



Universidade de Brasília – UnB
Faculdade de Ciências e Tecnologias em Engenharia – FCTE
Engenharia de Software

NattyPet: Um WebApp para Geração de Receitas Balanceadas Nutricionalmente para Pets

Autor: Lucas Felipe Soares e João Victor Teixeira Batista

Orientadora: Profa. Dra. Milene Serrano

Brasília, DF

2025



Lucas Felipe Soares e João Victor Teixeira Batista

NattyPet: Um WebApp para Geração de Receitas Balanceadas Nutricionalmente para Pets

Monografia submetida ao curso de graduação
em Engenharia de Software da Universidade
de Brasília, como requisito parcial para ob-
tenção do Título de Bacharel em Engenharia
de Software.

Universidade de Brasília – UnB

Faculdade de Ciências e Tecnologias em Engenharia – FCTE

Orientador: Profa. Dra. Milene Serrano

Coorientador: Prof. Dr. Maurício Serrano

Brasília, DF

2025

Lucas Felipe Soares e João Victor Teixeira Batista

NattyPet: Um WebApp para Geração de Receitas Balanceadas Nutricionalmente para Pets/ Lucas Felipe Soares e João Victor Teixeira Batista. – Brasília, DF, 2025-

120 p. : il. (algumas color.) ; 30 cm.

Orientador: Profa. Dra. Milene Serrano

Coorientador: Prof. Dr. Maurício Serrano

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB

Faculdade de Ciências e Tecnologias em Engenharia – FCTE , 2025.

1. Pets. 2. Alimentação Natural. I. Profa. Dra. Milene Serrano. II. Universidade de Brasília. III. FCTE - Faculdade de Ciências e Tecnologias em Engenharia IV. NattyPet: Um WebApp para Geração de Receitas Balanceadas Nutricionalmente para Pets

CDU 02:141:005.6

Lucas Felipe Soares e João Victor Teixeira Batista

NattyPet: Um WebApp para Geração de Receitas Balanceadas Nutricionalmente para Pets

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 24 de Julho de 2025 :

Profa. Dra. Milene Serrano
Orientadora

Prof. Dr. Maurício Serrano
Coorientador

Prof. M.e. Ricardo Ajax Dias Kosloski
Convidado 1

Tomáz Felipe Rodrigues Martins
Convidado 2

Brasília, DF
2025

*Este trabalho é dedicado aos nossos pais, cujas histórias nos ensinaram o valor da
dedicação e trabalho árduo.*

Agradecimentos

Agradecemos a Deus por nos conceder saúde e força para superar os desafios e seguir em constante evolução.

Agradecemos à Universidade de Brasília e seus docentes do curso de Engenharia de Software, que abriram mais portas que imaginávamos ser possível.

Agradecemos imensamente aos nossos pais, tios(as), avós, avôs e irmãos pelo apoio incondicional, incentivo e estrutura que nos permitiram adquirir conhecimento e perseguir nossos objetivos. O amor, a paciência e a sabedoria de cada um de vocês foram fundamentais para chegarmos até aqui. Agradecemos por nos darem a mão, nos guiarem e nos darem força nos momentos desafiadores. Cada abraço, cada palavra de conforto e cada ensinamento nos fortaleceram e nos motivaram a seguir em frente. Agradecemos por acreditarem em nós e por nos darem as ferramentas para trilharmos o nosso caminho. Vocês são a base da nossa vida e a inspiração para continuarmos sonhando e conquistando nossos sonhos. Amamos muito vocês!

Agradecemos de todo o coração aos nossos queridos orientadores, Prof.^a Dr.^a Milene Serrano e Prof. Dr. Maurício Serrano, não apenas pelos valiosos ensinamentos transmitidos em suas disciplinas, mas, sobretudo, pelo carinho, paciência, dedicação e zelo que demonstraram ao nos guiar na jornada deste Trabalho de Conclusão de Curso. Agradecemos por nos inspirarem a trilhar o caminho do conhecimento com paixão e por nos darem o apoio necessário para superar os desafios. Deixamos, assim, a nossa mais profunda e sincera gratidão aos nossos orientadores.

Gostaríamos de expressar nossa imensa gratidão por todos os amigos, colegas e professores que nos guiaram e apoiaram ao longo desta jornada. Sem vocês, não teríamos chegado tão longe. O nosso muito obrigado é sincero e eterno.

Resumo

Desde o início da década passada, um nicho tem crescido fortemente no mercado alimentício para animais de estimação (*pets*). Fala-se da Alimentação Natural, sendo essa uma dieta caseira que consiste em oferecer ao animal refeições compostas de ingredientes naturais, como carnes, frutas e vegetais, ao invés do uso de rações artificiais e altamente processadas. Esse tipo de alimentação apresenta vários benefícios para os *pets*, tais como: o aumento da digestibilidade, diminuição do consumo de conservantes químicos, uso de ingredientes frescos e de qualidade, e maior controle na formulação da dieta para animais com doenças. Entretanto, um dos principais desafios e fonte de críticas da Alimentação Natural é a dificuldade que tutores têm para elaborar refeições balanceadas que atendam às necessidades de seus animais. A inadequada formulação de receitas pode incorrer em desequilíbrios nutricionais, e causar problemas de saúde ao *pet*, sendo muitas vezes recomendado o auxílio de profissionais especializados. O presente Trabalho de Conclusão de Curso tem como intuito facilitar o emprego da Alimentação Natural por tutores de cães e gatos. Para tanto, desenvolveu-se um *WebApp* (em português, Aplicativos da Web Progressivos) que auxilia usuários a criarem, de maneira fácil e intuitiva, receitas balanceadas que cumpram os requisitos nutricionais previstos pelas principais organizações regulamentadoras de nutrição animal. Visando uma maior adequação da programação ao problema em questão, optou-se pelo uso da Programação Linear para otimizar o desempenho do processo de balanceamento das receitas.

Palavras-chave: Alimentação Natural. Animais de Estimação (Cães e Gatos). Nutrição Animal. Organizações Regulamentadoras de Nutrição Animal. Software. Programação Linear. Aplicativos de Web Progressivos.

Abstract

Since the beginning of the last decade, a niche has been rapidly growing in the pet food market. This refers to Natural Feeding, a homemade diet that consists of providing pets with meals made from natural ingredients, such as meats, fruits, and vegetables, instead of artificial and highly processed kibble. This type of diet offers several benefits for pets, including increased digestibility, reduced consumption of chemical preservatives, the use of fresh and high-quality ingredients, and greater control over diet formulation for animals with specific health conditions. However, one of the main challenges and sources of criticism of natural feeding is the difficulty that pet owners face in creating balanced meals that meet their animals' nutritional needs. Poorly formulated recipes can lead to nutritional imbalances and cause health issues for pets, often making the guidance of specialized professionals necessary. This project aims to facilitate the adoption of natural feeding by dog and cat owners. To achieve this, developed a web application that assists users in easily and intuitively creating balanced recipes that meet the nutritional requirements established by the leading animal nutrition regulatory organizations. To ensure that the software is well-suited to the problem at hand, Linear Programming was chosen as an optimization approach to enhance the performance of the recipe-balancing process.

Key-words: Natural Feeding. Animal Nutrition. Pets. Animal Nutrition Regulatory Organizations. Software. Linear Programming. Progressive Web Apps.

Lista de ilustrações

Figura 1 – Fluxo de Atividades do Desenvolvimento da Aplicação	59
Figura 2 – Fluxo de Atividades do Versionamento e Análise da Aplicação	60
Figura 3 – Fluxo de Atividades do TCC	61
Figura 4 – Persona 1	67
Figura 5 – Persona 2	68
Figura 6 – Persona 3	68
Figura 7 – Antipersona	69
Figura 8 – Análise SWOT do Projeto NattyPet	70
Figura 9 – <i>Backlog</i> do NattyPet	73
Figura 10 – Legenda em Cores das Versões do NattyPet	73
Figura 11 – Cores Principais e seus Tons	75
Figura 12 – Paleta do NattyPet Orientando-se pelo Sistema do Material 3	75
Figura 13 – Logotipo do NattyPet	76
Figura 14 – Tela de Perfil do <i>Pet</i>	77
Figura 15 – Tela de Detalhes da Receita	78
Figura 16 – Tela de Criação de Receitas	79
Figura 17 – Visão Geral da Arquitetura de Interação do Sistema NattyPet	80
Figura 18 – Diagrama Entidade-Relacionamento do NattyPet	81
Figura 19 – Diagrama Lógico de Dados do NattyPet	82
Figura 20 – Tela de Acesso (Login)	83
Figura 21 – Tela de Cadastro de Usuário	84
Figura 22 – Tela Home Após Login	84
Figura 23 – Tela de Gerenciamento de Pets	85
Figura 24 – Tela de Criação de Receita	86
Figura 25 – Tela de Visualização das Receitas Salvas	87
Figura 26 – Gráfico da Faixa Etária dos Participantes.	97
Figura 27 – Gráfico de Familiaridade dos Usuários com o Domínio de Alimentação Natural.	97
Figura 28 – Gráfico de Familiaridade dos Usuários com o Preparo de Rações.	98
Figura 29 – Gráfico de Familiaridade dos Usuários com Aplicações Similares.	98
Figura 30 – Avaliação da Experiência de Uso (Parte 1).	99
Figura 31 – Avaliação da Experiência de Uso (Parte 2).	99
Figura 32 – Classificação Geral da Facilidade de Uso do Aplicativo.	100
Figura 33 – Probabilidade de Recomendação do Aplicativo.	100
Figura 34 – Resultados da Análise de Qualidade de Código no SonarQube.	101

Figura 35 – Resultados dos Testes de Cobertura do <i>Frontend</i>	102
Figura 36 – Resultados dos Testes de Cobertura do <i>Backend</i>	102
Figura 37 – <i>Benchmarking</i> com Aplicações Existentes	117

Lista de tabelas

Tabela 1 – Níveis Nutricionais Mínimos Recomendados para Cães	23
Tabela 2 – Níveis Nutricionais Mínimos Recomendados para Gatos	25
Tabela 3 – Guia para Escore de Condição Corporal em Gatos	28
Tabela 4 – Guia para Escore de Condição Corporal em Cães	29
Tabela 5 – Recomendações de NED para Cães em Relação à Atividade	32
Tabela 6 – Requisitos Médios Diários de Energia para Gatos Adultos	32
Tabela 7 – Recomendações Práticas para NEM em Cães de Diferentes Idades. . .	33
Tabela 8 – Necessidades Energéticas durante Crescimento e Reprodução em Cães	34
Tabela 9 – Necessidades Energéticas durante Crescimento e Reprodução em Gatos	34
Tabela 10 – Comparação entre Programação Dinâmica (MKP) e Programação Li- near (Simplex)	44
Tabela 11 – Tecnologias Utilizadas no Trabalho	51
Tabela 12 – <i>Strings</i> de Busca Utilizadas	57
Tabela 13 – Médias de Micronutrientes e Tempo de Execução por Perfil de <i>Pet</i> . .	91
Tabela 14 – Resultados Consolidados do Ciclo 1 de Testes de Usabilidade no Pro- tótipo.	94
Tabela 15 – Resultados Consolidados do Ciclo 2 de Testes de Usabilidade no Pro- tótipo.	95
Tabela 16 – Andamento Consolidado das Atividades do Projeto	104

Lista de abreviaturas e siglas

API	<i>Application Programming Interface</i>
BF	<i>Body Fat</i>
BPMN	<i>Business Process Model and Notation</i>
CD	<i>Continuous Delivery</i>
CI	<i>Continuous Integration</i>
CLI	<i>Command Line Interface</i>
CSC	<i>Compressed Sparse Column</i>
CSR	<i>Client Side Rendering</i>
DER	Diagrama Entidade-Relacionamento
DLD	Diagrama Lógico Relacional
ECC	Escore de Condição Corporal
EM	Energia Metabolizável
HiGHS	<i>High-Performance Interior Gateway Protocol for High-Speed Networks</i>
HTTP	<i>Hypertext Transfer Protocol</i>
JSON	<i>JavaScript Object Notation</i>
MVP	<i>Minimum Viable Product</i>
NED	Necessidade Energética Diária
NEM	Necessidade Energética de Manutenção
PC	Peso Corporal
POO	Programação Orientada a Objetos
PWA	<i>Progressive Web App</i>
SO	Sistema Operacional
SSG	<i>Static Site Generation</i>

SSR	<i>Server Side Rendering</i>
SWOT	<i>Strengths, Weaknesses, Opportunities, and Threats</i>
TCC	Trabalho de Conclusão de Curso
TCLE	Termo de Consentimento Livre e Esclarecido
UI	<i>User Interface</i>
USDA	<i>United States Department of Agriculture</i>
WCAG	<i>Web Content Accessibility Guidelines</i>

Sumário

1	INTRODUÇÃO	17
1.1	Contextualização	17
1.2	Justificativa	18
1.3	Questões de Pesquisa e Desenvolvimento	19
1.4	Objetivos	19
1.4.1	Objetivo Geral	19
1.4.2	Objetivos Específicos	20
1.5	Organização da Monografia	20
2	REFERENCIAL TEÓRICO	21
2.1	Nutrição Animal	21
2.1.1	Órgãos Regulamentadores	21
2.1.2	Alimentação Natural	22
2.1.3	Guias Nutricionais	23
2.1.4	Necessidades Energéticas	27
2.1.4.1	Peso Metabólico	27
2.1.4.2	Escore de Condição Corporal (ECC)	27
2.1.4.3	Relevância do Peso Metabólico e ECC	31
2.1.5	Fatores de Influência	31
2.1.5.1	Atividade Física	31
2.1.5.2	Idade	32
2.1.5.3	Fases de Crescimento e Reprodução	33
2.1.6	Cálculo dos Nutrientes	34
2.2	Engenharia de Software	36
2.2.1	Requisitos de Software	36
2.2.2	Usabilidade	37
2.2.2.1	Testes de Usabilidade	38
2.2.2.2	Pesquisa por Formulário	39
2.2.3	<i>Progressive Web Apps</i>	39
2.2.3.1	PWAs vs Websites vs Apps Nativos	39
2.2.3.2	Características Técnicas dos PWAs	41
2.2.4	Algoritmos	42
2.2.4.1	Programação Linear	42
2.2.4.2	O Algoritmo Simplex	43
2.2.4.3	Análise Comparativa de Abordagens	44

2.3	Resumo do Capítulo	45
3	SUPORTE TECNOLÓGICO	46
3.1	Gerência e Configuração de Software	46
3.1.1	Visual Studio Code	46
3.1.2	GIT	46
3.1.3	GitHub	47
3.1.4	SonarQube	47
3.2	Desenvolvimento	47
3.2.1	Svelte & SvelteKit	47
3.2.2	Java & Spring	48
3.2.3	Go	48
3.2.4	Docker	49
3.2.5	PostgreSQL	49
3.3	Modelagem	49
3.3.1	Figma	49
3.3.2	Eraser.io	50
3.4	Análise de Resultados	50
3.4.1	Maze	50
3.4.2	Google Forms	50
3.5	Implantação de Software	50
3.5.1	Hetzner (Servidor VPS com CPU Compartilhada)	51
3.6	Resumo do Capítulo	51
4	METODOLOGIA	53
4.1	Classificação da Pesquisa	53
4.1.1	Abordagem	53
4.1.2	Natureza	54
4.1.3	Objetivos	54
4.1.4	Procedimentos	55
4.1.5	Método Investigativo	55
4.2	Método de Desenvolvimento	57
4.2.1	Processo de Desenvolvimento	58
4.3	Método de Análise dos Resultados	59
4.4	Fluxo de Atividades	61
4.5	Cronograma	63
4.6	Resumo do Capítulo	64
5	NATTYPET	65
5.1	Contextualização	65

5.2	Sobre o NattyPet	66
5.2.1	Público Alvo	66
5.2.2	Análise SWOT	69
5.3	Análise de Risco	70
5.4	Backlog do Produto	72
5.5	Identidade Visual	74
5.5.1	Tipografia	74
5.5.2	Paleta de Cores	74
5.5.3	Logotipo	75
5.5.4	Protótipo de Alta Fidelidade	76
5.5.4.1	Tela de Perfil do <i>Pet</i>	77
5.5.4.2	Tela de Detalhes de uma Receita	77
5.5.4.3	Tela de Criação de Receita	78
5.6	Arquitetura da Solução	79
5.6.1	Dados dos Alimentos	80
5.6.2	Visão de Dados	81
5.7	Funcionalidades	83
5.7.1	Cadastro e Autenticação de Usuário	83
5.7.2	Tela Home	83
5.7.3	Gerenciamento de Pets	85
5.7.4	Gerenciamento de Receitas	85
5.8	Algoritmo de Balanceamento de Receitas	87
5.8.1	Modelagem do Problema	87
5.8.1.1	Variáveis de Decisão e Função Objetivo	87
5.8.1.2	Restrições Nutricionais e de Formulação	88
5.8.2	Implementação com Go e <i>Solver</i> HiGHS	89
5.8.2.1	Visão Geral da Arquitetura do Serviço	89
5.8.2.2	Interface com HiGHS via CGO	89
5.8.2.3	Estrutura de Dados e Fluxo de Execução	90
5.8.2.4	Resultados e Desempenho	90
5.9	Resumo do Capítulo	91
6	ANÁLISE DOS RESULTADOS	93
6.1	Ciclo 1: Validação do Protótipo Inicial	93
6.1.1	Coleta e Análise dos Dados	93
6.1.2	Plano de Ação e Divulgação dos Resultados	94
6.2	Ciclo 2: Refinamento e Otimização da Usabilidade	95
6.2.1	Coleta e Análise dos Dados	95
6.2.2	Plano de Ação e Divulgação dos Resultados	95

6.3	Ciclo 3: Validação da Aplicação Final	96
6.3.1	Coleta e Análise dos Dados	96
6.3.2	Plano de Ação Pós-Ciclo 3	98
6.4	Análise de Qualidade de Código e Testes	101
6.4.1	Análise Estática de Código com SonarQube	101
6.4.2	Testes de Cobertura	102
6.5	Resumo do Capítulo	103
7	CONCLUSÃO	104
7.1	Trabalho Realizado	104
7.2	Resultados Obtidos	104
7.3	Trabalhos Futuros	106
	REFERÊNCIAS	108
	APÊNDICES	115
	APÊNDICE A – BENCHMARKING DOS RECURSOS DAS APLICAÇÕES	116
	APÊNDICE B – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO	118

1 Introdução

Nesse capítulo, com base nos tópicos de interesse de pesquisa e desenvolvimento desse Trabalho de Conclusão de Curso (TCC), será apresentada uma breve [Contextualização](#), abordando o tema da Alimentação Natural para *pets*. Consoante à contextualização, têm-se as dificuldades para geração de receitas no contexto da Alimentação Natural, e a respectiva [Justificativa](#) para o presente trabalho, com foco no desenvolvimento de um software para tutores de *pets* que auxilia na introdução de Alimentação Natural através da geração de receitas nutricionalmente balanceadas. Na sequência, têm-se a [Questões de Pesquisa](#), e os objetivos, [Objetivo Geral](#) e [Objetivos Específicos](#), procurando esclarecer sobre os principais pontos levados em consideração para o cumprimento do trabalho. Por fim, é apresentada a [Organização da Monografia](#).

1.1 Contextualização

O Brasil tem a terceira maior população de animais de estimação do mundo e o segundo maior mercado de *pet food* do mundo, superado somente pelos Estados Unidos. Este mercado, representa entorno de 0,36% do PIB brasileiro, com faturamento anual superior a 20 bilhões de reais em 2017, dos quais o setor de *pet food* é responsável por 74% ([ABINPET, 2021](#)). Esses dados ilustram a importância que os animais domésticos e o mercado *pet* têm no Brasil.

Essa tendência é reforçada pelo estilo de vida solitário visto nos centros urbanos, onde cães e gatos se tornam ainda mais importantes para seus tutores. A proximidade entre animais de estimação e seres humanos, resulta na transferência de tendências humanas para o mercado *pet* ([MAZON; MOURA, 2017](#)). Esse fenômeno, conhecido como "humanização", eleva os *pets* de "animais de companhia" a membros da família, fazendo com que os tutores tenham uma maior preocupação com sua saúde e invistam mais na sua alimentação de forma a promover uma boa nutrição e longevidade. Uma pesquisa sobre o mercado de alimentação *pet* no Brasil, publicada em [Silva et al. \(2021\)](#), teve como resultado que para 74% dos respondentes é relevante que a alimentação do seu *pet* seja natural.

Em meados de 2007, foi realizado nos Estados Unidos um *recall* de rações após ser detectado que muitos lotes tinham sido contaminados por melamina. O incidente fez com que uma grande quantidade de *pets* fosse contaminada e contraísse diversos problemas de saúde, resultando, em muitos casos, em óbitos ([Sociedade Brasileira de Zootecnia, 2010](#)). Esse evento foi um marco importante para o crescente interesse por

novas alternativas alimentares para cães e gatos que substituam as rações comerciais convencionais (FRANÇA, 2009). Desde então, diversos tutores e profissionais na área de nutrição animal dedicam-se a elaborar receitas personalizadas para seus animais de estimação.

A Alimentação Natural refere-se ao oferecimento de uma dieta elaborada com ingredientes tradicionalmente consumidos por seres humanos, como carnes, frutas e vegetais (OLIVEIRA; FISCHER, 2022). Este tipo de nutrição vai muito além das sobras do almoço, ou de petiscos normalmente oferecidos aos *pets*. Baseia-se em uma dieta balanceada e adaptada às necessidades de cada animal.

