

LUCAS BASTOS FRANCO

**PROJETO DE SOFTWARE:
HAPPY PAWS**

**SÃO FRANCISCO DE GOIAS
2025**

LUCAS BASTOS FRANCO

**PROJETO DE SOFTWARE:
HAPPY PAWS**

Projeto de software para obtenção do
título de engenheiro de software

**SÃO FRANCISCO DE GOIAS
2025**

CONTROLE DE VERSÃO

<u>Versão</u>	<u>Data de Início</u>	<u>Data de Fim</u>
1.0	01/01/2026	03/01/2026

LUCAS BASTOS FRANCO

**PROJETO DE SOFTWARE:
HAPPY PAWS**

Projeto de software para obtenção do
título de engenheiro de software

Aprovado em:

Lucas Bastos Franco

Dedico este trabalho a Deus; sem ele eu
não teria capacidade para desenvolver
este trabalho.

AGRADECIMENTOS

Em primeiro lugar, a Deus, que fez com que meus objetivos fossem alcançados, durante todos os meus anos de estudos.

Aos meus pais e irmãos, que me incentivaram nos momentos difíceis e compreenderam a minha ausência enquanto eu me dedicava à realização deste trabalho.

Aos amigos, que sempre estiveram ao meu lado, pela amizade incondicional e pelo apoio demonstrado ao longo de todo o período de tempo em que me dediquei a este trabalho.

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

Alan Turing

RESUMO

O software proposto tem como objetivo atender à crescente demanda das clínicas veterinárias por organização, controle e confiabilidade das informações relacionadas aos atendimentos e à saúde dos animais. A solução visa centralizar e padronizar o gerenciamento de dados de tutores, pets, atendimentos, serviços e profissionais, garantindo um histórico clínico completo, integrado e seguro. Ao substituir registros físicos e informações dispersas por um sistema unificado, o software reduz falhas operacionais, melhora a continuidade dos tratamentos e auxilia na tomada de decisões mais assertivas pelos profissionais. Dessa forma, o sistema contribui para a eficiência dos processos internos, a segurança das informações e a melhoria da qualidade do atendimento, refletindo positivamente no bem-estar dos animais e na confiança dos tutores.

Palavras-chave: Sistema de gestão veterinária; Prontuário eletrônico animal; Clínicas veterinárias; Gestão de atendimentos; Centralização de dados; Saúde animal.

ABSTRACT

The proposed software aims to address the growing demand of veterinary clinics for organization, control, and reliability of information related to animal healthcare and appointments. The solution focuses on centralizing and standardizing the management of data concerning pet owners, animals, consultations, services, and professionals, ensuring a complete, integrated, and secure clinical history. By replacing physical records and scattered information with a unified system, the software reduces operational errors, improves continuity of care, and supports more informed decision-making by veterinary professionals. As a result, the system enhances internal management efficiency, strengthens data security, and improves the overall quality of service, positively impacting professionals, pet owners, and, most importantly, animal welfare.

Keywords: Veterinary management system; Electronic pet medical record; Veterinary clinics; Appointment management; Data centralization; Animal health.

SUMÁRIO

1	INTRODUÇÃO DO SOFTWARE	11
2	DESENVOLVIMENTO DA IDEIA	11
2.1	Matriz GUT	11
2.2	Partes Interessadas.....	12
3	PERSONA DE EMPATIA	13
4	REQUISITOS E METRICAS	13
4.1	Requisitos Funcionais.....	13
4.2	Requisitos Não Funcionais	14
5	PROCESSO DE NEGÓCIO.....	15
6	PADRÕES DE DESGINER	15
6.1	Tipografia	15
6.2	Ícones.....	16
6.3	Imagens	16
6.4	Paleta de Cores	17
6.5	Fundamentos de UX	17
6.6	Espaçamentos	17
6.7	Arredondamento de Bordas	17
7	ESTRUTURA ANALÍTICA DO PROJETO	18
8	METODOLOGIA DE DESENVOLVIMENTO	18
9	MODELO DER.....	19
10	DESIGN ARQUITETURAL	19
11	FRONT-END	20
12	BANCO DE DADOS	20
13	BACK-END	21
14	VERSIONAMENTO	21

