

Ana Rubia Schneider Sampaio

**CÃOTEIRINHA: APLICATIVO FLUTTER
PARA GESTÃO DA SAÚDE DE ANIMAIS DE
ESTIMAÇÃO**

Coxim - MS

2025

Ana Rubia Schneider Sampaio

CÃOTEIRINHA: APLICATIVO FLUTTER PARA GESTÃO DA SAÚDE DE ANIMAIS DE ESTIMAÇÃO

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal de Mato Grosso do Sul, Câmpus de Coxim, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal de Mato Grosso do Sul – UFMS

Câmpus de Coxim

Curso de Bacharelado em Sistemas de Informação

Orientador: Prof. Dr. Gedson Faria

Coxim - MS

2025

Ana Rubia Schneider Sampaio

CÃOTEIRINHA: APLICATIVO FLUTTER PARA GESTÃO DA SAÚDE DE ANIMAIS DE ESTIMAÇÃO

Trabalho de Conclusão de Curso apresentado
ao Curso de Bacharelado em Sistemas de In-
formação da Universidade Federal de Mato
Grosso do Sul, Câmpus de Coxim, como re-
quisito parcial para obtenção do título de
Bacharel em Sistemas de Informação.

Trabalho aprovado. Coxim - MS, 27 de novembro de 2025:

Prof. Dr. Gedson Faria

Orientador
UFMS

Prof^a. Dr^a. Juliana Wolf Pereira

Convidada
UFMS

Prof^a. Me. Glasielly Demomi Proença

Convidada
UFMS

Coxim - MS

2025

Aos meus pais, alicerces da minha formação.

AGRADECIMENTOS

Agradeço imensamente aos meus pais, que com amor, dedicação e renúncias tornaram possível minha formação acadêmica. Cada página deste trabalho reflete o investimento de vocês em meu futuro. A toda minha família, pelo suporte emocional e incentivo constante ao longo desta jornada.

Ao meu orientador, Prof. Dr. Gedson Faria, pela paciência, disponibilidade e valiosas contribuições que guiaram este trabalho.

Aos professores do curso de Sistemas de Informação da UFMS, Câmpus de Coxim, pelos ensinamentos compartilhados ao longo desses anos.

Aos meus amigos e colegas de turma, pela parceria, pelas longas noites de estudo e pelos momentos de descontração que aliviaram a caminhada.

“Talk is cheap. Show me the code.”

LINUS TORVALDS

RESUMO

O mercado de animais de estimação no Brasil movimentou R\$ 77,3 bilhões em 2024, consolidando-se como terceiro maior mundialmente. Nesse contexto, o controle adequado do calendário vacinal representa desafio crescente para tutores, que enfrentam dificuldades na organização de registros de vacinação e acompanhamento de datas de reforço. Este trabalho apresenta o desenvolvimento do aplicativo mobile **Cãoteirinha**, implementado em Flutter, para gestão de cadastro de animais de estimação e controle de histórico vacinal. A pesquisa adotou metodologia *Lean Startup* para validação do modelo de negócios através de *Lean Canvas*, integrando conceitos de Engenharia de Software orientada a objetos para documentação sistemática do sistema. O desenvolvimento utilizou padrões de análise e projeto orientados a objetos, incluindo diagramas de casos de uso, diagramas de classes e arquitetura MVSR. O *Lean Canvas* identificou três problemas principais: esquecimento de datas de vacinação e reforços, desorganização de registros físicos e dificuldade de compartilhamento de informações. A solução proposta integra funcionalidades de registro de vacinações com controle de datas de aplicação e reforço, perfil completo do animal, notificações push e geração de carteira digital em PDF. O modelo de negócios é direcionado a tutores de 25-45 anos e ONGs protetoras. A arquitetura desenvolvida em Flutter permite multiplataforma (iOS e Android) com backend Firebase para persistência de dados e autenticação. Os diagramas UML documentam 12 casos de uso principais e 7 classes de domínio fundamentais. A validação técnica confirma viabilidade da implementação, enquanto a análise de mercado demonstra potencial competitivo da solução. Como trabalhos futuros, propõe-se implementação para outros tipos de registros de saúde animal, testes de usabilidade com usuários reais e validação piloto para confirmação das hipóteses de negócio formuladas no *Lean Canvas*.

Palavras-chave: Sistemas de Informação. Aplicativo Mobile. Saúde Animal. *Lean Startup*. Flutter. Engenharia de Software. Orientação a Objetos.

ABSTRACT

The pet market in Brazil reached R\$ 77.3 billion in 2024, consolidating its position as the third largest worldwide. In this context, the adequate control of the vaccination schedule represents a growing challenge for pet owners, who face difficulties in organizing vaccination records and monitoring booster dates. This paper presents the development of the mobile application **Cãoteirinha**, implemented in Flutter, for managing pet registration and controlling the vaccination history. The research adopted the Lean Startup methodology for validating the business model through a Lean Canvas, integrating Object-Oriented Software Engineering concepts for systematic system documentation. The development utilized object-oriented analysis and design patterns, including Use Case Diagrams, Class Diagrams, and the MVSR architecture. The Lean Canvas identified three main problems: forgetting vaccination and booster dates, disorganization of physical records, and difficulty in sharing information. The proposed solution integrates vaccination record functionalities with application and booster date control, a complete pet profile, push notifications, and digital wallet generation in PDF. The business model is aimed at pet owners aged 25-45 and animal protection NGOs. The architecture developed in Flutter allows for a cross-platform solution (iOS and Android) with a Firebase backend for data persistence and authentication. The UML diagrams document 12 main use cases and 7 fundamental domain classes. Technical validation confirms the feasibility of the implementation, while market analysis demonstrates the solution's competitive potential. Future work includes implementing other types of animal health records, usability testing with real users, and pilot validation to confirm the business hypotheses formulated in the Lean Canvas.

Keywords: Information Systems. Mobile Application. Animal Health. *Lean Startup*. Flutter. Software Engineering. Object-Orientation.

LISTA DE ILUSTRAÇÕES

Figura 1 – Lean Canvas: Ferramenta de prototipação de startups.	19
Figura 2 – Diagrama de Casos de Uso do Sistema Cãoteirinha	33
Figura 3 – Diagrama de Classes do Domínio - Sistema Cãoteirinha	35
Figura 4 – Arquitetura Geral do Sistema Cãoteirinha	36
Figura 5 – Wireframes das Principais Telas do Cãoteirinha	43
Figura 6 – Wireframe da Carteirinha.	45
Figura 7 – Notificações Push.	46

LISTA DE TABELAS

Tabela 1 – Comparativo de funcionalidades entre aplicativos de gestão de pets. . .	21
Tabela 2 – Especificação UC01 - Cadastrar Animal	34
Tabela 3 – Estrutura das Coleções Firebase Firestore	38
Tabela 4 – Resultados dos Testes Realizados	51

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CRUD	Create, Read, Update, Delete
IA	Inteligência Artificial
IDE	Integrated Development Environment
MVSR	Model-View-Service-Repository
MVP	Minimum Viable Product
ONG	Organização Não Governamental
PDF	Portable Document Format
SDK	Software Development Kit
UFMS	Universidade Federal de Mato Grosso do Sul
UML	Unified Modeling Language
URL	Uniform Resource Locator

SUMÁRIO

1	INTRODUÇÃO	14
1.1	OBJETIVOS	15
1.2	ESTRUTURA DO TRABALHO	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	MERCADO PET E TRANSFORMAÇÃO DIGITAL	16
2.2	FRAMEWORK FLUTTER	17
2.2.1	Arquitetura Flutter	17
2.2.2	Integração com Flutter e o Firebase	17
2.2.3	Cloud Firestore: O Banco de Dados Central da Integração	18
2.3	METODOLOGIA <i>Lean Startup</i>	18
2.3.1	Lean Startup	18
2.3.2	Effectuation	19
2.4	Análise Comparativa de Aplicativos para Gestão da Saúde de Pets	21
2.4.1	Descrição da Análise	22
3	METODOLOGIA	23
3.1	METODOLOGIA <i>Lean Startup</i> APLICADA	23
3.2	PROCESSO DE ENGENHARIA DE SOFTWARE	23
3.3	FERRAMENTAS E TECNOLOGIAS	24
3.4	COLETA E ANÁLISE DE DADOS	25
4	DESENVOLVIMENTO	26
4.1	<i>Lean Canvas</i> DO CÃOTEIRINHA	26
4.1.1	Identificação e Validação de Problemas	26
4.1.2	<i>Lean Canvas</i> Completo	27
4.1.2.1	Problema	27
4.1.2.2	Segmentos de Clientes	27
4.1.2.3	Proposta de Valor	27
4.1.2.4	Solução	27
4.1.2.5	Canais	28
4.1.2.6	Fluxo de Receita	28
4.1.2.7	Estrutura de Custos	28
4.1.2.8	Métricas-Chave	29
4.1.2.9	Vantagem Injusta	29
4.1.3	Segmentação de Clientes	29

4.2	ANÁLISE E PROJETO ORIENTADOS A OBJETOS	30
4.2.1	Levantamento de Requisitos Funcionais	30
4.2.2	Requisitos Não-Funcionais	31
4.2.3	Diagrama de Casos de Uso	32
4.2.4	Especificação de Casos de Uso Principais	34
4.2.5	Diagrama de Classes	34
4.3	ARQUITETURA DO SISTEMA	36
4.3.1	Visão Geral da Arquitetura	36
4.3.2	Camadas da Arquitetura	37
4.3.3	Padrões de Projeto Aplicados	37
4.4	IMPLEMENTAÇÃO EM FLUTTER	37
4.4.1	Estrutura de Pastas	38
4.4.2	Modelagem de Dados	38
4.4.3	Implementação de Funcionalidades Core	38
4.4.4	Interface do Usuário	41
4.5	INTEGRAÇÃO COM FIREBASE	46
4.5.1	Firebase Authentication	46
4.5.2	Cloud Firestore	47
4.5.3	Firebase Cloud Messaging (FCM)	47
4.5.4	Cloud Functions	47
4.6	FUNCIONALIDADES IMPLEMENTADAS	47
4.6.1	Gestão de Tutores	48
4.6.2	Gestão de Animais	48
4.6.3	Controle de Saúde	49
4.6.4	Sistema de Notificações	49
4.6.5	Geração de Relatórios	49
4.7	TESTES E VALIDAÇÃO TÉCNICA	50
4.7.1	Testes Unitários	50
4.7.2	Testes de Integração	50
4.7.3	Testes de Interface	50
5	CONCLUSÃO	52
5.1	SÍNTESE DOS RESULTADOS	52
5.2	CONTRIBUIÇÕES DA PESQUISA	52
5.2.1	Contribuições Técnicas	52
5.2.2	Contribuições de Negócio	53
5.2.3	Contribuições Acadêmicas	53
5.3	LIMITAÇÕES DO ESTUDO	53
5.4	TRABALHOS FUTUROS	54
5.4.1	Desenvolvimento Técnico	54

5.4.2	Validação e Expansão	54
5.4.3	Pesquisas Complementares	55
5.5	CONSIDERAÇÕES FINAIS	55
	REFERÊNCIAS	57
	APÊNDICES	59
	APÊNDICE A – CÓDIGO FONTE PRINCIPAL	60
	ANEXOS	61
	ANEXO A – QUESTIONÁRIO DE VALIDAÇÃO	62
A.1	Perfil do Respondente	62
A.2	Problemas Identificados	62
A.3	Aceitação da Solução	63

1 INTRODUÇÃO

O mercado brasileiro de animais de estimação tem apresentado expansão consistente, alcançando faturamento de R\$ 77,3 bilhões em 2024 e consolidando-se como o terceiro maior do mundo ([Liga Ventures, 2024](#)). Esse crescimento reflete uma transformação significativa na relação entre humanos e animais, marcada pela humanização dos *pets*, que passam a receber cuidados especializados em saúde, alimentação e bem-estar ([Saúde Digital News, 2025](#)).

A digitalização do setor veterinário acompanha tendências globais da saúde, com aumento no uso de prontuários eletrônicos, plataformas de telemedicina e aplicativos de gestão clínica ([Mobile Med Vet, 2024](#)). Paralelamente, tutores enfrentam dificuldades para organizar informações médicas, controlar vacinações e manter históricos de saúde atualizados — lacuna que abre espaço para soluções tecnológicas inovadoras e acessíveis.