1.2 Justificativa

Embora apresente inúmeros benefícios, alguns problemas podem surgir com o uso dessa dieta, caso a mesma seja empregada sem o devido conhecimento e cuidado. Isso ocorre porque a maioria das pessoas não possui conhecimentos específicos sobre nutrição animal e comete erros ao tentar adotar a Alimentação Natural sem o apoio de profissionais ou o uso de ferramentas produzidas sob a orientação desses especialistas. Um estudo brasileiro verificou que 40% dos tutores avaliados não controlaram adequadamente a quantidade de ingredientes nas receitas (OLIVEIRA et al., 2014).

Diante do exposto, descuidos como os acordados anteriormente podem causar desequilíbrios nutricionais no animal e acarretar em problemas de saúde. A partir do momento que há falta ou excesso de nutrientes essenciais, como vitaminas, minerais, aminoácidos ou ácidos graxos, o *pet* pode apresentar, num período relativamente curto, piora de sua qualidade de vida pelo aparecimento de uma ou mais disfunções metabólicas, que vão se agravando com o passar do tempo (OLIVEIRA; FISCHER, 2022).

Outros problemas comuns quando se fala de Alimentação Natural são os inúmeros casos de receitas caseiras mal formuladas difundidas na internet. Um estudo brasileiro avaliou a composição de minerais de 100 alimentos caseiros para animais adultos saudáveis produzidos de acordo com receitas obtidas de *websites*, sendo 75 para cães e 25 para gatos. Das 100 dietas avaliadas, nenhuma atingiu os valores recomendados dos nutrientes avaliados, e mais de 84% das dietas apresentaram três ou mais nutrientes abaixo da recomendação (PEDRINELLI et al., 2019). Essa tendência também pode ser verificada em outros países. Nos Estados Unidos, pesquisadores analisaram dietas caseiras publicadas na internet para cães com câncer, e constataram que nenhuma atendia os níveis nutricionais mínimos exigidos pelo NRC (HEINZE; GOMEZ; FREEMAN, 2012). Um estudo similar avaliou 67 dietas caseiras para cães e gatos com doença renal crônica, e concluiu que nenhuma alcançou os níveis mínimos recomendados pelo NRC (LARSEN et al., 2012).

Esses resultados são alarmantes e apontam a necessidade do uso de ferramentas apropriadas quando se deseja praticar a Alimentação Natural. Para a correta formulação de dietas caseiras, é relevante o uso de produtos de software destinados para essa finalidade (OLIVEIRA; FISCHER, 2022). Claramente, já existem ferramentas que buscam auxiliar na elaboração de receitas naturais para animais domésticos, tais como: Nutro VET, Feed Real Calculator, Funcional Pet, entre outros. Porém, após ser realizado um *Benchmarking* destas aplicações pelos autores (disponível no [Apêndice A](#)), foi verificado que nenhuma delas atende devidamente o seguinte caso de uso: simultaneamente, elas não são destinadas para os tutores dos *pets*, nem mesmo utilizam-se das guias nutricionais oficiais como base para elaboração das receitas.

Nesse cenário, torna-se pertinente a criação de uma aplicação que auxilie tutores de cães e gatos interessados em praticar a Alimentação Natural a formular receitas nutricionalmente balanceadas, seguindo os principais guias internacionais de nutrição para animais domésticos de maneira fácil e intuitiva.

1.3 Questões de Pesquisa e Desenvolvimento

Este trabalho tem como principal intuito apoiar os tutores de *pet* que desejam prover uma Alimentação Natural para os seus *pets*, sendo essa balanceada e orientada de acordo com os principais guias de nutrição para animais domésticos. Para isso, desenvolveu-se uma aplicação que procura responder aos seguintes questionamentos:

- Como viabilizar um produto de software que auxilie tutores de *pets* que praticam Alimentação Natural a criarem receitas customizadas e nutricionalmente balanceadas para seus *pets*?
- Como balancear receitas para *pets* de maneira automatizada e em atendimento aos principais requisitos nutricionais vigentes?
- Como garantir que o produto tenha uma boa usabilidade para o público alvo, que não possui conhecimentos técnicos no domínio teórico da nutrição animal?

1.4 Objetivos

Buscou-se cumprir tanto com o Objetivo Geral quanto com os Objetivos Específicos, conforme descrito nos subtópicos a seguir.

1.4.1 Objetivo Geral

Apoio computacional para tutores de *pets* que desejam prover uma Alimentação Natural para os seus *pets*, sendo essa balanceada e orientada de acordo com os principais guias de nutrição para animais domésticos.

1.4.2 Objetivos Específicos

1. Levantamento do domínio de interesse do projeto, com destaque aos requisitos para a elaboração de dietas de Alimentação Natural para *pets* orientada aos principais guias de nutrição vigentes;
2. Estudo quanto às principais particularidades dos cálculos envolvidos no balanceamento das receitas, procurando uma solução algorítmica;
3. Estudo sobre *Progressive Web Apps* como alternativa a aplicações específicas de plataforma;
4. Desenvolvimento do produto de software em atendimento aos levantamentos e estudos anteriores;
5. Análise preliminar dos resultados obtidos, e
6. Documentação da solução, revelando levantamento, estudos, desenvolvimento e análise realizados.

1.5 Organização da Monografia

Este Trabalho de Conclusão de Curso está subdividido nos seguintes capítulos:

- **Capítulo 2 - Referencial Teórico:** apresenta os fundamentos teóricos do trabalho, especialmente em relação à Alimentação Natural para *pets* e à Nutrição Animal, além de tópicos mais técnicos de Engenharia de Software (ex. Algoritmos de Programação Dinâmica; Requisitos; Usabilidade de Software e *Progressive Web App*(PWAs));
- **Capítulo 3 - Suporte Tecnológico:** detalha as tecnologias empregadas no trabalho, abrangendo o desenvolvimento de software, a gestão do projeto, e a configuração de software;
- **Capítulo 4 - Metodologia:** apresenta os aspectos metodológicos sobre o levantamento bibliográfico, o desenvolvimento do software e a análise de resultados, além de apresentar o cronograma de atividades;

- **Capítulo 5 - NattyPet:** descreve em detalhes a solução, sendo essa um software dedicado;
- **Capítulo 6 - Análise de Resultados:** apresenta os resultados obtidos com o desenvolvimento do software, e
- **Capítulo 7 - Conclusão:** apresenta as considerações finais desse trabalho, bem como propostas para trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo tem como intuito mostrar os fundamentos teóricos em relação à área de Nutrição Animal, com ênfase na alimentação de *pets*, e conceitos da área de Engenharia de Software que foram utilizados no decorrer deste Trabalho de Conclusão de Curso. Na seção de [Nutrição Animal](#), aborda-se a fundamentação teórica acerca da área de Nutrição Animal. Nesta seção, são definidos conceitos de nutrição animal e Alimentação Natural; quais os principais órgãos regulamentadores de nutrição de animais domésticos e suas diretrizes; como realizar a leitura de tabelas nutricionais, e o como se dá o cálculo de nutrientes para dietas balanceadas. Em [Engenharia de Software](#), consta o material teórico referente ao domínio de Engenharia de Software. Nesta seção, são definidos: conceitos de requisitos de software; usabilidade; o que são *Progressive Web Apps* (PWA) e algoritmos orientados à Programação Linear. Ao final, tem-se o [Resumo do Capítulo](#).

2.1 Nutrição Animal

Seguem detalhes sobre [Órgãos Regulamentadores](#) no contexto da nutrição de animais domésticos; definições sobre o regime de [Alimentação Natural](#); e as principais orientações das [Guias Nutricionais](#) oficiais sobre o balanceamento da dieta de cães e gatos.

2.1.1 Órgãos Regulamentadores

Com o crescente interesse por uma alimentação mais natural e saudável para animais de estimação, o setor de *pet food* tem se expandindo para atender a demandas cada vez mais exigentes por padrões de qualidade e segurança ([SILVA et al., 2021](#)). Com isso, Organizações Regulamentadoras desempenham um papel relevante, garantido que os produtos e as receitas estejam alinhados com as necessidades nutricionais de animais domésticos como cães e gatos ([OLIVEIRA; FISCHER, 2022](#)). Dentre essas organizações, destacam-se três referências principais: a FEDIAF, Federação Europeia da Indústria de Alimentos para Animais de Estimação; a AAFCO, *Association of American Feed Control Officials*; e o NRC, *National Research Council*, dos Estados Unidos.

A FEDIAF é uma referência em termos de padronização e orientação de boas práticas em fabricação, elaboração e comercialização de produtos alimentícios para animais. A organização estabelece diretrizes que asseguram que alimentos para *pets* sigam normas de qualidade, segurança e transparência, colaborando para um setor que preza pela saúde e pelo bem-estar dos animais ([FEDIAF, 2024](#)).

A AAFCO é uma organização sem fins lucrativos nos Estados Unidos, que desempenha um papel fundamental na regulamentação da indústria de alimentos para animais, incluindo *pets*. Embora não crie leis, a AAFCO estabelece diretrizes e padrões nutricionais que são amplamente adotados como referências por órgãos reguladores estaduais e federais (AAFCO, 2013).

Por fim, o NRC é responsável pelas principais publicações sobre nutrição animal no mundo, e sua edição de recomendações nutricionais para cães e gatos tornou-se uma das principais referências na medicina veterinária (National Research Council (NRC), 2006).

2.1.2 Alimentação Natural

O termo "alimento completo", no contexto da nutrição animal, refere-se a um tipo de alimento que, devido à sua composição, é suficiente para fornecer uma porção diária (REGULATION..., 2009 apud FEDIAF, 2024). Sendo assim, quando um alimento completo é fornecido como a única fonte de nutrientes por um período prolongado (cobrindo todo o estágio de vida), ele irá suprir todas as necessidades nutricionais de animais da espécie e o estado fisiológico para os quais o alimento é destinado (FEDIAF, 2024).

No Brasil, o que se conhece popularmente como Alimentação Natural, trata-se de uma alimentação preparada com ingredientes inteiros, e tradicionalmente voltados ao consumo humano, como carnes, ossos, frutas e vegetais. De maneira geral, procura-se evitar a utilização de ingredientes ultra-processados e com aditivos químicos, como rações comerciais tradicionais, dando preferência aos ingredientes mais frescos e naturais (BUFF et al., 2014 apud OLIVEIRA; FISCHER, 2022).

Contudo, vale ressaltar que o emprego do termo "natural", no contexto de alimentos para *pets*, pode variar entre tutores, empresas e órgãos regulamentadores. Na União Europeia, a FEDIAF define que o termo "natural" pode ser utilizado somente para se referir a ingredientes de origem vegetal, mineral ou animal desde que os mesmos não possuam aditivos químicos, mas podendo ter passado por processos físicos como congelamento, pasteurização ou secagem para torná-los adequados para consumo animal (FEDIAF, 2018). Já nos Estados Unidos, diversos estados seguem a regulamentação da AAFCO, que também defende o uso do termo para se referir a alimentos ou ingredientes de fontes vegetais ou animais em estado bruto ou submetidos a processos físicos, térmicos, de purificação, hidrólise ou fermentação, porém permitindo quantidades mínimas de substâncias sintéticas ou aditivos químicos, seguindo boas práticas de fabricação (AAFCO, 2013).

Diante do exposto, o software desenvolvido no presente trabalho procura viabilizar a formulação de receitas que atendam à definição de alimento completo e que sejam

compostas por ingredientes considerados naturais.

2.1.3 Guias Nutricionais

A FEDIAF, assim como a AAFCO e o NRC, proporciona guias nutricionais oficiais contendo as quantidades mínimas recomendadas para cada um dos nutrientes que são avaliados para determinar a adequação nutricional de um alimento. Tais nutrientes estão listados no Quadro 1.

Quadro 1 – Nutrientes Considerados na Avaliação de Receitas

Nutrientes principais	Proteína	Gordura	
Ácidos graxos	Ácido linoleico	Ácido araquidônico (gatos)	Ácido alfa-linolênico
	Ácido eicosapentaenoico (EPA)	Ácido docosahexaenoico (DHA)	
Aminoácidos	Arginina	Histidina	Isoleucina
	Cistina	Tirosina	Lisina
	Fenilalanina	Treonina	Triptofano
	Leucina	Metionina	Valina
Minerais	Cálcio	Fósforo	Potássio
	Sódio	Cobre	Ferro
	Cloro	Magnésio	Iodo
	Manganês	Zinco	Selênio
Vitaminas	Vitamina A	Vitamina D	Vitamina E
	Tiamina	Riboflavina	Ácido pantotênico
	Niacina	Vitamina B6 (Piridoxina)	Biotina
	Cobalamina	Ácido fólico	Vitamina K
Substâncias semelhantes a vitaminas	Taurina (gatos)	Colina	

Fonte: [FEDIAF \(2024\)](#) (Formatação: Autores)

As recomendações oficiais, expressas nas tabelas a seguir (Tabela 1 e Tabela 2), informam as recomendações mínimas de nutrientes para cães e gatos levando em consideração diferentes estágios de vida e fases de reprodução. Além das recomendações mínimas, as tabelas também informam os limites legais máximos para determinados nutrientes.

Tabela 1 – Níveis Nutricionais Mínimos Recomendados para Cães

Nutriente		Nível Mínimo Recomendado				Nível Máximo
		Adulto com NEM de		Crescimento inicial e Reprodução	Crescimento final	(L) = Limite legal da UE (N) = Nutricional
		95 kcal/kg ^{0,75}	110 kcal/kg ^{0,75}			
Proteína	g	52,10	45,00	62,50	50,00	-
Arginina	g	1,51	1,30	2,04	1,84	-

Nutriente		Nível Mínimo Recomendado				Nível Máximo
		Adulto com NEM de		Crescimento inicial e Reprodução	Crescimento final	(L) = Limite legal da UE
		95 kcal/kg ^{0,75}	110 kcal/kg ^{0,75}			(N) = Nutricional
Histidina	g	0,67	0,58	0,98	0,63	-
Isoleucina	g	1,33	1,15	1,63	1,25	-
Leucina	g	2,37	2,05	3,23	2,00	-
Lisina	g	1,22	1,05	2,20	1,75	Crescimento: 7,00 (N)
Metionina	g	1,16	1,00	0,88	0,65	-
Metionina + Cistina	g	2,21	1,91	1,75	1,33	-
Fenilalanina	g	1,56	1,35	1,63	1,25	-
Fenilalanina + Tiro-sina	g	2,58	2,23	3,25	2,50	-
Treonina	g	1,51	1,30	2,03	1,60	-
Triptofano	g	0,49	0,43	0,58	0,53	-
Valina	g	1,71	1,48	1,70	1,40	-
Gordura	g	13,75	13,75	21,25	21,25	-
Ácido Linoleico (ω -6)*	g	3,82	3,27	3,25	3,25	Crescimento inicial: 16,25(N)
Ácido Araquidônico (ω -6)	mg	-	-	75,00	75,00	-
Ácido Alfa-linolênico (ω -3)*	g	-	-	0,20	0,20	-
EPA + DHA (ω -3)*	g	-	-	0,13	0,13	-
Minerais						
Cálcio	g	1,45	1,25	2,50	2,50	Adulto: 6,25(N) Crescimento inicial: 4(N) Crescimento final: 4,50(N)
Fósforo	g	1,16	1,00	2,25	1,75	Adulto: 4,00(N) Adulto: 2/1(N) Crescimento inicial e reprodução: 1,6/1(N) Crescimento final: 1,8/1
Relação Ca/P				1/1		
Potássio	g	1,45	1,25	1,10	1,10	-
Sódio	g	0,29	0,25	0,55	0,55	-
Cloreto	g	0,43	0,38	0,83	0,83	-
Magnésio	g	0,20	0,18	0,10	0,10	-
Microelementos*						
Cobre	mg	2,08	1,80	2,75	2,75	(L)
Iodo	mg	0,30	0,26	0,38	0,38	(L)
Ferro	mg	10,40	9,00	22,00	22,00	(L)
Manganês	mg	1,67	1,44	1,40	1,40	(L)

Nutriente		Nível Mínimo Recomendado				Nível Máximo
		Adulto com NEM de		Crescimento inicial e Reprodução	Crescimento final	(L) = Limite legal da UE
		95 kcal/kg ^{0,75}	110 kcal/kg ^{0,75}			(N) = Nutricional
Selênio	mg	87,00	75,00	100,00	100,00	(L)
Zinco	mg	20,80	18,00	25,00	25,00	(L)
Vitaminas						
Vitamina A	UI	1.754,00	1.515,00	1.250,00	1.250,00	100.000 (N)
Vitamina D	UI	159,00	138,00	138,00	138,00	(L) 800 (N)
Vitamina E	UI	10,40	9,00	12,50	12,50	-
Tiamina	mg	0,62	0,54	0,45	0,45	-
Riboflavina	mg	1,74	1,50	1,05	1,05	-
Ácido Pantotênico	mg	4,11	3,55	3,00	3,00	-
Vitamina B6 (Piridoxina)	mg	0,42	0,36	0,30	0,30	-
Vitamina B12	μg	9,68	8,36	7,00	7,00	-
Niacina	mg	4,74	4,09	3,40	3,40	-
Ácido Fólico	μg	74,70	64,50	54,00	54,00	-
Biotina	μg	-	-	-	-	-
Colina	mg	474,00	409,00	425,00	425,00	-
Vitamina K	μg	-	-	-	-	-

Fonte: [FEDIAF \(2024\)](#) (Formatação: Autores)

Tabela 2 – Níveis Nutricionais Mínimos Recomendados para Gatos

Nutriente		Nível Mínimo Recomendado			Nível Máximo
		Adulto com NEM de		Crescimento e Reprodução	(L) = Limite legal da UE
		75 kcal/kg ^{0,67}	100 kcal/kg ^{0,67}		(N) = Nutricional
Proteína	g	83,30	62,50	70/75	-
Arginina	g	3,30	2,50	2,68/2,78	Crescimento: 8,75(N)
Histidina	g	0,87	0,65	0,83	-
Isoleucina	g	1,44	1,08	1,35	-
Leucina	g	3,40	2,55	3,20	-
Lisina	g	1,13	0,85	2,13	-
Metionina	g	0,57	0,43	1,10	Crescimento: 3,25 (N)
Metionina + Cistina	g	1,13	0,85	2,20	-
Fenilalanina	g	1,33	1,00	1,25	-
Fenilalanina + Tirosina	g	5,11	3,83	4,78	-
Treonina	g	1,73	1,30	1,63	-
Triptofano	g	0,44	0,33	0,40	Crescimento: 4,25 (N)
Valina	g	1,70	1,28	1,60	-
Taurina (alimento úmido)	g	0,67	0,50	0,63	-
Taurina (alimento seco)	g	0,33	0,25	0,25	-
Gordura	g	22,50	22,50	22,50	-
Ácido Linoleico (ω -6)	g	1,67	1,25	1,38	-
Ácido Araquidônico (ω -6)	mg	20,00	15,00	50,00	-

Nutriente		Nível Mínimo Recomendado			Nível Máximo
		Adulto com NEM de		Crescimento e Reprodução	(L) = Limite legal da UE
		75 kcal/kg ^{0,67}	100 kcal/kg ^{0,67}		(N) = Nutricional
Ácido Alfa-linolênico (ω -3)	g	-	-	0,05	-
EPA + DHA (ω -3)	g	-	-	0,03	-
Minerais					
Cálcio	g	1,33	1,00	2,50	-
Fósforo	g	0,85	0,64	2,10	-
Relação Ca/P			1/1		Crescimento: 1,5/1 (N) Adulto: 2/1 (N)
Potássio	g	2,00	1,50	1,50	-
Sódio	g	0,25	0,19	0,40	-
Cloro	g	0,39	0,29	0,60	-
Magnésio	g	0,13	0,10	0,13	-
Microelementos					
Cobre	mg	1,67	1,25	2,50	(L)
Iodo	mg	0,43	0,33	0,45	(L)
Ferro	mg	26,70	20,00	20,00	(L)
Manganês	mg	1,67	1,25	2,50	(L)
Selênio (dietas úmidas)	mg	87,50	65,00	75,00	(L)
Selênio (dietas secas)	mg	70,00	52,50	75,00	(L)
Zinco	mg	25,00	18,80	18,80	(L)
Vitaminas					
Vitamina A	UI	1111,00	833,00	2250,00	Adulto e crescimento: 100.000 (N) Reprodução: 83.325 (N)
Vitamina D	UI	83,30	62,50	70,00	(L) 7.500,00 (N)
Vitamina E	UI	12,70	9,50	9,50	-
Tiamina	mg	1,47	1,10	1,40	-
Riboflavina	mg	1,05	0,80	0,80	-
Ácido Pantotênico	mg	1,92	1,44	1,43	-
Vitamina B6 (Piridoxina)	mg	0,83	0,63	0,63	-
Vitamina B12	μ g	5,87	4,40	4,50	-
Niacina	mg	10,50	8,00	8,00	-
Ácido Fólico	μ g	183,30	137,50	140,00	-
Biotina	μ g	2,67	2,00	2,00	-
Colina	mg	2,67	2,00	2,00	-
Vitamina K	μ g	-	-	-	-

Fonte: [FEDIAF \(2024\)](#) (Formatação: Autores)

Os valores mínimos e máximos (quando presentes) estão expressos nas tabelas em função de uma média diária de ingestão de energia. Com isso, as quantidades recomendadas de cada nutriente ocorrem de acordo com a quantidade de calorias recomendada para determinado animal, ou seja, de acordo com suas necessidades energéticas. Nas tabelas, são utilizadas médias de 95 kcal/kg^{0,75} ou 110 kcal/kg^{0,75} para cães, e 75 kcal/kg^{0,67} ou 100 kcal/kg^{0,67} para gatos. No entanto, esses valores são apenas referências para casos

gerais, e a demanda calórica de cada animal deve ser calculada levando em consideração suas características particulares. Essas particularidades são detalhadas nas próximas seções.

