar com as ações que realizam o *autocomplete* dos campos. Todas as classes citadas herdam de **BaseController** conforme o diagrama da Figura 5.17.

Figura 5.17 – Diagrama de classes - Controladores principais



Fonte: próprio autor - gerado no phUML

5.4.5.1 Componentes

O Yii possui um componente que é executado quando determinados eventos ocorrerem em uma classe, chamado de *behavior*. Esses eventos podem ser um salvamento de um dado no banco, uma busca de registros no banco, entre outros.

Nos modelos que utilizam o *Active Record*, foi utilizado um *behavior* chamado *CamposControllerBehavior* que serve para atribuição automática dos campos de controle como IP, Data e Usuário de última atualização, antes da realização de uma inserção ou atualização de um registro no banco de dados.

5.4.6 Ambiente de produção

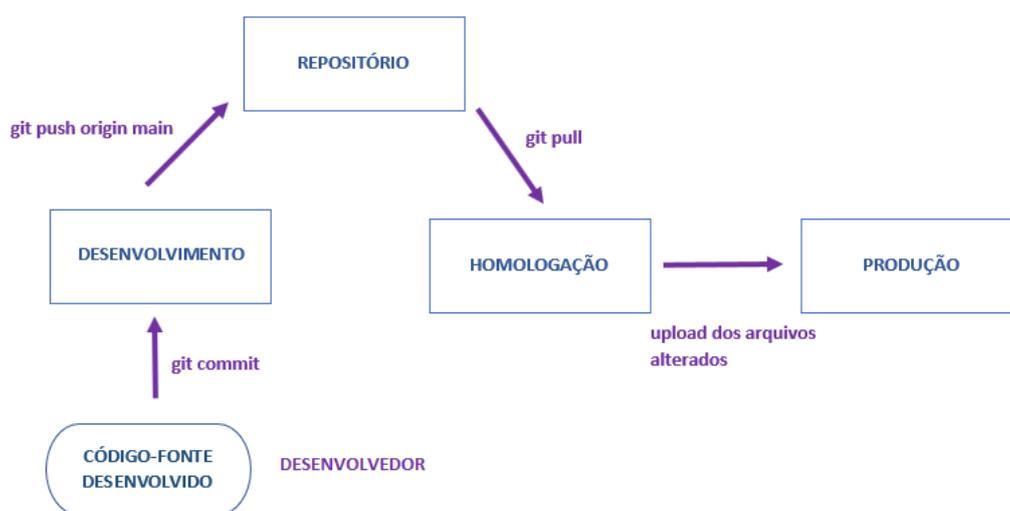
Para a realização do experimento de usabilidade foi necessário fazer a hospedagem do código-fonte para os testes com o usuário. Nesse caso, a hospedagem teve algumas configurações diferentes do ambiente de desenvolvimento do computador local.

- MariaDB 10.5: banco de dados relacional.
- PHP 8.0.7: versão PHP.
- phpMyAdmin: ferramenta para acesso ao banco de dados.

5.4.7 Protocolo de publicação

Para melhor controle e organização do desenvolvimento se adotou seguir o seguinte fluxo de publicação, conforme a Figura 5.18.

Figura 5.18 – Fluxo de publicação



Fonte: próprio autor

Temos assim a divisão de três ambientes de configurações distintas: **desenvolvimento** onde é realizada as alterações de nível de codificação, **homologação** onde são realizados os testes mais robustos e de validação das funcionalidades e finalmente **produção** que é o servidor localizado em um serviço de hospedagem na internet para uso real dos usuários.

A implantação desse fluxo se deu através da criação de duas pastas *desenvolvimento* e *homologacao* no servidor Web local, onde cada pasta tinha um clone do projeto

Tabela 5.2 – Configuração ambientes de trabalho

<i>Ambiente</i>	<i>Composer</i>	<i>Debug</i>	<i>Envio e-mail</i>	<i>Teste usabilidade</i>
Desenvolvimento	Completo	Sim	Não	Não
Homologação	Necessário	Não	Sim	Não
Produção	Necessário	Não	Não	Sim

Fonte: próprio autor

do sacv. Para adotar esse modelo é imprescindível o uso de um repositório remoto para manter o controle de versionamento, nesse caso foi adotado o GitHub.

Entre os ambientes podemos citar as seguintes diferenças conforme a Tabela 5.2.

Toda a alteração de código era realizada em *desenvolvimento/sacv*, sendo os *git commit* e *git push origin main* realizados nessa pasta. Em homologação era somente dado o comando *git pull* para atualizar a pasta local do repositório e finalmente em produção os arquivos que foram alterados eram publicados via gerenciador de arquivos.

6 DEMONSTRAÇÃO

Neste capítulo são demonstradas as funcionalidades implementadas da aplicação Web, sendo apresentados os passos necessários para realização das ações, bem como as telas apresentadas ao usuário da aplicação. As seções deste capítulo são abordadas no estilo de manual do usuário.

6.1 Tela de entrada

Para realizar a entrada no sistema deve ser preenchido o nome de usuário e a senha e clicar no botão **Entrar**, conforme a Figura 6.1.

Figura 6.1 – Tela inicial de login

SACV [Sobre](#) [Login](#)

[Página Inicial](#) / Login

SACV

Preencha os campos abaixo para acessar o sistema de agendamento:

Usuário

 ⓘ
Usuário não pode ficar em branco.

Senha

Lembre-me

Entrar

Criar conta

[Recuperar senha](#)

Fonte: próprio autor

Temos também nessa tela as ações de Criar conta e Recuperar Senha que serão vistas na sequência.

6.2 Criar conta

Na tela inicial temos o botão **Criar conta** que levará a tela da Figura 6.2. O usuário deverá preencher os dados solicitados e clicar em salvar, conforme a Figura 6.3.

O formulário realiza a busca dos campos de endereço conforme o CEP através do

uso de uma API. Caso o usuário não saiba o CEP tem a opção através do botão **Procurar CEP**.

Figura 6.2 – Tela de cadastro de tutor

A imagem mostra a interface de usuário para a criação de uma conta. No topo, há uma barra de navegação com o texto 'SACV' e links para 'Sobre' e 'Login'. Abaixo, há uma barra de progresso com 'Página Inicial' e 'Criar conta'. O título principal é 'Criar conta'. Um aviso em uma caixa azul indica: 'Preencha os dados solicitados abaixo. Campos com asterisco (*) são de preenchimento obrigatórios.' A seção 'Dados pessoais' contém campos para 'Nome do Tutor *' (com o placeholder 'Digite seu nome completo'), 'Data de Nascimento *' (formato dd/mm/aaaa) e 'CPF'. A seção 'Dados de endereço' contém um campo para 'CEP *' com um botão azul 'Procurar CEP' ao lado, e campos parciais para 'Endereço' e 'Número'.

Fonte: próprio autor

Figura 6.3 – Tela de cadastro de tutor - botão Salvar

A imagem mostra a seção 'Dados de login' da interface de cadastro. No topo, há uma barra de navegação com 'SACV' e links para 'Sobre' e 'Login'. Abaixo, há um campo para 'Telefone'. A seção 'Dados de login' contém campos para 'Nome de usuário *' (placeholder: 'Digite um nome de usuário'), 'Senha *' (placeholder: 'Digite uma senha') e 'Confirma Senha *' (placeholder: 'Digite sua senha novamente'). Abaixo dos campos, há uma caixa de seleção desmarcada com o texto: 'Ao realizar meu cadastro estou de acordo com os [termos de uso e política de privacidade](#)'. Um botão verde 'Salvar' está posicionado abaixo da caixa de seleção. No rodapé, há o texto '© SACV - Adriano Gebert Gomes 2023' e 'Desenvolvido com Yii Framework'.

Fonte: próprio autor

6.3 Recuperar senha

Na tela inicial temos o link **Recuperar senha** que levará a tela da Figura 6.4. O usuário deverá preencher os dados solicitados e clicar em enviar.

Figura 6.4 – Tela recuperar senha

Fonte: próprio autor

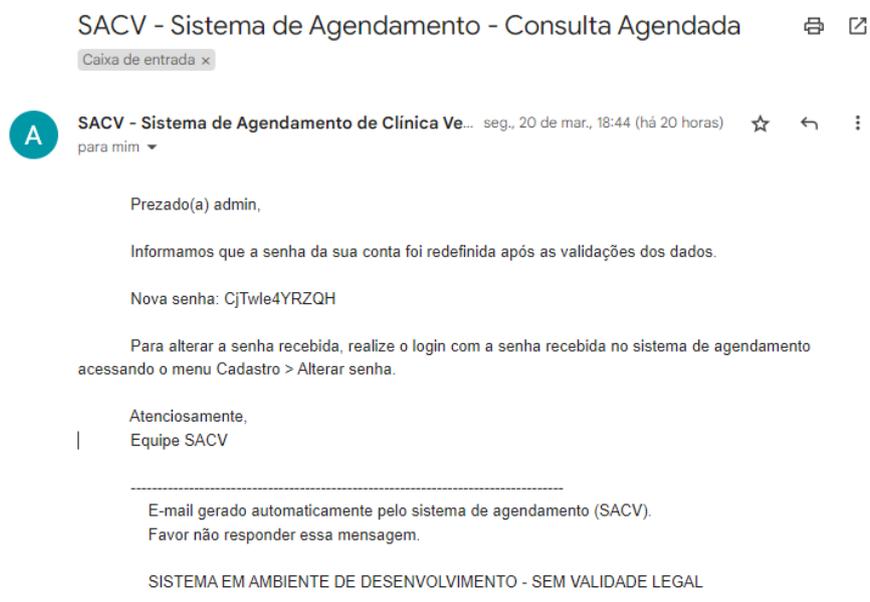
Havendo o preenchimento dos dados, independente de estar correto ou não, será mostrada uma mensagem informando que o e-mail de redefinição de senha será enviado para o e-mail cadastrado, conforme a Figura 6.5.

Figura 6.5 – Tela recuperar senha - mensagem de envio

Fonte: próprio autor

Após isso, será enviado um e-mail com uma nova senha provisória gerada pela aplicação Web, conforme a Figura 6.6.

Figura 6.6 – E-mail enviado pelo sistema com a nova senha



Fonte: próprio autor

6.4 Página inicial

Após o usuário entrar na aplicação Web será mostrada uma página inicial, denominada **Início** no menu, que será personalizada conforme o perfil do usuário, se é tutor, funcionário ou médico veterinário. O perfil de Gerente de TI terá os mesmos acessos de um médico veterinário.

6.4.1 Tutor

Temos na Figura 6.7 a tela apresentada ao tutor quando ele está logado no sistema, onde será apresentada uma mensagem de boas-vindas, instruções, consultas agendadas e seus animais cadastrados.

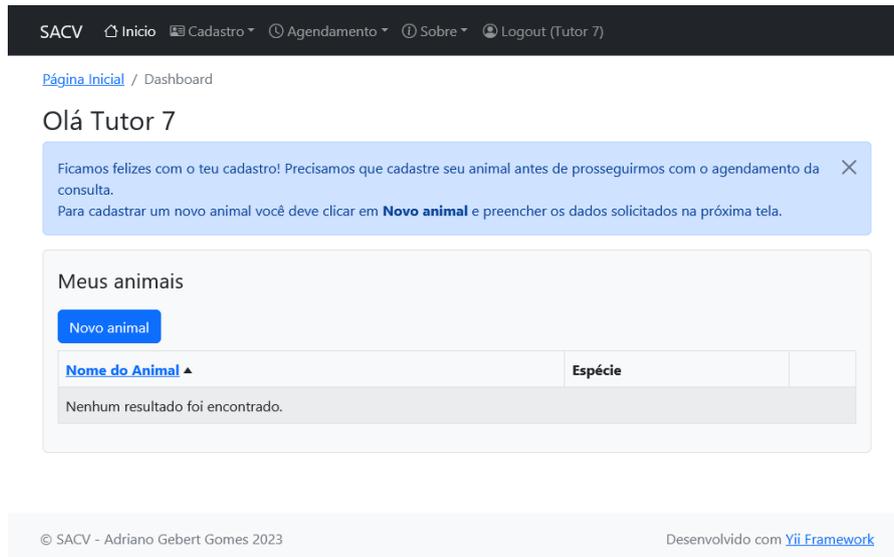
No caso do tutor não ter animal cadastrado, o bloco referente a consultas agendadas não será mostrado e haverá nas instruções que para realizar um agendamento deverá inicialmente cadastrar um novo animal, conforme a Figura 6.8.

Figura 6.7 – Tela da página inicial - Tutor

The screenshot displays the SACV web application interface for a Tutor user. At the top, there is a dark navigation bar with the following items: SACV, Início (with a home icon), Cadastro (with a list icon), Agendamento (with a clock icon), Sobre (with an information icon), and Logout (Tutor 7) (with a user icon). Below the navigation bar, the breadcrumb path is 'Página Inicial / Dashboard'. A green success message states 'Animal foi cadastro com sucesso.' with a close button. The main content area is titled 'Olá Tutor 7' and contains a blue informational message: 'Nesta tela é possível visualizar as consultas marcadas e realizar os cancelamentos de consultas que ainda não foram finalizadas. Caso precise realizar um novo agendamento de consulta, favor clicar em **Novo agendamento**. **Atenção:** Não serão permitidos cancelamentos de consultas com menos de 60 minutos de antecedência.' Below this, there is a section titled 'Meus agendamentos' with a blue 'Novo agendamento' button. A table with columns 'Animal', 'Espécie', 'Médico Veterinário', 'Data', and 'Horário' is shown, but it is empty with the message 'Nenhum resultado foi encontrado.'

Fonte: próprio autor

Figura 6.8 – Tela da página inicial sem bloco agendamento - Tutor

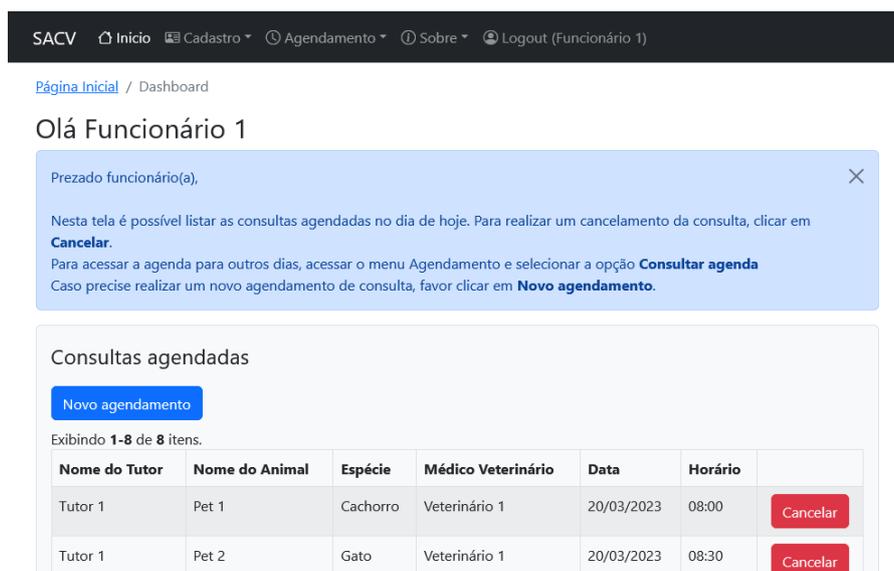


Fonte: próprio autor

6.4.2 Funcionário

Temos na Figura 6.9 a tela apresentada ao funcionário quando ele está logado no sistema, onde será apresentada uma mensagem de boas-vindas, instruções e as consultas agendadas do dia atual.