1 INTRODUÇÃO DO SOFTWARE

O crescimento do setor de clínicas veterinárias e pet shops nos últimos anos trouxe consigo uma demanda cada vez maior por organização, controle e confiabilidade das informações relacionadas aos atendimentos e à saúde dos animais. Diante de um cenário em que falhas na gestão de dados podem gerar impactos diretos na qualidade do atendimento e até implicações legais para as clínicas, torna-se evidente a necessidade de soluções tecnológicas que auxiliem na centralização e no gerenciamento adequado dessas informações.

Nesse contexto, o software proposto surge como uma solução voltada à organização, padronização e centralização do histórico completo dos pets atendidos em clínicas veterinárias. A aplicação permite o registro estruturado de dados dos tutores, informações clínicas dos animais, atendimentos realizados, serviços prestados e profissionais envolvidos, garantindo que todo o histórico do pet esteja disponível de forma integrada, segura e acessível durante os atendimentos.

Além de centralizar as informações, o sistema foi concebido para reduzir falhas operacionais causadas pela desorganização de documentos físicos ou registros dispersos em diferentes plataformas. Ao disponibilizar um histórico unificado e confiável, o software contribui para a tomada de decisões mais assertivas por parte dos profissionais, melhora a continuidade do tratamento dos animais e minimiza riscos associados à ausência ou inconsistência de dados clínicos.

Dessa forma, a solução proposta não apenas otimiza a gestão interna das clínicas veterinárias, como também promove maior segurança, transparência e eficiência nos processos de atendimento, impactando positivamente tanto os profissionais quanto os tutores e, principalmente, o bem-estar dos animais.

2 DESENVOLVIMENTO DA IDEIA

2.1 Matriz GUT

A matriz GUT é uma boa ideia para priorização de problemas de sistema, ele é levantado através de indicadores que estão atrelados a problemas que o projeto tenta mitigar, assim, trazendo dados reais e relevantes, como o sistema tenta tratar sobre **falta de histórico adequado do pet**, sua pontuação foi:

<u>CRITÉRIO</u>	<u>NOTA</u>	<u>JUSTIFICATIVA</u>
------------------------	--------------------	-----------------------------

Gravidade	5	Pode gerar lesões graves, internações, hipertermia, fraturas e até morte do animal. A clínica pode sofrer processos judicial.
Urgência	5	Falhas acontecem diariamente, no momento do atendimento, o que afeta todos os procedimentos associados a clínica
Tendência	5	O setor cresce rápido, sem controle, os erros aumentam exponencialmente. Com mais pets e serviços, o caos só aumenta
<u>NOTA FINAL</u>	125	Falhas no histórico levaram a casos de animais lesionados, hipertermia e erros no atendimento. Sem informações organizadas, o risco é crítico.

2.2 Partes Interessadas

Proprietários das Clínicas Veterinárias:

- Responsáveis pela tomada de decisão sobre a adoção do sistema.
- Interessados na melhoria da gestão, redução de riscos legais e aumento da eficiência operacional.

Médicos Veterinários:

- Interessados em acesso rápido ao histórico clínico, exames, tratamentos e evolução dos animais.
- Dependem da confiabilidade e integridade dos dados para a tomada de decisões clínicas.

Técnicos Veterinários:

- Responsáveis pelo registro de informações básicas, preparo de atendimentos e atualização de dados.
- Interessados em interfaces simples, padronizadas e ágeis para evitar erros operacionais.

Recepcionistas:

- Utilizam o sistema para cadastro de tutores, agendamentos e organização de atendimentos.
- Interessados em praticidade, rapidez no acesso às informações e redução de retrabalho.