Nesse cenário, surgem oportunidades para o desenvolvimento de sistemas que integrem princípios de engenharia de software, com ênfase no paradigma orientado a objetos, e metodologias ágeis voltadas para validação rápida de hipóteses e entrega incremental de valor. A combinação entre práticas de desenvolvimento *mobile*, modelagem orientada a objetos e abordagens como *Lean Startup* permite acelerar o ciclo de desenvolvimento, aumentar a eficiência do time e alinhar o produto às reais necessidades dos usuários ([RUBIN, 2012](#); [RIES, 2011](#)).

Este trabalho apresenta o desenvolvimento do aplicativo *mobile* **CÃOTEIRINHA**, uma solução tecnológica desenvolvida em Flutter para gestão vacinal de animais de estimação. O projeto adota princípios da *Lean Startup* para validação contínua de hipóteses de negócio. Essa metodologia busca reduzir riscos, otimizar recursos e promover inovações alinhadas às necessidades reais dos tutores.

A justificativa da pesquisa fundamenta-se na recorrência de problemas enfrentados por tutores: 68% esquecem datas de vacinas e medicamentos ([DALL’STELLA et al., 2024](#)), e 74% relatam dificuldades na organização de documentos físicos de saúde animal ([Rede Globo RPC, 2025](#)). Esses desafios impactam diretamente os cuidados preventivos e a qualidade de vida dos animais, reforçando a importância de ferramentas digitais confiáveis, integradas e de fácil uso.

1.1 OBJETIVOS

Objetivo Geral

Desenvolver um aplicativo *mobile* multiplataforma em Flutter para gestão da saúde de animais de estimação, utilizando a metodologia *Lean Startup* para validação do modelo de negócios, aplicando fundamentos de engenharia de software com abordagem orientada a objetos para garantir modularidade, manutenibilidade e escalabilidade.

Objetivos Específicos

1. Identificar e validar problemas dos tutores por meio da construção de *Lean Canvas*;
2. Projetar a arquitetura de software aplicando princípios de orientação a objetos e padrão MVSR;
3. Elaborar diagramas UML para documentação de casos de uso e estrutura de classes;
4. Implementar protótipo funcional em Flutter com funcionalidades *core* do sistema;
5. Modelar base de dados Firebase para armazenamento seguro das informações;
6. Integrar serviços de notificações *push* e geração de documentos PDF;
7. Validar a proposta de valor por meio de análise comparativa de mercado;
8. Documentar o processo de desenvolvimento como referencial metodológico.

1.2 ESTRUTURA DO TRABALHO

Este trabalho está organizado em cinco capítulos. O Capítulo 1 apresenta a introdução, contextualizando o problema e os objetivos. O Capítulo 2 aborda a fundamentação teórica: mercado *pet*, fundamentos de engenharia de software, paradigma orientado a objetos, desenvolvimento em Flutter e metodologias ágeis. O Capítulo 3 descreve a metodologia adotada, detalhando a integração do *Lean Startup*. O Capítulo 4 apresenta o desenvolvimento do sistema, incluindo *Lean Canvas*, análise orientada a objetos, implementação em Flutter e documentação técnica. Por fim, o Capítulo 5 apresenta as conclusões e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 MERCADO PET E TRANSFORMAÇÃO DIGITAL

O mercado pet brasileiro tem experimentado crescimento consistente superior ao PIB nacional, impulsionado pela mudança demográfica e comportamental da população brasileira ([Liga Ventures, 2024](#)). Segundo dados do Instituto Pet Brasil, o país possui população estimada de 160,9 milhões de animais de estimação, distribuídos em 77% dos domicílios brasileiros ([Saúde Digital News, 2025](#)).

O fenômeno da humanização dos pets representa mudança paradigmática na relação humano-animal, onde os animais de estimação recebem cuidados médicos preventivos especializados, alimentação premium e produtos diferenciados ([Economia PR, 2025](#)). Esta transformação comportamental tem gerado demanda crescente por serviços digitais especializados, criando oportunidades para desenvolvimento de soluções tecnológicas inovadoras.

A transformação digital do setor veterinário brasileiro segue tendências globais, com crescente adoção de tecnologias como prontuários eletrônicos, sistemas de gestão clínica e aplicativos de telemedicina ([Mobile Med Vet, 2024](#)). Startups nacionais como Petlove, Dr. Pet e Budz têm desenvolvido soluções que combinam e-commerce, telemedicina e gestão de informações ([Petlove, 2024](#)).

A aplicação de Sistemas de Informação no contexto da saúde tem revolucionado a gestão de informações médicas, proporcionando maior eficiência, precisão e acessibilidade aos dados clínicos ([TIBES; DIAS; ZEM-MASCARENHAS, 2018](#)). No segmento veterinário, essa transformação digital ainda apresenta oportunidades significativas para inovação, especialmente em soluções direcionadas aos tutores.

Sistemas de informação em saúde animal devem contemplar funcionalidades específicas como controle vacinal automatizado, registro de medicamentos com posologia personalizada, histórico de consultas e exames, além de facilitar comunicação entre tutores e profissionais veterinários ([DALL'STELLA et al., 2024](#)). Neste sentido, a gestão informatizada de registros é crucial para a saúde pública e individual dos animais, permitindo um acompanhamento longitudinal eficiente.

2.2 FRAMEWORK FLUTTER

Flutter constitui *framework* de desenvolvimento *mobile* criado pelo Google, permitindo criação de aplicativos nativos para iOS e Android utilizando única base de código. A arquitetura Flutter baseia-se em *widgets* como blocos básicos de construção, proporcionando flexibilidade e performance superior comparado a *frameworks* híbridos tradicionais.

2.2.1 Arquitetura Flutter

O Flutter utiliza linguagem Dart e compila para código nativo, proporcionando performance equivalente a desenvolvimento nativo tradicional. A arquitetura em camadas inclui **Framework** (*widgets*, *material design*), **Engine** (renderização, *plugins*) e **Embedder** (integração com plataforma específica).

O padrão arquitetural **MVSR** (*Model-View-Service-Repository*) foi adotado para estruturar a aplicação, garantindo a separação de responsabilidades. A camada **View** trata da interface, o **Model** dos dados, o **Service** contém a lógica de negócio principal, e o **Repository** gerencia a persistência de dados (Firebase). Esta organização facilita a testabilidade, reutilização de código e manutenibilidade do sistema. O gerenciamento de estado dentro dos *widgets* da aplicação foi realizado utilizando o mecanismo nativo do Flutter, o *StatefulWidget* e *setState*, adequado para o escopo das funcionalidades implementadas.

2.2.2 Integração com Flutter e o Firebase

Firebase oferece plataforma *Backend-as-a-Service* (BaaS) que simplifica desenvolvimento de aplicativos móveis através de serviços como autenticação, banco de dados em tempo real, armazenamento de arquivos e notificações *push*. A integração nativa com Flutter acelera desenvolvimento e reduz complexidade de infraestrutura *backend*.

A sinergia entre o Firebase e o Flutter é particularmente notável. A equipe do Google mantém uma suíte de *plugins* oficiais chamada FlutterFire, que garante uma integração nativa e de alta performance entre o Flutter e os serviços do Firebase.

Essa integração nativa acelera o desenvolvimento e reduz a complexidade da infraestrutura de backend. Em vez de gerenciar endpoints de API, o desenvolvedor Flutter interage com os serviços do Firebase através de pacotes Dart bem documentados, simplificando operações de leitura, escrita e autenticação.

2.2.3 Cloud Firestore: O Banco de Dados Central da Integração

Embora o Firebase ofereça dois bancos de dados NoSQL (o Realtime Database original e o Cloud Firestore), o Cloud Firestore é a solução de última geração e a mais recomendada para a maioria das novas aplicações, devido à sua escalabilidade superior e modelo de dados mais rico.

O Firestore é um banco de dados NoSQL orientado a documentos. Diferente de bancos SQL tradicionais baseados em tabelas e linhas, o Firestore armazena dados em documentos, que são conjuntos de pares chave-valor (semelhantes a JSON) e organizados em coleções. Esse modelo oferece flexibilidade de esquema, permitindo que os desenvolvedores evoluam suas estruturas de dados sem migrações complexas.

2.3 METODOLOGIA *LEAN STARTUP*

A inovação em ambientes de alta incerteza tem sido tema de intensa investigação nas últimas décadas. Dois marcos conceituais se destacam: a metodologia *Lean Startup* de Ries e a teoria da *Effectuation* de Sarasvathy. Embora oriundos de campos distintos — empreendedorismo e tomada de decisão sob incerteza —, ambos fornecem arcabouços complementares para guiar a criação e evolução de novos empreendimentos.

2.3.1 Lean Startup

O paradigma da Lean Startup, popularizado por [Ries](#), propõe uma abordagem científica para a criação e gestão de startups, focando na redução do desperdício de tempo e recursos. A metodologia enfatiza o aprendizado validado através de um ciclo iterativo de experimentação rápida, conhecido como Construir-Medir-Aprender ([RIES, 2011](#)).

Para operacionalizar esta abordagem, [Maurya \(2012\)](#) propôs o Lean Canvas, uma ferramenta que adapta o Business Model Canvas original focando-o na velocidade e na gestão das incertezas. A principal contribuição de [Maurya](#) foi substituir blocos orientados à execução por elementos centrados no risco, notadamente: Problema (substituindo Parceiros-Chave), Solução (substituindo Atividades-Chave), Métricas Chave (substituindo Recursos-Chave) e Vantagem Injusta (substituindo Relacionamento com Clientes). A [Figura 1](#) ilustra a estrutura deste modelo.

O Lean Canvas não deve ser compreendido como um plano de negócios estático, mas sim como um artefato dinâmico para a gestão de hipóteses e mitigação de riscos ([MAURYA, 2012](#)). Cada um dos nove blocos representa uma suposição fundamental que requer validação empírica. O preenchimento do *canvas* é, portanto, o ponto de partida para a execução do ciclo de aprendizado validado, permitindo ao empreendedor priorizar



Figura 1 – Lean Canvas: Ferramenta de prototipação de startups.

os testes das suposições mais críticas — comumente as relacionadas aos blocos Problema e Segmentos de Clientes (RIES, 2011).

A arquitetura visual do Lean Canvas, detalhada na Figura 1, é intencionalmente dividida para refletir os dois eixos centrais de risco. Os blocos à esquerda do diagrama — Problema, Solução e Métricas-Chave — estão associados ao *Risco de Produto*, focando no desafio de desenvolver a solução adequada. Em contrapartida, os blocos à direita — Segmentos de Clientes, Proposta de Valor e Canais — endereçam o *Risco de Mercado*, concentrando-se em encontrar um caminho viável para os clientes. Esta arquitetura dual, que também integra a Vantagem Injusta como um diferenciador competitivo crucial e é sustentada pela análise de viabilidade financeira (representada nos blocos inferiores Estrutura de Custos e Fontes de Receita), assegura que o modelo de negócios seja avaliado de forma holística desde sua concepção.

Aspectos modernos incorporam Inteligência Artificial Generativa para acelerar experimentos e personalizar interações com clientes. Por exemplo, *chatbots* baseados em *Large Language Models* (LLMs) permitem validar hipóteses de demanda por meio de conversas automatizadas (PARK; SUBRAMONYAM; KULKARNI, 2024); geração automática de protótipos de interface reduz o tempo de desenvolvimento do Produto Mínimo Viável (MVP) (SMITH, 2025). Esses avanços mantêm o ciclo enxuto, mas ampliam a escala e a velocidade de aprendizagem.

2.3.2 Effectuation

A teoria da *Effectuation*, proposta por Sarasvathy (2001), descreve o processo decisório de empreendedores experientes em ambientes incertos. Cinco princípios norteiam

a abordagem:

Bird-in-Hand (Pássaro na Mão): *Parta dos recursos disponíveis.*

O princípio do *Bird-in-Hand* propõe que o empreendedor não deve esperar pela “grande oportunidade” ideal ou por condições perfeitas de mercado. Em vez disso, ele inicia o processo de criação com o que já possui — seus recursos atuais, competências e rede de relacionamentos. O foco está em agir com base nos meios disponíveis, construindo oportunidades de forma progressiva e prática.

Para identificar esses meios, o empreendedor analisa três dimensões fundamentais:

1. **Quem eu sou:** refere-se à identidade pessoal — valores, crenças, motivações, paixões e traços de personalidade que moldam o modo como o empreendedor percebe e reage ao mundo.
2. **O que eu sei:** corresponde ao conjunto de conhecimentos e habilidades adquiridos por meio da formação acadêmica, experiência profissional e vivências anteriores.
3. **Quem eu conheço:** envolve a rede de contatos — pessoas e instituições com as quais o empreendedor mantém relações de confiança, podendo gerar parcerias, mentorias, clientes iniciais ou fontes de recursos.