Figura 6.9 – Tela da página inicial - Funcionário



Fonte: próprio autor

6.4.3 Médico Veterinário

Temos na Figura 6.10 a tela apresentada ao médico veterinário quando ele está logado no sistema, onde será apresentada uma mensagem de boas-vindas, instruções e as consultas agendadas para ele.

Figura 6.10 – Tela da página inicial - Veterinário

The screenshot shows the initial page for a veterinarian in the SACV system. The page features a dark navigation bar at the top with the following items: SACV, Início, Cadastro, Relatórios, Agendamento, Sobre, and Logout (Veterinário 1). Below the navigation bar, there is a breadcrumb trail: Página Inicial / Dashboard. The main heading is "Olá Veterinário 1". A blue message box contains the following text: "Prezado(a) Médico(a) Veterinário(a),", "Nesta tela é possível listar as consultas agendadas para você. Para realizar um cancelamento da consulta, clicar em **Cancelar**.", "Para edição do prontuário e conclusão do atendimento, favor clicar em **Finalizar**.", and "Caso precise realizar um novo agendamento de consulta, favor clicar em **Novo agendamento**". Below the message box, there is a section titled "Minhas consultas" with a "Novo agendamento" button. Underneath, it says "Exibindo 1-20 de 24 itens." and a table with the following data:

Tutor	Animal	Espécie	Data	Horário		
Tutor 1	Pet 1	Cachorro	20/03/2023	08:00	Cancelar	Finalizar
Tutor 1	Pet 2	Gato	20/03/2023	08:30	Cancelar	Finalizar

Fonte: próprio autor

6.5 Novo animal

Para realizar uma inclusão de um novo animal, poderá ser realizado através do botão **Novo animal** na página principal (no caso do tutor) ou através do caminho do menu *Cadastro->Novo Animal*.

Será mostrado o formulário para preenchimento dos dados, conforme a Figura 6.11, onde deverá ser preenchido o nome do tutor a ser vinculado ao cadastro de animal;

Caso a tela for acessada pelo tutor, não haverá a possibilidade de escolher o tutor, conforme a Figura 6.12.

Figura 6.11 – Tela novo animal - Veterinário

Fonte: próprio autor

Figura 6.12 – Tela novo animal - Tutor

Fonte: próprio autor

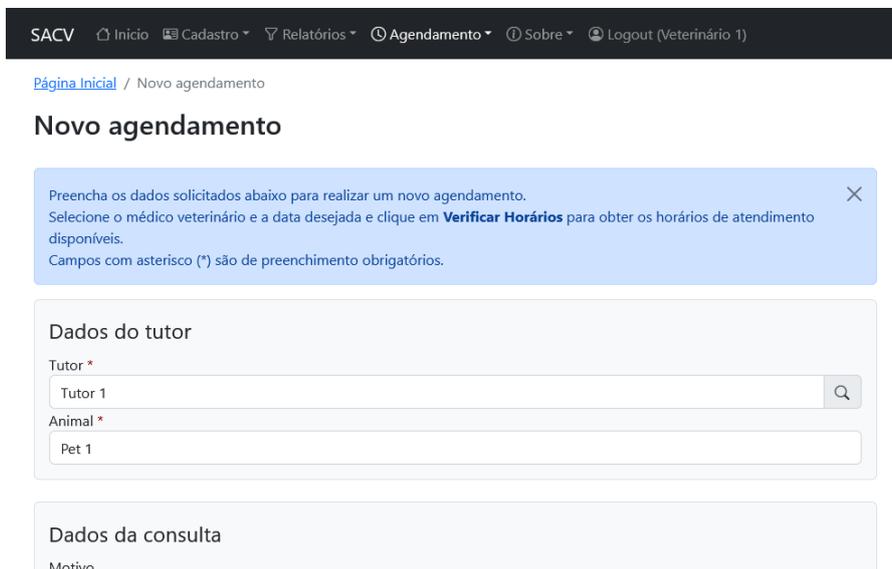
6.6 Novo agendamento

Um novo agendamento de consulta poderá ser realizado através do botão **Novo agendamento** na página principal ou através do caminho do menu *Agendamento->Novo Agendamento*.

Será apresentada a tela com um formulário para ser preenchido, conforme a Figura

6.13. No caso do acesso ser feito por um tutor, só mostrará para escolha os animais vinculados ao cadastro do tutor.

Figura 6.13 – Tela novo agendamento



SACV [Início](#) [Cadastro](#) [Relatórios](#) [Agendamento](#) [Sobre](#) [Logout \(Veterinário 1\)](#)

[Página Inicial](#) / Novo agendamento

Novo agendamento

Preencha os dados solicitados abaixo para realizar um novo agendamento.
Selecione o médico veterinário e a data desejada e clique em **Verificar Horários** para obter os horários de atendimento disponíveis.
Campos com asterisco (*) são de preenchimento obrigatórios.

Dados do tutor

Tutor *
Tutor 1

Animal *
Pet 1

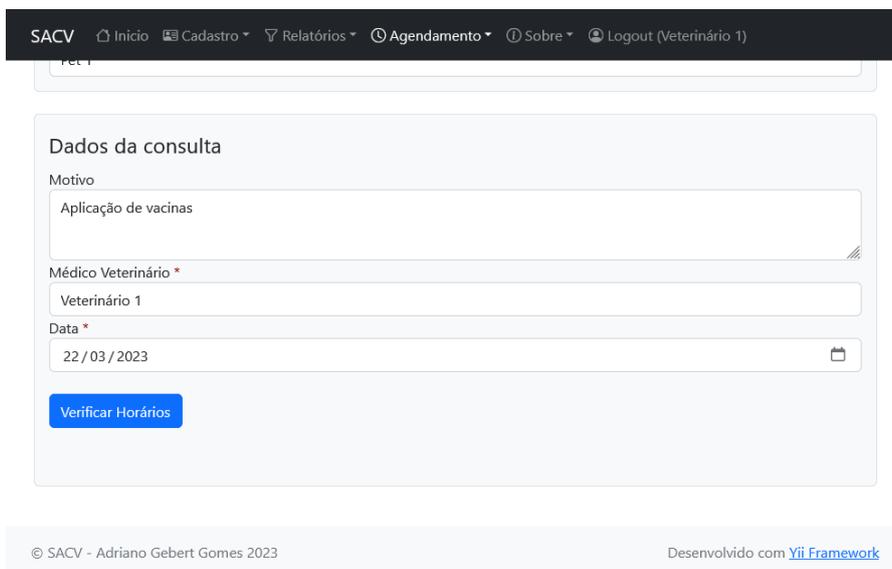
Dados da consulta

Motivo

Fonte: próprio autor

Na sequência, o usuário deverá escrever o motivo da consulta, escolher um médico veterinário e uma data, conforme a Figura 6.14. Após o preenchimento, o usuário deverá clicar no botão **Verificar horários** para mostrar as consultas disponíveis para agendamento.

Figura 6.14 – Tela novo agendamento com dados da consulta preenchidos



SACV [Início](#) [Cadastro](#) [Relatórios](#) [Agendamento](#) [Sobre](#) [Logout \(Veterinário 1\)](#)

Pet 1

Dados da consulta

Motivo
Aplicação de vacinas

Médico Veterinário *
Veterinário 1

Data *
22 / 03 / 2023

Verificar Horários

© SACV - Adriano Gebert Gomes 2023 [Desenvolvido com Yii Framework](#)

Fonte: próprio autor

Após o clique da ação, será mostrada a tela 6.15 onde o usuário deverá clicar em algum dos horários disponíveis e finalmente clicar no botão **Agendar**. Importante ressaltar que horários já passados não serão exibidos para escolha e horários já agendados estão desabilitados para seleção.

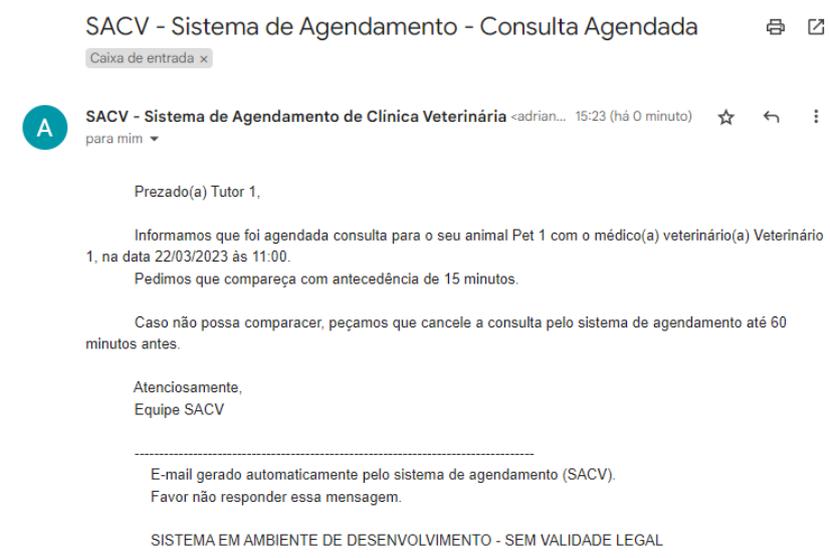
Figura 6.15 – Tela novo agendamento com horários para agendamento

A imagem mostra a interface de usuário do sistema SACV para agendamento de consultas. No topo, há uma barra de navegação com o menu "Agendamento" selecionado. Abaixo, há um campo "Data" com o valor "22/03/2023" e um botão "Verificar Horários". O formulário principal, rotulado "Horário da consulta", contém uma grade de botões de horários. O horário "11:00" está selecionado e destacado em azul escuro. Um botão "Agendar" verde está localizado na base do formulário.

Fonte: próprio autor

Após realizado o agendamento, será enviado um e-mail para o tutor informando a confirmação do agendamento, conforme a Figura 6.16.

Figura 6.16 – E-mail enviado confirmando o agendamento



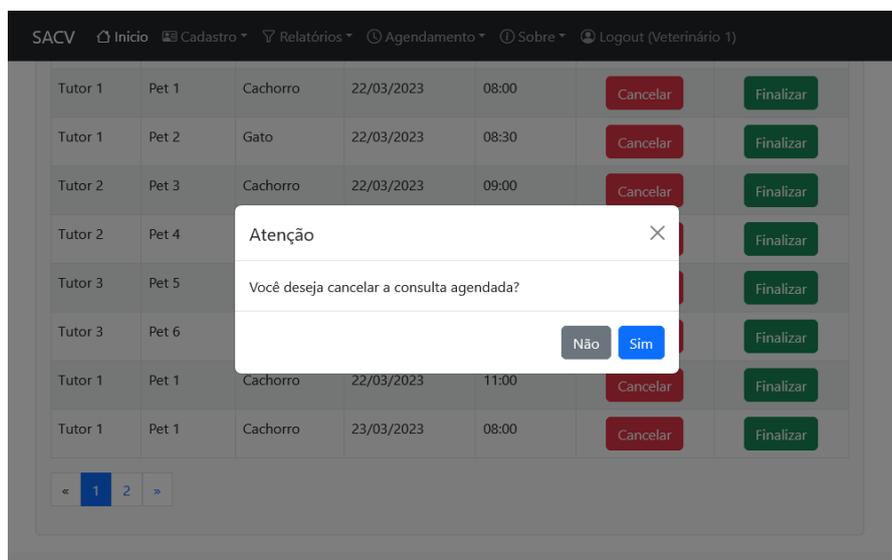
Fonte: próprio autor

6.7 Cancelar consulta

Para realizar um cancelamento de consulta agendada, existe mais de um caminho a ser realizado, conforme o perfil do usuário.

Para todos os usuários o cancelamento pode ser realizado clicando no botão vermelho **Cancelar consulta** na tela inicial que mostra as consultas. No caso dos usuários que não são tutores, o cancelamento também pode ser realizado pelo caminho *Agendamento > Consultar agenda* e clicando no botão vermelho **Cancelar** de alguma das consultas agendadas. Em todos os casos, ao clicar no botão será mostrada uma mensagem de confirmação, conforme a Figura 6.17.

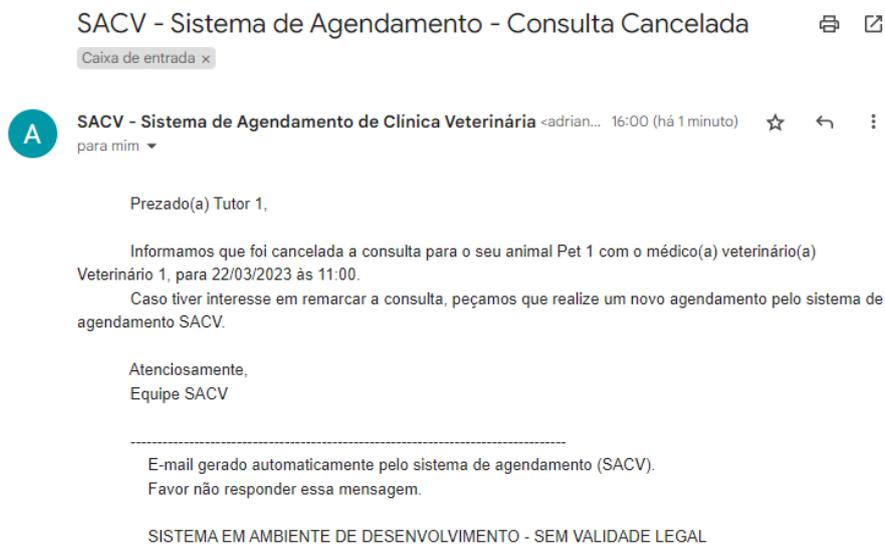
Figura 6.17 – Janela modal confirmando cancelamento



Fonte: próprio autor

Após a confirmação, será enviado um e-mail para o tutor confirmando o cancelamento da consulta, conforme a Figura 6.18.

Figura 6.18 – E-mail enviado confirmando o cancelamento



Fonte: próprio autor

6.8 Consultar agenda

Para consultar os atendimentos agendados, o usuário deverá acessar o caminho do menu *Agendamento->Consultar agenda*. O usuário deverá preencher os campos para filtrar os resultados que serão apresentados na tela, conforme a Figura 6.19.