2.1.4 Necessidades Energéticas

Para que as necessidades energéticas de cada animal sejam estimadas com precisão, destacam-se duas métricas importantes: o Peso Metabólico do Animal e seu Escore de Condição Corporal. Esta seção busca conceituar essas métricas e explicar sua relevância.

2.1.4.1 Peso Metabólico

Como colocado anteriormente, os requisitos nutricionais de animais domésticos estão diretamente relacionados às suas necessidades energéticas, ou seja, à quantidade de calorias consumidas diariamente. Essas necessidades, por sua vez, dependem do peso corporal de cada indivíduo.

No entanto, a definição das necessidades energéticas de cães e gatos com base somente no peso corporal não representa de forma exata a real necessidade destes animais. Assim, do ponto de vista fisiológico, a necessidade energética não está diretamente relacionada somente ao Peso Corporal (PC), mas ao peso corporal elevado à uma potência (p), que corresponde ao peso metabólico (PC^p) (CARCIOFI, 2019). De acordo com FEDIAF (2024), considera-se as unidades de peso metabólico como $kg^{0,75}$ para cães e $kg^{0,67}$ para gatos.

2.1.4.2 Escore de Condição Corporal (ECC)

Cerca de um terço dos cães e gatos acima de um ano de idade apresentados em visitas veterinárias nos Estados Unidos são considerados acima do peso ou obesos. Essa parcela aumenta para 50% para indivíduos entre 6 e 11 anos (LUND, 2005 apud OLIVEIRA; FISCHER, 2022). Na Europa, a situação é bastante similar (COLLIARD et al., 2006 apud OLIVEIRA; FISCHER, 2022).

Esses dados servem para ilustrar um fato importante: os requisitos energéticos dos animais devem ser baseados no peso ideal, que não necessariamente corresponde ao seu peso corporal atual. Embora o peso corporal (massa) seja uma medida importante, de maneira isolada, esse dado não fornece informações suficientes para determinar se o peso do indivíduo está ideal ou não. Dessa forma, analisar a condição corporal (*body condition*), em conjunto com o peso corporal, fornece uma forma de avaliação mais precisa da condição geral do animal e uma melhor base para determinar suas necessidades energéticas.

Um *body condition score*, ou Escore de Condição Corporal (ECC), trata-se de um método subjetivo e semiquantitativo de analisar a composição corporal de um ani-

mal, em específico para mensurar seu percentual de gordura corporal (%BF) e estimar o quanto o indivíduo está acima ou abaixo de seu peso ideal (FEDIAF, 2024). Ao longo dos anos, diferentes sistemas de ECC foram desenvolvidos. Um sistema em destaque é o que procura avaliar animais em uma escala entre 1 e 9. Esse sistema propõe uma série de categorias, onde cada uma corresponde a um determinado % BF (BURKHOLDER, 2000 apud FEDIAF, 2024). As Tabelas 3 e 4 mostram as categorias de ECC para cães e gatos juntamente com sua descrição, percentuais de BF correspondentes, e o quanto cada categoria está abaixo ou acima do peso corporal ideal. Este sistema foi validado tanto para cães como para gatos, demonstrando uma boa previsibilidade e repetibilidade (SILVA, 2021).

Tabela 3 – Guia para Escore de Condição Corporal em Gatos

Pontuação	Características da região	Gordura Corporal Estimada (%)	% do PC abaixo ou acima do ECC 5
1. Extremamente magro	Costelas e proeminências ósseas visíveis e facilmente palpáveis, sem cobertura de gordura. Grande curvatura abdominal e formato de ampulheta exagerado.	$\leq 10\%$	- $\geq 40\%$
2. Muito magro	Costelas e proeminências ósseas visíveis em gatos de pelagem curta, sem cobertura de gordura. Curvatura abdominal severa e forma de ampulheta acentuada.	5-15%	- 30-40%
3. Magro	Costelas e proeminências ósseas facilmente palpáveis com mínima cobertura de gordura. Curvatura abdominal acentuada e cintura visível.	10-20%	- 20-30%
4. Levemente abaixo do peso	Costelas e proeminências ósseas facilmente palpáveis com cobertura mínima de gordura. Curvatura abdominal e cintura bem proporcionada.	15-25%	- 10-15%
5. Ideal	Costelas e proeminências ósseas palpáveis com leve cobertura de gordura. Curvatura abdominal e cintura bem proporcionada.	20-30%	0%
6. Levemente acima do peso	Costelas e proeminências ósseas palpáveis sob cobertura moderada de gordura. Curvatura abdominal e cintura menos pronunciadas.	25-35%	+10-15%
7. Acima do peso	Costelas e proeminências ósseas palpáveis sob cobertura moderada de gordura. Sem curvatura abdominal, mas com gordura abdominal visível.	30-40%	+20-30%
8. Obeso	Costelas e proeminências ósseas difíceis de palpar, com espessa cobertura de gordura. Acúmulo abdominal visível.	35-45%	+30-40%
9. Grossamente obeso	Costelas e proeminências ósseas muito difíceis de palpar. Grande acúmulo abdominal com depósitos de gordura no rosto, pescoço e membros.	$> 45\%$	+ $> 40\%$

Fonte: FEDIAF (2024) (Formatação: Autores)

Tabela 4 – Guia para Escore de Condição Corporal em Cães

Pontuação	Características da região	Gordura Corporal Estimada (%)	% do abaixo ou do ECC 5	PC ou do
1. Extremamente magro	Costelas e outras proeminência ósseas: Visíveis à distância, facilmente palpáveis, sem camada de gordura. Abdômen: Visto da lateral, curvatura abdominal severa e visto por cima, apresenta forma de ampolheta exageradamente acentuada. Base da cauda: Estruturas ósseas em relevo e proeminentes, sem tecido entre a pele e o osso. Perda óbvia de massa muscular e sem gordura corporal perceptível.	$\leq 4\%$	- $\geq 40\%$	
2. Muito magro	Costelas e outras proeminência ósseas: Visíveis e facilmente palpáveis, sem camada de gordura sob a pele. Abdômen: Visto da lateral, curvatura abdominal marcada e visto por cima, apresenta forma acentuada de ampolheta. Base da cauda: Estruturas ósseas em relevo e proeminentes sem tecido entre a pele e o osso. Perda mínima de massa muscular.	4-10%	- 30-40%	
3. Magro	Costelas e outras proeminência ósseas: Perceptíveis e facilmente palpáveis, com uma fina camada de gordura. Abdômen: Visto da lateral, curvatura abdominal pronunciada e visto por cima, cintura acentuada. Base da cauda: Estruturas ósseas proeminentes com pouco tecido entre a pele e o osso.	5-15%	- 20-30%	
4. Levemente abaixo do peso	Costelas e outras proeminência ósseas: Facilmente palpáveis, com uma fina camada de gordura. Abdômen: Visto da lateral apresenta curvatura abdominal e visto por cima, cintura ligeiramente acentuada. Base da cauda: Estruturas ósseas em relevo com pouco tecido subcutâneo.	10-20%	- 10-15%	
5. Ideal	Costelas e outras proeminência ósseas: Costelas não visíveis, mas facilmente palpáveis, com fina camada de gordura. Outras proeminências ósseas são palpáveis com uma ligeira camada de gordura. Abdômen: Visto da lateral apresenta curvatura abdominal e visto por cima, cintura lombar bem proporcional. Base da cauda: Contornos suaves ou algum espessamento, estruturas ósseas apalpáveis sob uma fina camada de gordura subcutânea.	15-25%	0%	
6. Levemente acima do peso	Costelas e outras proeminência ósseas: s Palpáveis, com uma camada de gordura moderada. Abdômen: Visto da lateral, curvatura abdominal não muito evidente e visto por cima, cintura não muito acentuada. Base da cauda: Contornos suaves ou algum espessamento, estruturas ósseas permanecem palpáveis sob uma camada moderada de gordura subcutânea.	20-30%	+10-15%	

Pontuação	Características da região	Gordura Corporal Estimada (%)	% do PC abaixo ou do ECC 5
7. Acima do peso	Costelas e outras proeminência ósseas: Díficeis de palpar, com uma camada espessa de gordura. Abdômen: Visto da lateral, pouca curvatura abdominal e visto por cima, não tem muita cintura e a região lombar está ligeiramente alargada quando vista de cima. Base da cauda: Contornos suaves ou algum espessamento, estruturas ósseas permanecem palpáveis sob a camada de gordura subcutânea.	25-35%	+20-30%
8. Obeso	Costelas e outras proeminência ósseas: Muito difícil de palpar as costelas, com uma camada espessa de gordura. Outras proeminências ósseas apresentam-se distendidas, com amplos depósitos de gordura. Base da cauda: Apresenta-se espessada, com estruturas ósseas difíceis de palpar. Geral: Apresenta protuberância ventral no abdômen, sem cintura e região lombar alargada quando visto de cima. Depósitos de gordura na área lombar e no pescoço.	30-40%	+30-45%
9. Grossamente obeso	Costelas e outras proeminência ósseas: Muito difícil de palpar as costelas, com uma camada espessa de gordura; outras proeminências ósseas apresentam distensão com amplos depósitos de gordura entre ossos e pele. Base da cauda: Espessa, praticamente impossível palpar as estruturas ósseas. Geral: Apresenta protuberância ventral pendular no abdômen, sem cintura e região lombar acentuadamente alargada quando visto de cima. Depósitos de gordura sobre a área lombar, pescoço, face, membros e na região da virilha. Pode formar uma dobra nas costas, quando a gordura lombar e torácica avançam dorsalmente.	> 40%	+> 45%

Fonte: [FEDIAF \(2024\)](#) (Formatação: Autores)

Baseado no estudo de [Kealy et al. \(1992 apud SILVA, 2021\)](#), é recomendado que cães sejam alimentados a fim de manter uma pontuação entre 4 e 5 no sistema ECC de 9 pontos, visando saúde ideal e longevidade. Para gatos, é recomendado que os gatos castrados devem ser alimentados para manter uma pontuação de 4 pontos no mesmo sistema ([BJORNVAD et al., 2011](#)).

Desse modo, a combinação do peso corporal com o sistema ECC de 9 pontos fornece uma boa base para determinar requisitos energéticos. Nesse sentido, o sistema torna-se uma ferramenta útil para ajudar tutores, que muitas vezes falham em reconhecer que seus pets estão acima do peso ou mesmo obesos ([CARCIOFI, 2019](#)). O próprio NRC refere-se ao sistema ECC de 9 pontos como uma das referências utilizadas para determinar a Necessidade Energética de Manutenção (NEM) de gatos adultos ([National](#)

[Research Council \(NRC\), 2006](#)). A Associação Veterinária Mundial de Pequenos Animais (WSAVA) também faz menção a esse sistema, tendo incluído-o em guias nutricionais globais. Além disso, um estudo mostrou que tutores também podem ganhar experiência com sistemas ECC a ponto de alcançarem sozinhos avaliações com precisão satisfatória de seus *pets* ([GERMAN et al., 2006](#)).

2.1.4.3 Relevância do Peso Metabólico e ECC

Como demonstrado, a análise de condição corporal através do sistema ECC de 9 pontos, bem como a utilização do peso metabólico, mostram-se métricas essenciais para o devido cálculo de necessidades energéticas. O número de calorias diárias que um determinado animal deve consumir se dá em relação ao peso metabólico do animal. Este valor, por sua vez, deve ser calculado levando em consideração o peso corporal ideal do animal em questão. Ou seja, a partir da classificação de ECC, determina-se o peso corporal ideal do animal, que então é utilizado para calcular seu peso metabólico. Este peso é utilizado em conjunto com uma série de outros fatores que influenciam o cálculo da necessidade energética dos *pets*. A próxima seção busca explicar esses fatores.

2.1.5 Fatores de Influência

A Necessidade Energética de Manutenção (NEM) é a quantidade de energia consumida por um animal adulto moderadamente ativo. Consiste na Taxa Metabólica Basal (TMB) mais o gasto energético para obtenção, digestão e absorção de alimentos nas quantidades necessárias para manter o peso corporal ideal ([FEDIAF, 2024](#)).

Como exposto anteriormente, a quantidade total de calorias a serem consumidas diariamente é diretamente ligada ao peso metabólico do animal. Entretanto, independente do peso corporal, a NEM também é fortemente influenciada por outros fatores como idade, fase de reprodução e atividade física. Nas próximas seções, são apresentadas colocações sobre esses fatores.

2.1.5.1 Atividade Física

É evidente que o nível médio de atividade física exercido por um animal influencia de maneira significativa sua NEM. Basta considerar que, por exemplo, ficar em pé requer 40% mais energia do que ficar deitado ([MEYER; ZENTEK, 2005](#)). De forma análoga, um cão doméstico terá necessidades energéticas bem diferentes que outro cão de peso similar, mas que trabalha diariamente puxando trenós por longas distâncias. É importante considerar o cenário de cada indivíduo, e não somente utilizar a recomendação padrão de NEM disponibilizada nos guias oficiais. Afinal, a recomendação padrão pode ser alta demais para muitos cães domésticos, uma vez que quase um quarto dos tutores exercitam

seus cachorros menos de três horas por semana (SLATER et al., 1995 apud FEDIAF, 2024).

A Tabela 5 fornece exemplos de requisitos energéticos diários para cães com diferentes níveis de atividade física. Para gatos, a FEDIAF e o NRC consideram uma variação menor, especificando apenas duas categorias: gatos domésticos/castrados e gatos ativos (FEDIAF, 2024). Essa relação encontra-se na Tabela 6.

Tabela 5 – Recomendações de NED para Cães em Relação à Atividade

Nível de Atividade	kcal EM/kg ^{0,75}	kJ EM/kg ^{0,75}
Baixa atividade (<1 h/dia) (ex.: caminhada com coleira)	95	398
Atividade moderada (1 – 3 h/dia) (atividade de baixo impacto)	110	460
Atividade moderada (1 – 3 h/dia) (atividade de alto impacto)	125	523
Alta atividade (3 – 6 h/dia) (cães de trabalho, ex.: cães pastores)	150-175	628-732
Alta atividade em condições extremas (cães de trenó em corridas, 168 km/dia no frio extremo)	860-1240	3600-5190
Adultos propensos à obesidade	≤ 90	≤ 377
Diferenças específicas de raça:		
Dogues alemães	200 (200-250)	837 (837-1046)
Cães Terra-nova	105 (80-132)	439 (335-550)

Fonte: FEDIAF (2024) (Formatação: Autores)

Tabela 6 – Requisitos Médios Diários de Energia para Gatos Adultos

Condição	kcal EM/kg ^{0,67}	kJ EM/kg ^{0,67}
Gatos castrados e/ou de interior	52-75	215-314
Gatos ativos	100	418

Fonte: FEDIAF (2024) (Formatação: Autores)

2.1.5.2 Idade

Para gatos, o fator idade não se mostra tão relevante quanto para cães, de acordo com os guias da FEDIAF (2024). Uma vez que os filhotes chegam a um ano de vida, passa-se a determinar a NEM com base em fatores como nível de atividade (discutido anteriormente) e fase de reprodução.

Para cães, podem ser distinguidos três grupos em relação ao estágio de vida: cães com um a dois anos de idade; cães adultos comuns (três a sete anos), e cães com mais de sete anos de idade (FINKE, 1994 apud FEDIAF, 2024). Cães adultos jovens, com menos de dois anos de idade, necessitam de mais energia porque são mais ativos e, apesar de terem um peso semelhante ao de indivíduos mais velhos da mesma raça, ainda podem estar em desenvolvimento (MEYER; ZENTEK, 2005). Animais mais velhos precisam de menos calorias devido à diminuição da atividade: cães com mais de sete anos de idade podem precisar de 10 a 15% menos energia do que cães com três a sete anos (FEDIAF, 2024).

Portanto, as recomendações práticas devem estar sempre relacionadas com a idade (CARCIOFI, 2019). A Tabela 7 fornece NEM's para diferentes faixas etárias, sem levar em consideração diferentes níveis de atividade.

Tabela 7 – Recomendações Práticas para NEM em Cães de Diferentes Idades.

Idade (anos)	kcal EM/kg ^{0,75}	kJ EM/kg ^{0,75}
1 – 2	130 (125-140)	550 (523-585)
3 – 7	110 (95-130)	460 (398-545)
> 7 (cães idosos)	95 (80-120)	398 (335-500)

Fonte: FEDIAF (2024) (Formatação: Autores)

2.1.5.3 Fases de Crescimento e Reprodução

Uma estimativa precisa da necessidade de Energia Metabolizável (EM) é essencial para garantir recomendações alimentares ideais para apoiar o crescimento saudável e evitar o fornecimento excessivo de energia a animais jovens (FEDIAF, 2024).

Foi demonstrado que a superalimentação de filhotes leva ao crescimento acelerado, o que pode resultar em deformidades esqueléticas, especialmente em raças grandes e gigantes (DOBENECKER; KÖSTLIN; MATIS, 1998 apud FEDIAF, 2024). Consequentemente, filhotes nunca devem ser alimentados sem restrições.

A Tabela 8 fornece necessidades médias de energia durante o crescimento e reprodução em cães. No caso de filhotes, é recomendado que tutores pesem regularmente seus cachorros a fim de ajustar a alocação de alimentos (ou seja, energia), de modo a garantir um crescimento saudável (FEDIAF, 2024). Se tratando da fase de lactação, as necessidades energéticas dependem do tamanho da ninhada. A tabela fornece as equações para calcular as necessidades energéticas médias de cadelas lactantes em diferentes estágios da lactação. As informações da tabela são baseadas em FEDIAF (2024).

Para gatos, as necessidades médias de energia durante fases crescimento e repro-

dução são fornecidas na Tabela 9, orientando-se por (FEDIAF, 2024).

Tabela 8 – Necessidades Energéticas durante Crescimento e Reprodução em Cães

Filhotes	Idade	Necessidade Energética
	Filhotes recém-nascidos	25 kcal/100g PC
	8 semanas a 1 ano	$(254,1 - 135,0 \times \frac{PC_{atual}}{PC_{maduro}}) \times (PC_{atual})^{0,75}$
Cadelas	Fase de reprodução	Necessidade Energética
Gestação	primeiras 4 semanas	132 kcal/kg PC ^{0,75}
	últimas 5 semanas	132 kcal/kg PC ^{0,75} + 26 kcal/kgPC
Lactação ¹	1 a 4 filhotes	145 kcal/kg PC ^{0,75} + 24n × PC × L
	5 a 8 filhotes	145 kcal/kg PC ^{0,75} + (96 + 12n - 4) × PC × L

¹ n = número de filhotes; L = 0,75 na semana 1 de lactação; 0,95 na semana 2; 1,1 na semana 3 e 1,2 na semana 4

Fonte: FEDIAF (2024) (Formatação: Autores)

Tabela 9 – Necessidades Energéticas durante Crescimento e Reprodução em Gatos

Gatos filhotes	Idade	Fator de multiplicação da NEM
	Até 4 meses	2,00-2,50
	4 a 9 meses	1,75-2,50
	9 a 12 meses	1,50
Gatas	Fase de reprodução	Necessidade Energética
Gestação		140 kcal/kg PC ^{0,67}
	< 3 filhotes	100 kcal/kg ^{0,67} + 18 kcal × kg PC × L
Lactação ¹	3-4 filhotes	100 kcal/kg ^{0,67} + 60 kcal × kg PC × L
	> 4 filhotes	100 kcal/kg ^{0,67} + 70 kcal × kg PC × L

¹ L = 0,9 na primeira e segunda semanas de lactação; 1,2 na terceira e quarta semanas; 1,1 na quinta semana; 1 na sexta semana; e 0,8 na sétima semana.

Fonte: FEDIAF (2024) (Formatação: Autores)

2.1.6 Cálculo dos Nutrientes

Como mencionado na seção de Guias Nutricionais, organizações como a FEDIAF disponibilizam guias nutricionais que fornecem a recomendação dos principais nutrientes que devem ser considerados durante a elaboração de uma dieta balanceada para animais domésticos (em específico para cães e gatos). Essas recomendações podem ser visualizadas nas Tabelas 1 e 2. Essas tabelas contém as quantidades mínimas de cada nutriente para cada 1000 kcal de Energia Metabolizável (EM).

Para calcular o valor exato de cada nutriente que um determinado animal deve consumir, é preciso primeiro saber seu peso metabólico e a média de calorias por kg metabólico que ele deve consumir diariamente. O cálculo do peso metabólico consta na seção sobre [Peso Metabólico](#). Já a média de calorias por kg metabólico de cada indivíduo é calculada levando em consideração os fatores: atividade física, idade, e fase de crescimento/reprodução.