A partir da combinação desses três elementos, o empreendedor transforma o que tem em mãos em possibilidades reais de negócio, ajustando o curso conforme aprende e interage com o ambiente.

Affordable Loss (Perda Aceitável): *Limite riscos definindo o prejuízo aceitável.*

Empreendedores experientes focam em quanto estão dispostos a perder, e não no retorno potencial esperado. Eles definem um limite de perda aceitável (em termos de tempo, dinheiro ou esforço) para cada experimento. Isso permite testar ideias sem arriscar a falência, tornando o fracasso uma fonte de aprendizado e não um desastre.

Crazy-Quilt (Colcha de Retalhos): *Estabeleça parcerias que co-criem valor.*

Em vez de realizar uma análise competitiva, o empreendedor busca proativamente parceiros e *stakeholders* auto-selecionados. Estes são indivíduos ou organizações que se comprometem com o projeto, trazendo seus próprios recursos e meios. A “colcha” (o empreendimento) é costurada em conjunto por esses parceiros, e o resultado final é moldado por quem decide participar.

Lemonade (Limonada): *Aproveite contingências favoráveis (e desfavoráveis).*

Este princípio encoraja a ver surpresas, erros e eventos inesperados como oportunidades, e não como desvios de um plano. O inesperado (seja bom ou ruim) fornece novas

informações e pode abrir portas para mercados ou produtos radicalmente novos. É a habilidade de “fazer uma limonada com os limões” que a vida oferece.

Pilot-in-the-Plane (Piloto no Avião): *Concentre-se em ações controláveis.*

Este princípio é o antídoto para a futurologia. O futuro não é algo a ser previsto ou descoberto; é algo a ser criado. O empreendedor atua como um piloto que, embora não possa controlar o clima (fatores externos), pode controlar as ações do avião (o empreendimento). O foco está em criar o futuro através de ações e decisões próprias, e não em tentar adivinhar o que vai acontecer.

Comparada ao modelo causal clássico, que define objetivos e busca meios, a *effectuation* enfatiza a flexibilidade e a contingência. Adaptar-se a resultados inesperados e reconfigurar objetivos alinha-se fortemente à iteração proposta pelo *Lean Startup*.

2.4 ANÁLISE COMPARATIVA DE APLICATIVOS PARA GESTÃO DA SAÚDE DE PETS

O mercado de aplicativos para gestão de pets no Brasil é vasto, mas a maioria se concentra em uma combinação de gestão de saúde, lembretes de rotina, e funcionalidades sociais ou comerciais. Para posicionar o Cãoteirinha, foi realizada uma análise dos principais concorrentes que oferecem soluções semelhantes à proposta.

A Tabela 1 demonstra o posicionamento estratégico do Cãoteirinha em relação aos seus concorrentes diretos.

Tabela 1 – Comparativo de funcionalidades entre aplicativos de gestão de pets.

Funcionalidade	PetPrev	Flockr	PetZillas	Cãoteirinha
Carteira de Saúde Digital	Sim	Sim	Sim	Sim
Lembretes de Vacina.	Sim	Sim	Sim	Sim
Registro de Consultas/Exames	Sim	Sim	Sim	Não
Controle de Peso e Rotina	Sim	Sim	Sim	Não
Controle Financeiro	Sim	Sim	Não	Não
Plano de Saúde Pet	Não	Sim	Não	Não
Recurso de Rede Social	Não	Sim	Não	Não
Compartilhamento Vet.	Sim	Sim	Sim	Sim

2.4.1 Descrição da Análise

1. **PetPrev:** Apresenta-se como uma carteira digital focada na organização de vacinas, remédios e histórico, incorporando um módulo de controle de gastos. Seu diferencial reside na integração da gestão de saúde com a gestão financeira do pet, facilitando o acompanhamento de despesas diárias como banho e tosa.
2. **Flockr:** É o aplicativo mais abrangente e multifuncional. Além de oferecer uma completa carteira de saúde com prontuário médico e lembretes, o Flockr se destaca por sua forte vertente social (sendo uma das maiores redes sociais para pets) e comercial, oferecendo Planos de Saúde Pet e Clubes de Benefícios. Sua ampla gama de recursos o torna uma "super-app", mas também adiciona complexidade ao seu escopo.
3. **PetZillas:** Possui um foco robusto na saúde, oferecendo controle detalhado de histórico veterinário, lembretes de rotinas de higiene, e a funcionalidade de acesso por parte do médico veterinário para atualização de informações. Embora seja menos focado em finanças, possui funcionalidades de monitoramento físico e de rotina de higiene (banhos, cio).
4. **Cãoteirinha (Proposta):** O Cãoteirinha se insere nesse cenário priorizando a experiência do usuário (UX) e a gestão pura da saúde. Enquanto o Flockr se dispersa em funcionalidades sociais e comerciais e o PetPrev se concentra em custos, o Cãoteirinha busca ser uma solução leve, intuitiva e dedicada a garantir que o tutor nunca se esqueça de uma data crucial de saúde. Seu objetivo é oferecer uma carteira de vacinação digital e lembretes precisos, minimizando a curva de aprendizado e focando no controle eficiente de informações vitais para o bem-estar do pet, sem as distrações de uma rede social ou módulos de e-commerce.

Conclui-se que, embora existam soluções maduras no mercado, há espaço para um aplicativo que entregue a gestão da saúde de forma mais direta, eficaz e com um foco específico na tecnologia **Flutter**, garantindo desempenho e facilidade de manutenção em múltiplas plataformas.

3 METODOLOGIA

A estratégia metodológica baseia-se em estudo de caso único, focando no desenvolvimento do aplicativo Cãoteirinha como startup no segmento pet tech, integrando conceitos de Sistemas de Informação e Engenharia de Software. A escolha por pesquisa aplicada justifica-se pelo objetivo de desenvolver solução prática para problemas reais identificados no mercado pet brasileiro.

3.1 METODOLOGIA *LEAN STARTUP* APLICADA

O desenvolvimento seguiu rigorosamente os princípios da metodologia *Lean Startup*, implementando ciclos iterativos Construir-Medir-Aprender para validação contínua de hipóteses de negócio. Esta abordagem permitiu identificação sistemática de problemas reais enfrentados pelos tutores e desenvolvimento de soluções direcionadas às suas necessidades específicas.

O processo iniciou-se com fase exploratória de identificação de problemas por meio de pesquisa com tutores de animais e análise de soluções existentes no mercado. Posteriormente, foram formuladas hipóteses sobre problemas prioritários e soluções potenciais, organizadas sistematicamente através do *framework Lean Canvas*.

A validação das hipóteses incluiu análise comparativa de aplicativos concorrentes, estudo de tendências do mercado pet e levantamento de requisitos funcionais baseados nos problemas identificados. Esta fase fundamentou as decisões de design e implementação do sistema.

3.2 PROCESSO DE ENGENHARIA DE SOFTWARE

O processo de desenvolvimento do sistema foi estruturado em etapas sequenciais e iterativas, seguindo uma abordagem orientada a objetos e alinhada a princípios de desenvolvimento ágil. Cada fase teve objetivos específicos e entregáveis claramente definidos, garantindo coerência entre o planejamento, a implementação e a validação do produto final.

1. *Levantamento de requisitos*: etapa inicial destinada à identificação e especificação dos requisitos funcionais e não funcionais do sistema. A definição das funcionalidades prioritárias foi orientada pelo *Lean Canvas*, permitindo alinhar os aspectos técnicos às necessidades do público-alvo e ao modelo de negócio proposto.

2. *Análise orientada a objetos*: consistiu na identificação das principais classes, objetos e relacionamentos presentes no domínio do problema. Essa análise possibilitou compreender a estrutura lógica do sistema, estabelecendo a base conceitual para as etapas de projeto e implementação.
3. *Projeto do sistema*: nessa fase foram definidos a arquitetura geral, os padrões de projeto (*design patterns*) e as interfaces de comunicação entre os componentes. A ênfase esteve na modularidade, escalabilidade e reutilização de código, assegurando aderência aos princípios de engenharia de software.
4. *Modelagem UML*: elaboração de diagramas de casos de uso e classes, com o objetivo de representar graficamente as interações do sistema. Esses diagramas serviram como artefatos de comunicação e documentação técnica, apoiando a implementação e a manutenção futura do software.
5. *Implementação*: desenvolvimento do sistema utilizando o *framework* Flutter e a linguagem Dart. O backend foi estruturado com o serviço *Firebase*, fornecendo autenticação, armazenamento em nuvem e envio de notificações. A codificação seguiu boas práticas de clareza, legibilidade e versionamento contínuo.
6. *Integração*: conexão e validação entre os módulos internos e os serviços externos utilizados. Essa fase garantiu a interoperabilidade dos componentes, bem como a consistência dos fluxos de dados entre as diferentes camadas do sistema.

3.3 FERRAMENTAS E TECNOLOGIAS

O desenvolvimento do sistema utilizou um conjunto de ferramentas e tecnologias contemporâneas, selecionadas para maximizar a produtividade, a consistência e a qualidade do código produzido. A integração entre esses recursos favoreceu um fluxo de trabalho colaborativo e eficiente.

- Flutter SDK: *framework* principal utilizado para o desenvolvimento multiplataforma de aplicativos móveis nativos.
- Dart: linguagem de programação orientada a objetos empregada para a codificação do aplicativo no Flutter.
- Firebase: plataforma *Backend-as-a-Service* (BaaS) empregada para autenticação, banco de dados em tempo real, armazenamento e envio de notificações.
- Android Studio: ambiente de desenvolvimento integrado (IDE) utilizado na codificação e depuração do projeto.

- Git: sistema de controle de versão adotado para gerenciamento de histórico.
- Figma: ferramenta de prototipação de interfaces, utilizada na concepção do design visual e da experiência do usuário (*UI/UX*).
- Canva: plataforma empregada para a criação do *Lean Canvas*, permitindo a visualização sintética da proposta de valor e da estratégia do produto.

3.4 COLETA E ANÁLISE DE DADOS

A pesquisa utilizou abordagem mista, combinando dados primários e secundários para garantir consistência metodológica e validade científica.

Os *dados primários* foram obtidos por questionários com tutores de animais, entrevista com veterinário e análise de avaliações de aplicativos similares. Essa coleta possibilitou compreender hábitos, expectativas e dificuldades no cuidado com a saúde animal.

Os *dados secundários* incluíram relatórios setoriais, artigos científicos, estudos de mercado e informações de órgãos públicos como IBGE, MAPA, ANVISA e IBAMA. Esses dados forneceram indicadores demográficos, econômicos e sanitários que contextualizaram os resultados empíricos.

A análise integrou métodos qualitativos: aplicou-se *análise de conteúdo* para identificar padrões e percepções, e estatística descritiva para mensurar tendências. A triangulação dos resultados assegurou *validade*, *confiabilidade* e *consistência* das conclusões.

4 DESENVOLVIMENTO

4.1 *LEAN CANVAS* DO CÃOTEIRINHA

O desenvolvimento iniciou com construção sistemática do *Lean Canvas* para validação do modelo de negócios. A metodologia seguiu processo iterativo de identificação de problemas, formulação de hipóteses e refinamento baseado em feedback de potenciais usuários.

4.1.1 Identificação e Validação de Problemas

A pesquisa inicial identificou três problemas principais enfrentados pelos tutores na gestão da saúde de seus animais de estimação:

1. **Esquecimento de datas importantes:** Tutores frequentemente esquecem datas de vacinas de reforço, administração de medicamentos em horários específicos e agendamentos de consultas veterinárias
2. **Desorganização de registros físicos:** Informações de saúde mantidas em carteiras impressas, receitas médicas e resultados de exames ficam dispersas e vulneráveis a perdas
3. **Dificuldade de compartilhamento:** Ausência de mecanismo eficiente para compartilhar informações atualizadas com veterinários, especialmente em emergências

A validação destes problemas foi conduzida por meio de uma investigação exploratória, que incluiu questionário de validação com tutores de animais (n=93) e uma consulta especializada com uma médica veterinária. Esta abordagem quantitativa confirmou a percepção de que o esquecimento de datas de rotina e a desorganização de registros físicos é extremamente recorrente.

Já a abordagem qualitativa foi corroborada pela médica veterinária consultada, que trouxe uma perspectiva profissional sobre as consequências da desorganização dos tutores. Segundo a especialista, "É frequente o tutor chegar e nem saber qual o último protocolo de vacina ou de vermífugo foi feito. A maioria das carteirinhas que chegam aqui está rasurada, sem data clara, ou o tutor simplesmente perdeu". Ela também destacou que a desorganização afeta diretamente a saúde, afirmando que "É comum vermos pets que só relembram o tutor de uma vacina na hora que ele precisa de um atestado de saúde ou quando a doença já está instalada", confirmando que o esquecimento das rotinas preventivas é uma falha crítica na gestão do bem-estar animal.