Figura 6.19 – Tela Consultar agenda - Filtros

The image shows the 'Filtros' (Filters) section of the 'Consultar agenda' screen. The interface includes a navigation bar at the top with 'SACV' and menu items: 'Início', 'Cadastro', 'Relatórios', 'Agendamento', 'Sobre', and 'Logout (Veterinário 1)'. The filter form contains the following elements:

- Tutor:** A search input field with the placeholder 'Digite o nome do tutor' and a search icon.
- Animal:** A search input field.
- Veterinário:** A search input field.
- Data:** A date input field with the placeholder 'dd / mm / aaaa' and a calendar icon.
- Mostrar pagamento pendente:** A checkbox.
- Mostrar consultas finalizadas:** A checkbox.
- Buttons:** A green 'Buscar' button and a 'Limpar campos' button.

At the bottom of the page, there is a footer with the text: '© SACV - Adriano Gebert Gomes 2023' and 'Desenvolvido com Yii Framework'.

Fonte: próprio autor

Após o preenchimento dos dados o usuário deverá clicar no botão **Buscar**, e serão

apresentadas as consultas agendadas, conforme a Figura 6.20.

Figura 6.20 – Tela Consultar agenda - Resultados

Consultas agendadas

[Novo agendamento](#)

Exibindo 1-20 de 24 itens.

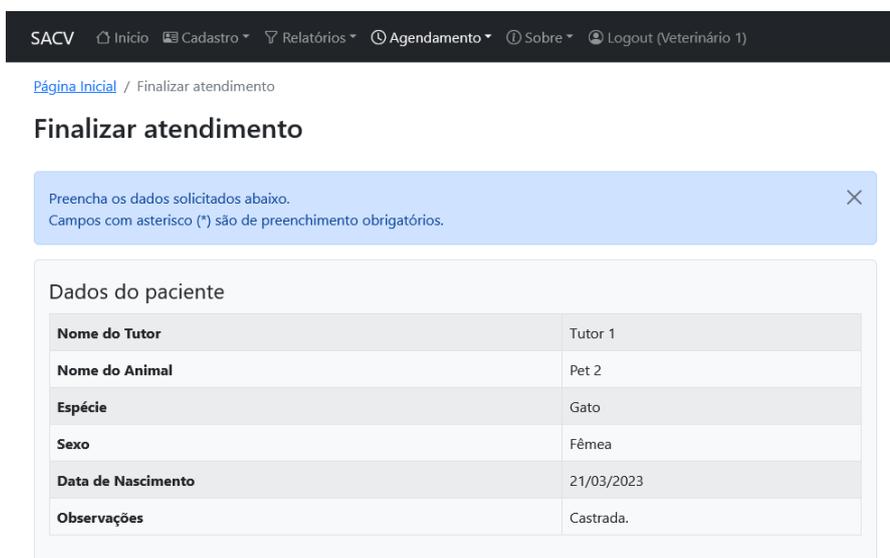
Tutor	Animal	Espécie	Médico Veterinário	Data	Horário		
Tutor 1	Pet 1	Cachorro	Veterinário 1	20/03/2023	08:00	Cancelar	Finalizar
Tutor 1	Pet 2	Gato	Veterinário 1	20/03/2023	08:30	Cancelar	Finalizar
Tutor 2	Pet 3	Cachorro	Veterinário 1	20/03/2023	09:00	Cancelar	Finalizar
Tutor 2	Pet 4	Gato	Veterinário 1	20/03/2023	09:30	Cancelar	Finalizar
Tutor 3	Pet 5	Cachorro	Veterinário 1	20/03/2023	10:00	Cancelar	Finalizar
Tutor 3	Pet 6	Gato	Veterinário 1	20/03/2023	10:30	Cancelar	Finalizar
Tutor 1	Pet 1	Cachorro	Veterinário 1	21/03/2023	08:00	Cancelar	Finalizar

Fonte: próprio autor

6.9 Finalizar atendimento

Para realizar a finalização de uma consulta, o médico veterinário deverá clicar no botão **Finalizar** na página inicial em "Consultas Agendadas" ou acessar o caminho do menu *Agendamento->Consultar agenda*. Nessa tela, serão apresentadas as informações sobre o paciente e dados iniciais do agendamento como data, hora e motivo, conforme a Figura 6.21.

Figura 6.21 – Tela finalizar atendimento - dados paciente



Dados do paciente	
Nome do Tutor	Tutor 1
Nome do Animal	Pet 2
Espécie	Gato
Sexo	Fêmea
Data de Nascimento	21/03/2023
Observações	Castrada.

Fonte: próprio autor

Conforme as Figuras 6.22 e 6.23, temos os campos referentes aos dados complementares e dados financeiros, como valor do atendimento e procedimentos e se o valor já foi pago. Após isso, para concluir o atendimento será necessário clicar no botão **Finalizar atendimento**.

Figura 6.22 – Tela finalizar atendimento - dados consulta

The screenshot shows the 'Dados da consulta' form in the SACV system. The form is located within a dark navigation bar at the top containing 'SACV', 'Início', 'Cadastro', 'Relatórios', 'Agendamento', 'Sobre', and 'Logout (Veterinário 1)'. The form itself has a light gray background and contains the following fields:

- Data:** A text input field containing '21/03/2023'.
- Horário:** A text input field containing '08:30'.
- Motivo:** A text area containing 'Aplicação de vacinas'.
- Diagnóstico:** An empty text area.
- Prognóstico:** An empty text area.
- Observações:** An empty text area.

Fonte: próprio autor

Figura 6.23 – Tela finalizar atendimento - dados financeiro

The screenshot shows the 'Dados financeiros' form in the SACV system. The form is located within a dark navigation bar at the top containing 'SACV', 'Início', 'Cadastro', 'Relatórios', 'Agendamento', 'Sobre', and 'Logout (Veterinário 1)'. The form has a light gray background and contains the following elements:

- Prognóstico:** An empty text area.
- Observações:** An empty text area.
- Dados financeiros:** A section containing:
 - Valor da Consulta + Procedimentos:** A text input field with a currency symbol 'R\$' and the value '0,00'.
 - Pagamento realizado:** A checkbox that is currently unchecked.
- Finalizar atendimento:** A green button with white text.

At the bottom of the page, there is a footer with the text '© SACV - Adriano Gebert Gomes 2023' on the left and 'Desenvolvido com [Yii Framework](#)' on the right.

Fonte: próprio autor

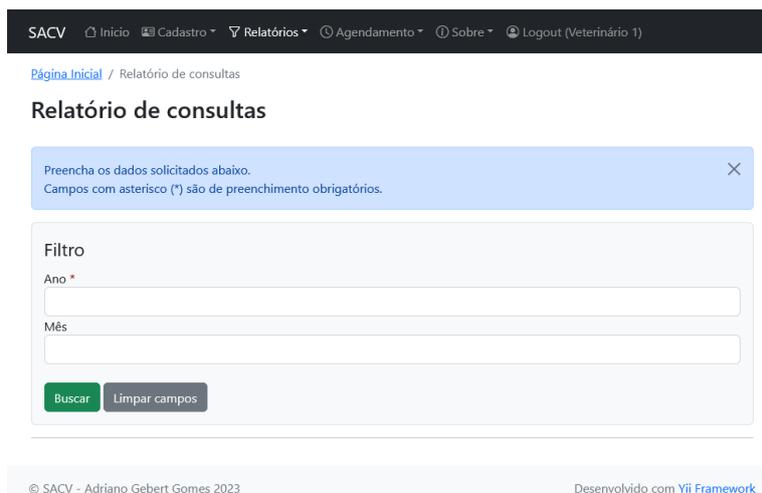
6.10 Relatórios

Nesta seção serão apresentadas as telas referentes ao menu de Relatórios, que incluem os relatórios das consultas, dados do tutor e dados do animal. Também serão vistas as funcionalidades de histórico do peso, evolução do peso, visualizar dados de um atendimento e editar dados de um atendimento.

6.10.1 Consultas

Para consultar o relatório de consultas dos atendimentos realizados pelo veterinário, o usuário deverá acessar o caminho do menu *Relatórios->Consultas* e preencher os dados de ano e mês e clicar em buscar, conforme a Figura 6.24.

Figura 6.24 – Relatório de consultas - Filtro



SACV Início Cadastro Relatórios Agendamento Sobre Logout (Veterinário 1)

[Página Inicial](#) / Relatório de consultas

Relatório de consultas

Preencha os dados solicitados abaixo.
Campos com asterisco (*) são de preenchimento obrigatórios.

Filtro

Ano *

Mês

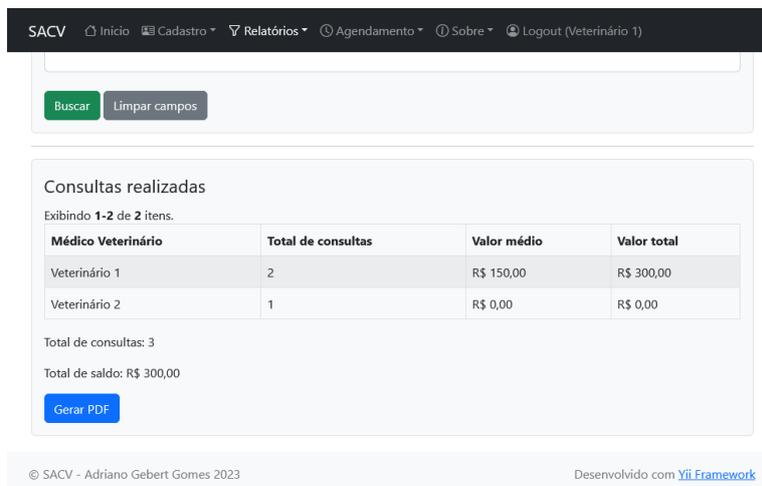
Buscar Limpar campos

© SACV - Adriano Gebert Gomes 2023 Desenvolvido com [Yii Framework](#)

Fonte: próprio autor

Após isso, será apresentada a tela com os resultados obtidos, onde é possível ver a relação dos veterinários, consultas totais, valor médio e valor total, conforme a Figura 6.25.

Figura 6.25 – Relatório de consultas - Resultados



SACV Início Cadastro Relatórios Agendamento Sobre Logout (Veterinário 1)

Buscar Limpar campos

Consultas realizadas

Exibindo 1-2 de 2 itens.

Médico Veterinário	Total de consultas	Valor médio	Valor total
Veterinário 1	2	R\$ 150,00	R\$ 300,00
Veterinário 2	1	R\$ 0,00	R\$ 0,00

Total de consultas: 3
Total de saldo: R\$ 300,00

Gerar PDF

© SACV - Adriano Gebert Gomes 2023 Desenvolvido com [Yii Framework](#)

Fonte: próprio autor

Se o usuário desejar gerar o relatório em formato PDF, deverá clicar no botão

Gerar PDF.

Um exemplo deste relatório PDF gerado pelo sistema pode ser encontrado no Apêndice E

6.10.2 Consultar tutor

Para consultar o cadastro do tutor, o usuário deverá acessar o caminho do menu *Relatórios->Consultar tutor* e preencher o nome do tutor a ser pesquisado, após deve clicar no botão **Buscar**, conforme a Figura 6.26.

Figura 6.26 – Consultar tutor - Filtro

A imagem mostra a interface de usuário para consultar o cadastro de tutores. No topo, há uma barra de navegação com o menu "Relatórios" selecionado e o sub-menu "Consultar tutor" ativo. Abaixo, há uma barra de título "Consultar tutor" e uma mensagem de instrução: "Preencha os dados solicitados abaixo. Campos com asterisco (*) são de preenchimento obrigatórios." O formulário principal, intitulado "Filtro", contém um campo de texto rotulado "Tutor*" com o valor "Tutor 1" inserido. Abaixo do campo, há dois botões: "Buscar" (verde) e "Limpar campos" (cinza). No rodapé, há informações de copyright: "© SACV - Adriano Gebert Gomes 2023" e "Desenvolvido com Yii Framework".

Fonte: próprio autor

Após isso, será possível termos o acesso ao cadastro do tutor, os animais cadastrados do tutor e o histórico de consultas dos animais do tutor, conforme as Figuras 6.27 e 6.28.

Figura 6.27 – Consultar tutor - Dados

SACV Início Cadastro Relatórios Agendamento Sobre Logout (Veterinário 1)

Cadastro do tutor

Dados do tutor

Nome do Tutor	Tutor 1
Endereço	Avenida Bento Gonçalves, 123 - Agronomia - 91501-970 - Casa
Cidade	Porto Alegre - RS
Email	adriano.gomes@inf.ufrgs.br
Telefone	(51) 00000000

Animais cadastrados

[Novo animal](#)

Exibindo 1-2 de 2 itens.

Animal	Espécie
Pet 1	Cachorro

Fonte: próprio autor

Figura 6.28 – Consultar tutor - Histórico consultas

SACV Início Cadastro Relatórios Agendamento Sobre Logout (Veterinário 1)

Data de Nascimento 01/03/2020

Observações Castrado.

[Histórico do peso](#) [Evolução do peso](#)

Histórico de consultas

Exibindo 1-4 de 4 itens.

Médico Veterinário	Data	Horário
Veterinário 1	20/03/2023	08:00
Veterinário 1	21/03/2023	08:00
Veterinário 1	22/03/2023	08:00
Veterinário 1	23/03/2023	08:00

© SACV - Adriano Gebert Gomes 2023 Desenvolvido com [Yii Framework](#)

Fonte: próprio autor

Temos a possibilidade de visualizar o atendimento, que será visto na seção 6.10.4 e de edição do atendimento, que será visto na seção 6.10.5.

6.10.3 Histórico do paciente

Para consultar o histórico do paciente, o usuário deverá acessar o caminho do menu *Relatórios->Histórico do paciente* e preencher o nome do tutor a ser pesquisado e escolher um animal, após isso deve clicar no botão **Buscar**, conforme a Figura 6.29.

Figura 6.29 – Consultar tutor - Filtro

© SACV - Adriano Gebert Gomes 2023 Desenvolvido com [Yii Framework](#)

Fonte: próprio autor

Nesta tela, é apresentado o histórico do paciente, onde é possível termos o acesso ao cadastro do tutor, o cadastro do animal, acompanhar as medições de peso e o histórico de consultas, conforme as Figuras 6.30 e 6.31.

Figura 6.30 – Histórico do paciente - Dados

Fonte: próprio autor

Figura 6.31 – Histórico do paciente - Histórico de consultas

Histórico de consultas

Exibindo 1-8 de 8 itens.