3 PERSONA DE EMPATIA



Maria Clara é médica veterinária e atua em uma clínica de pequeno a médio porte, onde realiza atendimentos clínicos, acompanha tratamentos contínuos e mantém contato direto com tutores em diferentes estágios de cuidado com seus animais. Sua rotina é intensa, marcada por consultas sucessivas, emergências inesperadas e pela necessidade constante de acessar informações clínicas confiáveis em pouco tempo. Para ela, ter um histórico completo e bem organizado do pet é essencial para garantir diagnósticos mais precisos e decisões rápidas durante os atendimentos.

As principais dores de Maria Clara estão relacionadas à perda de tempo com tarefas burocráticas, ao risco de inconsistências nos registros clínicos e à dificuldade de acesso rápido a informações essenciais durante atendimentos urgentes. Essas limitações geram estresse e a sensação de que o cuidado com os animais poderia ser ainda melhor se houvesse um sistema mais eficiente de gestão de dados.

4 REQUISITOS E METRICAS

4.1 Requisitos Funcionais

ID	DESCRÍÇÃO	METRICA
01	O usuário poderá realizar todas as operações fundamentais do CRUD para Tutor, assim, tendo informações dos donos do pet	Métrica RF2, 3, 4 e 5 + Todas as operações devem funcionar 100% de sucesso num tempo menos de 2 segundos em cada operação
02	O usuário poderá realizar todas as operações fundamentais do CRUD para Pet, assim, tendo informações dos animais associado a clínica	Métrica RF1, 3, 4 e 5 + Nenhum erro de integridade (FK) deve aparecer durante testes
03	O usuário poderá realizar as operações de CRUD para Serviços,	Métrica RF1, 2, 4 e 5 + Testes automatizados devem apresentar 0 falhas em 10 execuções segundas

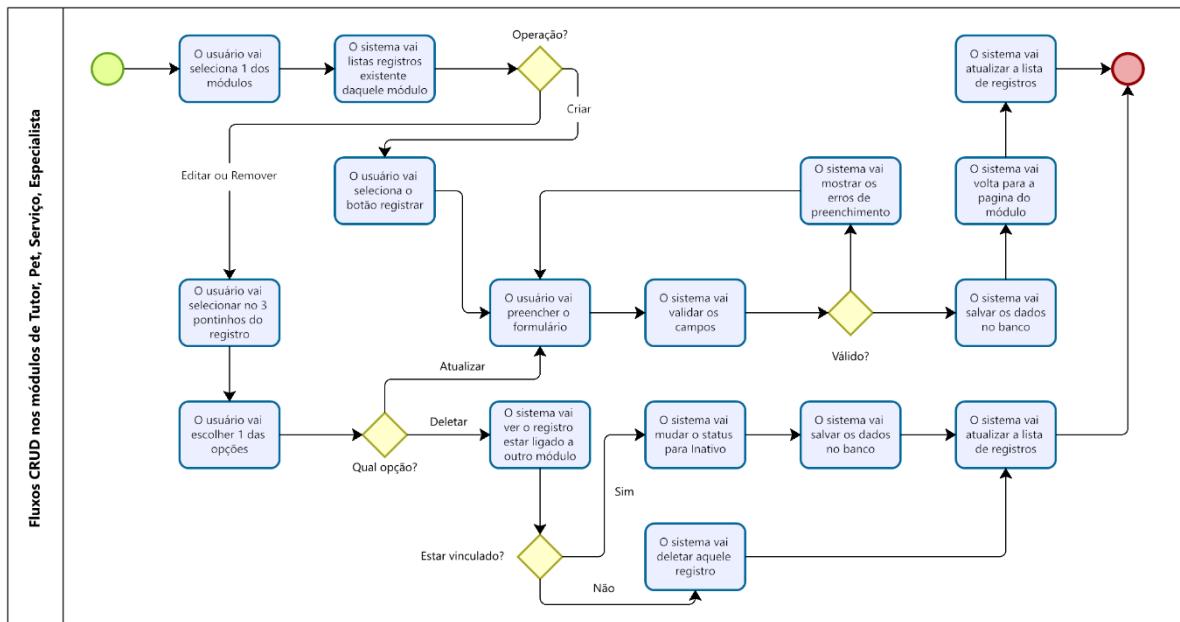
	assim, tendo informações dos tipos de serviços oferecidos.	
04	O usuário poderá realizar todas as operações fundamentais do CRUD para Especialistas, assim, tendo informações dos veterinários	Métrica RF1, 2, 3 e 5 + O sistema deve impedir acontecimentos de duplicatas de informações únicas
05	O usuário poderá realizar todas as operações fundamentais do CRUD para Atendimento, assim, tendo informações das consultas	Métrica RF1, 2, 3 e 4 + Todos os registros de relacionados de 1 pra muitos já registrados não podem ser deletados
06	O usuário poderá registrar, visualizar e baixar os relatórios completo do pet escolhido, incluindo os atendimentos e os prontuários.	Todos os atendimentos realizados devem aparecer no histórico do pet em 100% dos testes funcionais.
07	O usuário poderá recuperar a sua senha, caso ele ainda não tenha sido logado, através de uma notificação do e-mail.	O e-mail deve ser enviado em menos de 10 segundos é o link deve expirar em no máximo 1 hora.
08	O usuário poderá ver de forma fácil quais atendimentos ele terá no dia e no dia seguinte, falando o pet, serviço, data e hora	Todos os dados de atendimentos devem garantir que nenhum atendimento fora do período apareça no sistema.
09	O usuário poderá gerar relatórios em formato PDF sobre suas receitas mensais, incluindo os atendimentos, quantidade e valores.	O PDF não pode ser gerados em mais de 10 segundos e devem apresentar 0 erros de formatação em testes.