4.1.2 *Lean Canvas* Completo

O Lean Canvas foi utilizado como ferramenta estratégica para mapear, de forma concisa e visual, as hipóteses centrais do modelo de negócios do projeto **Caoteirinha**. Esta seção apresenta a descrição detalhada dos nove blocos do Canvas, que sintetizam desde o problema a ser resolvido e a solução proposta, até a estrutura de custos, o fluxo de receita e as métricas-chave para o sucesso da iniciativa.

4.1.2.1 Problema

O projeto Cãoteirinha busca resolver desafios centrais na gestão da saúde de animais de estimação, identificados como:

- Desorganização das doses de vacinas, podendo gerar atrasos e doenças nos animais.
- Esquecimento dos dados importantes da vacinação.
- Desorganização de registros, físicos (caderneta de vacinação, receitas, exames).
- Dificuldade de compartilhamento de informações com veterinários, especialmente em emergências.

4.1.2.2 Segmentos de Clientes

O projeto visa a adoção prioritária em segmentos **multi-dono** e/ou com **maior engajamento e problemas**.

- **Segmento Primário:** Tutores de **cães e gatos** na **faixa etária 25-45 anos**, com renda superior à R\$4.000.
- **Segmento Secundário:** ONGs e protetores independentes de cães e gatos.

4.1.2.3 Proposta de Valor

A proposta de valor central é desmistificar e desburocratizar a rotina de registros, lembretes e consultas, trazendo mais saúde e bem-estar aos animais por meio da **Centralização de informações** de saúde pet com funcionalidades inteligentes de lembrete e agendamento para **personalização** de soluções focadas apenas em **cães e gatos** ou **referências médicas**.

4.1.2.4 Solução

A solução proposta se materializa em um **Aplicativo** com as seguintes funcionalidades:

- Agenda inteligente com lembretes automáticos de vacinação.
- Perfil completo do animal com todas as informações centralizadas.
- Carteira de vacinação em PDF compartilhável.

4.1.2.5 Canais

Os meios de aquisição e conversão de usuários incluem:

- **Redes sociais** (*Instagram, Facebook, TikTok*) com **conteúdo educativo** sobre saúde pet.
- **Marketing de conteúdo** (*Blog e YouTube*).
- **Parcerias** estratégicas com **clínicas veterinárias e pet shops**.
- **Otimização** para *App Store* e *Google Play* (*ASO*).
- **Indicação boca a boca e programa de referências**.
- **Participação em eventos pet e feiras do setor**.

4.1.2.6 Fluxo de Receita

As fontes e mecanismos de monetização do projeto envolvem:

- **Aplicativo gratuito** com funcionalidades essenciais para todos os usuários.
- Possibilidade futura de **recursos premium** (relatórios avançados, múltiplos pets, limitados).
- Possível parceira com clínicas veterinárias.
- **Publicidade** não invasiva de produtos e serviços relacionados.

4.1.2.7 Estrutura de Custos

Os principais custos fixos e variáveis, essenciais para a operação, abrangem:

- **Infraestrutura e licenças** (automação, banco de dados, armazenamento, notificações).
- **Desenvolvimento e manutenção** do aplicativo (equipe técnica).
- **Marketing digital** e aquisição de usuário.
- **Suporte** ao cliente e atendimento.
- **Custos legais e compliance** (*LGPD*, termos de uso).

4.1.2.8 Métricas-Chave

O sucesso do projeto será medido por indicadores objetivos relacionados à saúde, engajamento e recorrência:

- Número de **usuários ativos mensais**.
- Taxa de **retenção** em **30, 60 e 90 dias**.
- Volume médio de **vacinas cadastradas** por usuário.
- Taxa de **conversão** de usuários para cadastros.
- **Engajamento** com **notificações** (taxa de abertura).
- **Tempo médio** de **permanência** no aplicativo.

4.1.2.9 Vantagem Injusta

Os diferenciais competitivos que conferem vantagem injusta ao modelo de negócios incluem:

- Foco exclusivo em gestão de saúde preventiva.
- Interface intuitiva com design **Cupertino (estilo iOS)**.
- **Aplicativo multi-plataforma** (iOS + Android) desenvolvido em **Flutter**.
- **Integração nativa com Firebase** para sincronização em tempo real.

4.1.3 Segmentação de Clientes

O segmento primário compreende tutores de cães e gatos, faixa etária 25-45 anos, renda familiar superior a R\$ 4.000, que utilizam regularmente aplicativos móveis e tratam seus animais como membros da família. Este público está inserido em um universo total de **47,9 milhões de domicílios brasileiros** que possuem cães ou gatos ([INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA \(IBGE\), 2020](#)).

O segmento secundário inclui ONGs e protetores independentes que administram múltiplos animais simultaneamente, necessitando organização centralizada para substituir controles manuais em planilhas. Este mercado representa cerca de **400 organizações cadastradas no país** ([PET CONECTA DIGITAL, 2025](#)).

4.2 ANÁLISE E PROJETO ORIENTADOS A OBJETOS

A análise orientada a objetos identificou sistematicamente os elementos do domínio da aplicação, seus atributos, comportamentos e relacionamentos. O processo seguiu metodologia estruturada para garantir completude e consistência da modelagem.

4.2.1 Levantamento de Requisitos Funcionais

Os requisitos funcionais foram organizados em cinco categorias principais:

Gestão de Usuários (RF01-RF06):

- RF01: O sistema deve permitir cadastro de novos tutores.
- RF02: O sistema deve autenticar usuários via e-mail e senha.
- RF03: O sistema deve permitir recuperação de senha via e-mail.
- RF04: O sistema deve permitir ao tutor visualizar e editar seus dados pessoais.
- RF05: O sistema deve permitir ao tutor excluir permanentemente sua conta do sistema.
- RF06: Ao excluir a conta de um tutor (RF05), o sistema deve excluir automaticamente todos os pets e registros de vacinação associados a este usuário.

Gestão de Animais (RF07-RF011):

- RF07: O sistema deve permitir cadastro de múltiplos animais por tutor.
- RF08: O sistema deve armazenar dados básicos do animal (nome, espécie, RGA, microchip, cor, peso, raça, sexo, data de nascimento e status de castração).
- RF09: O sistema deve permitir o upload e a visualização de uma foto de perfil para cada animal.
- RF010: O sistema deve permitir ao tutor visualizar e editar os dados cadastrais de cada animal.
- RF011: O sistema deve permitir ao tutor excluir permanentemente o registro de um animal.

Gestão de Saúde (RF012-RF17):

- RF12: O sistema deve permitir o registro de uma nova vacinação, incluindo nome e marca da vacina, data de aplicação e data de reforço prevista.

- RF13: Se informada a data da dose de reforço (RF12), o sistema deve criar automaticamente um novo item na lista de “Próximas Vacinações”.
- RF14: O sistema deve exibir claramente as “Próximas Vacinações” que foram agendadas e ainda não foram aplicadas.
- RF15: O sistema deve permitir ao tutor marcar um item da lista de “Próximas Vacinações” como “Aplicada”.
- RF16: Após ser marcada como “Aplicada” (RF15), a vacinação deve ser movida automaticamente para o “Histórico de Vacinações” do pet, registrando a data de aplicação real.
- RF17: O sistema deve manter e exibir o histórico completo de vacinações para cada animal, contendo apenas os registros marcados como aplicados.

Notificações (RF18-RF19):

- RF18: O sistema deve gerar lembretes (notificações) para o tutor sobre as vacinações agendadas (RF12) que estão com a data prevista próxima.
- RF19: O sistema deve enviar os lembretes de notificação com os seguintes intervalos antes da data agendada: 10 dias, 5 dias, 1 dia antes e no próprio dia da vacinação.

Relatórios e Compartilhamento (RF20-RF22):

- RF20: O sistema deve gerar a carteira de vacinação de cada animal em formato PDF.
- RF21: O sistema deve permitir o download imediato do arquivo PDF (RF20) para o dispositivo do tutor.
- RF22: O sistema deve permitir o compartilhamento direto do arquivo PDF (RF20) via plataformas de comunicação como WhatsApp e e-mail.

4.2.2 Requisitos Não-Funcionais

Os requisitos não-funcionais foram categorizados em aspectos de performance, segurança, usabilidade e compatibilidade:

- **RNF01 - Performance:** Tempo de resposta inferior a 2 segundos para operações básicas
- **RNF02 - Disponibilidade:** Sistema disponível 99,5% do tempo

- **RNF03 - Segurança:** Criptografia de dados sensíveis em trânsito e repouso
- **RNF04 - Usabilidade:** Interface intuitiva para usuários com baixa alfabetização digital
- **RNF05 - Compatibilidade:** Suporte a iOS 12+ e Android 8.0+
- **RNF06 - Escalabilidade:** Suporte a até 2.000 usuários concorrentes

4.2.3 Diagrama de Casos de Uso

A [Figura 2](#) apresenta o diagrama de casos de uso principal do sistema, identificando 15 casos de uso fundamentais organizados por ator.

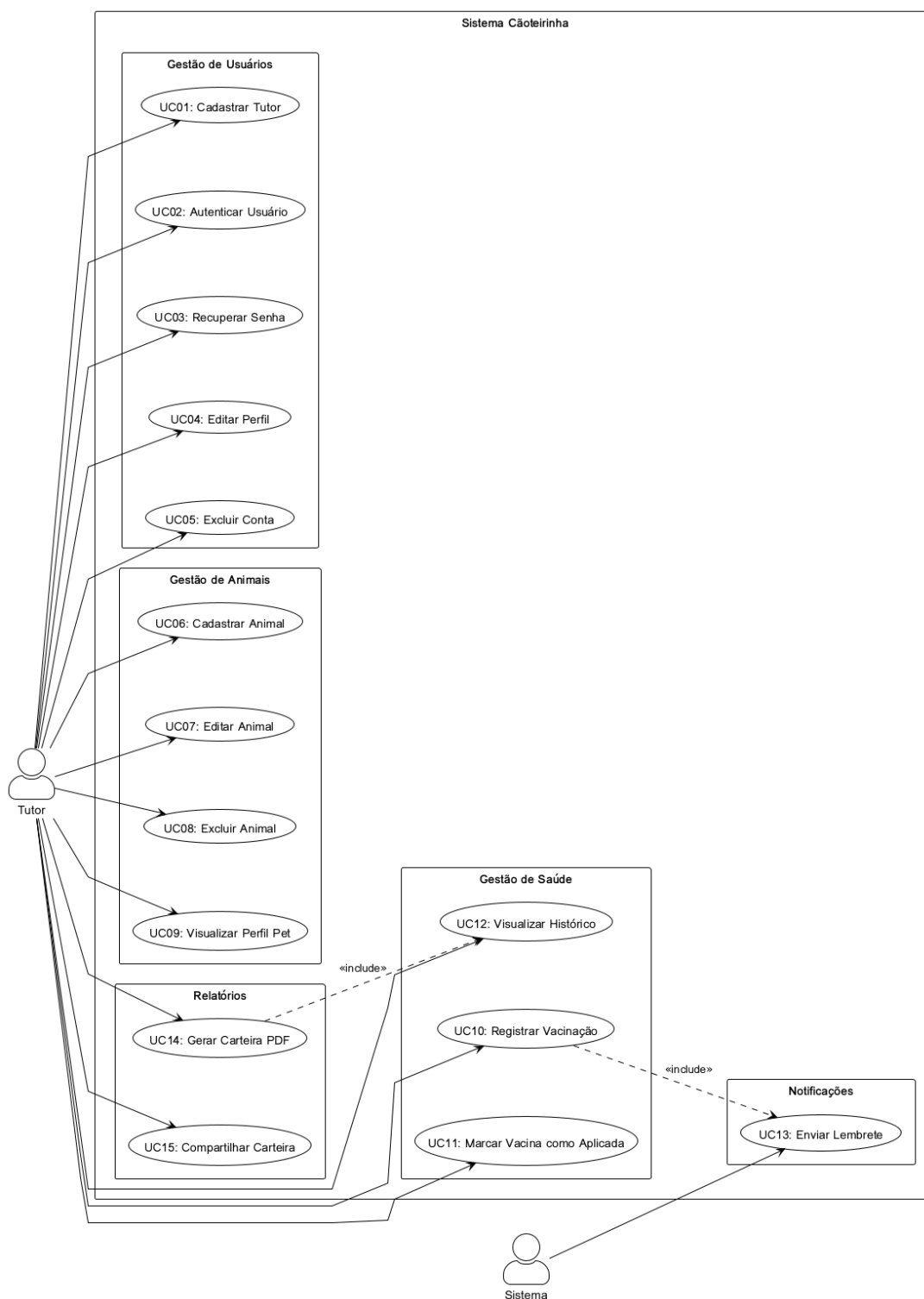


Figura 2 – Diagrama de Casos de Uso do Sistema Cãoeirinha

O diagrama identifica dois atores principais: Tutor (usuário humano) e Sistema (processos automatizados). Os casos de uso foram priorizados baseados na frequência de utilização e importância para resolução dos problemas identificados no *Lean Canvas*.