Médico Veterinário	Animal	Espécie	Data	Horário	
Veterinário 1	Pet 1	Cachorro	20/03/2023	08:00	
Veterinário 1	Pet 1	Cachorro	21/03/2023	08:00	
Veterinário 1	Pet 1	Cachorro	22/03/2023	08:00	
Veterinário 1	Pet 1	Cachorro	23/03/2023	08:00	
Veterinário 1	Pet 2	Gato	20/03/2023	08:30	
Veterinário 1	Pet 2	Gato	21/03/2023	08:30	
Veterinário 1	Pet 2	Gato	22/03/2023	08:30	
Veterinário 1	Pet 2	Gato	23/03/2023	08:30	

© SACV - Adriano Gebert Gomes 2023

Desenvolvido com [Yii Framework](#)

Fonte: próprio autor

Temos a possibilidade de realizar o acompanhamento do peso do animal através das funcionalidades de Histórico do Peso e a Evolução do Peso. Ao clicar em histórico do peso, será mostrada uma janela *modal* com as últimas medições realizadas, limitada aos dez registros mais recentes, conforme a Figura 6.32.

Figura 6.32 – Janela de histórico de pesos

Histórico do peso

Exibindo os últimos 10 registros.

Data	Peso (Kg)
21/03/2023	5,60
01/03/2023	5,60
15/02/2023	5,60
01/02/2023	5,50
15/01/2023	5,30
01/01/2023	5,40
15/12/2022	5,20
01/12/2022	5,00
15/11/2022	5,00
01/11/2022	4,90

Fechar

Fonte: próprio autor

Ao clicar em Evolução do peso, será mostrada uma janela *modal* que será rende-

rizada em um gráfico simples, conforme a Figura 6.33, com o peso medido e a data da medição do conjunto das medições recentes, considerando os dez registros mais recentes.

Figura 6.33 – Janela de evolução do peso

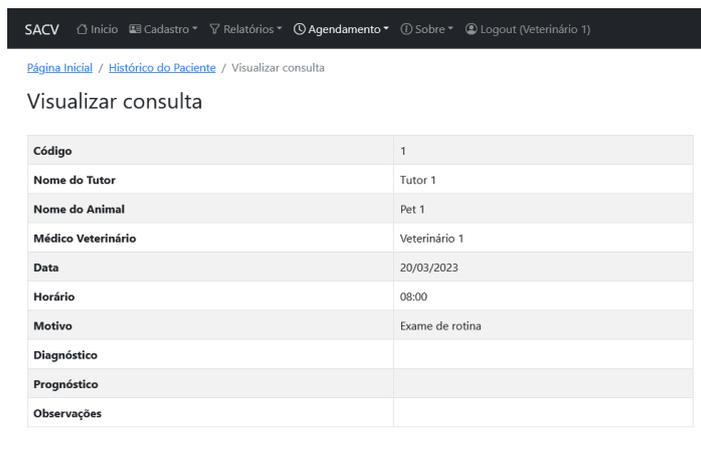


Fonte: próprio autor

6.10.4 Visualizar atendimento

Para acessar a tela de atendimento no modo visualização, deve-se estar na tela de Consultar tutor ou Histórico do Paciente, ir em histórico de consultas e clicar no ícone de visualizar (olho). Após isso, será mostrada a tela da Figura 6.34 para consulta dos dados da consulta.

Figura 6.34 – Tela de visualizar atendimento



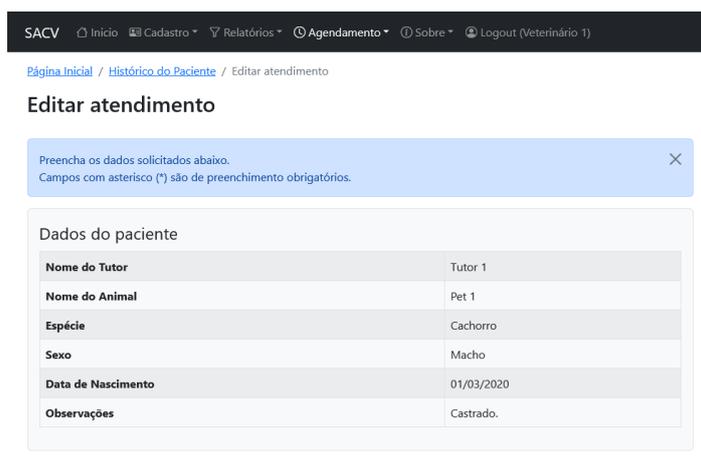
Código	1
Nome do Tutor	Tutor 1
Nome do Animal	Pet 1
Médico Veterinário	Veterinário 1
Data	20/03/2023
Horário	08:00
Motivo	Exame de rotina
Diagnóstico	
Prognóstico	
Observações	

Fonte: próprio autor

6.10.5 Editar atendimento

Para acessar a tela de atendimento no modo edição, deve-se estar na tela de Consultar tutor ou Histórico do Paciente, ir em histórico de consultas e clicar no ícone de edição (lápis). Após isso, será mostrada a tela da Figura 6.35 para edição dos dados da consulta.

Figura 6.35 – Tela de Editar atendimento



Preencha os dados solicitados abaixo.
Campos com asterisco (*) são de preenchimento obrigatórios.

Dados do paciente	
Nome do Tutor	Tutor 1
Nome do Animal	Pet 1
Espécie	Cachorro
Sexo	Macho
Data de Nascimento	01/03/2020
Observações	Castrado.

Fonte: próprio autor

6.10.6 Gerência

Ao gerente de TI é disponibilizada a possibilidade de gerência das tabelas do banco de dados pela aplicação. Para isso, deve-se acessar o menu CRUD e escolher a tabela que deseje listar os registros ou realizar alguma das ações de CRUD, que consiste em inserir, editar, excluir e visualizar um determinado registro de uma tabela do banco. Abaixo, temos um exemplo de tela de gerência para a tabela de login, conforme a Figura 6.36.

Figura 6.36 – Tela de gerência de logins

#	Id Login	Nome de usuário	Id Pessoa	Pessoa	Senha	Funcionário	Veterinário	Gerente TI	
1	1	admin	1	admin	Sim	Não	Não	Sim	
2	2	vet1	2	Veterinário 1	Sim	Não	Sim	Não	
3	3	vet2	3	Veterinário 2	Sim	Não	Sim	Não	
4	4	vet3	4	Veterinário 3	Sim	Não	Sim	Não	
5	5	fun1	5	Funcionário 1	Sim	Sim	Não	Não	
6	6	tut1	6	Tutor 1	Sim	Não	Não	Não	
7	7	tut2	7	Tutor 2	Sim	Não	Não	Não	
8	8	tut3	8	Tutor 3	Sim	Não	Não	Não	

Fonte: próprio autor

Abaixo, temos um exemplo de tela de gerência para a tabela de parâmetros de configuração do sistema, conforme a Figura 6.37.

Figura 6.37 – Tela de gerência de parâmetros

SACV [Início](#) [Cadastro](#) [CRUD](#) [Relatórios](#) [Agendamento](#) [Sobre](#) [Logout \(admin\)](#)

[Página Inicial](#) / Gerenciar Parâmetros

Gerenciar Parâmetros

[Inserir Parâmetro](#)

Exibindo 1-6 de 6 itens.

#	Id Parametro	Nome	Valor	Tipo	Descricao	
1	1	PARAM_HORARIO_INICIO	08:00	string	Horário de início da consulta	Ver Editar Excluir
2	2	PARAM_HORARIO_FIM	17:00	string	Horário de fim da consulta	Ver Editar Excluir
3	3	PARAM_DURACAO_CONSULTA	30	integer	Tempo (minutos) de duração da consulta	Ver Editar Excluir
4	4	PARAM_TOLERANCIA_CANC	60	integer	Tempo (minutos) de tolerância para cancelamento.	Ver Editar Excluir
5	5	PARAM_ENVIO_EMAIL	true	boolean	Ativa o envio de e-mail	Ver Editar Excluir
6	6	PARAM_TESTE_USUARIO				Ver Editar Excluir

Fonte: próprio autor

7 EXPERIMENTO DE USABILIDADE

Neste capítulo, são abordados o experimento de usabilidade, as etapas do experimento, lista de ações, perfil dos participantes, resultados obtidos e as considerações obtidas do experimento.

7.1 Metodologia adotada

O experimento aplicado consiste em um teste de usabilidade para avaliação de critérios de usabilidade, utilidade, facilidade e percepção de desempenho ao realizar uma lista de ações pré-estabelecidas na aplicação, possibilitando assim avaliar a qualidade e validar as funcionalidades implementadas.

A aplicação do experimento foi realizada de maneira remota e sem supervisão, sendo feito um convite para um grupo pequeno de pessoas, menos que 30 pessoas do círculo de convivência do autor desta monografia, mediante um *link* para acesso a um formulário com instruções, questões para coleta de dados e lista de ações a serem realizadas. A participação seria condicional ao aceite do termo de ciência sobre o experimento, uso dos dados e liberdade de desistir do experimento a qualquer momento. O experimento foi aplicado para 18 participantes. Importante ressaltar que o experimento não foi aplicado com médicos veterinários e funcionários de clínicas veterinárias.

Teve-se a preocupação em relação aos dados pessoais e privacidade dos participantes no experimento, para isso foi ativada na aplicação Web o modo “Teste de Usabilidade” através de um parâmetro de configuração. Esse modo altera alguns comportamentos da aplicação, tais como: envio de e-mail desativado, permite somente acesso ao cadastro de tutores previstos do teste (sem acesso aos cadastros de tutores participantes do experimento) e dados pessoais sensíveis (e.g. endereço, CPF) são mascarados pela aplicação. Também foi orientado durante o experimento que os participantes não preenchessem dados reais e sensíveis no cadastro de tutor.

Para a coleta de dados adotou-se: questões de múltipla escolha para critérios de identificação do perfil, experiências e dispositivo usado; questões na escala de 1 a 5 (escala Likert), sendo do negativo para o positivo, para critérios objetivos referente a avaliação da aplicação Web; espaço em aberto para comentários gerais (sugestões, elogios, dificuldades).

7.2 Etapas do experimento

A aplicação do experimento consistiu de um formulário único (coleta de dados, instruções e lista) dividido em três etapas, sendo elas:

- **Etapa 1:** etapa de pré-questionário para conhecimento do perfil do participante, onde foram coletadas informações sobre faixa etária, gênero, escolaridade, área do conhecimento, conhecimentos de informática, dispositivos de navegação e experiência com animais de estimação.
- **Etapa 2:** etapa prática com teste de usabilidade, onde foram disponibilizados para o participante duas listas de ações a serem executadas, a primeira lista com ações na persona de tutor e segunda lista na persona de médico veterinário, além do *link* e *qr-code* de acesso à aplicação Web em produção.
- **Etapa 3:** etapa de pós-questionário para avaliação da aplicação Web com questões objetivas com escala de 1 a 5 sobre a usabilidade, facilidade das ações, desempenho e utilidade da aplicação Web. Nessa etapa, também se coletou dados sobre o dispositivo que foi realizado o experimento, se recomendaria a aplicação e um espaço para livre manifestação sobre o experimento e a aplicação.

7.3 Teste de usabilidade

O teste de usabilidade terá como objetivo avaliar o quão fácil e intuitivo é realizar algumas ações na aplicação Web, cobrindo as principais funcionalidades da aplicação Web, tais como: agendamento, cancelamento, finalização e cadastro, além da avaliação das instruções, navegação e tempo de resposta das ações.

Para isso são apresentadas duas listas de ações propostas aos participantes, sendo a primeira lista como tutor e a segunda como veterinário. Ficou em aberto a possibilidade do participante realizar o teste de usabilidade em dispositivos convencionais com telas grandes, em dispositivos móveis ou em ambos.

7.3.1 Lista de ações do tutor

Temos abaixo a lista de ações a serem realizadas no sistema, no papel de um usuário tutor.

- Acessar o link da aplicação Web;
- Clicar no botão "Criar conta";
- Preencher os dados solicitados e clicar em "Salvar"; Observação: Não inserir dados sensíveis (como CPF, data de nascimento, endereço e e-mail) ; verdadeiros, apenas preencha algum dado, não haverá validação se estão corretos (como CPF);
- Clicar no botão "Novo animal";
- Preencher os dados solicitados de algum animal hipotético e clicar no botão "Salvar";
- Clicar no botão "Novo agendamento";
- Preencher os dados solicitados para uma consulta, escolher uma data, médico veterinário e clicar no botão "Verificar horários";
- Escolher algum horário disponível;
- Clicar no botão "Agendar";
- Após isso será mostrada uma tela com a consulta agendada em "Meus agendamentos". Clicar em "Cancelar consulta";
- Confirme o cancelamento escolhendo a opção "Sim"; Clique em Logout no menu superior da aplicação; Dispositivos móveis: para acessar o menu clicar no botão de contexto ("hamburger") para efetuar o Logout;
- O experimento como tutor terminou, você pode fechar a aba da aplicação Web e prosseguir para as atividades como veterinário no próximo tópico deste formulário.

7.3.2 Lista de ações do médico veterinário

Temos abaixo a lista de ações a serem realizadas no sistema, no papel de um usuário médico veterinário.

- Acessar o link da aplicação Web;
- Fazer a entrada no sistema utilizando o usuário "vet10" e a senha "vet10". Clicar em "Entrar";
- Será mostrada uma tela com consultas atribuídas para você em "Minhas consultas", favor clicar no botão "Finalizar" de alguma consulta desta tela;
- Preencha os dados solicitados (hipotéticos) e finalize a consulta clicando em "Finalizar atendimento";

- Agora você voltou para a tela inicial, realize o cancelamento de alguma consulta agendada em "Minhas consultas". Clique em "Cancelar";
- Confirme o cancelamento escolhendo a opção "Sim"
- Agora acesse no menu "Relatórios" a opção "Histórico do Paciente";
- Preencha os seguintes dados em Tutor: "Tutor 1" e depois em Animal escolha a opção "Pet 1". Clique em "Buscar";
- Será exibida uma tela com dados do tutor, do paciente e o histórico de consultas do animal. Acesse os detalhes de alguma consulta do histórico clicando no ícone de olho;
- Você verá dados da consulta selecionada, nesse momento você pode fechar a aplicação e o experimento terminou;
- Clique no botão Próxima deste formulário para prosseguir para a etapa seguinte.

7.4 Resultados obtidos

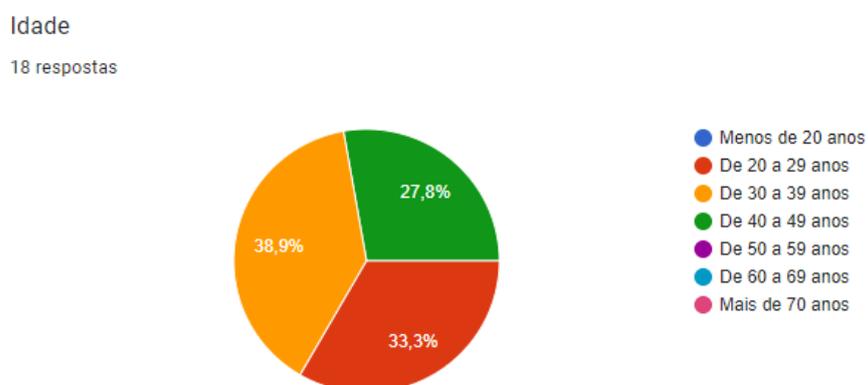
Nesta seção, serão apresentados os resultados da coleta de dados do formulário do experimento conforme as respostas fornecidas pelos participantes e os gráficos foram gerados pela ferramenta do Google Forms.