Em seguida, deve-se multiplicar o peso metabólico pela média de calorias, obtendo-se então o total de calorias que o animal deve consumir diariamente ([FEDIAF, 2024](#)). A determinação da quantidade equivalente de cada nutriente é realizada por meio de um cálculo de proporcionalidade linear, onde a relação entre as variáveis é representada pela expressão a seguir, conforme definido em [FEDIAF \(2024\)](#):

$$\frac{\text{Quantidade do Nutriente para } 1000 \text{ kcal}}{1000 \text{ kcal}} = \frac{\text{Quantidade Diária do Nutriente}}{\text{Total de Calorias Diárias}} \quad (2.1)$$

Aplicando na fórmula a quantidade de um nutriente X descrita na tabela (que corresponde ao valor para 1000 kcal) e o total de calorias previamente calculado, é possível determinar a quantidade diária do nutriente X que o *pet* deve consumir para uma dieta balanceada. O exemplo a seguir busca ilustrar esse cálculo.

Exemplo 1: Cão com 4 anos, 10kg, escore de condição corporal 5 (ideal – nem magro e nem gordo), passeia durante 15 minutos, 2 vezes ao dia (pouco ativo), macho e não-castrado:

- Baseado na idade e no nível de atividade física do animal, considera-se uma média de $95 \text{ kcal/kg}^{0,75}$;
- Sua NEM pode ser calculada por $\text{NEM} = 95 \text{ kcal} \times 10^{0,75}$;
- Logo, $\text{NEM} = 95 \times 5,62 = 534 \text{ kcal/dia}$;
- Para calcular, por exemplo, a quantidade de proteínas que este animal deve consumir diariamente, utiliza-se o valor correspondente na Tabela 1. Por fim, realiza-se o cálculo da equação 2.1:

$$\frac{52,10 \text{ g}}{1000 \text{ kcal}} = \frac{P}{534 \text{ kcal}}$$

- Conclui-se então que o animal necessita de 27,8 g de proteínas por dia.

2.2 Engenharia de Software

Por se tratar de um trabalho de desenvolvimento de um software, cabem reflexões sobre boas práticas da área de Engenharia de Software. Como o projeto tem como foco um domínio específico (nutrição animal) e é destinado para usuários sem conhecimento técnico neste domínio, é importante que o produto final seja bem especificado em termos de [Requisitos de Software](#) e [Usabilidade](#).

Além disso, cabem colocações sobre [Progressive Web Apps](#) (PWA), visto que essa tecnologia une a versatilidade e a acessibilidade de aplicações *web* com as vantagens de usabilidade de *apps* específicos de plataforma. Por fim, como o processo de balanceamento de dietas baseada em Nutrição Animal demanda a resolução de um problema de otimização, faz-se pertinente o estudo de algoritmos para este fim. Nesse contexto, este trabalho explora o uso da [Programação Linear](#), resolvida através do [Algoritmo Simplex](#), como técnica para prover uma solução otimizada para o problema de balanceamento. Detalhes complementares de cada tópico serão tratados nas seções a seguir.

2.2.1 Requisitos de Software

Os requisitos de software são descritos como as funcionalidades, os critérios qualitativos e as restrições que um sistema deve atender ou evitar ([LUTOWSKI, 2005](#)). No contexto da engenharia de requisitos, etapa inicial no processo de engenharia de software descrita por [Sommerville \(2016\)](#), o foco está em coletar, compreender, registrar, validar e gerenciar esses requisitos. Dentro desse processo, os requisitos são geralmente classificados em dois grandes grupos: funcionais e não funcionais. Os requisitos funcionais descrevem as funcionalidades esperadas do sistema, baseando-se no conhecimento fornecido pelo usuário sobre o domínio do negócio, como destacado por [Machado \(2011\)](#). Por outro lado, os requisitos não funcionais não estão diretamente relacionados às funcionalidades do sistema, mas sim às suas características e restrições de qualidade, tais como: desempenho, segurança, usabilidade e confiabilidade, também abordados pelo autor.

A definição e a classificação dos requisitos são etapas fundamentais para garantir que o sistema atenda às expectativas e necessidades do usuário final. Contudo, a forma como esses requisitos são documentados e gerenciados varia de acordo com a metodologia de desenvolvimento empregada. Neste trabalho, optou-se por utilizar metodologias ágeis, nas quais os requisitos funcionais são modelados como histórias de usuário.

De acordo com [Machado \(2011\)](#), histórias de usuário são simples descrições sob a perspectiva do usuário ou de alguma parte envolvida, representando funcionalidades como expressões de intenção do que se deseja alcançar. Essas histórias são compreensíveis tanto para os desenvolvedores, quanto para os usuários, e servem como uma ferramenta eficaz para definir o comportamento do sistema de maneira clara e colaborativa.

Para elicitación dos requisitos, sejam esses modelados usando histórias de usuário ou de outra forma, há necessidade de se aplicar técnicas de elicitación (ALFLEN; PRADO, 2021). Algumas técnicas muito conhecidas na comunidade de software são: entrevistas, questionários, casos de uso, protótipos, entre outros. Os autores sugerem o uso de protótipos, pois eles permitem validar e refinar os requisitos de software de forma iterativa e colaborativa com os *stakeholders*. Através de protótipos, é possível identificar possíveis alterações que devem ser incorporadas na próxima versão (ALFLEN; PRADO, 2021).

Já na modelagem, ao usar histórias de usuário, deve-se ter em mente a elaboração de um *Backlog* do Produto. Nesse caso, há boas práticas a serem seguidas, conforme consta em Sedano, Ralph e Péraire (2019). Dentre essas boas práticas, destacam-se:

- Utilização de níveis de granularidade na especificação do *Backlog* do Produto, como por exemplo: Tema, Épico e História de Usuário. Cabe ressaltar que neste TCC serão utilizados 5 níveis de granularidades, sendo eles: Tema, Épico, História de Usuário, Tarefa e Critério de Aceitação;
- Escrita Padronizada, principalmente, das histórias de usuário. Segundo Aguiar e Caroli (2020), um bom formato mantém o foco em três aspectos principais: "Quem?", "O que?" e "Por que?". Neste TCC, será utilizado o formato: *Como ... Desejo ... Para* Ou ainda: *Eu Como «papel» desejo poder «função» para conseguir «valor de negócio»*, e
- Priorização dos requisitos. Um exemplo de técnica muito usual na área é *MoSCoW* (HUDAIB et al., 2018). Apesar de não ser uma técnica tão precisa ou escalável para projetos maiores, como tratado por Hudaib et al. (2018), o *MoSCoW* apresenta vantagens como: facilidade no uso, em especial quando a quantidade de requisitos não é grandiosa; confiabilidade média; e alto nível de tolerância a falhas. Entende-se que essas características atendem o perfil de projeto deste TCC.

2.2.2 Usabilidade

Segundo a norma ABNT NBR ISO/IEC 25010 (ABNT, 2011), que substituiu a antiga série ISO/IEC 9126, a usabilidade é definida como o "grau em que um produto ou sistema pode ser usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto de uso específico". Essa definição moderna reforça a importância de um software que seja não apenas funcional, mas também facilmente assimilado e utilizado pelo usuário final. A norma detalha a usabilidade através das seguintes sub-características:

- Reconhecimento de Adequação: capacidade que permite aos usuários compreender se o software é adequado às suas necessidades;

- Apreensibilidade: capacidade do produto que permite ao usuário aprender a utilizar suas funções de forma rápida e eficaz;
- Operacionalidade: capacidade do produto que permite ao usuário operá-lo e controlá-lo com facilidade;
- Proteção Contra Erros do Usuário: capacidade do sistema de evitar que os usuários cometam erros ou de facilitar a recuperação caso eles ocorram;
- Estética da Interface do Usuário: capacidade da interface de ser agradável e satisfatória para o usuário, e
- Acessibilidade: capacidade do produto de ser utilizado por pessoas com as mais variadas características e capacidades, incluindo pessoas com deficiência.

Levando em consideração a natureza deste trabalho, bem como o público alvo para o qual o projeto se destina, um ponto de relevância é a simplificação da complexidade de processos como cálculos nutricionais e geração de receitas para usuários sem formação técnica na área de nutrição animal. Portanto, é essencial que o software proposto seja convidativo e de fácil aprendizado. As novas sub-características, como a **Proteção Contra Erros do Usuário** e a **Apreensibilidade**, tornam-se guias ainda mais precisos para garantir que os usuários se sintam confortáveis e motivados a utilizá-lo. Dessa forma, é pertinente que, em termos de usabilidade, o trabalho tenha como foco de atenção atender minimamente a todas as sub-características listadas.

2.2.2.1 Testes de Usabilidade

Como proposto por [Sauro e Lewis \(2012\)](#), existem dois tipos de teste de usabilidade, sendo eles, o teste formativo e o teste somativo. O teste formativo é realizado com intuito de identificar problemas de usabilidade. O teste somativo, por sua vez, tem como objetivo descrever a usabilidade da aplicação utilizando métricas. Com isso, os principais pontos a serem definidos e observados com estes testes são:

- Possuir uma amostra de usuários grande, entretando, dependendo do tamanho do software é possível ter uma amostra entre dois e cinco usuários;
- Ter uma amostra que seja representativa;
- Ter um método de coleta de dados, podendo ser uma observação do usuário interagindo com a aplicação, uma gravação da interação do usuário, entre outros;
- Estabelecer uma taxa de conclusão para se obter a informação referente ao usuário ter conseguido ou não realizar dada ação;

- Definir qual o tempo esperado para realização de uma tarefa, e quanto tempo os usuários levaram para completá-la, ou não, e
- Coletar o índice de satisfação do usuário para entender qual a percepção do usuário quanto o sistema, ou tarefa, que testou.

2.2.2.2 Pesquisa por Formulário

Sauro e Lewis (2012) definem a pesquisa por formulário como uma das principais formas de se medir aspectos de usabilidade e experiência de usuário em uma aplicação. Segundo os autores, a principal vantagem deste tipo de teste é a facilidade para coletar vários tipos de respostas, como: respostas binárias, comentários, notas em uma escala, entre outros. As principais formas para coletar essas informações de maneira eficiente em formulário são:

- Escala de Classificação: onde o usuário atribui uma nota à atividade realizada. O recomendado é que essa nota seja de um a cinco, e seja acompanhada de uma palavra, como por exemplo a Escala Likert (NORMAN, 2010), para facilitar a visualização do usuário e ter um ponto neutro;
- *Net Promoter Scores*: onde os usuários atribuem uma nota de 0-10 para a aplicação. Com isso, é possível definir a taxa de usuários: (i) promotores, sendo aqueles que irão divulgar e falar bem sobre a aplicação; (ii) neutros, sendo usuários que utilizam a aplicação, mas não observam nada muito espetacular, e (iii) detratores, sendo usuários que possuem potencial de difamar a aplicação, e
- Comentários e Dados Abertos: onde devem ser classificados em categorias, sendo recomendado que possuam potencial para serem quantificados em análises futuras.

2.2.3 Progressive Web Apps

Uma *Progressive Web App* (PWA) é uma aplicação criada com tecnologias web e aprimoradas com *Application Programming Interface* (APIs) para oferecer uma experiência de usuário similar à de uma aplicação específica de plataforma (MDN Web Docs, 2024b). Assim como um *website*, uma PWA pode ser executada em várias plataformas e dispositivos a partir de uma única base de código. Esse tipo de aplicação combina o amplo alcance de *apps* da *Web* com características de *apps* específicos da plataforma, podendo ser instaladas nos dispositivos dos usuários, além de operar *offline* e em segundo plano (WEB.DEV, 2024).

2.2.3.1 PWAs vs Websites vs Apps Nativos

Aplicações nativas, ou seja, específicas de alguma plataforma, são desenvolvidas para um sistema operacional específico e/ou para um tipo de dispositivo, como um dispositivo iOS ou Android, geralmente fazendo uso de algum SDK (*Software Development Kit*) disponibilizado pelos fornecedores da plataforma. Estas aplicações são tradicionalmente distribuídas através de lojas digitais dos próprios fornecedores do sistema, onde podem ser encontradas e instaladas. Segundo a documentação do [MDN Web Docs \(2024b\)](#), os benefícios de *apps* nativos são:

- **Facilidade de acesso:** eles obtêm seu próprio ícone no dispositivo, facilitando a localização e a abertura por parte dos usuários;
- **Operações *offline* e em segundo plano:** são capazes de operar quando o usuário não está interagindo diretamente com eles e quando o dispositivo está *offline*;
- **Integração com SO:** eles podem ser integrados ao sistema operacional *host*, podendo, dessa forma, se integrar com outras aplicações no dispositivo e acessar recursos como câmera, GPS ou acelerômetro, e
- **Integração com lojas de *apps*:** eles são distribuídos usando lojas digitais, oferecendo aos usuários um local centralizado para encontrá-los e uma maneira consistente de decidir se desejam instalá-los.

De acordo com [MDN Web Docs \(2024b\)](#), *websites* tradicionalmente atuam mais como um “lugar que o usuário visita” do que “algo que o usuário tem”. Normalmente, as aplicações web: (i) não estão presentes nos dispositivos dos usuários quando eles as acessam; (ii) só podem ser acessadas através de um navegador, e (iii) são altamente dependentes de conectividade de rede ([MDN Web Docs, 2024b](#)).

No entanto, [MDN Web Docs \(2024b\)](#) lista alguns benefícios de *websites* em relação aos aplicativos específicos de plataforma, como:

- **Base de código única:** como a *web* é inerentemente multiplataforma, um *site* pode ser executado em diferentes dispositivos e sistemas operacionais a partir de uma única base de código, e;
- **Distribuição via *web*:** a *web* é uma relevante plataforma de distribuição. *Sites* são indexados por mecanismos de busca e podem ser compartilhados e acessados facilmente por meio de URLs. Um *site* pode ser distribuído sem necessidade de inscrição na loja de aplicativos de qualquer fornecedor.

PWAs combinam vantagens de ambos, *websites* tradicionais e aplicativos específicos de plataforma. Os PWAs têm os benefícios dos *sites*: PWAs são aplicações *web*, e PWAs podem ser acessadas por qualquer dispositivo através da *web* e com apenas uma base de código ([WEB.DEV, 2024](#)).

Segundo a documentação de [MDN Web Docs \(2024b\)](#), PWAs também oferecem muitos dos benefícios de aplicativos específicos de plataforma, incluindo:

- PWAs podem ser instalados no dispositivo;
- Os PWAs podem operar em segundo plano e *offline*. Enquanto um *site* típico só fica ativo no momento em que a página é carregada no navegador, um PWA pode:
 - Operar enquanto o dispositivo não tiver conectividade de rede;
 - Atualizar conteúdo em segundo plano;
 - Responder a mensagens *push* do servidor, e
 - Exibir notificações usando o sistema de notificações do sistema operacional.
- Os PWAs podem utilizar a tela inteira do dispositivo, ao invés de executar dentro da UI do navegador;
- Os PWAs podem ser integrados ao dispositivo, registrando-se como alvos e fontes de compartilhamento e acessando recursos do dispositivo, e
- Os PWAs podem ser distribuídos em lojas de aplicativos, bem como abertamente pela web.

2.2.3.2 Características Técnicas dos PWAs

Como dito anteriormente, uma PWA é uma aplicação *web*, na qual foram implementadas certas configurações, conferindo funcionalidades como as mencionadas na seção supracitada. Dentre essas configurações, com base em [MDN Web Docs \(2024b\)](#), as principais são:

- Um arquivo de manifesto *web app*, e
- Um *service worker* para fornecer funcionamento *offline*.

Um manifesto *web app* é um arquivo JSON que contém todas as informações necessárias ao navegador sobre como instalar a PWA, bem como esta vai se comportar quando instalada no computador ou dispositivo móvel do usuário. Este arquivo inclui informações como: o nome do *app*, os ícones que o *app* precisa usar, e a URL que será aberta quando o app for iniciado ([MDN Web Docs, 2024b](#)).

Um *service worker* é um *script* que o navegador executa em segundo plano, separado da página da *web*, possibilitando recursos que não precisam de uma página ou de interação do usuário. Essencialmente, eles atuam como servidores *proxy* que ficam entre os aplicativos da *web*, o navegador, e a rede (quando disponível). Servem para, entre outras atuações, permitir a criação de experiências *offline* eficazes, interceptar solicitações de rede e tomar medidas apropriadas com base na disponibilidade da rede, e atualizar recursos presentes no servidor (MDN Web Docs, 2024a).

2.2.4 Algoritmos

Algoritmos podem ser entendidos como sequências organizadas e finitas de passos com o objetivo de resolver um problema ou realizar uma tarefa (Byte Magazine, 1995). Com isso, diversos problemas computacionais são resolvidos utilizando a abordagem de algoritmos. Os algoritmos exercem um papel primordial em tarefas como busca e organização de dados, alocação de recursos, otimização de processos, dentre outras.

Entretanto, à medida que a complexidade dos problemas aumenta, métodos convencionais podem se mostrar inadequados, resultando em desafios relacionados ao desempenho, ao consumo de recursos, e ao tempo necessário para encontrar uma solução. Para lidar com esses cenários, surgem algoritmos especializados que exploram estratégias mais avançadas, permitindo a obtenção de soluções eficientes e otimizadas. É nesse contexto que emergem os estudos de Pesquisa Operacional, em particular a Programação Linear (HILLIER; LIEBERMAN, 2015).

2.2.4.1 Programação Linear

A Programação Linear (PL) é uma técnica matemática de otimização utilizada para encontrar a melhor solução para um problema, dados um conjunto de requisitos ou restrições lineares. Trata-se de uma das ferramentas mais importantes da Pesquisa Operacional, aplicada em uma vasta gama de domínios, como logística, finanças, produção e, notavelmente, na formulação de dietas (HILLIER; LIEBERMAN, 2015).

Um problema de Programação Linear é composto por três elementos principais:

- **Função Objetivo:** Uma expressão matemática linear que se deseja maximizar (e.g., lucro, qualidade nutricional) ou minimizar (e.g., custo, desperdício);
- **Variáveis de Decisão:** São as incógnitas do problema, ou seja, as quantidades que podem ser controladas e cujos valores ótimos se deseja encontrar, e
- **Restrições:** Um conjunto de inequações ou equações lineares que limitam os valores que as variáveis de decisão podem assumir. Elas representam as limitações físicas, técnicas ou lógicas do problema.

No contexto do NattyPet, o balanceamento de uma receita é modelado como um problema de Programação Linear clássico. As **variáveis de decisão** (x_i) representam a quantidade de cada ingrediente i a ser incluída na receita. A **função objetivo** é minimizar a soma total das quantidades dos ingredientes, buscando uma receita eficiente, representada como:

$$\text{Minimizar } Z = \sum_{i=1}^n x_i$$

As **restrições** são as metas nutricionais que devem ser atingidas. Para cada nutriente (calorias, proteínas, gorduras, etc.), uma inequação é formulada para garantir que a combinação final atenda ao requisito mínimo. Por exemplo:

- **Calorias:** $c_1x_1 + c_2x_2 + \dots + c_nx_n \geq \text{Calorias Mínimas}$
- **Proteínas:** $p_1x_1 + p_2x_2 + \dots + p_nx_n \geq \text{Proteínas Mínimas}$
- **Gorduras:** $f_1x_1 + f_2x_2 + \dots + f_nx_n \geq \text{Gorduras Mínimas}$

Essa abordagem é amplamente validada e utilizada na indústria de nutrição animal para a formulação de rações de custo mínimo que atendem a todas as exigências nutricionais (SANTOS et al., 2018).

2.2.4.2 O Algoritmo Simplex

Para resolver os problemas modelados pela Programação Linear, o **Algoritmo Simplex**, desenvolvido por George Dantzig, é o método mais fundamental. Ele funciona navegando pelos vértices de uma região de soluções viáveis até encontrar o ponto ótimo. O processo iterativo do Simplex pode ser resumido nas seguintes etapas:

1. **Modelagem e Forma Padrão:** O problema é formulado e as restrições de inequação são convertidas em equações pela adição de variáveis de folga ou artificiais;
2. **Construção da Tabela Simplex:** O sistema de equações é representado em um *tableau* inicial, que organiza os coeficientes das variáveis e das restrições;
3. **Iteração para Otimização:** O algoritmo seleciona iterativamente variáveis para entrar e sair da base, atualizando a tabela a cada passo para encontrar uma solução progressivamente melhor, e
4. **Convergência:** O processo se repete até que nenhuma melhoria na função objetivo seja possível, indicando que a solução ótima foi alcançada. O resultado final é então lido diretamente da tabela.

A aplicação do Algoritmo Simplex no NattyPet permite, uma vez definido o problema, calcular de forma computacionalmente eficiente as quantidades ideais de cada ingrediente para compor uma receita balanceada e otimizada.

2.2.4.3 Análise Comparativa de Abordagens

Para o problema de balanceamento nutricional, uma abordagem alternativa utilizando **Programação Dinâmica (PD)**, especificamente uma variação do Problema da Mochila conhecida como **Mochila Multidimensional (MKP - *Multidimensional Knapsack Problem*)**, foi considerada. No entanto, uma análise comparativa revela vantagens significativas da Programação Linear (PL) com o método Simplex para este caso de uso. A Tabela 10 resume os principais pontos de comparação entre as duas abordagens.

Tabela 10 – Comparação entre Programação Dinâmica (MKP) e Programação Linear (Simplex)

Critério	Programação Dinâmica (MKP)	Programação Linear (PL) com Simplex
Precisão	Garante a solução ótima para o problema formulado.	Garante a solução ótima para o problema formulado.
Complexidade Temporal	Alta: $O(N \times C \times P \times F)$, onde C, P, F são os limites dos nutrientes. Torna-se impraticável com valores altos.	Polinomial: Frequentemente $O(N^3)$ em casos práticos, dependendo principalmente do número de ingredientes (N).
Complexidade Espacial	Alta: $O(C \times P \times F)$, exigindo grande quantidade de memória para a matriz multidimensional.	Boa: $O(N^2)$ para armazenar o <i>tableau</i> Simplex, escalando melhor.
Escalabilidade	Ruim. O desempenho degrada rapidamente com o aumento dos valores das restrições ou com a adição de novas restrições.	Boa. Consegue lidar eficientemente com um grande número de ingredientes e restrições.
Flexibilidade	Baixa. Adicionar uma nova restrição nutricional exige adicionar uma nova dimensão à matriz, aumentando drasticamente a complexidade.	Alta. Adicionar uma nova restrição se traduz em apenas adicionar uma nova linha ao problema, o que é facilmente gerenciável pelo Simplex.