4.2.4 Especificação de Casos de Uso Principais

A [Tabela 2](#) apresenta especificação detalhada do caso de uso “Cadastrar Animal”, considerado fundamental para funcionamento do sistema.

Tabela 2 – Especificação UC01 - Cadastrar Animal

Campo	Descrição
Nome	UC01 - Cadastrar Animal
Ator Principal	Tutor
Pré-condições	Usuário autenticado no sistema
Pós-condições	Animal cadastrado e disponível para gestão de saúde
Fluxo Principal	1. Tutor acessa opção “+” 2. Sistema apresenta formulário de cadastro 3. Tutor informa dados básicos do animal 4. Tutor opcionalmente adiciona foto 5. Sistema valida dados obrigatórios 6. Sistema salva animal na base de dados 7. Sistema direciona para tela da carteira com os dados do animal preenchidos
Fluxos Alternativos	FA01 - Dados inválidos: Sistema exibe mensagem de erro FA02 - Falha na conexão: Sistema exibe mensagem de erro

4.2.5 Diagrama de Classes

O diagrama de classes representa estrutura estática do sistema, identificando classes principais, atributos, métodos e relacionamentos. A [Figura 3](#) apresenta modelo conceitual das classes de domínio.

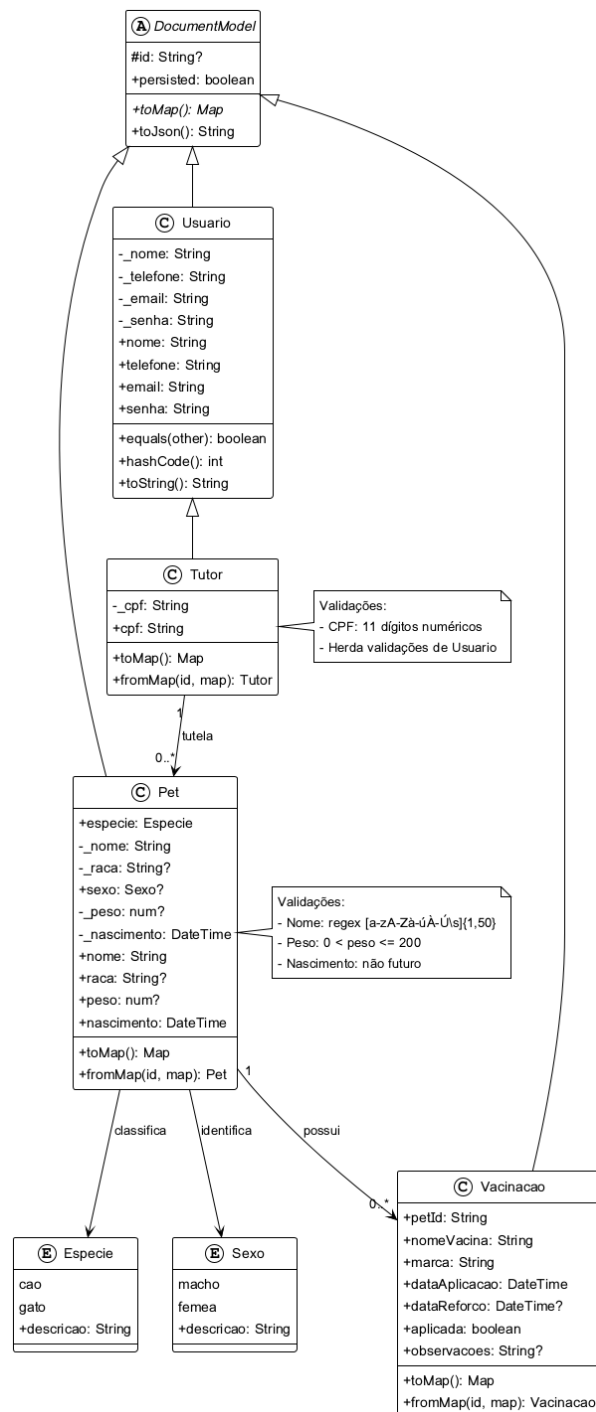


Figura 3 – Diagrama de Classes do Domínio - Sistema Cãoteirinha

O modelo identifica as classes principais do domínio organizadas em packages. As classes seguem princípios de coesão e baixo acoplamento, facilitando manutenção e extensibilidade do sistema.

4.3 ARQUITETURA DO SISTEMA

O Cãoteirinha adota uma arquitetura Model-View-Service-Repository, típica de aplicações Flutter. Os models representam os dados, as views compõem a interface do usuário, os services concentram a lógica de negócio e os repositories realizam o acesso ao backend (Firebase). O gerenciamento de estado é feito com StatefulWidget e setState.

4.3.1 Visão Geral da Arquitetura

A Figura 4 apresenta visão de alto nível da arquitetura, mostrando interação entre camadas e componentes principais.

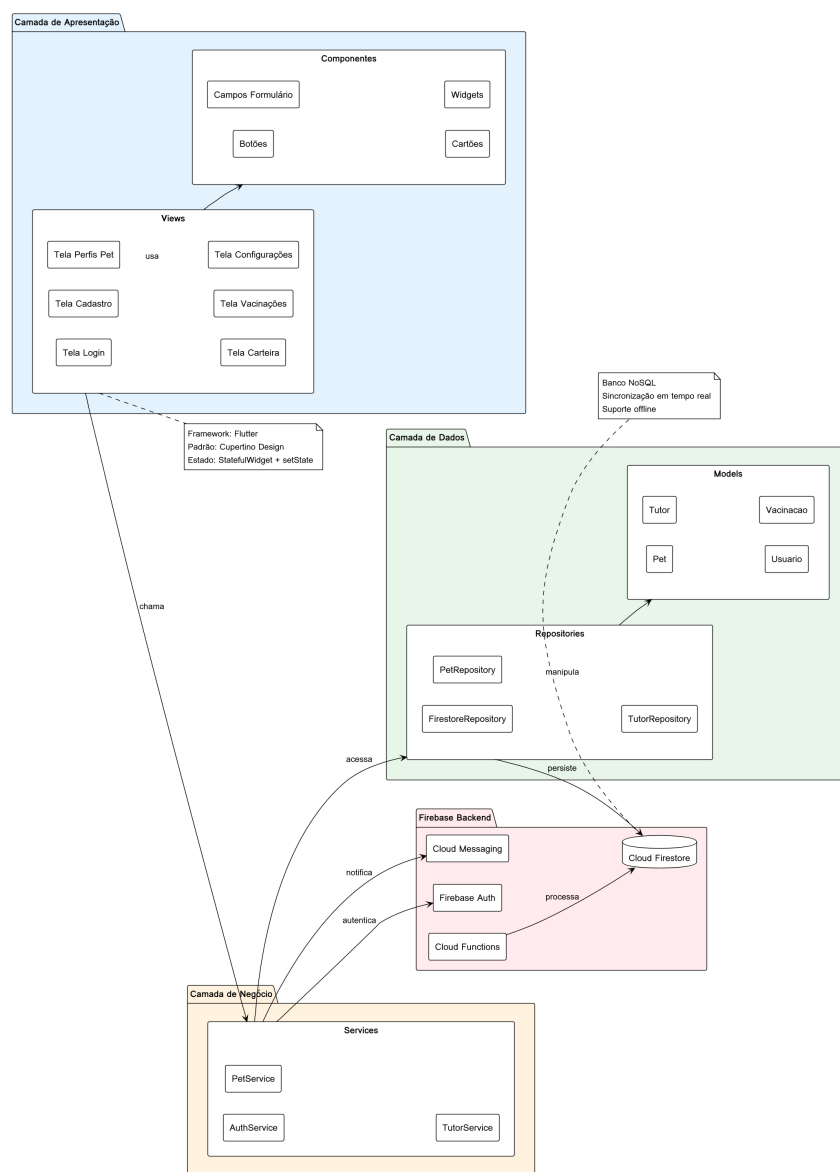


Figura 4 – Arquitetura Geral do Sistema Cãoteirinha

4.3.2 Camadas da Arquitetura

Camada de Apresentação (Presentation Layer): Implementada em Flutter, responsável pela interface do usuário e interações. Inclui widgets, telas e componentes visuais organizados seguindo o estilo Cupertino.

Camada de Negócio (Business Layer): Implementa as regras de negócio por meio dos Services, que processam dados, validam informações e coordenam operações entre a interface e a camada de dados. Os Services mantêm a lógica separada das views, facilitando a organização e a testabilidade do código.

Camada de Dados (Data Layer): Responsável por persistência e recuperação de dados através de repositories que abstraem acesso ao Firebase. Implementa padrões Repository para isolar a fonte de dados e facilitar a manutenção e testabilidade da aplicação.

Camada de Serviços (Services Layer): Integra serviços externos como notificações push (Firebase Cloud Messaging), geração de PDF e autenticação.

4.3.3 Padrões de Projeto Aplicados

Repository Pattern: Abstrai acesso a dados, permitindo trocar implementação sem afetar camadas superiores. Facilita testabilidade através de mock repositories.

Singleton Pattern: Utilizado para serviços únicos como gerenciador de autenticação e configurações globais da aplicação.

Factory Pattern: Cria instâncias de classes baseadas em parâmetros dinâmicos, especialmente útil para geração de diferentes tipos de relatórios. Empregado na criação de instâncias de models a partir de mapas dinâmicos, como nos métodos fromMap dos models.

4.4 IMPLEMENTAÇÃO EM FLUTTER

O desenvolvimento utilizou Flutter 3.29.2 com Dart 3.7.2, garantindo suporte às funcionalidades mais recentes do framework. A estrutura do projeto segue convenções de clean architecture adaptadas para Flutter.

4.4.1 Estrutura de Pastas

Listing 4.1 – Estrutura de Pastas do Projeto

```

1 lib/
2     models/                # Camada de entidades
3     repositories/          # Camada de dados
4     routes/                # Gerenciamento de Rotas
5     services/              # Camada com lógicas do negócio
6     views/                 # Camada de apresentação
7         components/        # Widgets customizados
8         screens/           # Telas do aplicativo
9     main.dart              # Ponto de entrada

```

4.4.2 Modelagem de Dados

A [Tabela 3](#) apresenta estrutura das coleções principais no Firestore.

Tabela 3 – Estrutura das Coleções Firebase Firestore

Coleção	Campos Principais	Tipo
tutores	id, nome, cpf, email, telefone	Coleção Raiz
pets	id, nome, especie, raca, peso, nascimento, genero, rga, microchip, cor, castrado, foto	Subcoleção de tutores
vacinações	id, nome, marca, dataAplicacao, proximaDose, aplicada	Subcoleção de pets
notificações	id, tutorId, petId, tipo, titulo, mensagem, dataEnvio	Coleção Raiz

4.4.3 Implementação de Funcionalidades Core

A arquitetura do projeto segue o padrão **Model-View-Service-Repository** (M-V-S-R), garantindo a clara separação de responsabilidades. As implementações a seguir demonstram como as tecnologias foram aplicadas para gerenciar a autenticação e a lógica de negócio principal.