4.2 Requisitos Não Funcionais

ID	TIPO	DESCRÍÇÃO	METRICA
01	Usabilidade	O sistema deve adaptar automaticamente o tema (claro/escuro) conforme a preferência do navegador do usuário.	Testes em navegadores e dispositivos diferentes, verificando que o tema muda automaticamente em 100% dos cenários.
02	Confiabilidade	O sistema deve garantir a persistência dos dados armazenados, evitando perdas de informações	Testes de falhas devem comprovar que nenhum dado é perdido após reinicialização do sistema
03	Segurança	O sistema deve garantir na segurança dos dados do usuário e demais entidades, controlando acessos e protegendo informações	As senhas devem ser armazenadas utilizando HASH criptográfico.
04	Escalabilidade	O sistema deve suportar o crescimento contínuo do número de usuários sem degradação perceptível de desempenho.	Testes de estresse com crescimento de registros, mantendo o tempo de respostas do sistema em menos tempo possível.

05	Manutenibilidade	O sistema deve permitir manutenção e evolução facilitadas, possibilitando correções e inclusão de novas funcionalidades.	O código tem que ser escrito totalmente em inglês e organizada de acordo com a arquitetura de software definida
06	Compatibilidade	O sistema deve ser compatível com os principais navegadores modernos.	Funcionamento correto nos navegadores Chrome, Firefox e Edge.
07	Portabilidade	O sistema deve funcionar tanto em desktop, quanto em dispositivos móveis também, sem perda de funcionalidade	Testes de responsividade em todas as telas. Se utilizando do Lighthouse, vamos atingir mais de 85% de pontuação

5 PROCESSO DE NEGÓCIO



6 PADRÕES DE DESIGNER

Os padrões de designer adotados neste projeto foram definidos para garantir consistência visual, experiência do usuário aprimorada e uma identidade clara e profissional em todas as interfaces. Cada decisão foi pensada para alinhar estética, funcionalidade e acessibilidade, resultando em um produto coerente e de alta qualidade.