Fonte: Autores

A principal desvantagem da Programação Dinâmica neste cenário é sua complexidade. A solução de MKP requer a construção de uma matriz cujo tamanho é o produto

dos limites de cada restrição nutricional (e.g., calorias, proteínas, gorduras). Se os valores desses limites forem grandes, o consumo de tempo e, principalmente, de memória torna-se proibitivo.

Em contrapartida, a complexidade do método Simplex depende majoritariamente do número de ingredientes (N) e de restrições, e não dos seus valores numéricos. Como o número de ingredientes em uma receita tende a ser muito menor que os valores numéricos das metas nutricionais (Por exemplo, $N = 20$ ingredientes vs. $C = 500$ calorias), a Programação Linear apresenta uma solução com complexidade temporal e espacial muito inferior.

Adicionalmente, a flexibilidade da Programação Linear é um fator decisivo. No futuro, caso seja necessário considerar novos nutrientes (por exemplo, cálcio, fósforo, fibras), basta adicionar novas inequações ao modelo. Na abordagem de Programação Dinâmica, isso implicaria em aumentar a dimensionalidade da tabela, o que rapidamente a tornaria computacionalmente inviável.

Dessa maneira, a implementação do Método Simplex para resolver o problema de Programação Linear formulado fornece uma solução mais robusta, escalável e flexível, sendo a mais adequada para o problema de balanceamento nutricional do NattyPet.

2.3 Resumo do Capítulo

Este capítulo buscou apresentar o referencial bibliográfico dos temas necessários ao desenvolvimento deste Trabalho de Conclusão de Curso, que teve por finalidade o desenvolvimento de um *WebApp* que facilita a criação de receitas balanceadas nutricionalmente para *pets*.

Para a apresentação desses conceitos, fez-se necessária uma breve explicação acerca do domínio de [Nutrição Animal](#) e da área de [Engenharia de Software](#) como um todo. A seção de [Nutrição Animal](#) abordou os conceitos fundamentais e regulamentações que orientam a alimentação de *pets*, essenciais para orientar o balanceamento adequado das receitas pelo software. Já a seção de [Engenharia de Software](#) discutiu tecnologias e boas práticas que serão empregadas no desenvolvimento da aplicação, como definições de [Requisitos de Software](#) e [Usabilidade](#), o emprego de [Progressive Web Apps](#) (PWA), e estudos de [Algoritmos](#) para o balanceamento das receitas utilizando técnicas de [Programação Linear](#).

Cada tópico confere particularidades que foram abordadas ao longo do capítulo e foram primordiais para o desenvolvimento prático do projeto.

3 SUPORTE TECNOLÓGICO

Nesse capítulo, serão tratados os principais apoios de cunho tecnológico usados no trabalho, seja para [Gerência e Configuração de Software](#); [Desenvolvimento](#); [Modelagem](#), [Análise de Resultados](#) ou [Implantação de Software](#). Por fim, consta o [Resumo do Capítulo](#), com a apresentação de um quadro que reúne esses apoios, conferindo descrição e versão.

3.1 Gerência e Configuração de Software

Como o principal entregável desse trabalho é um software no domínio de Nutrição de *Pets*, mais especificamente um *WebApp*, ocorreu a necessidade de ferramentas capazes de prover um ambiente de programação, configurável para lidar com diferentes linguagens, além de apoio à hospedagem e ao versionamento. Nesse contexto, segue uma visão geral sobre essas ferramentas.

3.1.1 Visual Studio Code

O Visual Studio Code (VSCode) é um editor de código-fonte gratuito, leve e altamente customizável desenvolvido pela Microsoft ([Microsoft, 2024](#)). Ele funciona como um ambiente de desenvolvimento integrado (IDE), oferecendo suporte nativo a várias linguagens de programação, bem como integração com ferramentas como GIT e extensões que ampliam suas funcionalidades, como depuradores e *linters*. É uma escolha vantajosa para projetos de qualquer escala devido à sua interface amigável, ao bom desempenho, e ao suporte a ambientes de desenvolvimento remoto. Além das vantagens já mencionadas, os autores desse trabalho justificam sua escolha com base na produtividade proporcionada por esse editor via recursos como terminal embutido, controle de versão integrado e colaboração em tempo real (via extensões).

3.1.2 GIT

O GIT é um sistema de versionamento distribuído de código aberto, amplamente utilizado para gerenciar mudanças de código-fonte ([Git Community, 2024](#)). Ele permite, através de sua *Command Line Interface* (CLI), colaboração eficiente, rastreamento de alterações, e recuperação de versões anteriores com facilidade. Além disso, possui uma documentação extensa e abrangente sobre suas funcionalidades. Em projetos, sua escolha costuma ser vantajosa, pois permite trabalhar em funcionalidades diferentes de maneira isolada. Isso tende a garantir a integridade do código, e possibilita a implementação de

workflows, como *branches* para *features* ou correções de *bugs*, além de integração com plataformas de hospedagem de repositórios como o GitHub.

3.1.3 GitHub

O GitHub é uma plataforma de hospedagem de repositórios em nuvem baseada em GIT que facilita o versionamento e o desenvolvimento colaborativo de um produto de software ([GitHub Docs, 2024](#)). Além da função supracitada, a plataforma oferece soluções para diferentes *workflows* de desenvolvimento; ferramentas de integração e *deploy* contínuo como GitHub Actions, quadro de tarefas, e controle de permissões e acessos. O Github é amplamente adotado em projetos devido à acessibilidade; suporte a repositórios privados e públicos, e integração com várias ferramentas externas. Todas essas funcionalidades foram consideradas pelos autores ao escolherem essa plataforma para apoiar a hospedagem e o versionamento desse projeto.

3.1.4 SonarQube

SonarQube é uma plataforma de código aberto que permite a análise contínua da qualidade do código-fonte ([SonarSource, 2024](#)). Esta ferramenta pode ser inserida no *workflow* de uma aplicação para gerar uma análise contínua de problemas no código fonte, como *bugs*, vulnerabilidades, códigos duplicados, entre outros. Além mais, Esta ferramenta pode ser integrada ao *Continuous Delivery* e *Command Line Interface*(CI/CD) de um projeto a fim analisar os *Pull Requests* criados, e com isso, toda nova submissão de código terá sua qualidade analisada.

3.2 Desenvolvimento

Como colocado anteriormente, o software a ser produzido nesse trabalho é um *WebApp*, ou seja, um *Progressive Web App* (PWA). Nesse sentido, envolverá ferramentas de apoio ao desenvolvimento do *frontend* e do *backend*, conforme apresentado nas próximas seções.

3.2.1 Svelte & SvelteKit

Svelte é um *framework* para construção de interfaces de usuário na *web*. Com ele, é possível escrever componentes declarativos em HTML, CSS e JavaScript ([Svelte Team, 2024a](#)). Diferente de outros *frameworks*, Svelte faz uso de um compilador para prover interfaces interativas e reativas, eliminando a necessidade de um ambiente de *runtime* pesado.

O SvelteKit é um *framework* para desenvolver aplicações *web* robustas e de alto desempenho usando Svelte. Enquanto Svelte se preocupa com a renderização de componentes UI, Sveltekit fornece um conjunto de soluções para criar aplicações *web* completas. Ele oferece desde funcionalidades básicas, como um roteamento interno entre páginas, até recursos mais avançados como: otimizações de construção de código para carregar apenas o código mínimo necessário; suporte *offline*; pré-carregamento de páginas antes da navegação do usuário; e suporte às diferentes estratégias de renderização configuráveis (ex. SSR, CSR e SSG) (Svelte Team, 2024b). A escolha desse *framework* para o projeto justificou-se pelo seu desempenho, sua simplicidade na escrita de código, e sua alta customização, sendo esses aspectos necessários para atender aos anseios do presente trabalho.

3.2.2 Java & Spring

Java é uma linguagem de Programação Orientada a Objetos (POO). Trata-se de um das linguagens mais populares do mercado. Conhecida por suas portabilidade, segurança e capacidade de se adaptar a diferentes contextos, Java é uma linguagem comumente utilizada para o desenvolvimento de aplicações *web* (Amazon Web Services, 2024).

O Java Spring Framework é um *framework* de software livre, desenvolvido para construir aplicações Java completas de maneira rápida e segura. O *framework* fornece um conjunto de ferramentas que facilitam e agilizam o desenvolvimento de aplicações como APIs, microsserviços e sistemas distribuídos. O Spring simplifica a configuração, a gerência de dependências, além de oferecer um ecossistema consistente para desenvolvimento de aplicações de nível empresarial. Sua escolha é recomendada para projetos que demandam escalabilidade, confiabilidade e desenvolvimento ágil (Spring Team, 2024). Sendo assim, os autores optaram pelo seu uso nesse trabalho.

3.2.3 Go

Go, ou Golang, é uma linguagem de programação moderna, criada pelo Google, projetada para ser eficiente, simples e de alto desempenho (Go Developers, 2024). Trata-se de uma linguagem multiparadigma de propósito geral, compilada e fortemente tipada. É especialmente adequada para sistemas distribuídos, aplicações em nuvem e microsserviços, devido ao seu rápido tempo de compilação, à facilidade de implantação, ao gerenciamento de concorrência, e ao uso eficiente de recursos.

Neste projeto, a linguagem foi escolhida para implementar o algoritmo de balanceamento de receitas devido ao seu desempenho alto, sintaxe simples e robustez na gestão de concorrência. Esses atributos garantiram otimização no processamento, além de redução do tempo de execução.

3.2.4 Docker

O Docker é uma plataforma de virtualização baseada em contêineres, que permite criar, distribuir e executar aplicações em ambientes isolados ([Docker, Inc., 2024](#)). Ele é ideal para projetos que requerem consistência entre ambientes de desenvolvimento, teste e produção. Sua escolha é justificada pela redução de problemas relacionados a dependências e pelo suporte à escalabilidade, permitindo entregas mais rápidas e compartilhamento facilitado do ambiente entre os desenvolvedores. Além disso, o Docker foi utilizado pelo autores desse projeto para garantir consistência/padronização entre os ambientes de desenvolvimento dos membros do trabalho.

3.2.5 PostgreSQL

O PostgreSQL é um sistema de gerenciamento de banco de dados relacional e de código aberto ([PostgreSQL Global Development Group, 2024](#)). Destaca-se pela conformidade com padrões SQL, extensibilidade e suporte a transações de Atomicidade, Consistência, Isolamento e Durabilidade (ACID). É ideal para projetos que necessitam de consistência de dados, consultas complexas e grande volume de dados. Os autores justificam a escolha do PostgreSQL com base na disponibilidade de recursos avançados oferecidos pelo sistema, bem como na facilidade de integração com boa parte das tecnologias utilizadas no mercado, incluindo as usadas na elaboração desse projeto.

3.3 Modelagem

A seguir são apresentadas as principais ferramentas de modelagem usadas: (i) na elaboração do protótipo de alta fidelidade do software, e (ii) na especificação de modelos inrentes à documentação do software (ex. diagramas para compreensão da arquitetura), bem como aos elementos dessa monografia (ex. fluxogramas relacionados às atividades realizadas nesse trabalho, orientando-se por notações específicas).

3.3.1 Figma

O Figma é uma ferramenta de *design* colaborativa baseada na nuvem, amplamente utilizada para criar interfaces de usuário (UI) e prototipagem de aplicações. Sua principal vantagem é a colaboração em tempo real, permitindo que *designers* e desenvolvedores trabalhem juntos, diretamente nos arquivos ([Figma, Inc., 2024](#)). Essa ferramenta é ideal para projetos que demandam iteração rápida e *feedback* constante, tornando o processo de *design* mais integrado ao fluxo de desenvolvimento. Como exemplo de uso do Figma no presente projeto, pode-se mencionar a modelagem do Protótipo de Alta Fidelidade, disponível na [seção 5.5.4 Protótipo de Alta Fidelidade](#).

3.3.2 Eraser.io

O Eraser.io é uma ferramenta visual utilizada para criar fluxogramas, *wireframes* e diagramas arquiteturais (Eraser, Inc., 2024). A ferramenta possui uma interface intuitiva e com recursos que auxiliam no planejamento e em esboços de projetos. Dentre esses recursos, cabe mencionar um canvas interativo e a capacidade de editar diagramas através de sua própria linguagem declarativa. Sua utilização no trabalho versa sobre o desenho de fluxos e diagramações, tal como ocorre no [Capítulo 4 - Metodologia](#). Nesse capítulo, constam fluxos desenhados especificamente para esclarecer sobre as várias atividades planejadas e cumpridas, no intuito de concluir esse projeto.

3.4 Análise de Resultados

A seguir são apresentadas as principais ferramentas de análise de resultados utilizadas na validação do Protótipo de Alta Fidelidade do software, nas métricas de qualidade do código fonte, bem como na coleta de dados dos usuários finais.

3.4.1 Maze

Maze é uma ferramenta de testes de usabilidade que permite coletar *feedbacks* de usuários; analisar comportamentos, e medir a eficácia de um produto digital (Maze, Inc., 2024). Além disso, o Maze oferece integração com o [Figma](#), permitindo que o protótipo desenhado no [Figma](#) seja testado e validado no Maze. Tal ferramenta, coleta os dados dos testes realizados pelos usuários e os disponibiliza, tornando possível a verificação da usabilidade de uma determinada aplicação.

3.4.2 Google Forms

Google Forms é uma ferramenta *online* fornecida pelo Google, que permite criar formulários personalizados para coletar informações de forma fácil e rápida (Google, Inc., 2024). A ferramenta foi escolhida para elaboração de questionários, cujo o objetivo foi coletar dados dos *stakeholders* e validar a aplicação em termos de usabilidade.

3.5 Implantação de Software

No que diz respeito à implantação do software, foram escolhidas as seguintes ferramentas para hospedagem e gerenciamento do código fonte e do banco de dados.

3.5.1 Hetzner (Servidor VPS com CPU Compartilhada)

A Hetzner é uma provedora europeia de infraestrutura em nuvem que oferece servidores virtuais privados (VPS) com recursos dedicados ou compartilhados ([Hetzner, Inc., 2025](#)). Para este projeto, foi utilizada uma instância de VPS com CPU compartilhada, que representa uma solução econômica e eficiente para ambientes de desenvolvimento e testes. Esses servidores permitem o controle total do ambiente, com acesso root, suporte a múltiplos sistemas operacionais e capacidade de configuração personalizada. A escolha pela Hetzner se deu devido ao seu excelente custo-benefício, desempenho consistente mesmo em configurações de CPU compartilhada, e pela facilidade de automação e escalabilidade via sua API de gerenciamento. Assim, a Hetzner se mostrou uma alternativa robusta e flexível para suportar as demandas técnicas do projeto.

3.6 Resumo do Capítulo

Na Tabela 11, constam as tecnologias de maior relevância para o cumprimento do trabalho, sendo apresentadas com Nome, Descrição e Versão.

Tabela 11 – Tecnologias Utilizadas no Trabalho

Nome	Descrição	Versão
VS Code	IDE de edição de texto e desenvolvimento colaborativo. Utilizado tanto para criação dos softwares desenvolvidos neste trabalho, quanto para escrita da sua documentação.	1.94.2
GIT	Sistema de controle de versionamento de código.	2.39.3
GitHub	Plataforma de hospedagem de repositórios integrada ao GIT. Utilizada para armazenar os códigos dos serviços do projeto.	-
Svelte	Framework de criação de componentes de UI para a <i>web</i> .	5.4.0
Sveltekit	Framework para criação de aplicações <i>web</i> completas.	2.9.0
Java	Linguagem de POO robusta conhecida por sua portabilidade e segurança. Utilizada para construir serviços <i>web</i> como APIs e microsserviços.	17
Spring Framework	Framework para construir aplicações Java de maneira rápida e eficaz. Ele simplifica a configuração, gerência de dependências e oferece um ecossistema maduro para desenvolvimento de aplicações de nível empresarial.	6.2.0

Nome	Descrição	Versão
Go	Linguagem compilada de propósito geral conhecida por sua simplicidade, alto desempenho e gerenciamento de concorrência. Utilizada para desenvolvimento do algoritmo de balanceamento.	1.23.3
Docker	Plataforma de virtualização baseada em contêineres que permite criar, distribuir e executar aplicações em ambientes isolados. Utilizada para configurar e padronizar os ambientes de desenvolvimento dos softwares do trabalho.	26.0.0
PostgreSQL	Sistema de gerenciamento de banco de dados relacional poderoso e de código aberto. Escolhido por sua confiabilidade, recursos avançados e facilidade de integração com outras tecnologias.	17.2
Figma	Ferramenta de design colaborativa utilizada para realizar a prototipagem da aplicação e fluxos de usuário.	-
Eraser.io	Ferramenta de diagramação interativa utilizada para criar fluxogramas, <i>wireframes</i> e diagramas arquiteturais do projeto.	-
Maze	Plataforma de testes de usabilidade	-
SonarQube	Ferramenta de análise de qualidade de código	-
Hetzner	Plataforma de computação em nuvem, para implantação e hospedagem de código	-

Fonte: Autores

4 METODOLOGIA

Este capítulo tem como objetivo detalhar os métodos adotados no desenvolvimento teórico e prático deste Trabalho de Conclusão de Curso. Constam três seções principais, sendo: a [Classificação da Pesquisa](#), incluindo o [Método Investigativo](#); o [Método de Desenvolvimento](#) e [Método de Análise de Resultados](#). Há ainda o [Fluxo de Atividades](#) e o [Cronograma](#), ambos planejados e cumpridos para a execução deste trabalho. Por fim, tem-se o [Resumo do Capítulo](#).

4.1 Classificação da Pesquisa

De acordo com [Gerhardt e Silveira \(2009\)](#), a Metodologia pode ser classificada como o estudo da organização, dos caminhos a serem trilhados, para se realizar uma pesquisa ou um estudo, ou para produzir ciência. Já a pesquisa científica, refere-se ao resultado de “um inquérito ou exame minucioso, realizado para resolver um problema, recorrendo a procedimentos científicos” ([GERHARDT; SILVEIRA, 2009](#)).

A partir das definições de [Gerhardt e Silveira \(2009\)](#), esta seção visa categorizar a pesquisa deste trabalho quanto à [abordagem](#), à [natureza](#), aos [objetivos](#) e aos [procedimentos](#). O Quadro 2 procura apresentar um resumo dessa classificação para o caso deste trabalho.

Quadro 2 – Resumo de Classificação da Metodologia

Categoria	Classificação
Abordagem	Quantitativa e Qualitativa
Natureza	Aplicada
Objetivos	Exploratórios
Procedimentos	Pesquisa Bibliográfica e Pesquisa-ação

4.1.1 Abordagem

A pesquisa realizada, neste trabalho, pode ser classificada como **quantitativa** e **qualitativa**. A pesquisa quantitativa centra-se na objetividade, isto é, busca realizar a análise de dados brutos, coletados de modo padronizado, e sem demasiados procedimentos estatísticos. Por outro lado, a pesquisa qualitativa concentra-se em aspectos que não podem ser quantificados, ou seja, que dizem respeito às relações, aos processos e aos fenômenos sociais que não podem ser reduzidos à operacionalização de variáveis ([GERHARDT; SILVEIRA, 2009](#)).

Neste trabalho, coletou-se métricas qualitativas sobre a satisfação do usuário ao usar a ferramenta, compreendendo melhor a usabilidade da plataforma *NattyPet*, e sobre a qualidade do código (ex. manutenibilidade). Além disso, foram coletadas métricas quantitativas, mais associadas, por exemplo, ao desempenho da ferramenta e/ou às métricas mensuráveis sobre a qualidade do código, dentre elas: número de *bugs*. De toda forma, cabe colocar que a análise foi, predominantemente, qualitativa. Portanto, mesmo as métricas quantificáveis foram utilizadas como base para prover uma análise qualitativa mais embasada.

4.1.2 Natureza

A natureza de pesquisa deste trabalho pode ser classificada como **aplicada**. A pesquisa aplicada visa gerar conhecimentos para aplicação prática, sendo direcionada à resolução de problemas específicos. Por outro lado, a pesquisa básica propõe gerar conhecimentos novos, mas sem aplicação prática prevista (GERHARDT; SILVEIRA, 2009).

O presente trabalho acorda o desenvolvimento de uma aplicação projetada para tratar de preocupações de um domínio específico, no caso, a nutrição de animais domésticos. O desenvolvimento dessa plataforma foi orientado a fim de propor o *Minimum Viable Product* (MVP), ou seja, uma versão do produto com todas as características mais básicas necessárias para ser considerado como “um entregável de valor” aos *stakeholders*. Sendo assim, trata-se de uma primeira versão da ferramenta, por meio da qual tutores de *pets*, que seguem o regime de Alimentação Natural, podem formular receitas nutricionalmente balanceadas para seus animais.