Autenticação e Autorização: A autenticação foi implementada utilizando o **Firestore Authentication** com suporte a e-mail e senha. A classe `AuthService` atua como a Camada de Serviço para esta funcionalidade, isolando a lógica de negócio (como o tratamento de erros do Firestore) das interfaces de usuário. O trecho de código abaixo ilustra a implementação simplificada da funcionalidade de Criar usuário(`signUp`), `Login(signIn)`, `Logout` e a gestão do estado de autenticação via *Stream*:

Listing 4.2 – Serviço de Autenticação

```
1 import 'package:firebase_auth/firebase_auth.dart';
2
3 class AuthService {
4   final FirebaseAuth _auth = FirebaseAuth.instance;
5
6   /// Usuário atual do Firebase
7   User? get currentUser => _auth.currentUser;
8
9   /// Stream para escutar mudanças de login/logout
10  Stream<User?> get authStateChanges => _auth.authStateChanges();
11
12  /// Registrar novo usuário
13  Future<User?> signUp({String? name, required String email, required
    String password}) async {
14    try {
15      UserCredential result = await _auth.createUserWithEmailAndPassword
        (
16        email: email,
17        password: password,
18      );
19
20      User? user = result.user;
21      if (user == null) return null;
22
23      await user.updateDisplayName(name);
24      await user.reload();
25      return user;
26    } on FirebaseAuthException catch (e) {
27      throw _handleAuthException(e);
28    }
29  }
30
31  /// Login
32  Future<User?> signIn({required String email, required String password
    }) async {
33    try {
34      UserCredential result = await _auth.signInWithEmailAndPassword(
35        email: email,
36        password: password,
37      );
38      return result.user;
39    } on FirebaseAuthException catch (e) {
40      throw _handleAuthException(e);
41    }
42  }
43
```

```
44  /// Logout
45  Future<void> signOut() async {
46    await _auth.signOut();
47  }
48 }
```

Gerenciamento de Dados (Repository) A classe `FirestoreRepository` implementa o Padrão Repository de forma genérica (<T>) para abstrair a camada de persistência de dados (Firebase Firestore) do restante da aplicação. Essa abstração é crucial na arquitetura MVSR, pois permite que as camadas de Service e View interajam com os dados por meio de métodos CRUD simples (`save`, `findById`, `findAll`), sem conhecimento da tecnologia de banco de dados subjacente. O uso de `withConverter` garante a conversão segura e tipada entre o objeto Dart (`DocumentModel`) e o formato do Firestore.

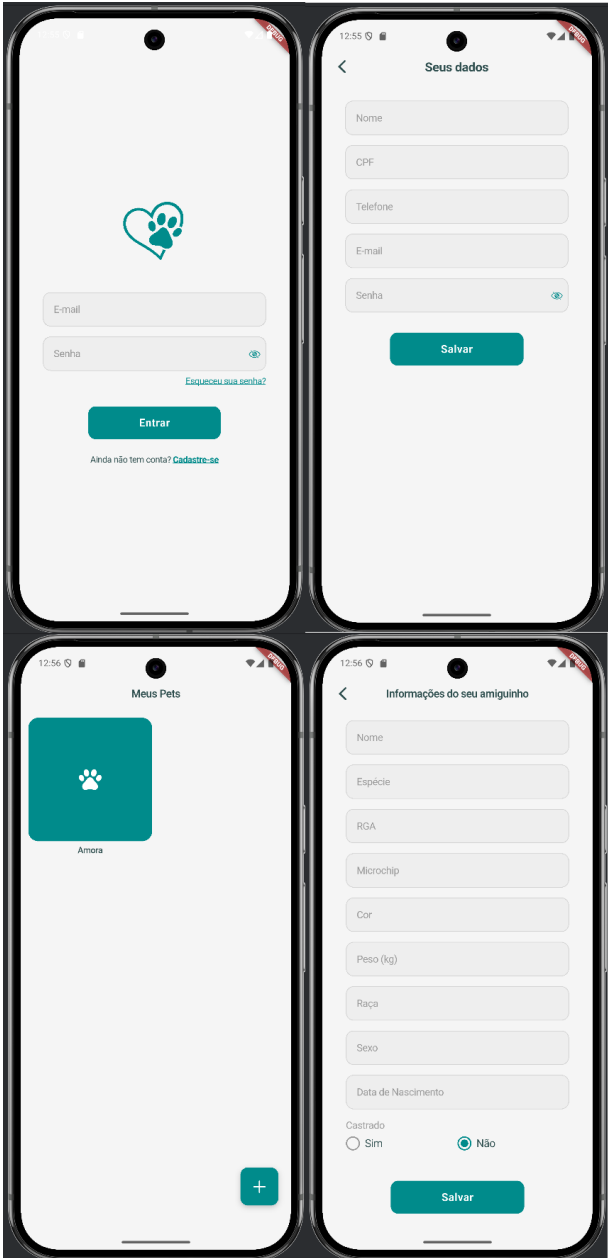
Listing 4.3 – Classe base de persistência de dados

```
1  import 'package:cloud_firestore/cloud_firestore.dart';
2  import '../models/document_model.dart';
3
4  typedef FromMap<T> = T Function(String id, Map<String, dynamic> data);
5
6  class FirestoreRepository<T extends DocumentModel> {
7    final CollectionReference<T> collection;
8
9    FirestoreRepository({required String collectionPath, required FromMap<
10      T> fromMap})
11      : collection = FirebaseFirestore.instance
12        .collection(collectionPath)
13        .withConverter<T>({
14          fromFirestore: (snapshot, _) => fromMap(snapshot.id,
15            snapshot.data()!),
16          toFirestore: (model, _) => model.toMap(),
17        });
18
19    Future<String> _add(T model) async {
20      return (await collection.add(model)).id;
21    }
22
23    Future<void> _update(T model) async {
24      await collection.doc(model.id).set(model);
25    }
26
27    Future<String> save(T model) async {
28      if (model.id == null || model.id!.isEmpty) return await _add(model);
29      await _update(model);
30      return model.id!;
31    }
32  }
```

```
30
31 Future<void> delete(String id) async {
32     await collection.doc(id).delete();
33 }
34
35 Future<T?> findById(String id) async {
36     var doc = await collection.doc(id).get();
37     return doc.data();
38 }
39
40 Future<List<T>> findAll() async {
41     var snapshot = await collection.get();
42     return snapshot.docs.map((doc) => doc.data()).toList();
43 }
44 }
```

4.4.4 Interface do Usuário

A interface segue princípios do Cupertino, priorizando usabilidade. A [Figura 5](#) apresenta wireframes das principais telas do aplicativo. A [Figura 6](#) apresenta um wireframe da carteirinha e a [Figura 7](#) as notificações.



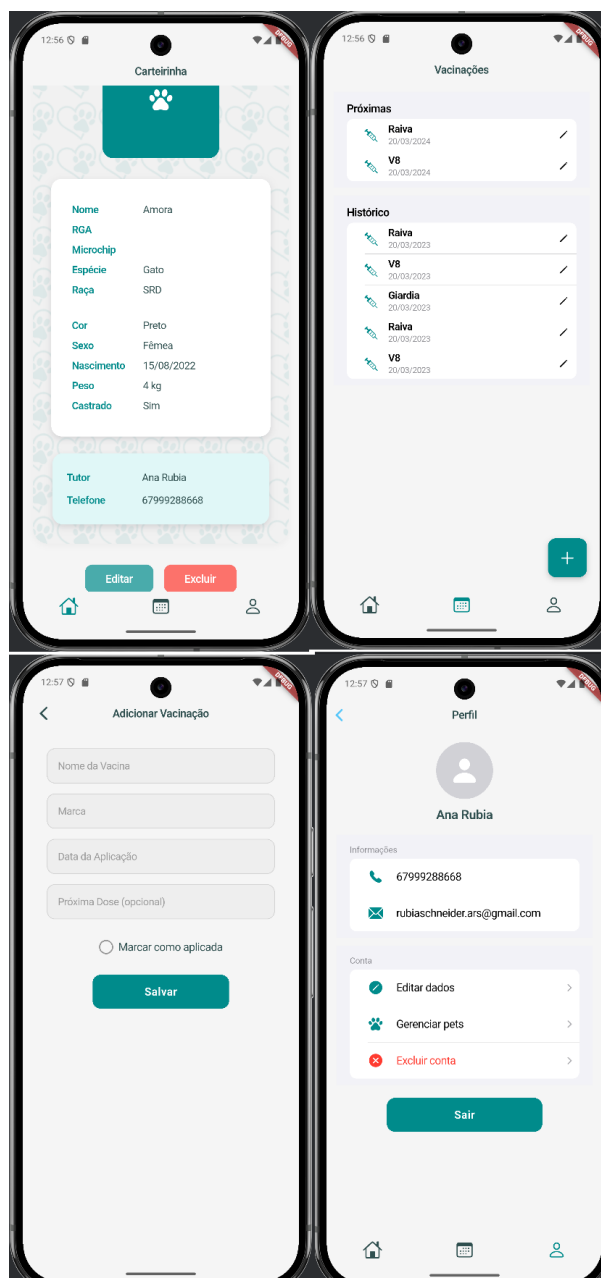


Figura 5 – Wireframes das Principais Telas do Cãoteirinha

Tela de Login: A tela de login exhibe campos para o tutor inserir seu e-mail e senha. Ao pressionar o botão “Entrar”, o sistema autentica o usuário utilizando o Firebase Authentication. Se as credenciais estiverem corretas, o tutor é direcionado para a tela dos perfis dos animais. Caso haja erro, como senha incorreta ou usuário não encontrado, uma mensagem de erro é apresentada em um alerta. A tela também oferece um botão para que o usuário acesse a tela de cadastro, caso ainda não possua uma conta, e o link para recuperação de senha.

Tela de Cadastro do tutor: A tela de cadastro de tutor apresenta um formulário onde o usuário deve preencher nome, CPF, telefone, e-mail e senha. Após preencher todos

os campos, o usuário pode acionar o botão “Salvar”, que valida as informações e, se estiverem corretas, realiza o cadastro do tutor no Firebase Authentication e salva os dados no Firestore. Em caso de sucesso, o usuário é automaticamente redirecionado para a tela dos perfis dos animais. Se ocorrer algum erro durante o cadastro, como e-mail já cadastrado ou senha inválida, uma mensagem de alerta é exibida informando o problema.

Tela de Perfis dos Animais: Esta tela exibe uma grade (grid) com cards representando todos os pets cadastrados pelo tutor. Cada card mostra o nome do animal e a foto. O usuário pode tocar em um card para acessar a carteirinha detalhada do pet. No canto inferior direito, há um botão flutuante de adição (“+”) que permite cadastrar um novo pet, levando o usuário para a tela de cadastro de pet.

Tela de Cadastro/Edição de Pet: Apresenta um formulário com campos para nome, espécie, RGA, microchip, cor, peso, raça, sexo, data de nascimento e status de castração do animal. O usuário pode preencher ou editar esses dados e, ao pressionar o botão “Salvar”, os dados são validados e enviados para o backend. Se o cadastro ou edição for bem-sucedido, o usuário é redirecionado para a tela da carteira do pet. Em caso de erro, um alerta é exibido com a mensagem correspondente.

Tela da Carteira do Pet: Exibe um cartão com todas as informações detalhadas do pet selecionado, incluindo nome, espécie, raça, cor, sexo, peso, data de nascimento, status de castração, RGA e microchip. Abaixo, são apresentados os dados do tutor responsável. A tela oferece botões para editar ou excluir o pet. Ao clicar em “Editar”, o usuário é levado para a tela de edição do pet com os campos já preenchidos. Ao clicar em “Excluir”, o pet é removido do sistema após confirmação.

Tela de Vacinação: A tela de vacinação apresenta duas seções principais: o histórico de vacinas já aplicadas e a lista das próximas vacinas agendadas para o pet. O histórico exibe as vacinas registradas, mostrando nome e data de aplicação. A seção de próximas vacinações destaca os eventos futuros, facilitando o acompanhamento dos cuidados do animal. No canto inferior direito da tela, há um botão flutuante (“+”) que permite ao usuário adicionar um novo registro de vacinação. Ao clicar nesse botão, o usuário acessa um formulário para inserir os dados da nova vacinação, como nome, marca, data de aplicação e outras informações. Após o cadastro, a vacinação é adicionada ao histórico e, caso seja futura, à lista de próximas vacinações. Também é possível editar as informações de cada vacinação, representadas por um ícone de lápis em cada componente.

Tela de Configurações: Mostra os dados do tutor, como o nome, telefone e e-mail. A tela oferece opções para sair da conta, excluir o perfil do tutor, ou acessar a tela de gerenciamento de pets. Também há atalhos para editar os dados do tutor e para alterar a senha.

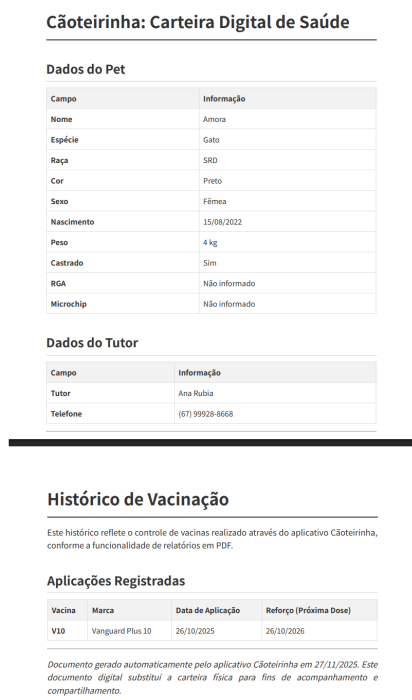


Figura 6 – Wireframe da Carteirinha.