6.1 Tipografia

A tipografia principal do projeto utiliza a fonte Madimi One ...

ABC - 123 - @Ã?

Escolhida por sua legibilidade e modernidade. Para manter a hierarquia visual clara, foram definidos tamanhos específicos:

- Texto super grande (span - 60px).
- Título (h1 - 36px).
- Texto padrão (p - 24px).
- Texto menores (p - 18px).

Esses padrões criam uma hierarquia clara e garantem consistência em toda a interface.

6.2 Ícones

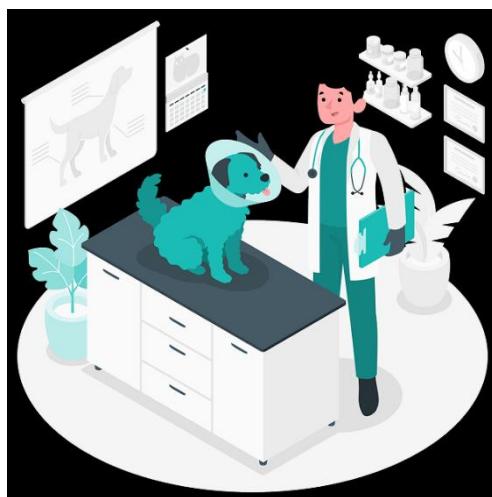
Os ícones utilizados seguem o estilo Design Circle Glyph ...



Que se caracteriza por formas sólidas. Esse estilo garante clareza visual e mantém uma aparência limpa e profissional em todos os componentes gráficos.

6.3 Imagens

As imagens foram padronizadas no estilo Amico...



Oferecendo ilustrações amigáveis e visualmente envolventes. Esse estilo equilibra modernidade e clareza.

6.4 Paleta de Cores

A paleta do projeto é centrada em tons da cor principal definido: azul ciano, ...



Assim, transmitindo calma, clareza e comunicação.

6.5 Fundamentos de UX

Todos os elementos de design seguem os 10 Fundamentos das Heurísticas de Nielsen, tendo como os principais pilares:

- Correspondência entre sistema e mundo real
- Controle e liberdade do usuário
- Reconhecimento ao invés de memorização
- Flexibilidade e eficiência de uso
- Recuperação diante de erros

Esse compromisso assegura que a experiência do usuário seja intuitiva, clara e acessível, reduzindo fricções e aumentando a eficácia da interface.

6.6 Espaçamentos

O layout utiliza espaçamentos entre itens, inputs e componentes variando entre 25px e 50px, garantindo respiro visual adequado. Esse espaçamento melhora a leitura, reduz a sobrecarga cognitiva e ajuda na organização dos elementos na tela.