4.1.3 Objetivos

Em relação aos objetivos, a pesquisa deste trabalho é classificada como **Exploratória**, uma vez que se enquadra no contexto de inovação e na recente difusão do tema no meio tecnológico. A pesquisa exploratória busca proporcionar maior familiaridade com o problema, com o objetivo de destacá-lo ou de formular hipóteses (GERHARDT; SILVEIRA, 2009).

Como mencionado anteriormente, a aplicação foi desenvolvida para explorar o domínio da nutrição natural para cães e gatos com o objetivo de gerar receitas balanceadas que atendam às necessidades energéticas desses animais. Com isso, o desenvolvimento foi conduzido em colaboração com *stakeholders*, e embasado em descobertas e hipóteses obtidas a partir da literatura especializada. A proposta é possibilitar uma interação mais intuitiva e facilitada entre o usuário e a aplicação, garantindo que ela esteja alinhada às recomendações da literatura especializada.

4.1.4 Procedimentos

Quanto aos procedimentos, a pesquisa deste trabalho é classificada como **pesquisa-ação** e **bibliográfica**. A pesquisa-ação envolve a colaboração planejada do pesquisador na situação a ser investigada, com objetivo de diagnosticar o problema e buscar um resultado prático (GERHARDT; SILVEIRA, 2009). A Pesquisa bibliográfica é realizada por meio de um conjunto de referenciais teóricos disseminados em diferentes formatos como, livros, revistas, artigos, materiais disponíveis na internet, entre outros.

A pesquisa bibliográfica foi elaborada orientando-se pelo **Método Investigativo**, aprofundado na próxima seção deste capítulo. Já a pesquisa-ação será abordada na seção de **Método de Análise dos Resultados**.

4.1.5 Método Investigativo

A pesquisa bibliográfica é uma sequência de etapas que podem variar de acordo com a natureza do problema, entre outros fatores. Uma pesquisa bibliográfica geralmente contém as seguintes etapas: escolha do tema; levantamento bibliográfico preliminar; formulação do problema; elaboração do plano provisório de assunto; busca das fontes; leitura do material; fichamento; organização do assunto, e redação do texto (GIL et al., 2002).

Neste trabalho, após a definição do tema, foi realizado um levantamento das principais literaturas referentes ao domínio de pesquisa, além de diversos artigos pertinentes ao assunto estudado, com o objetivo de criar uma base de conhecimento e garantir suporte ao presente trabalho. A fim de fornecer um maior embasamento técnico ao trabalho, foram reunidos documentos oficiais, disponibilizados pelas principais Organizações Regulamentadoras de nutrição de animais domésticos. Além disso, foi realizada a busca de materiais em bases de dados como Google Acadêmico e ACM Digital Library. Outras bibliotecas digitais foram consultadas, mas não produziram resultados significativos.

A Tabela 12 retrata a elaboração e o refinamento das *strings* de busca, bem como as respectivas bases de dados utilizadas para pesquisa e a quantidade de artigos retornados pelas mesmas.

Após a triagem inicial do material bibliográfico obtido na pesquisa, iniciou-se o processo de leitura detalhada e seleção dos trabalhos mais relevantes. Nesse estágio, realizou-se uma filtragem para identificar os materiais mais adequados ao presente trabalho, com base em critérios específicos estabelecidos para a seleção. A leitura exploratória desempenhou um papel relevante nesse processo, sendo realizada por meio de uma análise de resumos e identificação de palavras-chave (GIL et al., 2002). Como resultado, definiram-se os seguintes critérios de inclusão:

- Trabalhos relacionados à Nutrição Natural para *pets*;

- Estudos sobre a geração de receitas de Alimentação Natural para cães e gatos;
- Materiais que abordem os benefícios da Alimentação Natural para *pets*;
- Documentos oficiais sobre os Órgãos Regulamentadores voltados à Alimentação Natural para *pets*;
- Trabalhos que abordem os requisitos nutricionais de cães e gatos, e
- Aplicações que gerem receitas de Alimentação Natural para cães ou gatos.

De maneira similar, foram estabelecidos os seguintes critérios para exclusão:

- Trabalhos focados no tema de mercado *pet* ao invés de nutrição animal;
- Estudos aprofundados sobre os efeitos da Alimentação Natural na biologia dos animais;
- Trabalhos que abordem outros tipos de dieta, que diferem da Alimentação Natural, e
- Estudos que abordem a outros animais, que diferem de cães e gatos.

Com a aplicação dos critérios descritos, foram selecionados os seguintes artigos como embasamento para o tema proposto:

- *FEDIAF. Nutritional Guidelines For Complete and Complementary Pet Food for Cats and Dogs. 2024.* (FEDIAF, 2024);
- *OLIVEIRA, L. D.; FISCHER, M. M. Alimentação Natural para Cães. São Paulo: [s.n.], 2022.* (OLIVEIRA; FISCHER, 2022);
- *AAFCO. Official Publication. Champaign, IL: Association of American Feed Control Officials, 2013* (AAFCO, 2013);
- *ALIMENTAÇÃO Natural para Cães e Gatos. Revista Brasileira de Zootecnia, v. 39, p. supplspe, 2010.* (Sociedade Brasileira de Zootecnia, 2010), e
- *SILVA, L. da et al. MERCADO DA ALIMENTAÇÃO NATURAL PET NO BRASIL: Um estudo sobre a oferta e demanda por produtos naturais no mercado brasileiro. In: Anais do Fórum Internacional On-line de Empreendedorismo e Inovação no Agro 2021. Piracicaba, SP, Brasil: Galoá, 2021.* (SILVA et al., 2021).

Claramente, outros materiais foram consultados para embasamento de pontos correlatos aos tópicos de interesse do trabalho. Todas as referências bibliográficas constam ao final da monografia.

Tabela 12 – *Strings* de Busca Utilizadas

Base de dados	<i>String</i> de busca	Quantidade obtida
Google Acadêmico	alimentação natural para cães	36.500
Google Acadêmico	alimentação natural AND software	985
Google Acadêmico	alimentação natural AND pets	249
Google Acadêmico	mercado pet AND alimentação natural	129
Google Acadêmico	alimentação natural AND pets AND software	60
Google Acadêmico	alimentação natural AND pets AND receita	50
Google Acadêmico	alimentação natural AND pets AND aplicativo	36
ACM	natural feeding AND pet	19.098
ACM	natural feeding AND pet AND software	9.925
ACM	natural feeding AND pet AND app	4.568
ACM	natural diet AND dog	946

4.2 Método de Desenvolvimento

Foi adotado um método de desenvolvimento prático da aplicação, com base no critério de afinidade dos autores, combinando algumas práticas de duas metodologias ágeis: *Scrum* e *Kanban*. Esta seção tem como função descrever estas metodologias e explicar como suas principais práticas foram adaptadas de acordo com as necessidades do trabalho.

O *Scrum* é um método ágil voltado para a gestão de projetos e desenvolvimento iterativo de produtos (SOMMERVILLE, 2016). Tal metodologia não fornece qualquer técnica em especial para a fase de desenvolvimento, apenas estabelece um conjunto de regras e práticas gerenciais que devem ser adotadas para o sucesso de um projeto. Com isso, o seu processo é dividido em três fases principais. A primeira etapa trata-se do planejamento geral e da definição dos objetivos do projeto. Em seguida, ocorre uma sequência de ciclos de *sprint*, em que cada iteração resulta em um novo incremento ao produto. Por fim, a terceira fase consiste no encerramento do projeto e na sua respectiva documentação (SOMMERVILLE, 2016).

A partir do *Scrum*, estabelece-se, principalmente, o uso do *Product Backlog*, uma lista priorizada dos requisitos de software, sujeita às alterações de acordo com a evolução do projeto (SCHWABER; BEEDLE, 2002).

Em relação ao *Kanban*, trata-se de uma metodologia de controle de fluxo de atividades através da organização de cada demanda em seções de um quadro (AHMAD; MARKKULA; OIVO, 2013). Como proposto por Anderson (2010), utiliza-se o *Kanban* em complemento ao *Product Backlog*, pois o mesmo possibilita uma visualização contínua do estado e do progresso do desenvolvimento, contribuindo para uma melhor compreensão, gestão e fluidez do fluxo de trabalho.

O processo de desenvolvimento da aplicação orientou-se por metodologias ágeis.

Com isso, definiu-se o *Product Backlog* com base nos fundamentos do *Scrum*, e os incrementos e as iterações do produto orientaram-se pelo quadro *Kanban*. Seguem descrições adicionais:

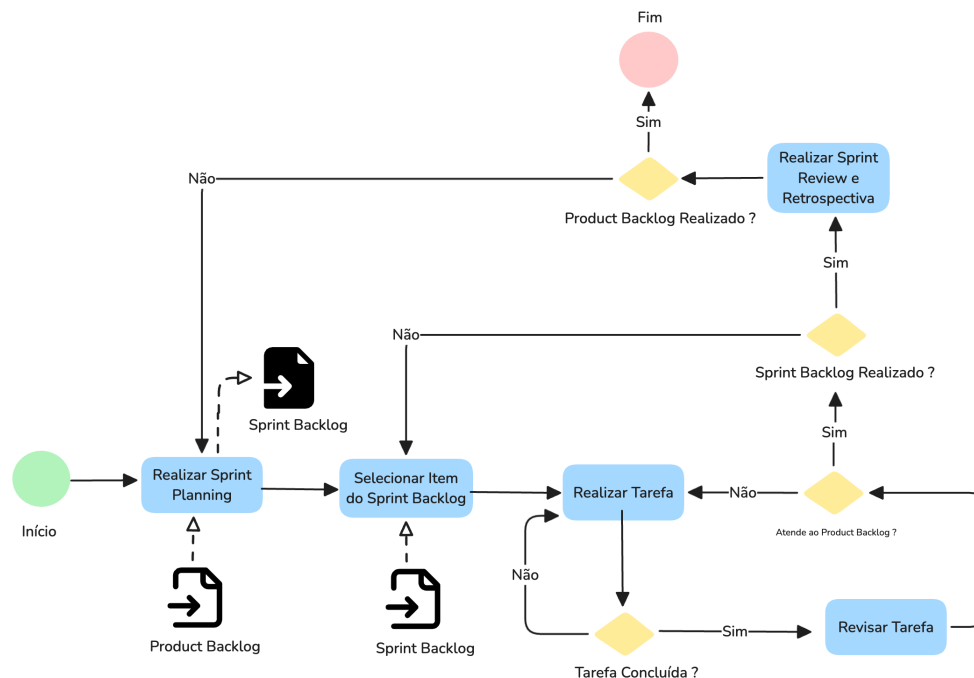
- *Product Backlog*: lista de tarefas priorizada para o desenvolvimento da aplicação;
- *Sprint*: ciclos de trabalho que podem ter duração de 1 a 4 semanas. Possuem duração fixa, tendo o trabalho concluído ou não;
- *Sprint Backlog*: lista de tarefas alocadas para a *Sprint* atual;
- Pendente: tarefas alocadas ou não para a *Sprint* atual que ainda não foram iniciadas;
- Em Andamento: tarefas alocadas para a *Sprint* atual que já foram iniciadas;
- Em Revisão: tarefas que foram finalizadas e estão passando pelo processo de Revisão para a devida validação dos critérios de aceitação, e
- Concluído: tarefas revisadas e aceitas, que cumprem o critério de aceitação.

4.2.1 Processo de Desenvolvimento

O processo de desenvolvimento escolhido está disponível na Figura 1, elaborado com base na notação BPMN (*Business Process Model and Notation*), como um detalhamento do processo de desenvolvimento da aplicação contendo as seguintes etapas:

- **Realizar *Sprint Planning***: a cada fim de *Sprint*, foi realizado um planejamento das tarefas a serem cumpridas na *Sprint* seguinte. Essas tarefas foram retiradas do *Product Backlog*, e selecionadas com base no grau de priorização das histórias de usuário. Como resultado, definiu-se o *Sprint Backlog*;
- **Selecionar Item do *Sprint Backlog***: com a definição do *Sprint Backlog*, os responsáveis foram alocados nas tarefas disponíveis;
- **Realizar a Tarefa**: etapa de desenvolvimento das tarefas selecionadas durante a *Sprint*;
- **Revisar a Tarefa**: processo de revisão realizado pelo membro que não fez parte do desenvolvimento da tarefa, sendo responsável por analisar a entrega, entendendo se ela atendeu o que fora definido nos critérios de aceitação, e
- **Realizar a *Sprint Review* e a Retrospectiva**: ao finalizar uma *Sprint*, os membros realizaram essa atividade de revisão, a fim de elencar os pontos negativos e positivos da *Sprint* com objetivos de melhoria para o ciclo seguinte.

Figura 1 – Fluxo de Atividades do Desenvolvimento da Aplicação



Fonte: Autores

4.3 Método de Análise dos Resultados

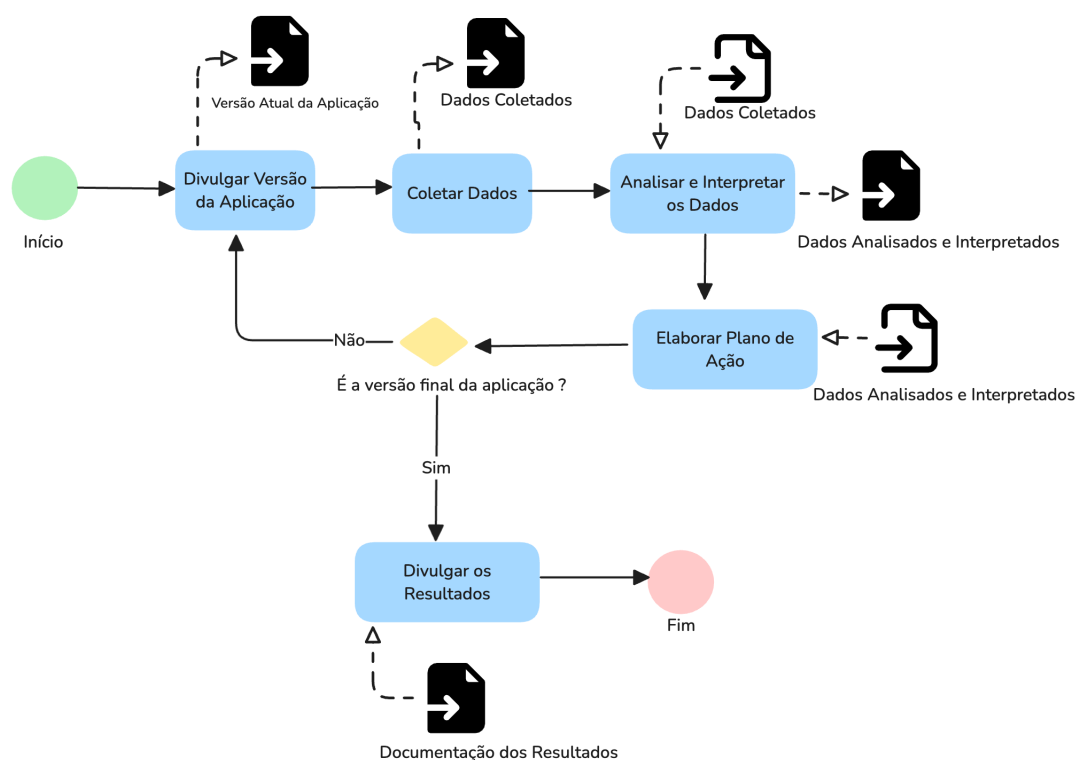
Como foi abordado na Seção 4.2.4, a análise de resultados deste trabalho foi conduzida pelo método de pesquisa-ação. A pesquisa-ação é uma abordagem de pesquisa investigativa, que adota um processo metodológico empírico e envolve a identificação de problemas dentro de um contexto social (KOERICH et al., 2009). Esse método possui particularidades variadas entre as fases, que são determinadas pela dinâmica dos pesquisadores com seu objeto de análise, sendo passível de adaptações (GERHARDT; SILVEIRA, 2009).

O protocolo adotado para esse trabalho foi adaptado, podendo ser visualizado na Figura 2, e a pesquisa-ação utilizada foi orientada pelas seguintes etapas:

- **Divulgar Versão da Aplicação:** etapa que consistiu na divulgação da última versão disponível para os *stakeholders* utilizarem e testarem a aplicação;
- **Coletar Dados:** etapa destinada à coleta dos dados qualitativos por meio de *feedbacks* originados a partir da interação dos *stakeholders* com a aplicação;
- **Analisar e Interpretar os Dados:** consistiu na análise dos dados coletados e a realização de exposição dos resultados obtidos;

- **Elaborar Plano de Ação:** etapa destinada à elaboração de um plano de ação que visou mitigar os problemas alavancados a partir dos dados analisados, e
- **Divulgar os Resultados:** consistiu na divulgação dos resultados obtidos e suas respectivas documentações. Esses insumos foram validados com os *stakeholders*.

Figura 2 – Fluxo de Atividades do Versionamento e Análise da Aplicação



Fonte: Autores

Ressalta-se ainda que, na etapa de Coletar Dados, ocorreram a coleta de dados quantitativos, seja número de *bugs*, seja tempo para a realização de uma ação por parte do usuário, seja outro. Nesses casos, na etapa de Analisar e Interpretar Dados, esses dados mensuráveis foram analisados de forma a apresentar um relato qualitativo mais embasado. Por exemplo, se o tempo para obter o cálculo de uma dieta foi X segundos, foi validado se isso era aceito ou não pelo usuário, ou seja, se isso influenciava na sua satisfação. Nesses casos, foi usada a escala likert (NORMAN, 2010)

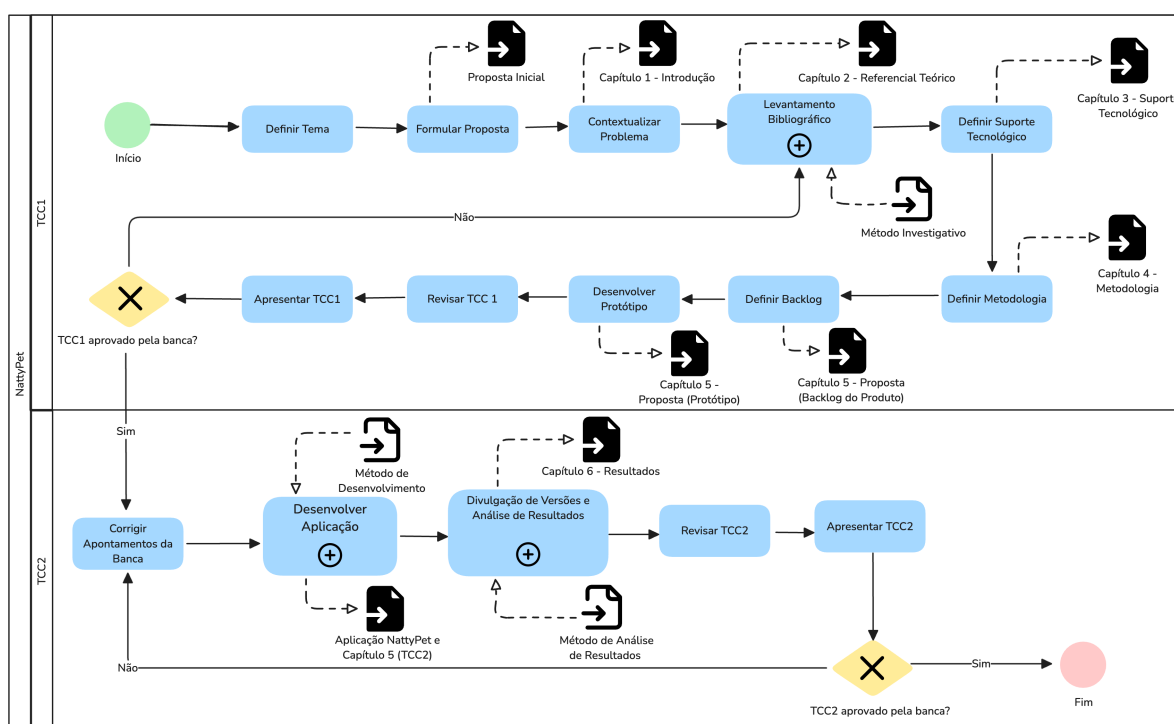
Essa escala é comumente utilizada para coletar as opiniões dos usuários. Nesse contexto, os usuários selecionam as respostas em uma escala que varia em diferentes níveis de concordância e discordância. O formato típico de um item Likert é: Discordo Totalmente; Discordo Parcialmente; Indiferente; Concordo Parcialmente, e Concordo Totalmente. Há plena satisfação (ex. Satisfação Excelente), se a resposta for "Concordo Totalmente", e nada de satisfação (ex. Satisfação Nula), se a resposta for "Discordo Totalmente". Além

disso, graduações nas satisfações para os casos: "Discordo Parcialmente"(ex. Satisfação Regular); "Indiferente"(ex. Satisfação Neutra); "Concordo Parcialmente"(ex. Satisfação Boa).

4.4 Fluxo de Atividades

Para a condução deste trabalho, fez-se necessário o estabelecimento de algumas etapas. Essas etapas podem ser visualizadas na Figura 3, baseado na notação BPMN. A descrição de cada etapa ocorre na sequência.