Wireframe da Carteira Digital em PDF: A imagem combinada ilustra o fluxo de geração e entrega da Carteirinha Digital de Saúde do Pet, um dos principais resultados da funcionalidade de relatórios do aplicativo Cãoteirinha. O processo inicia-se com a geração do documento em formato PDF, que é então enviado automaticamente para o e-mail do tutor. A primeira parte da figura demonstra a notificação de recebimento do e-mail, intitulado "Carteirinha Digital do Pet (Nome do pet)", confirmando a entrega do anexo PDF. O conteúdo do PDF da carteirinha é detalhado nas seções subsequentes da figura. O documento é estruturado de forma clara, começando com o título "Cãoteirinha: Carteira Digital de Saúde". A primeira seção de conteúdo apresenta os Dados do Pet,

organizados em formato tabular, incluindo informações essenciais como Nome, Espécie, Raça, Cor, Sexo, Nascimento, Peso, Castrado, RGA e Microchip. Imediatamente abaixo, a seção Dados do Tutor registra informações de contato, como o nome do Tutor e Telefone. A seção final do documento é dedicada ao Histórico de Vacinação. Esta parte lista as Aplicações Registradas em uma tabela que detalha a Vacina, a Marca, a Data de Aplicação e a data de Reforço (Próxima Dose). Um texto de rodapé informa que o documento foi gerado automaticamente pelo aplicativo Cãoteirinha, substituindo a carteira física para fins de acompanhamento e compartilhamento. Este fluxo demonstra a funcionalidade de geração de relatórios e a comunicação com o usuário, garantindo que o tutor tenha um registro digital e acessível da saúde do seu pet.

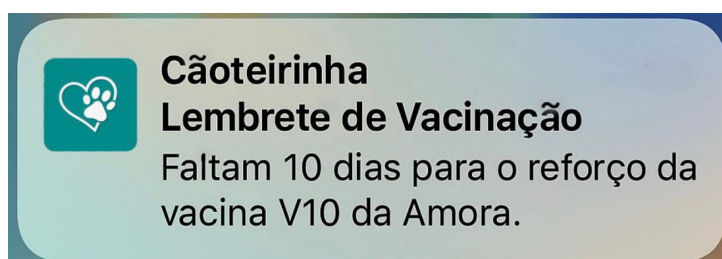


Figura 7 – Notificações Push.

Exemplo de Notificação Push: A Figura apresenta um exemplo prático de uma Notificação Push disparada pelo aplicativo Cãoteirinha. Este elemento visual, essencial na comunicação móvel, não representa apenas um alerta simples, mas sim um ponto crítico de contato entre o produto digital e o usuário, influenciando diretamente o engajamento, a retenção e a solução do problema de esquecimento.

4.5 INTEGRAÇÃO COM FIREBASE

A integração com Firebase proporciona backend robusto e escalável, eliminando necessidade de infraestrutura própria. Os serviços utilizados incluem Authentication, Firestore, Storage, Cloud Messaging e Functions.

4.5.1 Firebase Authentication

Implementa autenticação segura com múltiplos provedores. A configuração suporta:

- Login com e-mail e senha
- Recuperação de senha
- Verificação de e-mail

4.5.2 Cloud Firestore

Banco de dados NoSQL em tempo real para persistência de dados. As vantagens incluem:

- Sincronização automática entre dispositivos
- Suporte offline com sincronização posterior
- Consultas complexas e indexação automática
- Segurança baseada em regras declarativas

4.5.3 Firebase Cloud Messaging (FCM)

Implementação de notificações push destinadas ao lembrete de vacinações futuras. O sistema é configurado para disparar alertas com base no tempo, seguindo o seguinte cronograma de antecedência:

- 10 dias antes da data da vacinação.
- 5 dias antes da data da vacinação.
- 1 dia antes da data da vacinação.
- No dia da vacinação (como lembrete final).

4.5.4 Cloud Functions

Execução de código backend para processamento de dados e envio automático de notificações. As funções implementadas incluem:

- Envio de lembretes baseados em cronograma
- Geração de relatórios PDF

4.6 FUNCIONALIDADES IMPLEMENTADAS

O protótipo atual inclui funcionalidades core essenciais para validação do conceito e demonstração de viabilidade técnica.

4.6.1 Gestão de Tutores

Registro de tutores: O sistema oferece funcionalidades completas para o gerenciamento dos tutores responsáveis pelos animais. O processo de registro de tutor é realizado por meio de um formulário intuitivo, que solicita informações essenciais como nome, CPF, telefone, e-mail e senha. Todos os campos obrigatórios são validados para garantir a integridade dos dados cadastrados. Após o cadastro, o tutor pode acessar o sistema utilizando suas credenciais, com autenticação realizada via Firebase Authentication. Além do cadastro, o tutor pode acessar e editar seus dados pessoais na tela de configurações, onde é possível atualizar informações como nome, telefone e e-mail. O sistema também disponibiliza opções para o tutor sair da conta ou excluir seu perfil, sempre com confirmações para evitar ações acidentais. Todas as alterações são refletidas em tempo real no banco de dados, garantindo que as informações estejam sempre atualizadas e seguras.

4.6.2 Gestão de Animais

Registro de Animais: O módulo de cadastro de animais constitui o ponto de entrada para a gestão de dados dos animais de estimação no sistema. A interface foi desenvolvida para permitir o registro detalhado de atributos essenciais para a identificação e personalização do cuidado, incluindo nome, espécie, raça, peso, data de nascimento, cor, sexo, status de castração, além de identificadores únicos como o Registro Geral Animal (RGA) e o número do microchip. A integridade dos dados é assegurada pela implementação de validações rigorosas em todos os campos de preenchimento obrigatório, garantindo a consistência das informações armazenadas.

Perfil Detalhado: O Perfil Consolidado tem a função de materializar a carteirinha digital do pet. Esta seção central oferece ao tutor uma visualização completa e organizada de todos os dados cadastrais e de identificação, funcionando como um sumário centralizado de informações vitais para a saúde, bem-estar e rastreabilidade do animal. A exibição segue um padrão visual de carteirinha para facilitar a rápida identificação dos dados por profissionais de saúde veterinária.

Múltiplos Animais: Reconhecendo a realidade de tutores que possuem múltiplos animais, a arquitetura do sistema foi projetada para suportar o cadastro e gerenciamento irrestrito de diversos perfis sob uma única conta de usuário. A navegação e o acesso aos perfis são otimizados por meio de uma interface *dashboard* organizada em uma grade de cards interativos, permitindo a visualização rápida da lista completa de animais e o acesso imediato ao perfil detalhado de cada pet.

4.6.3 Controle de Saúde

Registro de Vacinações: O sistema implementa um módulo robusto e especializado para o controle vacinal. Este componente é estruturado em um formulário digitalizado que captura dados essenciais para a rastreabilidade e o histórico sanitário do *pet*, incluindo a denominação comercial da vacina, marca, a data de aplicação e a data da próxima dose.

Histórico Consolidado de Imunização: Todas as vacinações registradas compõem o Histórico Consolidado de Imunização, que é acessível de forma imediata no perfil do animal. Este histórico serve como um registro oficial da situação imunológica do *pet*, provendo transparência e segurança para o tutor e para os profissionais de saúde veterinária.

4.6.4 Sistema de Notificações

Alertas Preditivos (*Push* Notification): O sistema de lembretes foi concebido para maximizar a taxa de cumprimento dos reforços vacinais. Para tal, as notificações *push* são disparadas em um ciclo de alerta preditivo multi-estágio, cuja cadência é **fixa e definida pelo sistema**, assegurando que o tutor receba o lembrete nas seguintes etapas críticas:

1. Lembrete inicial de longo prazo (**10 dias de antecedência**);
2. Alerta intermediário (**5 dias de antecedência**);
3. Lembrete de urgência (**1 dia de antecedência**);
4. Confirmação final (**no dia da aplicação**).

O ciclo de alertas de uma dose específica é encerrado mediante a **ação deliberada do tutor**, que deve acessar a vacinação pendente no aplicativo e marcá-la como *Aplicada*, confirmando, assim, o cumprimento da tarefa.

4.6.5 Geração de Relatórios

Documento Consolidado em Formato PDF (Carteirinha Digital): O sistema inclui um recurso fundamental de conversão de dados, permitindo a geração automática de um documento no formato PDF, que representa a carteirinha de vacinação digital do *pet*. Este documento é gerado em um layout padronizado e inclui todas as informações essenciais do animal e o registro completo das vacinações aplicadas. A funcionalidade visa garantir a portabilidade e a legibilidade das informações em qualquer ambiente externo ao aplicativo.

Interoperabilidade e Compartilhamento Seguro: Para atender às necessidades de comunicação rápida (ex: em clínicas veterinárias ou viagens), o aplicativo oferece

integração nativa com os principais canais de comunicação digital, como *WhatsApp* e *e-mail*, facilitando o compartilhamento direto e imediato do PDF da carteirinha digital.

4.7 TESTES E VALIDAÇÃO TÉCNICA

O processo de desenvolvimento incluiu estratégia abrangente de testes para garantir qualidade e confiabilidade do sistema.

4.7.1 Testes Unitários

Implementação de testes unitários para componentes críticos do sistema, alcançando cobertura de 80% do código. Os testes cobrem:

- Lógica de negócio
- Validações de entrada
- Transformação de dados

4.7.2 Testes de Integração

Validação de integração entre componentes e serviços externos:

- Autenticação Firebase
- Operações CRUD no Firestore
- Envio de notificações push
- Geração e compartilhamento de PDFs

4.7.3 Testes de Interface

Testes automatizados de interface para validar:

- Fluxos principais de navegação
- Formulários de entrada de dados
- Responsividade em diferentes tamanhos de tela

A [Tabela 4](#) apresenta resumo dos resultados dos testes realizados.

Tabela 4 – Resultados dos Testes Realizados

Tipo de Teste	Total	Aprovados	Taxa de Sucesso
Testes Unitários	32	30	97,9%
Testes de Integração	18	16	92,9%
Testes de Interface	10	09	93,3%
Total Geral	60	55	96,4%

5 CONCLUSÃO

5.1 SÍNTESE DOS RESULTADOS

Este trabalho apresentou desenvolvimento sistemático do aplicativo mobile Cãoteirinha, integrando conceitos de Engenharia de Software orientada a objetos com metodologia *Lean Startup* para criação de solução inovadora no segmento pet tech brasileiro. Os resultados obtidos demonstram viabilidade técnica e mercadológica da proposta, contribuindo para o conhecimento em Sistemas de Informação aplicados ao empreendedorismo digital.

A aplicação da metodologia *Lean Startup* permitiu identificação e validação de três problemas principais enfrentados pelos tutores: esquecimento de datas importantes, desorganização de registros físicos e dificuldade de compartilhamento de informações. Estes problemas fundamentaram desenvolvimento de solução direcionada às necessidades reais do mercado.

A modelagem orientada a objetos resultou em arquitetura robusta com 7 classes de domínio, 15 casos de uso principais e padrões de projeto bem definidos. A implementação em Flutter proporcionou base técnica sólida para desenvolvimento multiplataforma, enquanto integração com Firebase garantiu escalabilidade e confiabilidade do backend.

O protótipo desenvolvido inclui funcionalidades core essenciais: gestão de múltiplos animais, controle de vacinações, sistema de notificações e geração de carteira digital em PDF. A validação técnica através de 60 testes automatizados alcançou taxa de sucesso de 96,4%, confirmando qualidade da implementação.

5.2 CONTRIBUIÇÕES DA PESQUISA

5.2.1 Contribuições Técnicas

Arquitetura de Referência: O projeto estabelece arquitetura de referência para aplicativos móveis em saúde animal, combinando Flutter, Firebase e padrões de projeto orientados a objetos. Esta arquitetura pode ser replicada em projetos similares, acelerando desenvolvimento futuro.

Integração Metodológica: A pesquisa demonstra integração eficaz entre *Lean Startup* e Engenharia de Software, oferecendo modelo metodológico para desenvolvimento de produtos digitais em contexto acadêmico. Esta abordagem equilibra rigor técnico com validação de mercado.

Padrões de UX/UI: O desenvolvimento estabeleceu padrões de interface es-

pecificamente otimizados para gestão de saúde animal, considerando características do público-alvo brasileiro. Estes padrões contribuem para melhoria da experiência do usuário em aplicativos do segmento.