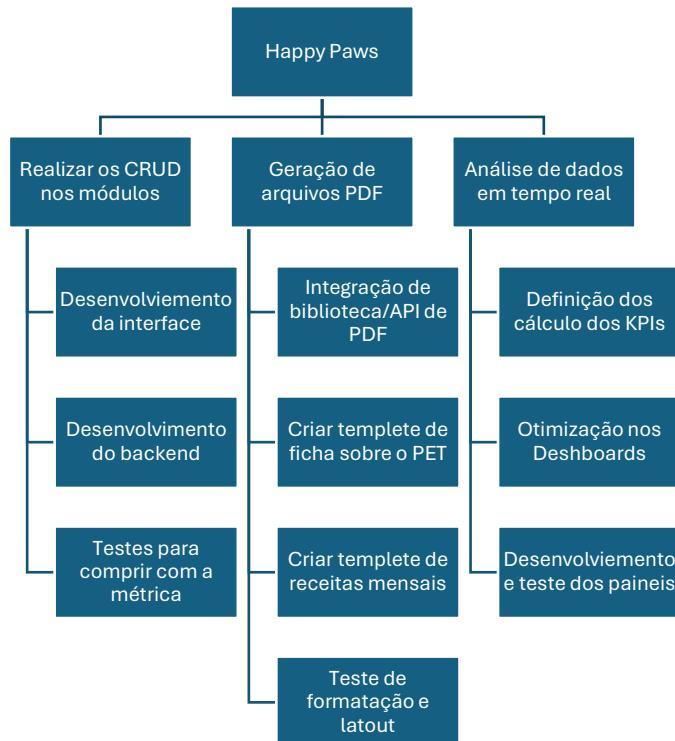
6.7 Arredondamento de Bordas

As bordas seguem uma padronização progressiva:

- Campo principal: 12px
- Campo interno dentro de outro campo: 6px
- Subcampos ainda mais internos: 3px

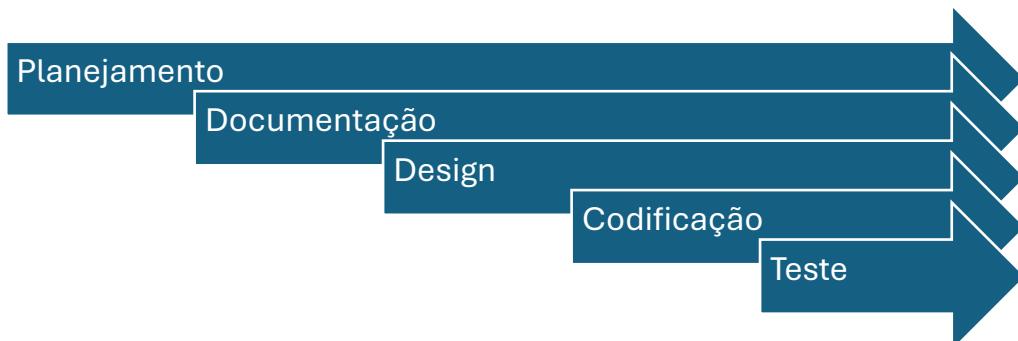
Esse sistema cria uma linguagem visual coerente e hierárquica, facilitando a identificação dos níveis de profundidade e agrupamentos dentro da interface.