Figura 3 – Fluxo de Atividades do TCC



Fonte: Autores

- **Definir Tema:** atividade destinada à escolha do tema de desenvolvimento do trabalho. O tema surgiu como uma ideia de um dos autores, que foi totalmente aceita pelo outro autor, e validada com a orientadora. *Status:* Concluído;
- **Formular Proposta:** atividade destinada ao fechamento do escopo do trabalho, bem como conferir contextualização, identificar questões de pesquisa, estabelecer os objetivos gerais e específicos, e conferir as justificativas. *Status:* Concluído;
- **Contextualizar Problema:** atividade de desenvolvimento de uma visão mais ampla da área de interesse, que no caso foi a elaboração de dietas naturais balanceadas

para *pets*, bem como o aprofundamento da escrita quanto aos tópicos contextualização, justificativa, questionamentos e objetivos desse trabalho. *Status*: Concluído; Resultado: [Capítulo 1](#);

- **Levantamento Bibliográfico**: subprocesso que visa acordar os materiais bibliográficos centrados no tema de interesse do trabalho. Fez-se uso do [Método Investigativo](#) *Status*: Concluída. Resultado: [Capítulo 2](#);
- **Definir Suporte Tecnológico**: atividade voltada para definição dos apoios tecnológicos para a realização deste trabalho. *Status*: Concluído. Resultado: [Capítulo 3](#);
- **Definir Metodologia**: atividade, na qual os autores se dedicaram à idealização de um processo pelo qual ocorreu a condução do trabalho. *Status*: Concluído. Resultado: [Capítulo 4](#);
- **Definir Backlog**: atividade para a definição de forma clara e objetiva dos principais requisitos da aplicação. *Status*: Concluído, em versão preliminar. Resultado: [Capítulo 5](#);
- **Desenvolver Protótipo da Aplicação**: atividade para elaborar um Protótipo de Alta Fidelidade com base nos requisitos elencados no *Backlog*. O protótipo tem o objetivo de demonstrar de forma visual a proposta da aplicação, podendo ser visto como uma prova de conceito, em relação à proposta, uma vez que se orienta por e procura atender vários detalhamentos em termos de funcionalidades e fluxos pretendidos na aplicação. *Status*: Concluída, em versão preliminar. Resultado: [Capítulo 5](#);
- **Revisar TCC1**: atividade de revisão do trabalho como um todo, incluindo desde sua escrita até os artefatos produzidos. Tal processo foi feito tanto por meio de ciclos de discussões entre os autores, bem como entre os autores e a orientadora. *Status*: Concluída. Resultado: Própria Monografia;
- **Apresentar TCC1**: atividade destinada à aprovação da proposta, junto à Banca Avaliadora. *Status*: Concluída. Resultado: Própria Monografia;
- **Corrigir Apontamentos da Banca**: atividade destinada a revisão da Monografia e de seus artefatos com base nos *feedbacks* realizados pela Banca Avaliadora, fazendo as correções solicitadas. *Status*: Concluída. Resultado: Própria Monografia;
- **Desenvolvimento da Aplicação**: subprocesso destinado ao desenvolvimento da aplicação. O detalhamento desse processo é ilustrado na Figura 1, orientando-se pelo Método de Desenvolvimento. *Status*: Concluída. Resultado: [Capítulo 5](#);

- **Divulgação de Versões e Análise de Resultados:** subprocesso que consiste em aplicar a Pesquisa-Ação, conforme consta na seção [Análise dos Resultados](#), com objetivo de obter uma análise híbrida, qualitativa (predominantemente) e quantitativa (complementarmente), e realizar evoluções na aplicação com base nessa análise. É desejado, nesse processo, que seja feita a validação da aplicação com os usuários finais por meio de *feedbacks* que serão realizados com base nas versões da aplicação, seja via protótipos, seja via aplicação implementada. *Status:* Concluída. Resultado: [Capítulo 6](#);
- **Revisar TCC2:** atividade que procura revisar cada aspecto do trabalho, incluindo escrita e elaboração de artefatos inerentes ao escopo de atividades da segunda etapa do TCC. Foi utilizado o mesmo processo iterativo, conduzido para o escopo da primeira etapa do TCC. *Status:* Concluída. Resultado: Monografia Final e artefatos em geral, e
- **Apresentar TCC2:** atividade destinada à aprovação do trabalho em sua totalidade, junto à Banca Avaliadora. *Status:* A ser realizada.

4.5 Cronograma

Os Quadros [3](#) e [4](#) apresentam os cronogramas da primeira e da segunda etapas do TCC, respectivamente. Nos quadros, as atividades e os subprocessos são descritos de acordo com suas datas de implementação, baseando-se no fluxo de atividades acordado na seção anterior.

Quadro 3 – Cronograma de Atividades/Subprocessos da Primeira Etapa do TCC

Atividade	Out/24	Nov/24	Dez/24	Jan/25	Fev/25
Definir Tema	X				
Formular Proposta	X				
Contextualizar Problema		X			
Levantamento Bibliográfico		X			
Definir Suporte Tecnológico			X		
Definir Metodologia			X		
Definir <i>Backlog</i>			X	X	
Revisar TCC1				X	
Apresentar TCC1					X

Quadro 4 – Cronograma de Atividades/Subprocessos da Segunda Etapa do TCC

Atividade	Mar/25	Abr/25	Mai/25	Jun/25	Jul/25
Corrigir Apontamentos da Banca	X				
Desenvolvimento da Aplicação	X	X	X	X	
Divulgação de Versões e Análise de Resultados		X	X	X	
Revisar TCC2					X
Apresentar TCC2					X

4.6 Resumo do Capítulo

Este capítulo buscou apresentar os métodos adotados para a condução deste Trabalho de Conclusão de Curso. Inicialmente, foi realizada a [Classificação da Pesquisa](#), onde a pesquisa foi categorizada quanto à [abordagem](#)(Qualitativa e quantitativa), [natureza](#)(Aplicada), [objetivos](#)(Exploratórios) e [procedimentos](#)(Pesquisa Bibliográfica e Pesquisa-Ação). Em seguida, detalhou-se o [Método Investigativo](#), que orientou a pesquisa bibliográfica. Posteriormente, foi descrito o [Método de Desenvolvimento](#), que combinou práticas das metodologias ágeis *Scrum* e *Kanban*. Também foi apresentado o [Método de Análise dos Resultados](#), que foi conduzido pelo procedimento de pesquisa-ação. Além disso, foi estabelecido o [Fluxo de Atividades](#) necessário para a condução do trabalho, detalhando cada etapa do processo. Por fim, foi apresentado o [Cronograma](#) de atividades/subprocessos, sendo esse organizado considerando a primeira e a segunda etapas do TCC.

5 NattyPet

Este capítulo tem como objetivo apresentar, de forma detalhada, a proposta de solução desenvolvida no presente Trabalho de Conclusão de Curso (TCC), que visa auxiliar tutores de *pets* na geração de receitas de Alimentação Natural de forma balanceada. Para isso, as seções a seguir detalham os conceitos, objetivos e particularidades do projeto. A seção [Contextualização](#) retoma a problemática que motivou o trabalho. Em seguida, a seção [Sobre o NattyPet](#) apresenta o projeto, seu [Público-Alvo](#), [Análise SWOT](#) e [Análise de Risco](#). O [Backlog do Produto](#) e a [Identidade Visual](#) definem o escopo e o design da aplicação, respectivamente. A [Arquitetura da Solução](#) detalha a estrutura técnica implementada. Constan ainda as [Funcionalidades](#) da aplicação desenvolvida. Por fim, a seção [Algoritmo de Balanceamento de Receitas](#) aprofunda a implementação do método de otimização utilizado, e as conclusões do capítulo são apresentadas no [Resumo do Capítulo](#).

5.1 Contextualização

O Brasil destaca-se como um dos países com maior população de animais de estimação, ocupando a terceira posição mundial, e possui o segundo maior mercado de *pet food*, atrás apenas dos Estados Unidos ([ABINPET, 2021](#)). Essa métrica é acompanhada do fenômeno de humanização dos *pets*, uma tendência cada vez mais presente, na qual a proximidade entre tutores e animais promove uma maior preocupação com a saúde e o bem-estar dos *pets* ([MAZON; MOURA, 2017](#)). Esse cenário reflete a crescente relevância dos animais de estimação na sociedade brasileira, onde são frequentemente tratados como membros da família.

Nesse sentido, observa-se uma demanda crescente por alternativas alimentares que priorizem a qualidade nutricional dos *pets*. Uma pesquisa, por exemplo, indicou que 74% dos tutores brasileiros entrevistados consideraram importante que a alimentação dos seus animais de estimação seja natural ([SILVA et al., 2021](#)). Em 2007, uma crise de lotes de ração contaminados nos Estados Unidos, que resultou em problemas de saúde e óbitos de muitos animais, foi um marco para o interesse e a procura por alternativas alimentares para cães e gatos ([Sociedade Brasileira de Zootecnia, 2010](#)).

Desde então, alternativas como a Alimentação Natural ganharam destaque. Essa abordagem envolve a formulação de dietas com ingredientes frescos, como carnes, frutas e vegetais, adaptadas às necessidades nutricionais específicas de cada animal, promovendo maior qualidade de vida e longevidade ([OLIVEIRA; FISCHER, 2022](#)).

Com base na crescente popularidade da Alimentação Natural e nos desafios associ-

ados à sua implementação segura e eficaz, este trabalho propôs o desenvolvimento de uma aplicação voltada para tutores de cães e gatos. O objetivo é o apoio computacional para tutores de *pets* que desejam prover uma Alimentação Natural para os seus *pets*, sendo essa balanceada e orientada de acordo com os principais guias de nutrição para animais domésticos. Para tanto, foi desenvolvida uma ferramenta que auxilia na formulação de receitas nutricionalmente balanceadas.

Adicionalmente, a ferramenta foi desenvolvida com o diferencial de priorizar tanto a usabilidade para o público leigo; quanto a adesão rigorosa às diretrizes nutricionais reconhecidas internacionalmente. Com isso, a aplicação busca mitigar a lacuna identificada no *benchmarking*, disponível no [Apêndice A](#), realizado pelos autores, destacando-se como uma solução inovadora e assertiva no segmento de Alimentação Natural para animais domésticos.

5.2 Sobre o NattyPet

A ferramenta NattyPet trata-se de um *WebApp* que tem como objetivo auxiliar tutores de cães e gatos na geração de receitas de Alimentação Natural de forma balanceada. A aplicação fornece uma solução para a complexidade do processo de balanceamento nutricional de dietas para animais de diferentes perfis. Ainda, um dos anseios do projeto foi tornar a plataforma acessível, intuitiva e prática, de modo que a mesma fosse facilmente utilizada pelos próprios tutores de *pets*, que não possuem conhecimentos técnicos na área de nutrição animal. Com isso, o NattyPet procura conferir uma solução computacional viável para este público alvo.

A aplicação foi projetada para permitir que os tutores insiram informações sobre seus animais, como peso, idade, nível de atividade física e eventuais restrições alimentares. A partir desses dados, o sistema gera ou auxilia na geração de receitas personalizadas que atendam às necessidades nutricionais específicas de cada *pet*, reduzindo os riscos associados a dietas mal formuladas e promovendo a saúde e o bem-estar dos animais.

5.2.1 Público Alvo

Visando uma compreensão mais adequada quanto ao público alvo do *WebApp* NattyPet, foram especificadas personas e antipersonas ([FERREIRA et al., 2018](#)).

A primeira, identificada como Raquel, representa o perfil de usuário que já tem experiência com a Alimentação Natural, mas precisa de ajuda para inventar receitas novas, buscando ainda uma forma mais fácil de realizar os cálculos nutricionais. A segunda persona, Amanda, representa o perfil de usuário que já pratica Alimentação Natural, mas ainda é iniciante e utilizava modelos informais que encontrou na internet como base para

montagem de receitas. Sua principal necessidade é encontrar uma forma segura de criar receitas para seus *pets*. A terceira pessoa, João das Neves, representa o usuário que não é praticante e só recentemente descobriu a Alimentação Natural, interessando-se pelos seus benefícios. As Figuras 4, 5 e 6 representam respectivamente as personas Raquel, Amanda e João.

Também foi identificada uma antipersona, especificada como Seu Bernardo, representando o público que não configura usuários da plataforma. Sendo assim, esse usuário é a representação simbólica daqueles que aderem firmemente à alimentação baseada na utilização de rações comerciais tradicionais e não possuem interesse em adotar a Alimentação Natural. A antipersona é ilustrada na Figura 7.

Figura 4 – Persona 1

	PERSONA Raquel		
PERFIL • Solteira • 29 anos • Dona de cães e gatos • Empresária • Formada em Adm	COMPORTAMENTO • Pessoa Ativa • Preocupada com saúde e alimentação • Tem muita experiência de Alimentação Natural	NECESSIDADES • Tem dificuldades em pensar em novas receitas para seus <i>pets</i> • Tem dificuldades em calcular e dosar as porções de ingredientes quando prepara as refeições dos <i>pets</i>	

Fonte: Autores

Figura 5 – Persona 2

 PERSONA Amanda		
PERFIL • Solteira • 27 anos • Dona de gatos • Formada em Letras • Trabalha em loja de livros	COMPORTAMENTO • Pessoa caseira • Gosta de cozinhar • Evita ao máximo alimentos industrializados • Trata seus <i>pets</i> como se fossem filhos • Praticante de Alimentação Natural • Faz receitas utilizando modelos que achou na internet	NECESSIDADES • Procura uma forma mais segura de montar receitas para seus <i>pets</i>

Fonte: Autores

Figura 6 – Persona 3

 PERSONA João das Neves		
PERFIL • Casado • 74 Anos • Dono de cachorro • Ex-servidor público • Formado em Psicologia	COMPORTAMENTO • Antenado nas novidades • Trata seus pet como companheiro. • Adora pets • Busca inserir dietas baseadas em Alimentação Natural para seu cachorro	NECESSIDADES • Dificuldade em iniciar com Alimentação Natural para seu cachorro por falta de conhecimentos técnicos

Fonte: Autores

Figura 7 – Antipersona

	PERSONA Seu Bernardo		
PERFIL • Divorciado • 52 anos • Dono de cães e gatos • Contador • Formado em Ciências Contábeis	COMPORTAMENTO • Conservador em seus costumes. • Pouco antenado aos avanços da tecnologia • Gosta de <i>pets</i> mas mantém uma relação menos humanizada com o mesmo. • Alimenta <i>pets</i> com rações tradicionais	NECESSIDADES	

Fonte: Autores

5.2.2 Análise SWOT

A análise SWOT é uma ferramenta de planejamento estratégico utilizada para identificar Forças (*Strengths*), Fraquezas (*Weaknesses*), Oportunidades (*Opportunities*) e Ameaças (*Threats*) relacionadas à competição de negócios ou planejamento de projetos. Para o NattyPet, esta análise permite uma avaliação aprofundada dos fatores internos e externos que podem influenciar o sucesso do projeto. A Figura 8 apresenta a análise realizada para a aplicação, destacando como pontos fortes a adesão a guias nutricionais oficiais e o foco na usabilidade, enquanto aponta como fraqueza a dependência de dados externos (USDA). As oportunidades residem no mercado crescente de humanização de *pets*, e as ameaças incluem a concorrência e a desinformação online sobre o tema.

Figura 8 – Análise SWOT do Projeto NattyPet

Forças • Multiplataformas • Baseado em guias nutricionais • Acesso a base de dados de alimentos americana • Foco no Tutor Leigo • Receitas Personalizadas	Oportunidades • Mercado Pet crescente • Ausência de ferramentas destinadas a tutores. • Aumento da Procura por Alimentação Natural • Tendência de Humanização dos Pets
Fraquezas • Base de dados limitada • Orçamento limitado • Não atender outros tipos de pet • Dependência de dados externos (USDA)	Ameaças • Pesquisas não recomendarem implementação de AN (baixas chances) • Guias oficiais defasados (baixas chances) • Base de dados ser descontinuada (baixas chances)

Fonte: Autores

5.3 Análise de Risco

A gestão de riscos é um processo relevante para antecipar e mitigar potenciais problemas que possam impactar o sucesso do projeto. Para o NattyPet, os riscos foram categorizados em três áreas principais: Técnicos, de Gerenciamento de Projeto e de Produto/Usuário. O Quadro 5 detalha os riscos identificados, sua probabilidade e impacto, e as estratégias de mitigação adotadas.

Quadro 5 – Matriz de Análise de Risco do Projeto NattyPet

Risco Identificado	Prob.	Impacto	Estratégia de Mitigação
Riscos Técnicos			
Imprecisão ou ausência de dados nutricionais para certos alimentos na base de dados externa (USDA).	Média	Alto	Utilizar fontes de dados oficiais e validadas. Permitir, em versões futuras, que usuários profissionais (veterinários) possam cadastrar ou ajustar dados de ingredientes.
Baixo desempenho do <i>solver</i> de otimização ao lidar com receitas muito complexas (muitos ingredientes e restrições).	Baixa	Médio	Utilização de um <i>solver</i> de alta performance (HiGHS) e formatação de matrizes de restrições de forma esparsa (CSC) para otimizar o consumo de memória e processamento.
Riscos de Gerenciamento de Projeto			
Escopo de projeto crescente (<i>scope creep</i>), com adição de funcionalidades não planejadas para o MVP.	Média	Alto	Aderência estrita ao <i>backlog</i> priorizado, com validação constante junto à orientadora. Novas funcionalidades devem ser planejadas para iterações futuras.
Riscos de Produto e Usuário			
Entrada incorreta de dados do <i>pet</i> pelo usuário (e.g., peso, idade, nível de atividade), levando a cálculos errôneos.	Alta	Alto	Implementação de validações robustas no <i>backend</i> (pacote validator) e fornecimento de instruções claras e <i>tooltips</i> na interface do usuário.
Interpretação incorreta dos resultados pelo tutor, que pode assumir que a ferramenta substitui a orientação veterinária.	Média	Alto	Exibição de avisos e <i>disclaimers</i> claros na aplicação, informando que a ferramenta é um auxílio e não substitui a consulta a um profissional. Exigir aceite dos termos de uso.

Fonte: Autores

5.4 Backlog do Produto

O *Backlog* do Produto, conforme abordado no [Capítulo de Metodologia](#), trata-se de um artefato que concentra a lista de requisitos de software modelados como histórias de usuário necessárias para a compreensão das funcionalidades e características desejadas para o produto ([SUTHERLAND, 2010](#)). Cabe destacar que o *Backlog* é sequenciado de acordo com a prioridade de seus itens, sendo estes analisados de maneira individual.

Para realizar a priorização, atividade que busca ordenar a lista de requisitos de acordo com a importância de cada um, foi utilizada a técnica *MOSCOW*. Esta técnica consiste em classificar os requisitos em quatro categorias: *Must Have* (deve ter), *Should Have* (deveria ter), *Could Have* (poderia ter) e *Won't Have* (não terá). O *MOSCOW* é uma técnica de priorização muito utilizada para gerenciamento e análise do negócios, ou projetos de desenvolvimento de software, para alinhar o entendimento comum entre as partes interessadas sobre a importância atribuída à cada entrega de versão ([MIRANDA, 2022](#)).

Os autores optaram por utilizar o *Lean Inception* como metodologia de concepção do produto, uma abordagem colaborativa que visa alinhar as ideias da equipe e definir um MVP (*Minimum Viable Product*) de maneira ágil e eficiente ([CAROLI, 2018](#)). Essa metodologia facilita a convergência de diferentes perspectivas em um plano único e organizado, promovendo uma visão clara sobre as prioridades do produto e o valor entregue aos usuários. Como resultado desse processo, foi elaborado o *Backlog* do Produto do NattyPet, ilustrado na Figura 9, contendo as histórias de usuário elicitadas e priorizadas de acordo com a técnica *MOSCOW*.

Cabe ressaltar que cada *feature* foi alocada para diferentes versões de entrega do trabalho. A versão MVP corresponde à versão funcional inicial da aplicação; enquanto as demais são definidas como incrementos sequenciais.