7.4.1 Perfil dos participantes

São apresentados na sequência os resultados referente a primeira etapa para conhecimento do perfil e experiências pessoais dos participantes do experimento.

Na Figura 7.1 temos o percentual referente a faixa etária dos participantes, que foram de 6 com idades entre 20 a 29 anos, 7 com idades entre 30 a 39 anos e 5 com idades entre 40 a 49 anos.

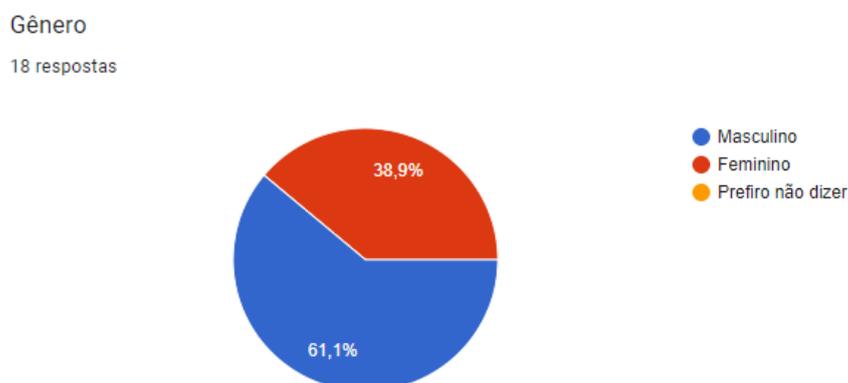
Figura 7.1 – Gráfico - Faixa etária



Fonte: próprio autor

Na Figura 7.2 é apresentado o percentual referente ao gênero, que foram de 11 homens e 7 mulheres que participaram do experimento.

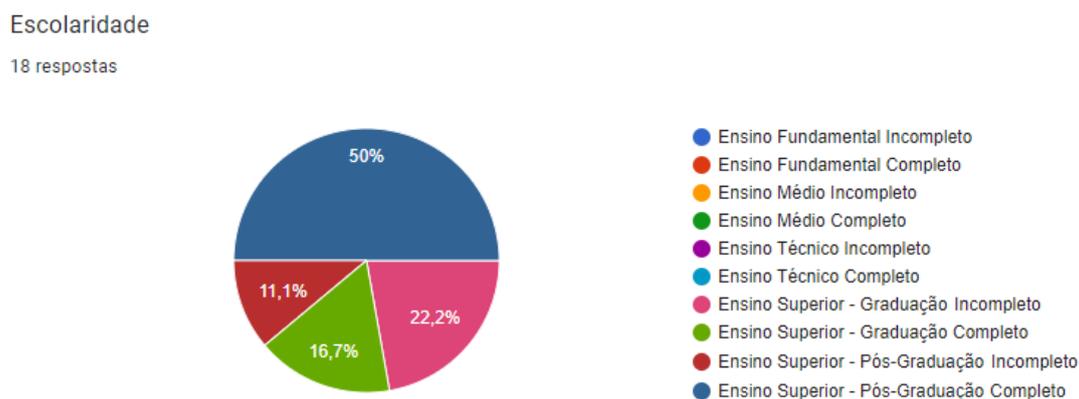
Figura 7.2 – Gráfico - Gênero



Fonte: próprio autor

Temos na Figura 7.3 o percentual referente a escolaridade, sendo que 4 possuem graduação incompleta, 3 possuem graduação completa, 2 possuem pós-graduação incompleta e 9 possuem pós-graduação completa. Todos os participantes possuem alguma formação superior, mesmo que incompleta.

Figura 7.3 – Gráfico - Escolaridade



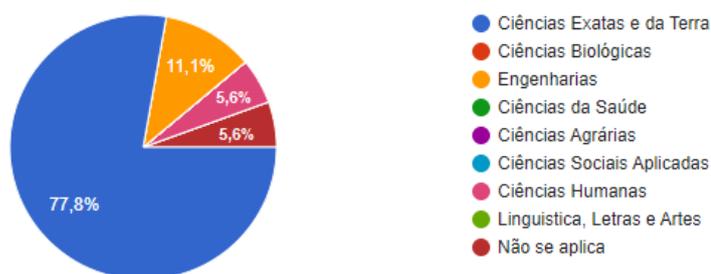
Fonte: próprio autor

Conforme a Figura 7.4, tivemos a maioria de participantes da área de ciências exatas e da terra com 14 participantes, seguido da área de engenharias com 2 participantes, área de ciências humanas 1 participante e não se aplica 1 participante.

Figura 7.4 – Gráfico - Área de formação

Caso você possua escolaridade de nível técnico ou superior, indique em qual área do conhecimento é sua formação:

18 respostas



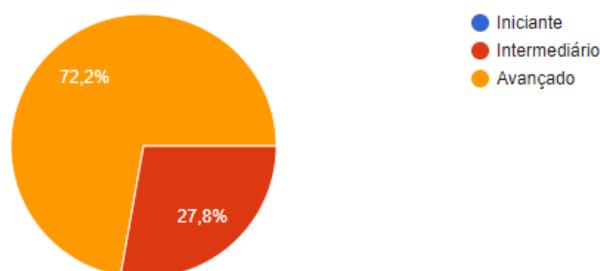
Fonte: próprio autor

Conforme a Figura 7.5, tivemos a maioria dos participantes que consideram que possuem conhecimento avançado em informática com 13 participantes. Tivemos 5 participantes que consideram ter conhecimento médio em informática.

Figura 7.5 – Gráfico - Conhecimentos em informática

Como autoavaliao meus conhecimentos em informática

18 respostas



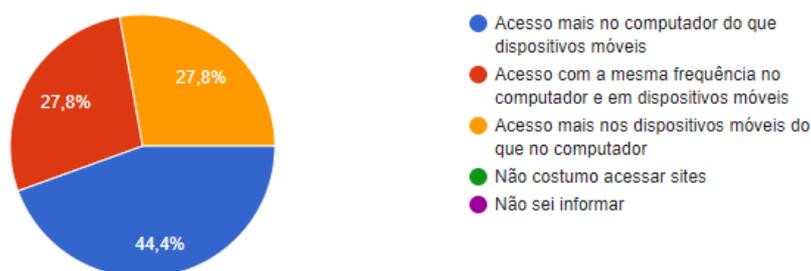
Fonte: próprio autor

Conforme a Figura 7.6, tivemos um equilíbrio de dois grupos que acessam mais computadores e acessam mais celulares. Predomina mais o uso de computadores, sendo que 8 participantes usam mais o computador, 5 participantes usam mais dispositivos móveis e 5 participantes usam computador e dispositivos móveis com a mesma frequência.

Figura 7.6 – Gráfico - Uso dispositivos móveis

Com qual frequência você acessa sites ou páginas da Web em dispositivos móveis ?

18 respostas



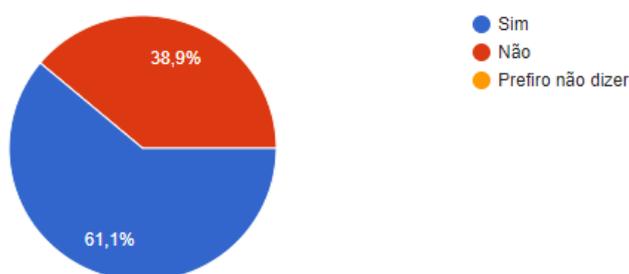
Fonte: próprio autor

Conforme a Figura 7.7, a maioria dos participantes possui animais de estimação, sendo que 11 participantes possuem animais e 7 participantes não possuem animais.

Figura 7.7 – Gráfico - Pessoas que são tutoras

Você é uma pessoa tutora de algum animal ?

18 respostas



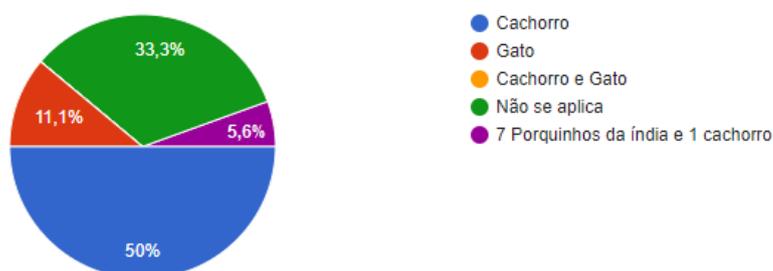
Fonte: próprio autor

Conforme a Figura 7.8, a maioria dos participantes possui cachorro e um percentual bem relevante de pessoas que não possuem animais. Tivemos 9 participantes que possuem cachorro, 2 participantes possuem gato, 1 participante que possui porquinho-da-índia e cachorro e 6 que responderam não se aplica.

Figura 7.8 – Gráfico - Animais que pessoas tutoras possui

Quais animais você cuida ?

18 respostas



Fonte: próprio autor

Conforme a Figura 7.9, tivemos um percentual relevante que realiza o agendamento via mensagem de texto, e também um percentual relevante de não se aplica. Um caso curioso é que nenhum dos participantes relata o uso de uma aplicação para agendamento de consultas. Tivemos 6 participantes que mandam mensagem de texto, 6 participantes que marcaram não se aplica, 4 participantes que levam presencialmente na clínica e 2 participantes que ligam por telefone.

Figura 7.9 – Gráfico - Atendimento veterinário

Quando precisa de atendimento veterinário para o animal que está cuidando, você

18 respostas



Fonte: próprio autor

7.4.2 Avaliação da usabilidade

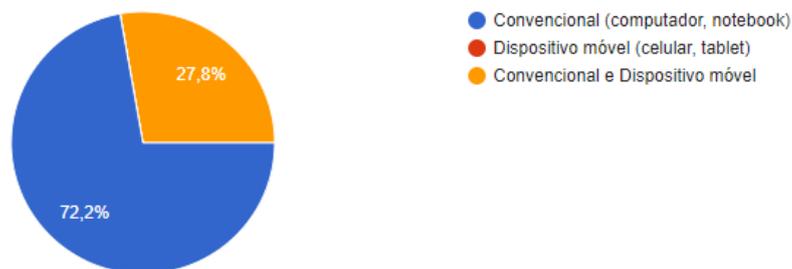
São apresentados na sequência os resultados referente a terceira etapa de usabilidade após a realização das ações propostas na segunda etapa com o uso prático da aplicação Web.

Conforme a Figura 7.10, tivemos que mais da metade dos participantes realizaram o experimento em dispositivo convencional (e.g. computador ou notebook), enquanto pouco mais de 25% realizaram o experimento tanto em tela convencional e dispositivo móvel. Tivemos 13 participantes que testaram no computador e 5 participantes que testaram tanto no computador como em dispositivo móvel.

Figura 7.10 – Gráfico - Dispositivo usado no experimento

Em qual(is) dispositivo(s) você acessou a aplicação Web ?

18 respostas



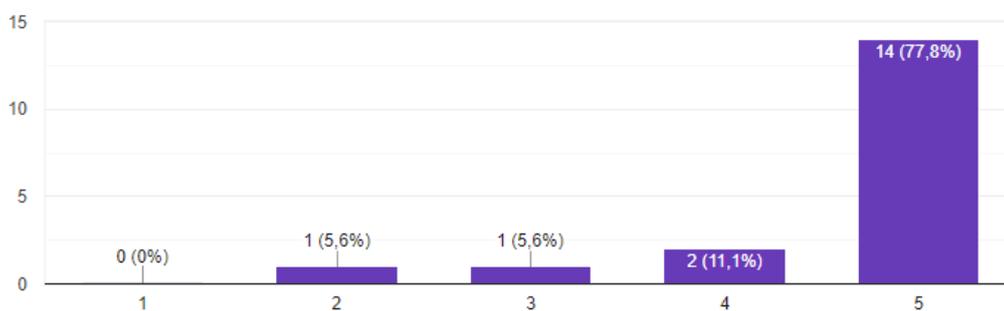
Fonte: próprio autor

Conforme a Figura 7.11, a maioria dos participantes achou que as instruções e dicas ajudaram no uso da aplicação Web. Tivemos uma participante que atribuiu nota 2, porém nos comentários relatou que ela achou o sistema tão fácil e intuitivo que as instruções não pareciam tão relevantes.

Figura 7.11 – Gráfico - Experiência das instruções

Quanto você considera que as instruções e dicas ajudaram no uso da aplicação Web ?

18 respostas



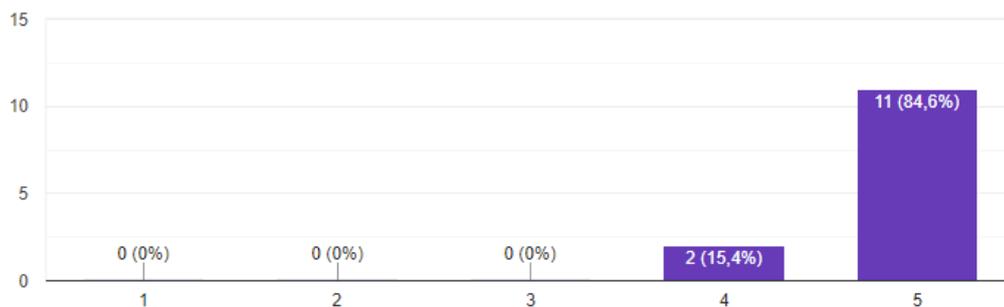
Fonte: próprio autor

Conforme a Figura 7.12, a ampla maioria dos participantes acharam excelente a responsividade da aplicação Web. Esse foi o único critério objetivo de preenchimento opcional e tivemos 13 respostas.

Figura 7.12 – Gráfico - Responsividade

Como você avalia a responsividade da aplicação Web ?

13 respostas



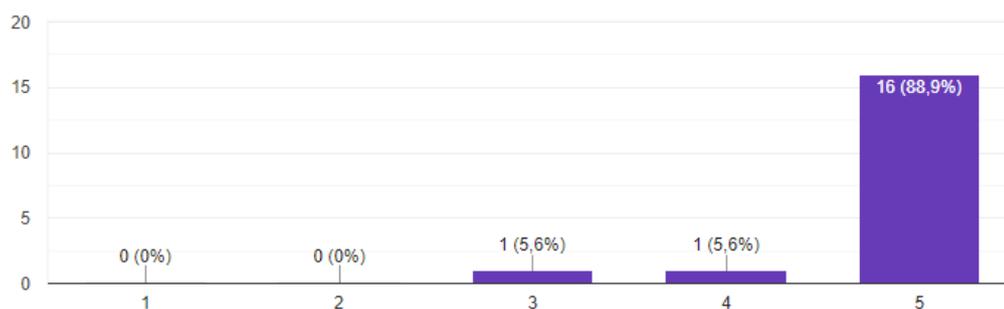
Fonte: próprio autor

Conforme a Figura 7.13, a ampla maioria dos participantes acharam muito fácil realizar a ação de cadastro de animais. Tivemos um participante que atribuiu nota 3, que relatou nos comentários que sentiu falta de outros tipos de animais para cadastro, além de cachorro e gato (e.g. porquinho-da-índia).