7 ESTRUTURA ANALÍTICA DO PROJETO



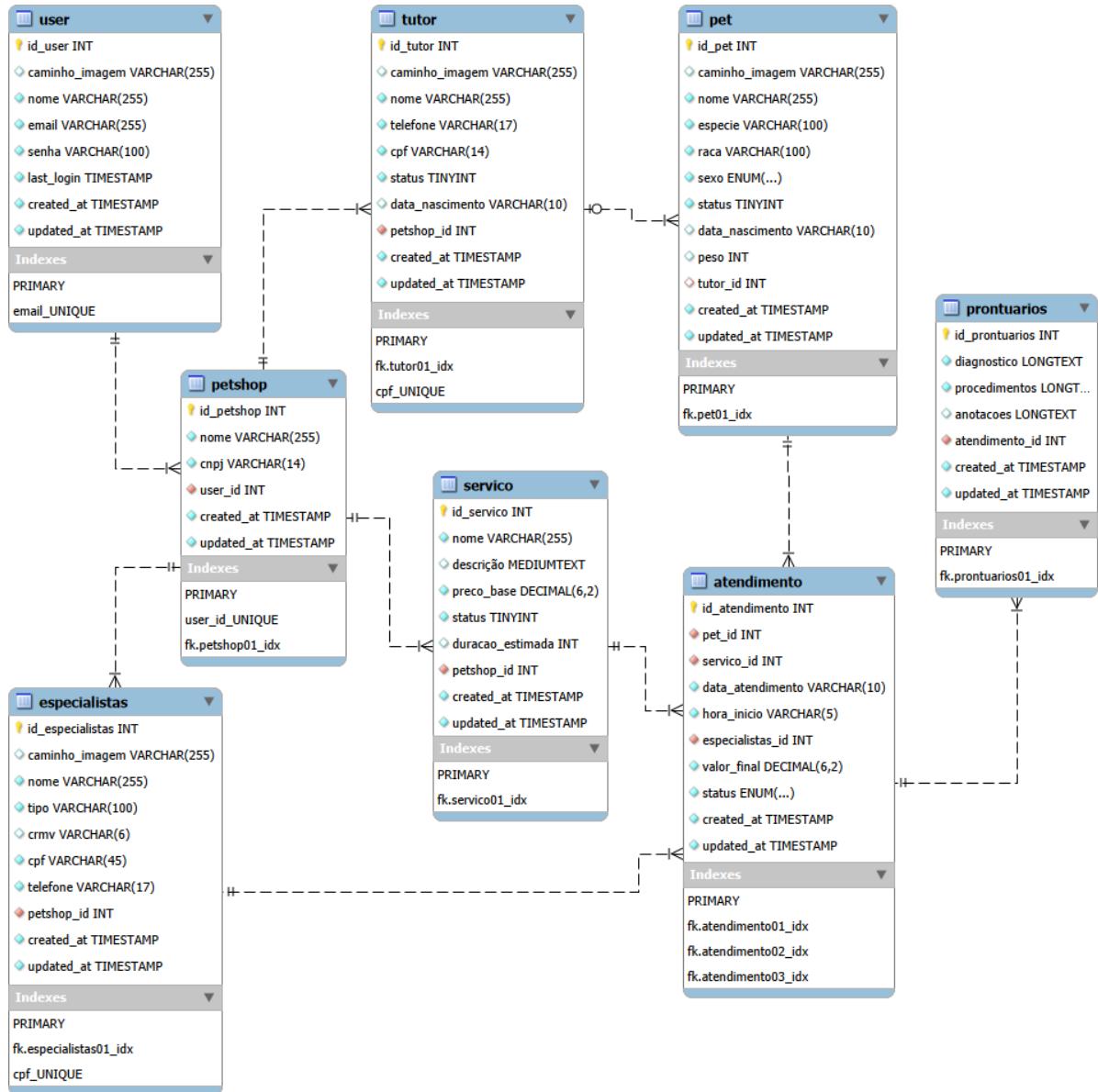
8 METODOLOGIA DE DESENVOLVIMENTO

A metodologia escolhida foi Waterfall, na qual aborda um gerenciamento de projetos em uma sequência linear. A ideia é que seus processos sigam apenas um fluxo, sendo ele de cima para baixo, sem voltar. Ou seja, cada etapa deve ser feita 100% antes de prosseguir para as próximas. O nosso fluxo será este:



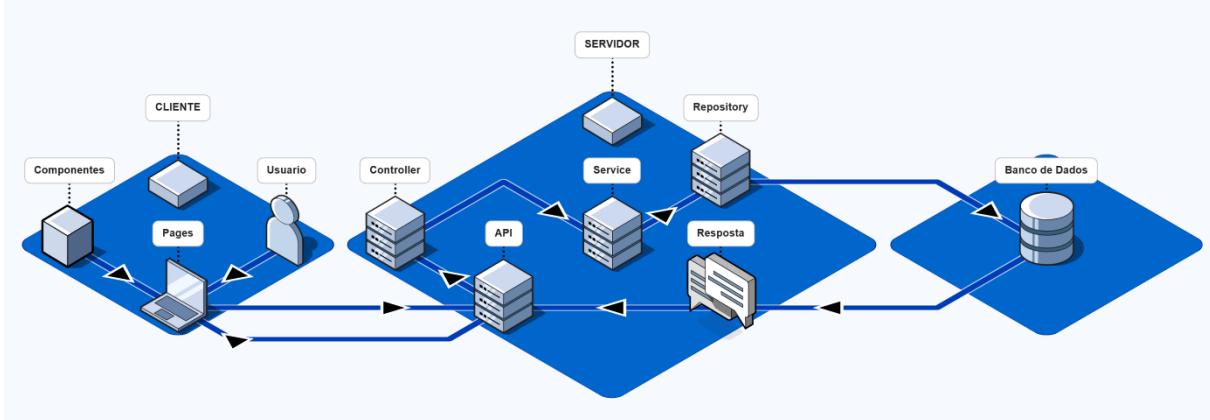
Nota de Observação: Embora eu já tenha começado o Design e a Codificação, após a escolha da metodologia, ou seja, na versão 1.0 do documento, eu sigo fielmente a metodologia Waterfall.