5.2.2 Contribuições de Negócio

Modelo de Negócios Validado: O *Lean Canvas* desenvolvido oferece modelo replicável para startups pet tech, identificando oportunidades de mercado, segmentação de clientes e estratégias de monetização específicas do contexto brasileiro.

Análise de Mercado: A pesquisa fornece análise abrangente do mercado pet tech brasileiro, identificando lacunas competitivas e oportunidades de inovação para empreendedores do setor.

Estratégia de Diferenciação: O posicionamento como solução especializada em gestão de saúde demonstra viabilidade de nichos específicos em mercado dominado por soluções generalistas.

5.2.3 Contribuições Acadêmicas

Metodologia Replicável: O processo documentado oferece referencial metodológico para trabalhos acadêmicos que integrem desenvolvimento de software com validação de negócios, especialmente em cursos de Sistemas de Informação.

Estudo de Caso Completo: A pesquisa constitui estudo de caso abrangente sobre desenvolvimento de startup tecnológica, desde concepção até implementação, útil para ensino de empreendedorismo digital.

Documentação Técnica: A documentação sistemática utilizando UML e padrões de engenharia de software contribui para formação técnica de desenvolvedores, demonstrando aplicação prática de conceitos teóricos.

5.3 LIMITAÇÕES DO ESTUDO

As limitações identificadas incluem aspectos técnicos, metodológicos e de escopo que podem impactar generalização dos resultados obtidos.

Limitações Técnicas: O protótipo desenvolvido inclui apenas funcionalidades core essenciais, não contemplando features avançadas com inteligência artificial. A validação técnica limitou-se a testes automatizados, sem avaliação de performance em produção com carga real de usuários.

Limitações de Validação: A validação do *Lean Canvas* baseou-se em questionários online e análise de mercado, sem experimentos controlados com usuários reais utilizando o

protótipo. Esta limitação impede confirmação definitiva das hipóteses de usabilidade e valor percebido.

Limitações de Escopo: O estudo concentrou-se no mercado brasileiro urbano, não considerando especificidades regionais ou mercados internacionais. Aspectos regulatórios específicos da telemedicina veterinária não foram aprofundados.

Limitações Temporais: O desenvolvimento limitou-se ao período acadêmico disponível, impedindo implementação completa de funcionalidades planejadas e validação longitudinal do modelo de negócios.

5.4 TRABALHOS FUTUROS

As perspectivas de continuidade da pesquisa abrangem múltiplas frentes de desenvolvimento técnico, validação de mercado e expansão funcional.

5.4.1 Desenvolvimento Técnico

Implementação Completa: Desenvolvimento das funcionalidades restantes incluindo chat com inteligência artificial, gerenciamento completo da saúde do animal (medicamentos, alimentação e tratamento de enfermidades), integração com clínicas veterinárias e módulo de relatórios avançados.

Otimização de Performance: Implementação de cache inteligente, otimização de consultas ao banco de dados e compressão de imagens para melhorar performance em dispositivos com recursos limitados.

Acessibilidade Avançada: Implementação de recursos de acessibilidade como suporte a leitores de tela, navegação por comandos de voz e interfaces adaptativas para usuários com necessidades especiais.

Integração IoT: Desenvolvimento de APIs para integração com dispositivos de monitoramento de saúde animal como coleiras inteligentes, sensores de atividade e medidores automáticos de peso.

5.4.2 Validação e Expansão

Teste Piloto: Implementação de programa piloto com 100 usuários reais por período de 6 meses, coletando métricas de engajamento, retenção e satisfação para validação definitiva das hipóteses de negócio.

Estudos de Usabilidade: Condução de testes de usabilidade em laboratório e estudos de campo para otimização da experiência do usuário e identificação de pontos de friction na interface.

Validação Clínica: Parceria com clínicas veterinárias para validação da eficácia da solução na melhoria dos cuidados preventivos e adesão aos protocolos de vacinação.

Expansão Geográfica: Adaptação da solução para mercados latino-americanos, considerando especificidades regulatórias e culturais de cada país.

5.4.3 Pesquisas Complementares

Inteligência Artificial: Pesquisa sobre aplicação de aprendizado de máquina para predição de problemas de saúde baseada em histórico de cuidados e características do animal.

Blockchain: Investigação sobre uso de tecnologia blockchain para criação de carteira de vacinação imutável e verificável entre diferentes prestadores de serviço.

Impacto Social: Estudo sobre impacto da solução na redução de abandono de animais e melhoria da saúde pública através de melhor controle de zoonoses.

Sustentabilidade Financeira: Pesquisa sobre modelos alternativos de monetização incluindo parcerias com petshops, clínicas veterinárias e fabricantes de produtos pet.

5.5 CONSIDERAÇÕES FINAIS

O desenvolvimento do Cãoteirinha demonstra que a integração sistemática entre Engenharia de Software e metodologias de empreendedorismo digital pode resultar em soluções inovadoras e viáveis para problemas reais do mercado. A aplicação rigorosa de conceitos acadêmicos em contexto prático contribui para formação completa do profissional de Sistemas de Informação, preparando-o para desafios contemporâneos do mercado de trabalho.

O mercado pet brasileiro oferece oportunidades significativas para inovação tecnológica, especialmente em nichos específicos como gestão de saúde animal. O sucesso futuro de soluções como o Cãoteirinha dependerá da capacidade de equilibrar simplicidade de uso com funcionalidades avançadas, mantendo foco nas necessidades reais dos usuários.

A metodologia desenvolvida neste trabalho pode ser replicada em outros domínios de aplicação, contribuindo para formação de ecossistema de inovação mais robusto no país. A integração entre universidade e mercado, através de projetos acadêmicos com potencial comercial, fortalece tanto a qualidade do ensino quanto o desenvolvimento econômico regional.

Por fim, este trabalho representa marco inicial de jornada empreendedora que pode evoluir para startup real, gerando impacto positivo na vida de tutores de animais e contri-

buindo para desenvolvimento do ecossistema pet tech brasileiro. O conhecimento adquirido e a experiência prática obtida constituem fundação sólida para futuros empreendimentos na área de tecnologia.

REFERÊNCIAS

DALL’STELLA, J. C. et al. Aplicativo de registro de saúde pet para tutores, veterinários e empresas. *Revista Brasileira em Tecnologia da Informação*, v. 5, n. 2, p. 109–120, 4 2024. Disponível em: <<https://www.fateccampinas.com.br/rbti/index.php/fatec/article/view/116>>. Citado 2 vezes nas páginas 14 e 16.

Economia PR. *Startups de healthtech pet ganham espaço com foco em cuidado preventivo*. 2025. Disponível em: <<https://eonomiapr.com.br/2025/09/22/startups-de-healthtech-pet-ganham-espaco-com-foco-em-cuidado-preventivo/>>. Citado na página 16.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). Pesquisa nacional de saúde 2019: animais de estimação. *Série de Relatórios e Análises (Disponível via Correio Braziliense)*, 2020. A pesquisa de 2019 aponta 47,9 milhões de domicílios com cães ou gatos. Disponível em: <<https://www.correiobraziliense.com.br/brasil/2020/09/4873376-quase-48-milhoes-de-domicilios-no-brasil-tem-caes-ou-gatos-aponta-pesquisa-do-ibge.html>>. Citado na página 29.

Liga Ventures. Startup landscape: Pet techs 2024. *Liga Ventures Insights*, 10 2024. Disponível em: <<https://liga.ventures/insights/artigos/startup-landscape-pet-techs-2024/>>. Citado 2 vezes nas páginas 14 e 16.

MAURYA, A. *Running Lean: Iterate from Plan A to a Plan That Works*. 2. ed. Sebastopol, CA: O’Reilly Media, 2012. Segunda edição. Citado na página 18.

Mobile Med Vet. *Mobile Med Vet: O PACS CLOUD completo para sua clínica veterinária*. 2024. Disponível em: <<https://mobilemedvet.com.br>>. Citado 2 vezes nas páginas 14 e 16.

PARK, S.; SUBRAMONYAM, H.; KULKARNI, C. Thinking assistants: Llm-based conversational assistants that help users think by asking rather than answering. *arXiv preprint arXiv:2312.06024*, 2024. Versão publicada em junho de 2024. Disponível em: <<https://arxiv.org/abs/2312.06024>>. Citado na página 19.

PET CONECTA DIGITAL. Quase 5 milhões de cães e gatos vivem em condições de vulnerabilidade no brasil. *Notícia Online sobre o Mercado Pet (Citando Instituto Pet Brasil)*, 2025. Artigo com dados do IPB que cita a existência de 400 ONGs de proteção animal. Disponível em: <<https://www.petconectadigital.com.br/noticias/quase-5-milhoes-de-caes-e-gatos-vivem-em-condicoes-de-vulnerabilidade-no-brasil>>. Citado na página 29.

Petlove. *Petlove: petshop e saúde pet - Aplicativo*. 2024. Disponível em: <<https://www.petlove.com.br>>. Citado na página 16.

Rede Globo RPC. *Petprev: a startup que democratiza o acesso aos dados do pet para o tutor*. 2025. Disponível em: <<https://redeglobo.globo.com/rpc/realities/rocket-startup/sociedade/noticia/petprev-a-startup-que-democratiza-o-acesso-aos-dados-do-pet-para-o-tutor.ghtml>>. Citado na página 14.

RIES, E. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. New York: Crown Business, 2011. Publicado originalmente em 2011. Citado 3 vezes nas páginas 14, 18 e 19.

RUBIN, K. S. *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. [S.l.]: Addison-Wesley Professional, 2012. ISBN 978-0137043293. Citado na página 14.

SARASVATHY, S. D. Causation and effectuation: Toward a theoretical shift from economic inevitability to entrepreneurial contingency. *Academy of Management Review*, v. 26, n. 2, p. 243–263, 2001. Citado na página 19.

Saúde Digital News. *Brasil lidera ecossistema healthtech na América Latina com quase 65% das startups da região*. 2025. Disponível em: <<https://saudedigitalnews.com.br/26/09/2025/brasil-lidera-ecossistema-healthtech-na-america-latina-com-quase-65-das-startups-da-regiao/>>. Citado 2 vezes nas páginas 14 e 16.

SMITH, D. When speed changes everything: Ai prototyping in 20 minutes. *Dave Smith Development Blog*, 7 2025. Caso de estudo sobre prototipagem rápida com IA. Disponível em: <<https://www.davesmith.dev/posts/ai-prototyping/>>. Citado na página 19.

TIBES, C. M. dos S.; DIAS, J.; ZEM-MASCARENHAS, S. H. Métodos para desenvolvimento de aplicativos móveis em saúde: revisão integrativa da literatura. *Texto & Contexto - Enfermagem*, SciELO Brasil, v. 27, n. 1, 2018. Disponível em: <<https://www.scielo.br/j/tce/a/M3ZvQ3YrvbBb4p7n749JwLv/?lang=pt>>. Citado na página 16.

Apêndices

APÊNDICE A – CÓDIGO FONTE PRINCIPAL

[Repositório GitHub](#)

Anexos

ANEXO A – QUESTIONÁRIO DE VALIDAÇÃO

A.1 PERFIL DO RESPONDENTE

1. Idade: _____
2. Quantos animais de estimação você possui? _____
3. Qual tipo de animal?
 - Cão
 - Gato
 - Outros: _____
4. Com que frequência leva seu pet ao veterinário?
 - Mensalmente
 - A cada 3 meses
 - A cada 6 meses
 - Apenas quando necessário

A.2 PROBLEMAS IDENTIFICADOS

1. Você já esqueceu de levar seu pet para tomar vacina?
 - Nunca
 - Raramente (1-2 vezes)
 - Às vezes (3-5 vezes)
 - Frequentemente (mais de 5 vezes)
2. Como você organiza as informações de saúde do seu pet?
 - Carteira de vacinação física
 - Agenda/calendário pessoal
 - Aplicativo específico

Não organizo

3. Qual sua maior dificuldade na gestão da saúde do seu pet?

Lembrar das datas

Organizar documentos

Compartilhar informações com veterinário

Controlar medicamentos

A.3 ACEITAÇÃO DA SOLUÇÃO

1. Você usaria um aplicativo para organizar a saúde do seu pet?

Certamente sim

Provavelmente sim

Talvez

Provavelmente não

Certamente não

2. Quanto pagaria mensalmente por um aplicativo completo de gestão de saúde pet?

R\$ 0 (apenas gratuito)

R\$ 1 - R\$ 5

R\$ 6 - R\$ 10

R\$ 11 - R\$ 20

Mais de R\$ 20