Figura 7.13 – Gráfico - Facilidade cadastro de animal

Como você avalia a facilidade para cadastrar um animal ?

18 respostas



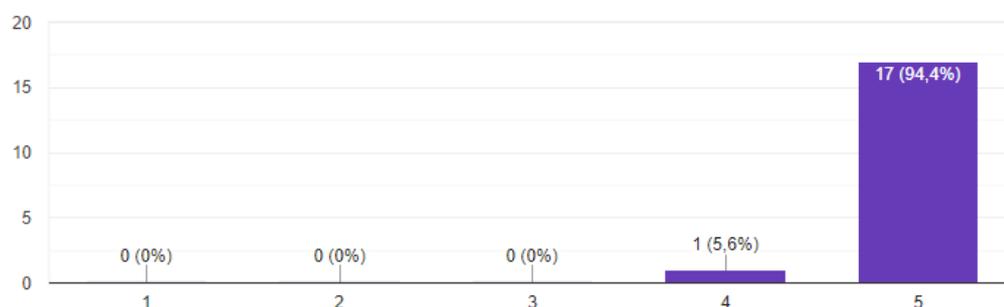
Fonte: próprio autor

Conforme a Figura 7.14, a ampla maioria dos participantes acharam muito fácil realizar a ação de agendar consulta.

Figura 7.14 – Gráfico - Facilidade de agendamento da consulta

Como você avalia a facilidade para agendar uma consulta ?

18 respostas



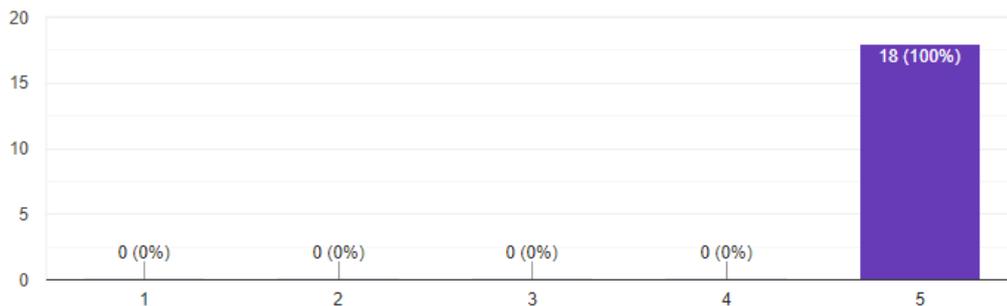
Fonte: próprio autor

Conforme a Figura 7.15, todos os participantes acharam muito fácil realizar a ação de cancelar consulta.

Figura 7.15 – Gráfico - Facilidade de cancelamento da consulta

Como você avalia a facilidade para cancelar uma consulta ?

18 respostas



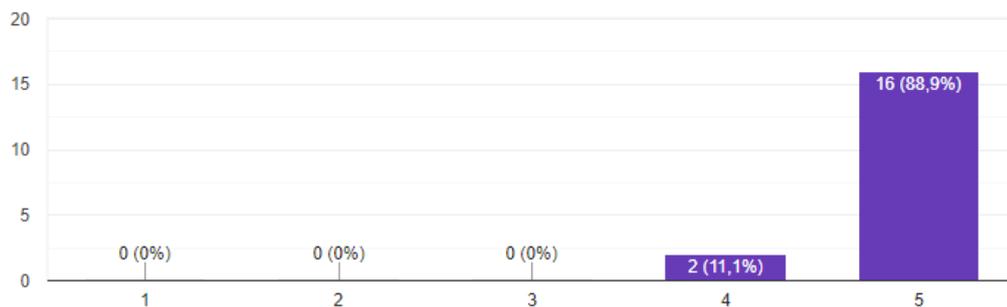
Fonte: próprio autor

Conforme a Figura 7.16, a ampla maioria dos participantes acharam muito fácil realizar a ação de finalizar consulta.

Figura 7.16 – Gráfico - Facilidade de finalizar a consulta

Como você avalia a facilidade para finalizar uma consulta ?

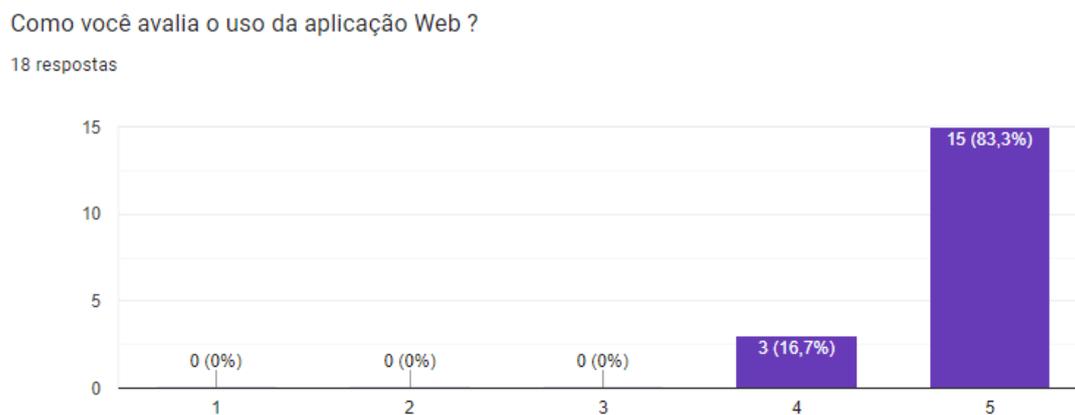
18 respostas



Fonte: próprio autor

Conforme a Figura 7.17, a ampla maioria dos participantes acharam excelente a usabilidade da aplicação Web.

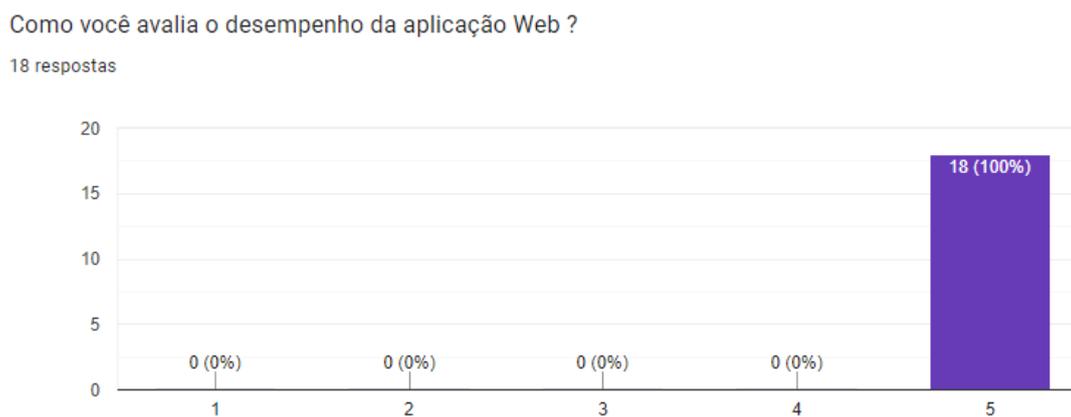
Figura 7.17 – Gráfico - Uso aplicação



Fonte: próprio autor

Conforme a Figura 7.18 todos os participantes acharam o desempenho da aplicação Web excelente.

Figura 7.18 – Gráfico - Desempenho aplicação



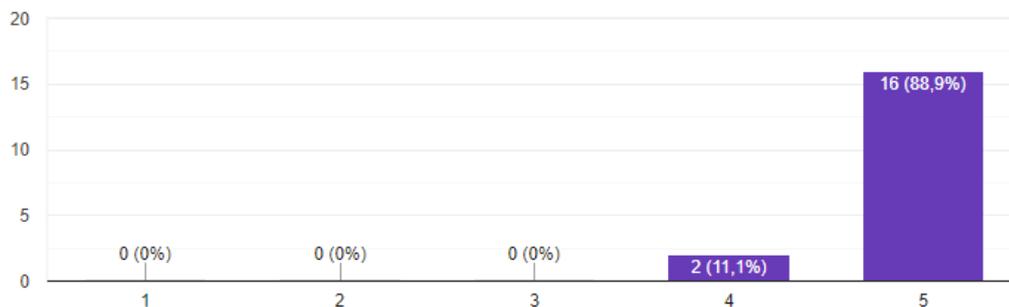
Fonte: próprio autor

Conforme a Figura 7.19 a ampla maioria dos participantes consideraram que a aplicação Web possui uma utilidade.

Figura 7.19 – Gráfico - Utilidade aplicação

Como você avalia a utilidade da aplicação Web ?

18 respostas



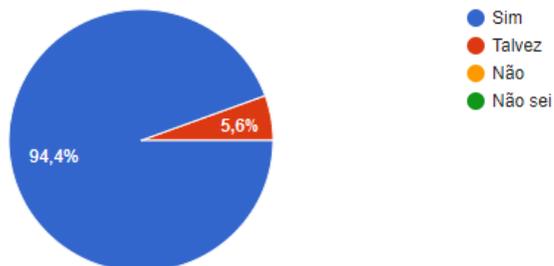
Fonte: próprio autor

Finalmente, temos na Figura 7.20 na quase unanimidade dos participantes recomendariam a aplicação Web para outra pessoa, apenas um participante marcou a opção talvez.

Figura 7.20 – Gráfico - Recomendação da aplicação

Você recomendaria a alguém essa aplicação Web?

18 respostas



Fonte: próprio autor

7.5 Considerações do experimento

Foram considerados os dados fornecidos pelos participantes, porém foi identificado algumas situações: um participante marcou que não tinha animal, mas em outro critério marcou que tinha um cachorro e na conferência da base de dados da aplicação em ambiente de produção, provavelmente 4 pessoas não realizaram todas as ações propostas no experimento. Tendo em vista que os participantes acessaram e inseriram dados na aplicação Web, as respostas fornecidas no experimento foram consideradas na análise.

Neste experimento, tivemos um perfil de participantes com elevada escolaridade, conhecimento intermediário-avançado em informática básica e em grande maioria da área de ciências exatas e da terra. Em relação ao uso de dispositivos, tivemos um certo equilíbrio, tendo uma pequena tendência ao uso maior de computador do que em dispositivos móveis para navegação de sites. Pouco mais da metade dos participantes possuem animais de estimação, sendo que a maioria destes possuem cachorro, além disso, quando necessário atendimento veterinário os meios são presenciais, telefone ou mensagem de texto.

Quase dois terços dos participantes realizaram o experimento no computador e pouco mais de um terço realizaram tanto no computador como em dispositivos móveis. Para os critérios de usabilidade, facilidade, desempenho e utilidade, a ampla maioria dos participantes avaliaram muito bem a aplicação, com resultados de muito bom a excelente. Foram ínfimas as respostas com pontuação 2 ou 3, totalizando três respostas do total.

Em relação aos comentários, grande parte foram de elogios e recomendações a aplicação Web como também de sugestões de melhorias com alguns campos não serem obrigatórios, aumento de ícones de ações, *slot* de horários agendados com outra cor. Também foram poucos relatos sobre dificuldades no campo de data, campo de moeda e acesso ao relatório.

Sendo assim, podemos constatar que a aplicação foi muito bem avaliada, tendo o experimento validado a proposta da aplicação Web, tendo resultados muito bons nos critérios objetivos e como todo o *software* está sujeita a melhorias, aperfeiçoamentos e possíveis correções relatadas pelos participantes do experimento.

8 CONCLUSÃO

Neste trabalho foi proposto o desenvolvimento de uma aplicação Web para o agendamento de clínica veterinária, demonstrando todas as etapas de revisão do estado da arte e de trabalhos prévios, levantamento de requisitos, definição do modelo de dados, definição de *layout*, implementação, publicação e validação com experimento com usuários.

Foi realizada uma revisão do modelo de dados e das funcionalidades de um trabalho prévio anterior de uma aplicação *Desktop* desenvolvida num projeto de curso Técnico em Informática, entre as principais melhorias é a possibilidade do tutor realizar agendamento e cancelamento de consultas em qualquer lugar através de um navegador Web.

Para a implementação da aplicação Web utilizou o *framework* Yii 2.0 em linguagem PHP, usando o Bootstrap 5 como *framework* de *frontend* e o banco de dados relacional MySQL. Durante o desenvolvimento desta aplicação, tentou-se aproveitar os recursos oferecidos pelo Bootstrap, em especial tornar as telas responsivas para dispositivos móveis, bem como tornar agradável e intuitivo para o usuário o uso da aplicação.

Foi adotado um protocolo de publicação, onde tivemos um ambiente de desenvolvimento para codificação e implementação das funcionalidades, um ambiente de homologação para realizar testes em um sistema sem as configurações e pacotes de desenvolvimento e finalmente um ambiente de produção onde foi hospedado a aplicação Web para uso pelos usuários. Para isso, foi usado um sistema de controle de versionamento Git para termos um repositório do projeto e possibilitar essa integração entre ambientes.

Foi realizado um experimento de teste de usabilidade, onde foi aplicado um questionário para coleta de dados e uma lista de ações para serem realizadas no sistema, com vistas para validação dos requisitos e avaliar a usabilidade da aplicação. Os resultados do experimento foram muito satisfatórios, onde a ampla maioria dos participantes avaliaram como excelente a usabilidade e não tiveram maiores dificuldades na realização das tarefas. Foram apontadas algumas sugestões de melhorias e detalhes que precisam ser ajustados, mas que faz parte de todo processo de desenvolvimento a implementação de melhorias e aperfeiçoamento.

Para o desenvolvimento deste trabalho de graduação, foram necessários os conhecimentos consolidados de diversas atividades de ensino da graduação, tais como: Algoritmos e Programação, Estrutura de Dados I, Banco de Dados I, Programação Orientada a Objetos, Programação para Web, Técnicas de Construção de Programas, Engenharia de Software I, Interação Homem-Computador, Métodos Ágeis e Laboratório de Sistema de

Software; ficando assim evidenciado a importância da formação superior obtida para o desenvolvimento deste trabalho. Também houve a contribuição significativa e relevante da formação técnica e experiência profissional como desenvolvedor Web do autor desta monografia que possibilitaram o desenvolvimento desta aplicação. Sendo assim, fica evidente que esse trabalho de graduação possibilitou demonstrar os conhecimentos obtidos na formação superior, desde a proposta até um produto final.