9 MODELO DER



10 DESIÇÃO ARQUITETURAL

O sistema utiliza a arquitetura Cliente-Servidor, onde o cliente é responsável pela interface e interação com o usuário, enquanto o servidor processa as regras do sistema e cuida da segurança dos dados. A comunicação entre cliente e servidor acontece por meio de uma API. No servidor, a aplicação é organizada em camadas para manter o código mais organizado e fácil de manter. Os dados são armazenados em um banco de dados, acessado apenas pelo servidor.



11 FRONT-END

A escolha do front-end em React foi motivada pela necessidade de construir uma interface moderna, responsiva e altamente interativa, alinhada aos requisitos funcionais e não funcionais do sistema. React permite a criação de interfaces baseadas em componentes reutilizáveis, o que facilita a manutenção e a evolução do sistema ao longo do tempo, atendendo diretamente ao requisito de manutenibilidade. Além disso, sua abordagem declarativa e eficiente na atualização da interface contribui para uma melhor experiência do usuário, essencial para operações frequentes de CRUD que devem responder em menos de dois segundos. A compatibilidade com os principais navegadores modernos e a facilidade de adaptação para dispositivos móveis também reforçam sua adequação aos requisitos de usabilidade, compatibilidade e portabilidade do projeto.

12 BANCO DE DADOS

O banco de dados MySQL foi selecionado por sua confiabilidade, desempenho e ampla adoção em sistemas transacionais que exigem integridade e consistência dos dados. Considerando que o sistema possui múltiplas entidades relacionadas, como tutores, pets, atendimentos e serviços, o modelo relacional do MySQL se mostra ideal para garantir o correto uso de chaves estrangeiras e evitar problemas de integridade referencial. Seu desempenho em operações de leitura e escrita contribui para atender às métricas de tempo de resposta exigidas, além de suportar o crescimento da base de dados sem perdas de informações, atendendo aos requisitos de confiabilidade e escalabilidade do projeto.

13 BACK-END

O back-end em Laravel foi escolhido por ser um framework robusto, maduro e amplamente utilizado no mercado para o desenvolvimento de aplicações web seguras e escaláveis. Laravel oferece uma arquitetura bem definida, alinhada ao padrão MVC, o que facilita a organização do código e a separação clara de responsabilidades, contribuindo diretamente para a manutenibilidade e confiabilidade do sistema. Seus recursos nativos para autenticação, criptografia de senhas, validação de dados e controle de acesso atendem de forma eficiente aos requisitos de segurança. Além disso, o suporte a testes automatizados e a facilidade de integração com APIs REST tornam o Laravel uma escolha adequada para garantir estabilidade, desempenho e evolução contínua do sistema.

14 VERSIONAMENTO

O versionamento de código utilizando Git com repositório no GitHub foi adotado para garantir controle, rastreabilidade e segurança no desenvolvimento do sistema. O uso de Git permite o gerenciamento eficiente das alterações no código-fonte, facilitando a identificação de erros, a implementação de correções e a inclusão de novas funcionalidades, o que está diretamente relacionado ao requisito de manutenibilidade. Já o GitHub possibilita a centralização do código, o acompanhamento da evolução do projeto e a apresentação profissional do sistema no portfólio, demonstrando boas práticas de engenharia de software, organização do código e aderência a padrões utilizados no mercado de desenvolvimento.