A aplicação Web desenvolvida possui algumas limitações que não foram implementadas, tais como: a possibilidade de anexar arquivos na finalização de consultas, a inclusão de fotos dos cadastros de pessoa e animais, acesso ao histórico e evolução de peso pelo tutor e carteira de vacinação do animal.

Como melhorias da aplicação Web para trabalhos futuros, considerando algumas sugestões dos participantes do experimento e de funcionalidades que não entraram no escopo deste trabalho, temos diversas possibilidades como: a implementação de recursos de acessibilidade Web; a inclusão de gráficos dos dados referentes aos atendimentos realizados; a inclusão de um cadastro de exames e procedimentos com valores pré-estabelecidos; a possibilidade do envio de arquivos e exames; a implementação das limitações citadas, entre outros. Também existe a possibilidade da criação de casos de testes de aceitação e testes de regressão desta aplicação para possíveis evoluções e manutenções, usando uma biblioteca de testes automatizados. Também seria interessante realizar uma validação da aplicação com médicos veterinários e funcionários de clínicas veterinárias, até para uma possível melhoria ou expansão das funcionalidades implementadas.

Concluindo, fica evidente que a aplicação Web para o agendamento de consultas permite uma facilidade para o agendamento de consultas, bem como o cuidado da saúde dos animais, trazendo uma maior autonomia para tutores e veterinários, restando assim a importância e a relevância das aplicações informatizadas para sociedade.

REFERÊNCIAS

- BOOTSTRAP. **Bootstrap**. 2023. Available from Internet: <<https://getbootstrap.com/>>.
- COMPOSER. **Composer**. 2023. Available from Internet: <<https://getcomposer.org/>>.
- DBEAVER. **DBeaver**. 2023. Available from Internet: <<https://dbeaver.io>>.
- DOCS, M. W. **MDN Web Docs**. 2023. Available from Internet: <<https://developer.mozilla.org/pt-BR/>>.
- GITHUB. **GitHub**. 2023. Available from Internet: <<https://docs.github.com/pt/>>.
- GOMES, A. G. Sistema de agendamento de clínica veterinária. Trabalho de conclusão de curso Técnico em Informática. 2014.
- HIGHCHARTS. **Highcharts**. 2023. Available from Internet: <<https://www.highcharts.com/>>.
- MICROSOFT. **Visual Studio Code**. 2023. Available from Internet: <<https://visualstudio.microsoft.com/pt-br/>>.
- MYSQL. **MySQL**. 2023. Available from Internet: <<https://www.mysql.com/>>.
- OLIVEIRA, J. P. M. de. Sistemas de informação e sociedade. **Ciência e Cultura**, 2003.
- PHP. **PHP**. 2023. Available from Internet: <<https://www.php.net>>.
- VALENTE, M. T. **Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade**. [s.n.], 2020. ISBN 978-65-00-00077-1. Available from Internet: <<https://engsoftmoderna.info/>>.
- VETBASE. **VetBase**. 2023. Available from Internet: <<https://vetbase.com.br>>.
- VETSOFT. **Suporte VetSoft**. 2023. Available from Internet: <<https://suporte.vetsoft.com.br/>>.
- VETSOFT. **VetSoft**. 2023. Available from Internet: <<https://www.vetsoft.com.br/>>.
- VETWORK. **Suporte VetWork - Login**. 2023. Available from Internet: <<https://suportevetwork.zendesk.com/hc/pt-br/articles/360000907843-Login>>.
- VETWORK. **VetWork**. 2023. Available from Internet: <<https://vetwork.com.br/>>.
- WIKIPÉDIA. **Wikipédia, a enciclopédia livre - Versão Língua Portuguesa**. 2023. Available from Internet: <<https://pt.wikipedia.org/>>.
- XAMPP. **XAMPP**. 2023. Available from Internet: <<https://www.apachefriends.org/pt-br/index.html>>.
- YII. **Guia Definitivo para Yii 2.0**. 2023. Available from Internet: <<https://www.yiiframework.com/doc/guide/2.0/pt-br>>.

APÊNDICE A — SCRIPT PARA GERAÇÃO DO ESQUEMA DE DADOS

```

/* *****
* UFRGS - Universidade Federal do Rio Grande do Sul
* Instituto de Informatica
* Curso de Ciencia da Computacao
* Trabalho de Graduacao
* Adriano Gebert Gomes - 209865
* *****/

-- Excluir banco para testes
-- drop database sacv;

-- Criacao do banco
create database sacv;

use sacv;

-- Criacao de tabelas --

create table pessoa (
    id_pessoa          integer primary key auto_increment,
    nome               varchar(70) not null,
    cpf                varchar(14),
    data_nascimento    date,
    uf                 varchar(2),
    cidade             varchar(50),
    num_endereco       varchar(8),
    endereco           varchar(50),
    cep                varchar(10),
    email              varchar(50),
    telefone           varchar(25),
    crmv               integer,
    crmv_uf            varchar(2),

```

```

bairro                varchar(50),
complemento           varchar(50),
data_registro         date,
id_pessoa_controle   integer,
ip_controle           varchar(40),
data_controle         datetime

-- foreign key (id_pessoa_controle) references pessoa(id_pessoa)
);

```

```

create table login (
  id_login            integer primary key auto_increment,
  id_pessoa           integer not null unique,
  nome_usuario        varchar(20) not null unique,
  senha              varchar(64),
  chave_autenticacao varchar(32),
  ind_funcionario     boolean default false,
  ind_veterinario     boolean default false,
  ind_gerente         boolean default false,
  id_pessoa_controle integer,
  ip_controle         varchar(40),
  data_controle       datetime,

  foreign key (id_pessoa) references pessoa(id_pessoa)
  -- foreign key (id_pessoa_controle) references pessoa(id_pessoa)
);

```

```

create table especie (
  id_especie          integer primary key auto_increment,
  nome                varchar(50) not null,
  nome_cientifico     varchar(50),
  id_pessoa_controle integer,
  ip_controle         varchar(40),
  data_controle       datetime
);

```

```

-- foreign key (id_pessoa_controle) references pessoa(id_pessoa)
);

create table animal (
  id_animal          integer primary key auto_increment ,
  id_pessoa          integer not null ,
  id_especie         integer not null ,
  nome               varchar(70) not null ,
  sexo               varchar(1) ,
  data_nascimento    date ,
  data_obito         date ,
  observacoes        varchar(300) ,
  id_pessoa_controle integer ,
  ip_controle        varchar(40) ,
  data_controle      datetime ,

  foreign key (id_pessoa) references pessoa(id_pessoa) ,
  foreign key (id_especie) references especie(id_especie)
  -- foreign key (id_pessoa_controle) references pessoa(id_pessoa)
);

create table peso (
  id_peso            integer primary key auto_increment ,
  id_animal          integer not null ,
  peso               float ,
  data_registro      date ,
  id_pessoa_controle integer ,
  ip_controle        varchar(40) ,
  data_controle      datetime ,

  foreign key (id_animal) references animal(id_animal)
  -- foreign key (id_pessoa_controle) references pessoa(id_pessoa)
);

```

```

create table consulta (
    id_consulta      integer primary key auto_increment ,
    id_animal        integer not null ,
    id_pessoa_registro integer ,
    id_medico        integer not null ,
    data_hora        datetime not null ,
    resumo           varchar(500) ,
    diagnostico      varchar(4000) ,
    prognostico      varchar(4000) ,
    observacoes      varchar(4000) ,
    ind_finalizada   boolean default false ,
    ind_pagamento    boolean default false ,
    valor_pagamento float ,
    id_pessoa_controle integer ,
    ip_controle      varchar(40) ,
    data_controle    datetime ,

    foreign key (id_animal)      references animal(id_animal) ,
    foreign key (id_pessoa_registro) references pessoa(id_pessoa) ,
    foreign key (id_medico)      references pessoa(id_pessoa)
    -- foreign key (id_pessoa_controle) references pessoa(id_pessoa)
);

```

```

create table parametro (
    id_parametro      integer primary key auto_increment ,
    nome              varchar(30) not null ,
    valor             varchar(30) ,
    tipo              varchar(10) ,
    descricao         varchar(100) ,
    id_pessoa_controle integer ,
    ip_controle       varchar(40) ,
    data_controle     datetime
    -- foreign key (id_pessoa_controle) references pessoa(id_pessoa)
);

```

);

APÊNDICE B — SCRIPT PARA INSERÇÃO DE DADOS NO BANCO

```

/*****
* UFRGS – Universidade Federal do Rio Grande do Sul
* Instituto de Informatica
* Curso de Ciencia da Computacao
* Trabalho de Graduacao
* Adriano Gebert Gomes – 209865
*****/

USE sacv;

/*****
* COMANDOS OBRIGATORIOS PARA A IMPLANTACAO DA APLICACAO WEB
*****/

-- Insere os registros de pessoa e login , para usuario
admin com acesso de gerente

INSERT INTO pessoa (nome) VALUES ( 'admin' );

INSERT INTO login (
    id_pessoa ,
    ind_gerente ,
    nome_usuario ,
    senha ,
    chave_autenticacao
)
VALUES (
    1 ,
    1 ,
    'admin' ,
    '$2y$13$co244Dr3Cn . Eeb . I2nZT5OARkgjdvm8TUw7nT/9 BA1qT379vljSi' ,
    '1PaM8exNoOsNFMtXUxPw4lrBNLCNG–Os'
)

```

);

-- Insere as especies de animais

```
INSERT INTO especie (nome, nome_cientifico)
VALUES('Cachorro', 'Canus_lupus_lupus');
```

```
INSERT INTO especie (nome, nome_cientifico)
VALUES('Gato', 'Felis_catus_catus');
```

-- Insere os parametros de configuracao do sistema ,
-- como os valores padrao

```
insert into parametro (nome, valor, tipo, descricao)
values (
'PARAM_HORARIO_INICIO',
'08:00',
'string',
'Horario_de_inicio_da_consulta');
```

```
insert into parametro (nome, valor, tipo, descricao)
values (
'PARAM_HORARIO_FIM',
'17:00', 'string',
'Horario_de_fim_da_consulta');
```

```
insert into parametro (nome, valor, tipo, descricao)
values (
'PARAM_DURACAO_CONSULTA',
'30', 'integer',
'Tempo_(minutos)_de_duracao_da_consulta');
```

```
insert into parametro (nome, valor, tipo, descricao)
values (
'PARAM_TOLERANCIA_CANC',
```

```
'60', 'integer',  
'Tempo_(minutos)_de_tolerancia_para_cancelamento.');
```

```
insert into parametro (nome, valor, tipo, descricao)  
values (  
'PARAM_ENVIO_EMAIL',  
'0',  
'boolean',  
'Ativa_o_envio_de_e-mail'  
);
```

APÊNDICE C — FORMULÁRIO TESTE DE USABILIDADE

Formulário do experimento de usabilidade de sistema de agendamento - TCC - UFRGS

Este formulário é parte de um experimento de teste de usabilidade de um sistema de agendamento do trabalho "**Sistema de Agendamento de Clínica Veterinária: desenvolvimento de uma aplicação Web utilizando *framework* Yii 2.0**" no contexto da atividade de ensino TRABALHO DE GRADUAÇÃO (TCC) do curso de bacharelado em Ciência da Computação da Universidade Federal do Rio Grande do Sul (UFRGS).

O experimento consiste de três etapas que serão:

Etapa 1: etapa para conhecimento do perfil do participante da pesquisa, onde serão coletadas informações sobre idade, gênero, escolaridade, área do conhecimento, conhecimentos em informática básica, navegação em celulares e tablets e experiências sobre animais de estimação.

Etapa 2: etapa do teste de usabilidade do sistema, nessa etapa o participante acessará um link para acesso ao sistema de agendamento e terá a disposição duas listas de ações a serem realizadas no sistema.

Etapa 3: etapa para avaliar o teste de usabilidade realizado, nessa etapa o participante responderá questões objetivas de múltipla escolha e de escala de 1 a 5 sobre a usabilidade do sistema, realização das tarefas, percepções e responsividade. Também nessa etapa terá um espaço em aberto para preenchimento de comentários sobre sugestões, críticas e dificuldades obtidas no experimento.

Termo de livre consentimento e esclarecimento:

Os dados coletados serão anonimizados e utilizados somente para os fins acadêmicos dentro do contexto desta atividade de TCC.

Ao participar desse experimento, o participante fica ciente de que pode desistir a qualquer momento sem precisar de nenhuma justificativa.

Tempo estimado de duração do experimento: 30 minutos

Acadêmico: Adriano Gebert Gomes

Orientadora: Profa. Dra. Renata de Matos Galante

Contato: adriano.gomes@inf.ufrgs.br

*Obrigatório

1. Aceite do experimento *

Marque todas que se aplicam.

- Estou ciente do experimento que irei participar, dos termos de livre consentimento e esclarecimento, que meus dados serão anonimizados e utilizados para o experimento e que posso a qualquer momento desistir de participar desse experimento.

ETAPA 1 - PERFIL PARTICIPANTE

Etapa para conhecimento do perfil do participante da pesquisa, onde serão coletadas informações sobre idade, gênero, escolaridade, área do conhecimento, conhecimentos em informática básica, navegação em celulares e tablets e experiências sobre animais de estimação.

2. Idade *

Marcar apenas uma oval.

- Menos de 20 anos
- De 20 a 29 anos
- De 30 a 39 anos
- De 40 a 49 anos
- De 50 a 59 anos
- De 60 a 69 anos
- Mais de 70 anos

3. Gênero *

Marcar apenas uma oval.

- Masculino
- Feminino
- Prefiro não dizer
- Outro: _____

4. Escolaridade *

Marcar apenas uma oval.

- Ensino Fundamental Incompleto
- Ensino Fundamental Completo
- Ensino Médio Incompleto
- Ensino Médio Completo
- Ensino Técnico Incompleto
- Ensino Técnico Completo
- Ensino Superior - Graduação Incompleto
- Ensino Superior - Graduação Completo
- Ensino Superior - Pós-Graduação Incompleto
- Ensino Superior - Pós-Graduação Completo

5. Caso você possua escolaridade de nível técnico ou superior, indique em qual área do conhecimento é sua formação: *

Em caso de mais de uma área, escolha a que possui mais experiência

Marcar apenas uma oval.

- Ciências Exatas e da Terra
- Ciências Biológicas
- Engenharias
- Ciências da Saúde
- Ciências Agrárias
- Ciências Sociais Aplicadas
- Ciências Humanas
- Linguística, Letras e Artes
- Não se aplica

6. Como autoavaliar meus conhecimentos em informática *

Marcar apenas uma oval.

- Iniciante
- Intermediário
- Avançado

7. Com qual frequência você acessa sites ou páginas da Web em dispositivos móveis ? *

Dispositivos móveis: celulares, tablets

Marcar apenas uma oval.

- Acesso mais no computador do que dispositivos móveis
- Acesso com a mesma frequência no computador e em dispositivos móveis
- Acesso mais nos dispositivos móveis do que no computador
- Não costumo acessar sites
- Não sei informar

8. Você é uma pessoa tutora de algum animal ? *

Pessoa tutora é a pessoa que possui um animal sobre sua responsabilidade e cuidados

Marcar apenas uma oval.

- Sim
- Não
- Prefiro não dizer

9. Quais animais você cuida ? *

Marcar apenas uma oval.

- Cachorro
- Gato
- Cachorro e Gato
- Não se aplica
- Outro: _____

10. Quando precisa de atendimento veterinário para o animal que está cuidando, você *

Marcar apenas uma oval.

- Leva presencialmente na clínica
- Liga por telefone para agendar uma consulta
- Manda uma mensagem de texto para agendar uma consulta
- Utiliza algum sistema ou aplicativo para agendar uma consulta
- Não se aplica

ETAPA 2 - EXECUÇÃO DO TESTE DE USABILIDADE

Etapa do teste de usabilidade do sistema, nessa etapa o participante acessará um link para acesso a aplicação Web para agendamento e duas listas de ações a serem realizadas na aplicação.

Recomenda-se nessa etapa que mantenha abertas duas abas do navegador: uma deste formulário de coleta de dados com a lista de atividades e uma outra aba para a aplicação Web.

Dispositivos móveis: Caso queira também avaliar o uso da responsividade, pedimos que mantenha o formulário de coleta de dados com a lista de atividades aberta no computador e somente a aplicação Web aberta em algum navegador no dispositivo móvel de sua preferência (e.g. celular, tablet). Para facilitar terá abaixo um QR-CODE para acessar a aplicação no seu dispositivo móvel.

Acesso a aplicação Web de agendamento (SACV)

Clique no link ou faça a leitura por QR-CODE para o acesso da aplicação Web - Sistema de Agendamento de Clínica Veterinária (SACV)

Link de acesso: <https://adriano-gomes.000webhostapp.com/sacv/web/>

QR-Code para acesso a aplicação Web de Agendamento (SACV)

Google Formulários

Acesso a lista de atividades - Persona Tutor

Você irá desempenhar o papel de um tutor e realizar as seguintes ações:

Dispositivos móveis: Ações de clique será tocar ou pressionar o botão na tela do dispositivo.

1. Acessar o link da aplicação Web ([aqui](#));
2. Clicar no botão "**Criar conta**";
3. Preencher os dados solicitados e clicar em "**Salvar**";
Observação: Não inserir dados sensíveis (como CPF, data de nascimento, endereço e e-mail) ;verdadeiros, apenas preencha algum dado, não haverá validação se estão corretos (como CPF);
4. Clicar no botão "**Novo animal**";
5. Preencher os dados solicitados de algum animal hipotético e clicar no botão "**Salvar**";
6. Clicar no botão "**Novo agendamento**";
7. Preencher os dados solicitados para uma consulta, escolher uma data, médico veterinário e clicar no botão "**Verificar horários**";
8. Escolher algum horário disponível;
9. Clicar no botão "**Agendar**";
10. Após isso será mostrada uma tela com a consulta agendada em "Meus agendamentos".
Clicar em "**Cancelar consulta**";
11. Confirme o cancelamento escolhendo a opção "**Sim**";
12. Clique em **Logout** no menu superior da aplicação;
Dispositivos móveis: para acessar o menu clicar no botão de contexto ("hamburger") para efetuar o Logout;
13. O experimento como tutor terminou, você pode fechar a aba da aplicação Web e prosseguir para as atividades como veterinário no próximo tópico deste formulário.

Acesso a lista de atividades - Persona Veterinário

Agora você irá desempenhar o papel de um veterinário e realizar as seguintes ações:

Dispositivos móveis: Ações de clique será tocar ou pressionar o botão na tela do dispositivo.

1. Acessar o link da aplicação Web ([aqui](#))
2. Fazer a entrada no sistema utilizando o usuário "**vet10**" e a senha "**vet10**". Clicar em "**Entrar**";
3. Será mostrada uma tela com consultas atribuídas para você em "Minhas consultas", favor clicar no botão "**Finalizar**" de alguma consulta desta tela;
4. Preencha os dados solicitados (hipotéticos) e finalize a consulta clicando em "**Finalizar atendimento**";
5. Agora você voltou para a tela inicial, realize o cancelamento de alguma consulta agendada em "Minhas consultas". Clique em "**Cancelar**";
6. Confirme o cancelamento escolhendo a opção "**Sim**"
7. Agora acesse no menu "**Relatórios**" a opção "**Histórico do Paciente**";
8. Preencha os seguintes dados em Tutor: "**Tutor 1**" e depois em Animal escolha a opção "**Pet 1**". Clique em "**Buscar**";
9. Será exibida uma tela com dados do tutor, do paciente e o histórico de consultas do animal. Acesse os detalhes de alguma consulta do histórico clicando na ícone de olho;
10. Você verá dados da consulta selecionada, nesse momento você pode fechar a aplicação e o experimento terminou;
11. Clique no botão Próxima deste formulário para prosseguir para a etapa seguinte.

ETAPA 3 - REAÇÃO E PERCEPÇÕES AO TESTE DE USABILIDADE

Etapa do teste de usabilidade para avaliar o uso e percepções da aplicação Web, nessa etapa o participante responderá questões sobre o uso e facilidade do sistema de agendamento que consistirá de perguntas de múltiplas escolhas e de escala linear de 1 a 5 abordando usabilidade, facilidade, realização de tarefas, percepções de desempenho e responsividade da tela. Ao final desta etapa será disponibilizado um espaço aberto para comentários sobre críticas, dificuldades encontradas, sugestões e elogios ao experimento e/ou sistema de agendamento.

11. Em qual(is) dispositivo(s) você acessou a aplicação Web ? *

Marcar apenas uma oval.

- Convencional (computador, notebook)
- Dispositivo móvel (celular, tablet)
- Convencional e Dispositivo móvel
- Outro: _____

12. Quanto você considera que as instruções e dicas ajudaram no uso da aplicação Web ? *

Marcar apenas uma oval.

- Pouco
- _____
- 1
- _____
- 2
- _____
- 3
- _____
- 4
- _____
- 5
- _____
- Muito
- _____

13. Como você avalia a responsividade da aplicação Web ?

Responsividade é a adaptação do tamanho da fonte e disposição das ações em dispositivos móveis como celulares e tablets.

Observação: Caso você não acessou a aplicação em dispositivo móvel, pode deixar sem resposta.

Marcar apenas uma oval.

Pouco

1

2

3

4

5

Muito

14. Como você avalia a facilidade para cadastrar um animal ? *

Marcar apenas uma oval.

Pouco

1

2

3

4

5

Muito

15. Como você avalia a facilidade para agendar uma consulta ? *

Marcar apenas uma oval.

Pouco

1

2

3

4

5

Muito

16. Como você avalia a facilidade para cancelar uma consulta ? *

Marcar apenas uma oval.

Pouco
=====

1

2

3

4

5

Muito
=====

17. Como você avalia a facilidade para finalizar uma consulta ? *

Marcar apenas uma oval.

Pouco
=====

1

2

3

4

5

Muito
=====

18. Como você avalia o uso da aplicação Web ? *

Considerar interação, sistema foi intuitivo, navegação, facilidade de executar as ações no sistema em geral

Marcar apenas uma oval.

Péssimo



1



2



3



4



5



Excelente



19. Como você avalia o desempenho da aplicação Web ? *

Considerar tempo de resposta das ações, tempo que precisou realizar as ações

Marcar apenas uma oval.

Péssimo

1

2

3

4

5

Excelente

20. Como você avalia a utilidade da aplicação Web ? *

Considerar se o sistema é interessante, se pode ter utilidade para tutores e veterinários

Marcar apenas uma oval.

Pouco

1

2

3

4

5

Muito

21. Você recomendaria a alguém essa aplicação Web? *

Marcar apenas uma oval.

- Sim
- Talvez
- Não
- Não sei

22. Você possui alguma crítica, dificuldades, sugestão, recomendação e/ou elogio * sobre o experimento e/ou a aplicação Web que gostaria de compartilhar conosco ?

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários

APÊNDICE D — DICIONÁRIO DE DADOS

Neste apêndice são apresentados os dicionários de dados revisados do novo esquema de banco de dados da aplicação Web.

D.1 Pessoa

Para a entidade Pessoa temos o seguinte dicionário de dados, conforme a Tabela D.1.

Tabela D.1 – Dicionário de dados da tabela Pessoa

Atributo	Tipo(tamanho)	Restrições	Descrição
id_pessoa	integer	PK	Identificador da tabela
nome	string(70)	NOT NULL	CPF
cpf	string(14)		CPF
data_nascimento	date		Data de nascimento
uf	string(2)		Sigla do estado
cidade	string(50)		Cidade
num_endereco	string(8)		Endereço
endereco	string(50)		Endereço
cep	string(10)		Número do CEP com o traço
email	string(50)		Email
telefone	string(25)		Telefone
crm	string(11)		Número do CRMV (Veterinário)
crmv_uf	string(2)		Estado do CRMV (Veterinário)
bairro	string(50)		Bairro
complemento	string(50)		Complemento do endereço
data_registro	date		Data de cadastro
id_pessoa_controle	integer		Pessoa que fez última atualização
ip_controle	string(40)		IP da última atualização (controle)
data_controle	datetime		Data da última atualização (controle)

Fonte: próprio autor

D.2 Login

Para a entidade Login temos o seguinte dicionário de dados, conforme a Tabela

D.2.

Tabela D.2 – Dicionário de dados da tabela Pessoa

Atributo	Tipo(tamanho)	Restrições	Descrição
id_login	integer	PK	Identificador da tabela
id_pessoa	integer	FK, NOT NULL, UNIQUE	Identificador da tabela Pessoa
nome_usuario	string(30)	NOT NULL, UNIQUE	Nome de usuário do sistema
senha	string(64)		Senha criptografada
chave_autenticacao	string(32)		chave de autenticação (Yii)
ind_funcionario	boolean		Indicador permissão funcionário
ind_veterinario	boolean		Indicador permissão veterinário
ind_gerente	boolean		Indicador permissão gerente
id_pessoa_controle	integer		Pessoa que fez última atualização
ip_controle	string(40)		IP da última atualização
data_controle	datetime		Data da última atualização

Fonte: próprio autor

D.3 Espécie

Para a entidade Espécie temos o seguinte dicionário de dados, conforme a Tabela

D.3.

Tabela D.3 – Dicionário de dados da tabela Espécie

Atributo	Tipo(tamanho)	Restrições	Descrição
id_especie	integer	PK	Identificador da tabela
nome	string(50)	NOT NULL	Nome da espécie
nome_cientifico	string(50)		Nome da espécie em latim
id_pessoa_controle	integer		Pessoa que fez última atualização
ip_controle	string(40)		IP da última atualização (controle)
data_controle	datetime		Data da última atualização (controle)

Fonte: próprio autor

D.4 Animal

Para a entidade Animal temos o seguinte dicionário de dados, conforme a Tabela

D.4.

Tabela D.4 – Dicionário de dados da tabela Animal

Atributo	Tipo(tamanho)	Restrições	Descrição
id_animal	integer	PK	Identificador da tabela
id_pessoa	integer	FK, NOT NULL	Identificador da tabela Pessoa - Tutor
nome	string(70)	NOT NULL	Nome do animal
sexo	string(1)		Sexo do animal
data_nascimento	date		Data de nascimento
data_obito	date		Data de óbito
observacoes	string(300)		Campo observações
id_pessoa_controle	integer		Pessoa que fez última atualização
ip_controle	string(40)		IP da última atualização (controle)
data_controle	datetime		Data da última atualização (controle)

Fonte: próprio autor

D.5 Peso

Para a entidade Peso temos o seguinte dicionário de dados, conforme a Tabela

D.5.

Tabela D.5 – Dicionário de dados da tabela Peso

Atributo	Tipo(tamanho)	Restrições	Descrição
id_peso	integer	PK	Identificador da tabela
id_animal	integer	FK, NOT NULL	Identificador da tabela Animal
peso	float		Valor numérico do peso em Kg
data_registro	date		Data de medição do peso
id_pessoa_controle	integer		Pessoa que fez última atualização
ip_controle	string(40)		IP da última atualização (controle)
data_controle	datetime		Data da última atualização (controle)

Fonte: próprio autor

D.6 Consulta

Para a entidade Consulta temos o seguinte dicionário de dados, conforme a Tabela

D.6.

Tabela D.6 – Dicionário de dados da tabela Consulta

Atributo	Tipo(tamanho)	Restrições	Descrição
id_consulta	integer	PK	Identificador da tabela
id_animal	integer	FK, NOT NULL	Identificador paciente (Animal)
id_pessoa_registro	integer	FK	Identificador responsável agendamento
id_medico	integer	FK, NOT NULL	Identificador médico (Pessoa)
data_hora	datetime		Data e hora da consulta
resumo	string(500)		Resumo do motivo da consulta
diagnostico	string(4000)		Texto diagnóstico
prognostico	string(4000)		Texto prognóstico
observacoes	string(4000)		Observações
observacoes	string(4000)		Senha criptografada
ind_finalizada	boolean		Indicador de consulta finalizada
ind_pagamento	boolean		Indicador de consulta paga
valor_pagamento	float		Valor da consulta e procedimentos
id_pessoa_controle	integer		Pessoa que fez última atualização
ip_controle	string(40)		IP da última atualização (controle)
data_controle	datetime		Data da última atualização (controle)

Fonte: próprio autor

D.7 Parametro

Para a entidade Parametro temos o seguinte dicionário de dados, conforme a Tabela D.7.

Tabela D.7 – Dicionário de dados da tabela Peso

Atributo	Tipo(tamanho)	Restrições	Descrição
id_parametro	integer	PK	Identificador da tabela
nome	string(30)	NOT NULL	Nome do parâmetro (constante)
valor	string(30)		Valor do parâmetro
tipo	string(10)		Tipo do parâmetro
descricao	string(100)		Descrição do parâmetro
id_pessoa_controle	integer		Pessoa que fez última atualização
ip_controle	string(40)		IP da última atualização (controle)
data_controle	datetime		Data da última atualização (controle)

Fonte: próprio autor

APÊNDICE E — RELATÓRIO GERADO PELA APLICAÇÃO

Figura E.1 – Relatório gerado pela aplicação

SACV Relatório 21/03/2023

SACV - SISTEMA DE AGENDAMENTO DE CLÍNICA VETERINÁRIA

RELATÓRIO DE CONSULTAS
REFERÊNCIA: 03/2023

Médico	Consultas realizadas	Preço médio	Preço total
Veterinário 1	2	R\$ 150,00	R\$ 300,00
Veterinário 2	2	R\$ 50,00	R\$ 100,00

Total de consultas: 4
Total de saldo: R\$ 400,00

Fonte: gerado pela aplicação