Figura 9 – Backlog do NattyPet

ÉPICOS	FEATURES	HISTÓRIAS DE USUÁRIO	PRIORIDADE	VERSÃO
Alimentos	Pegar dados de alimentos de uma base oficial	Como Usuário, quero ter acesso a informações nutricionais de diversos alimentos a partir de uma fonte confiável, para poder escolher ingredientes para minhas receitas	MUST	MVP
	Filtrar alimentos	Como Usuário quero poder pesquisar alimentos por critérios como animal de origem, nome ou categoria para poder encontrar os alimentos mais rapidamente	SHOULD	MVP
Criação de Receitas	Balancear uma receita a partir de uma lista de ingredientes selecionados	Como Usuário quero que o sistema seja capaz de balancear uma receita, dada uma seleção prévia de ingredientes, para criar uma receita com ingredientes que eu quero utilizar	MUST	MVP
	Indicar % de completude dos requisitos	Como usuário quero poder ver o quanto dos requisitos nutricionais foram completos com dado ingrediente durante a criação de uma receita para verificar o balanceamento da mesma	MUST	MVP
	Fixar/limitar quantidade de um ingrediente	Como usuário desejo fixar a quantidade de um ou mais ingredientes de modo que o balanceamento automático altere somente os demais ingredientes, para que consiga criar a receita com base em quantidades que eu tenha acesso	SHOULD	V2
	Indicar quando a receita ultrapassou em muito os requisitos	Como Usuário quero sinalizado quando uma receita em criação tiver passado muito dos critérios de aceite, para não correr o risco de alimentar exageradamente meu pet	SHOULD	V2
	Geração automatizada de receitas	Como usuário quero conseguir gerar receitas que atendam os requisitos nutricionais do meu pet de maneira automática, para poder alimentar meus pets de maneira adequada	COULD	V2
Receitas	Manter Receitas	Como usuário, quero poder criar uma receita, para oferecer uma alimentação natural e balanceada para meu pet	MUST	MVP
		Como usuário, quero poder visualizar as informações das minhas receitas para poder fazer um acompanhamento mais eficaz das minhas receitas	MUST	MVP
		Como usuário, quero poder editar minhas receita para caso queira adaptar algum ingrediente	MUST	MVP
		Como usuário, quero poder deletar uma receita para manter apenas as receitas que me interessem	MUST	MVP
	Apontar para quais pets a receita é apropriada	Como usuário quero ser capaz de saber para quais dos meus pets uma dada receita é balanceada, para saber se posso preparar a mesma receita para mais de um pet	SHOULD	V4
Pets	Manter Pets	Como usuário quero visualizar a tabela nutricional de uma receita para ter uma visão detalhada dos macro e micro nutrientes da mesma	COULD	V4
		Como usuário, quero poder cadastrar meus pets para poder criar minhas receitas de maneira individual para cada um	MUST	MVP
		Como usuário, quero poder visualizar os pets que tenho cadastrados	MUST	MVP
		Como usuário, quero poder editar os pets para melhor representa-los no sistema	MUST	MVP
	Calcular Requisitos Nutricionais dos Pets	Como usuário, quero poder deletar meus pets para manter apenas os pets que me interessem	MUST	MVP
Usuário	Facilita a realização do teste de ECC	Como usuário quero que o sistema calcule os requisitos nutricionais dos meus pets para conseguir balancear adequadamente as receitas	MUST	MVP
	Manter Usuário	Como usuário quero poder avaliar o ECC dos meus pets para ter uma noção melhor do seu peso ideal	COULD	V3
		Como usuário, quero poder cadastrar meu próprio perfil para que possa acessar o sistema	MUST	MVP
		Como usuário quero poder visualizar meu perfil para poder verificar as informações que tenho cadastradas como receitas e pets	MUST	MVP
		Como usuário, quero poder editar meu perfil para melhor representa-lo no sistema	MUST	MVP
PWA	Sessão de Usuário	Como usuário, quero poder deletar meu perfil para que não possua nenhum vínculo deste perfil com o sistema	MUST	MVP
		Como usuário quero ser capaz de acessar minha conta através de um sistema de login para poder ter minhas informações privadas	MUST	MVP
	Login social	Como usuário quero ser capaz de sair da minha conta logada para não deixar a conta aberta no dispositivo	MUST	MVP
	Aplicação instalável	Como usuário quero ser capaz de fazer login usando alguma das minhas redes sociais como fonte de acesso para que não tenha que criar um novo usuário e senha	COULD	V5
	Acesso offline	Como usuário, quero ser capaz de instalar o app em meus dispositivos pessoais para poder acessá-lo mais facilmente	SHOULD	V5
		Como usuário, quero ser capaz de acessar dados dos meus pets e das minhas receitas em modo offline para poder acessar minhas informações quando não tiver acesso a internet	COULD	V5

Fonte: Autores

Figura 10 – Legenda em Cores das Versões do NattyPet

MVP
V2 - Geração automatizada
V3 - Teste de ECC
V4 - Melhorias na tela de Receita
V5 - Configurações adicionais

Fonte: Autores

5.5 Identidade Visual

A identidade visual da aplicação NattyPet foi desenvolvida para refletir a missão e os valores do projeto, garantindo uma comunicação adequada com os usuários e uma experiência consistente em todos os aspectos visuais. Adiante, são detalhadas as escolhas realizadas em três pilares principais: tipografia, paleta de cores e logotipo. Em seguida, aborda-se sobre a prototipagem do projeto.

5.5.1 Tipografia

A fonte escolhida para a aplicação é a Inter, uma tipografia amplamente utilizada em interfaces digitais. A escolha da fonte foi baseada em diversos fatores, como explicado por [Andersson \(2023\)](#):

- **Legibilidade:** a fonte Inter é otimizada para ser altamente legível em telas de diferentes tamanhos e resoluções, desde dispositivos móveis até monitores maiores;
- **Flexibilidade:** com uma grande variedade de pesos e estilos (*light*, *regular*, *bold*, dentre outros), a Inter permite criar hierarquias visuais claras e destacadas, sem a necessidade de utilizar fontes adicionais;
- **Modernidade e Neutralidade:** seu *design* moderno e neutro alinha-se à proposta tecnológica e acolhedora do NattyPet, conferindo profissionalismo sem perder a sensação de proximidade com o usuário, e
- **Acessibilidade:** a fonte foi projetada com suporte às diretrizes de acessibilidade *Web Content Accessibility Guidelines* (WCAG), garantindo que os textos sejam fáceis de ler por pessoas com deficiências visuais.

Essa escolha visa proporcionar aos tutores de *pets* uma experiência fluida e agradável durante o uso da aplicação.

5.5.2 Paleta de Cores

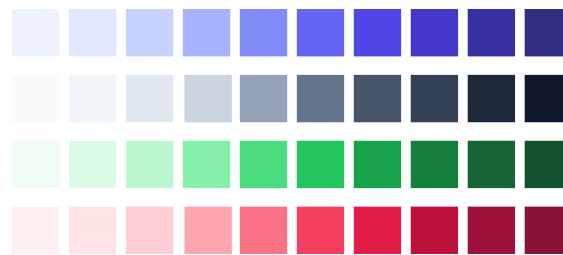
A paleta de cores da aplicação foi definida com base em princípios de *design* inspirados no sistema Material 3. Este sistema fornece diretrizes claras para a criação de paletas funcionais e harmônicas, garantindo que as cores desempenhem papéis específicos e contribuam para uma experiência visual equilibrada ([TEAM, 2023](#)).

O processo de definição da paleta consistiu nas seguintes etapas:

- **Cores Principais:** foram escolhidas cores para representar a identidade da marca e que estejam alinhadas à temática da aplicação;

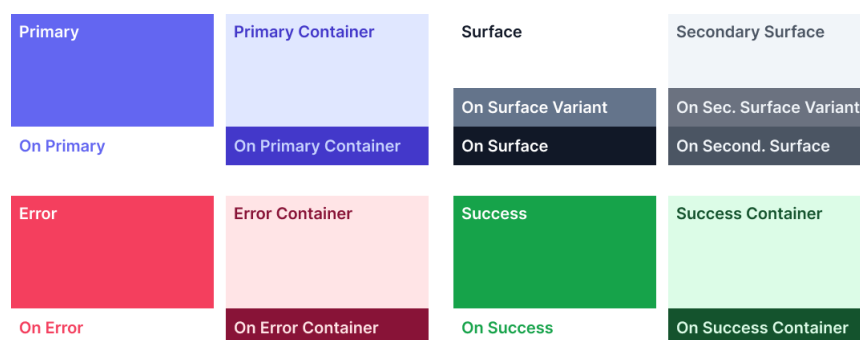
- Geração de Tons: a partir das cores principais, foram gerados tons claros e escuros, criando uma paleta completa e coerente, e
- Atribuição de Funções: cada cor foi atribuída a papéis específicos na aplicação, seguindo o modelo do Material 3:
 - *Primary* e *On Primary*: para destacar elementos principais, como botões de ação;
 - *Surface* e *On Surface*: para fundos e textos sobrepostos, garantindo legibilidade;
 - *Error* e *On Error*: para mensagens de erro, com contraste suficiente para chamar atenção, e
 - *Success* e *On Success*: para mensagens de sucesso, utilizando tons que transmitam tranquilidade e confirmação.

Figura 11 – Cores Principais e seus Tons



Fonte: Autores

Figura 12 – Paleta do NattyPet Orientando-se pelo Sistema do Material 3



Fonte: Autores

5.5.3 Logotipo

O logotipo da NattyPet, ilustrado na Figura 13, foi concebido para capturar a essência da aplicação de maneira simples e memorável. A escolha da simbologia foi baseada nos seguintes fatores:

- Relevância Temática: representa diretamente o conceito de criação de receitas saudáveis e balanceadas para *pets*;
- Apego Emocional: animais como gatos evocam uma conexão emocional forte com os tutores, enquanto o chapéu de chef reforça a ideia de cuidado especial com a alimentação, e
- Estilo Simpático e Acolhedor: o *design* foi mantido minimalista e amigável, criando uma identidade que é tanto profissional quanto acessível.

Figura 13 – Logotipo do NattyPet



Fonte: Autores

O logotipo é uma representação visual do compromisso da NattyPet com a qualidade e a satisfação dos seus usuários, reforçando a proposta de valor do projeto.

5.5.4 Protótipo de Alta Fidelidade

Um protótipo de alta fidelidade é uma representação detalhada e funcional de um software. Trata-se de uma etapa importante do processo de *design* de software, que permite que as funcionalidades do sistema sejam testadas antes de serem devidamente implementadas. O protótipo também permite aos *stakeholders* visualizarem e interagirem com a aparência e as funcionalidades do software, ajudando a validar ou ajustar suas expectativas e seus requisitos (RUDD; STERN; ISENSEE, 1996).

Como mencionado no Capítulo 3, o protótipo do presente trabalho foi construído com a ferramenta Figma, com o propósito de simular o comportamento real do software. O protótipo contendo as principais páginas da plataforma NattyPet está disponível para visualização no endereço: <https://www.figma.com/file/yPIp6udewaycbDjaiwfl0M?embed_host=share&kind=file&node-id=1427-10766&page-selector=0&viewer=1>.

Por se tratar de um protótipo envolvendo várias telas do *WebApp*, são tratadas com maior clareza três telas principais especificadas no protótipo, que serviram de base para o desenvolvimento final.

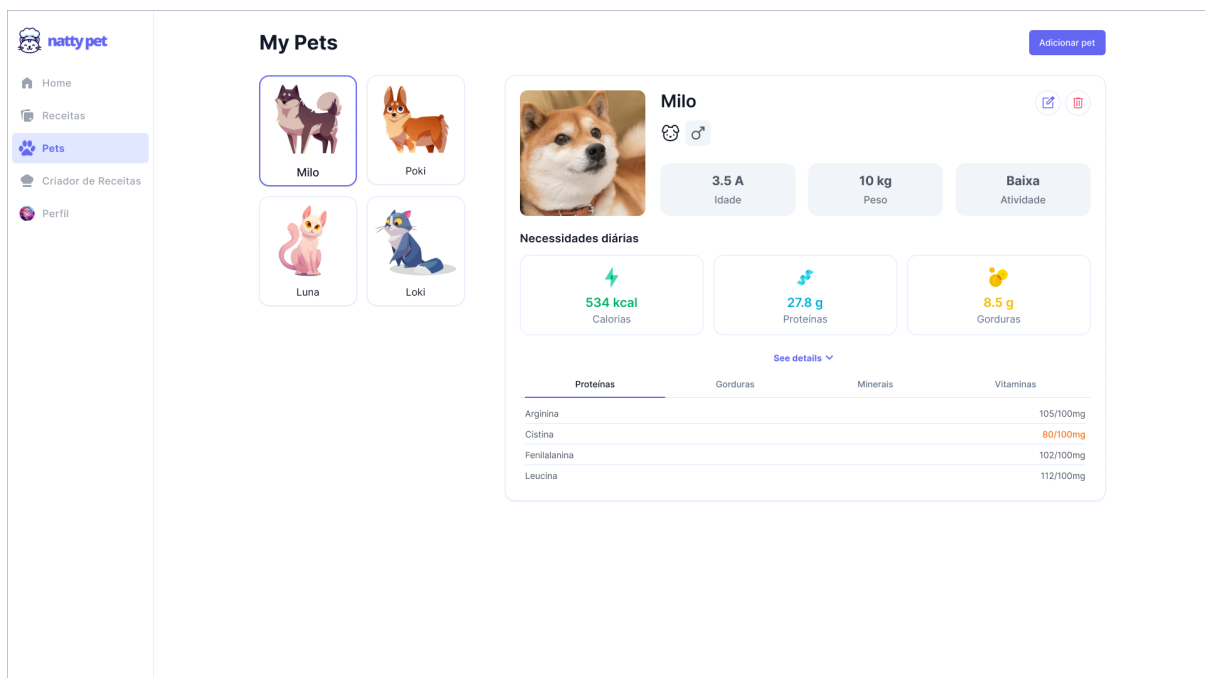
5.5.4.1 Tela de Perfil do *Pet*

A tela de perfil do *pet*, ilustrada na Figura 14, exibe as informações essenciais sobre cada animal cadastrado na aplicação. Os dados apresentados incluem:

- Nome, idade, atividade física e peso do *pet*, e
- Necessidades nutricionais diárias.

Essa tela serve como ponto de referência para a criação e a seleção de receitas, garantindo que cada refeição seja adequada às necessidades do *pet*.

Figura 14 – Tela de Perfil do *Pet*



Fonte: Autores

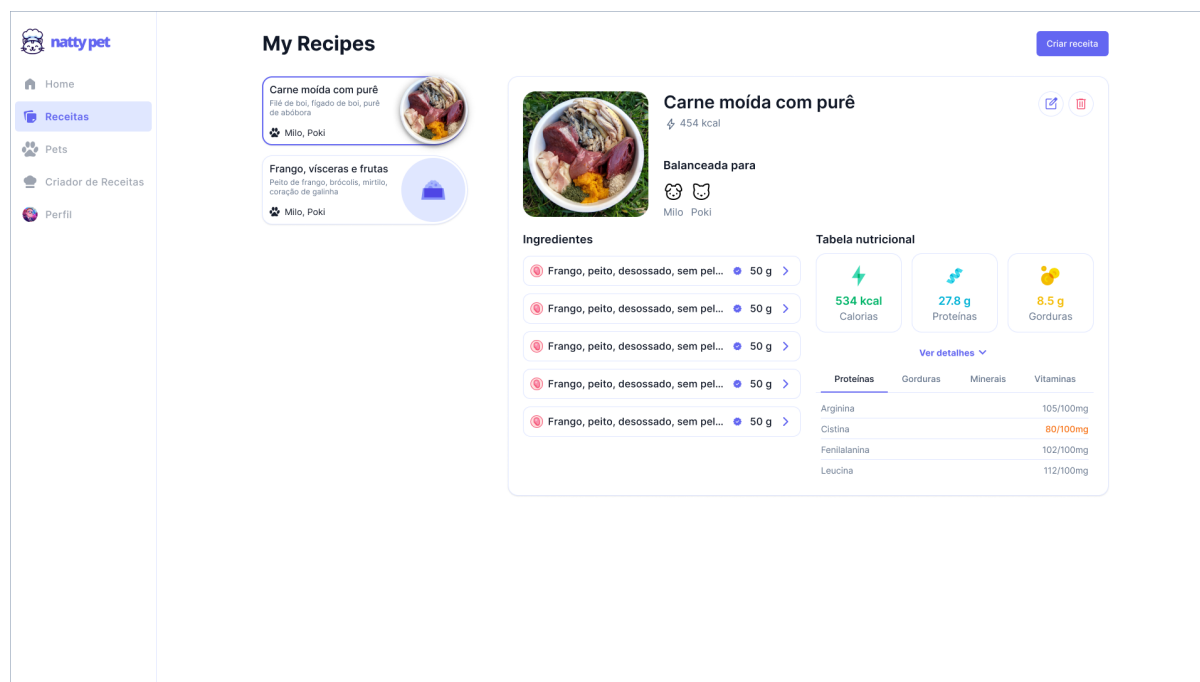
5.5.4.2 Tela de Detalhes de uma Receita

A tela de detalhes de uma receita, ilustrada na Figura 15, fornece todas as informações necessárias sobre uma refeição específica, permitindo que o tutor avalie sua adequação antes de prepará-la. Os elementos exibidos incluem:

- Lista de ingredientes e suas respectivas quantidades;
- Tabela nutricional detalhada, incluindo calorias, proteínas, gorduras, carboidratos, vitaminas e minerais presentes na receita, e
- Indicação dos *pets* para os quais a receita é apropriada.

Essa tela auxilia os tutores na escolha de receitas bem equilibradas e alinhadas às exigências nutricionais de seus animais.

Figura 15 – Tela de Detalhes da Receita



Fonte: Autores

5.5.4.3 Tela de Criação de Receita

A tela de criação de receita, ilustrada na Figura 16, versa sobre o processo interativo de seleção e balanceamento dos ingredientes. Essa funcionalidade permite que o tutor personalize receitas conforme as necessidades do *pet*, seguindo um fluxo estruturado:

- Seleção de ingredientes: o usuário escolhe os alimentos que deseja incluir na receita a partir do banco de dados da aplicação;
- Ajuste de quantidades: com base nos valores nutricionais de cada ingrediente, o sistema auxilia o tutor na definição das proporções adequadas para atender aos requisitos do *pet*;
- Análise nutricional: o sistema exibe um resumo com a distribuição de macronutrientes e micronutrientes da receita em relação às necessidades do *pet*, destacando possíveis excessos ou deficiências, e
- Salvamento: o tutor pode salvar a receita para uso posterior.

Essa tela oferece uma abordagem prática e interativa para a formulação de dietas personalizadas, garantindo que as refeições atendam aos padrões nutricionais recomendados.

Figura 16 – Tela de Criação de Receitas

Fonte: Autores

Por fim, cabe ressaltar que foi utilizada a ferramenta Maze, já explicada brevemente no [Capítulo 3](#), para realizar testes orientados à usabilidade, ainda na versão prototipada. Uma vez alinhada a proposta aos interesses reais dos usuários, foi conduzido o processo de desenvolvimento das telas. Para os testes de usabilidade, os usuários receberam Termos de Consentimento Livre e Esclarecido (TCLE). No [Apêndice B](#), há um template do TCLE elaborado para esse projeto.

5.6 Arquitetura da Solução

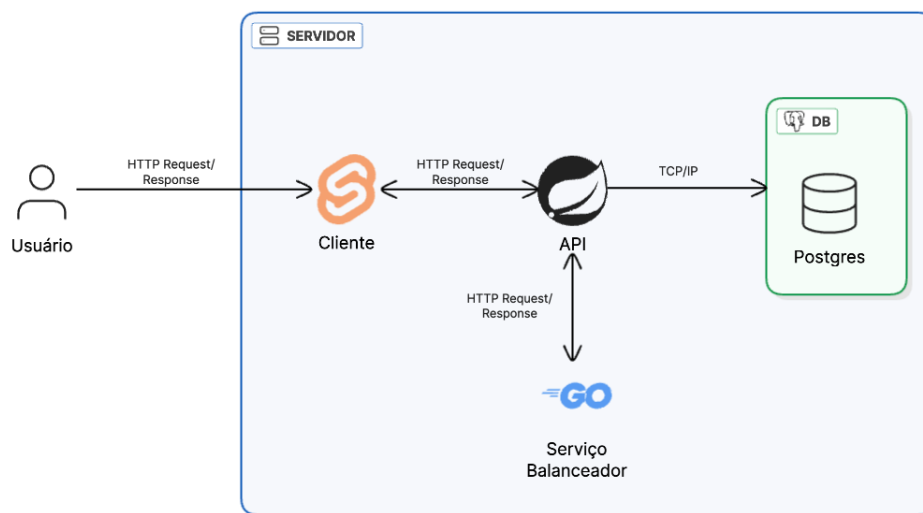
A arquitetura da aplicação NattyPet é estruturada em três módulos principais: Cliente (ou *Frontend*); Servidor (parte API interna do sistema), e Servidor (parte Serviço de Balanceamento de Receitas). O módulo do Cliente é responsável por fornecer a interface de interação com o usuário, constituindo-se de um monolito utilizando o *framework* SvelteKit. Quanto à API do sistema, esta é responsável por gerenciar os dados dos usuários, *pets*, receitas e alimentos, sendo desenvolvida na forma de uma API *RESTful* em Spring Boot. Por fim, o módulo do Serviço de Balanceamento foi desenvolvido com a linguagem Go, sendo responsável pelo cálculo dos requisitos nutricionais dos *pets* e pelo balanceamento das receitas.

Como abordado no Capítulo de [Suporte Tecnológico](#), a plataforma Vercel é utilizada para hospedar o *frontend* da aplicação. Já a API Spring Boot e o Serviço em Go são encapsulados em *containers* Docker e hospedados na plataforma Google Cloud

Run, garantindo escalabilidade automática e uma infraestrutura gerenciada. Finalmente, o banco de dados PostgreSQL é hospedado na plataforma Neon, que oferece uma solução otimizada e robusta para o armazenamento de dados.

Do ponto de vista de comunicação entre os módulos, o Cliente, sendo a interface com o usuário, comunica-se com a API Spring Boot para realizar operações específicas, tais como: gerenciamento de dados de usuários e informações de *pets* e receitas, além de consultas de alimentos. A API também realiza a intermediação entre o Cliente e o Serviço de Balanceamento para realizar o cálculo dos requisitos nutricionais dos *pets* e o balanceamento das receitas. Ambas as comunicações usam o protocolo HTTP. Detalhes dessas implementações podem ser vistos no Diagrama de Interação Arquitetural do Sistema NattyPet, no qual consta a Arquitetura do Sistema, disponível na Figura 17.

Figura 17 – Visão Geral da Arquitetura de Interação do Sistema NattyPet



Fonte: Autores

5.6.1 Dados dos Alimentos

As informações nutricionais dos alimentos disponibilizados pela aplicação são extraídos da API FoodCentral, disponibilizada pelo *United States Department of Agriculture* (USDA). No entanto, ao invés de consumir a API diretamente, alguns alimentos principais foram selecionados, processados e armazenados no banco de dados do projeto. Essa escolha foi feita por diversos motivos, como:

- Tradução e Idioma: a API do USDA fornece informações apenas em inglês. Para oferecer uma experiência mais amigável, considerando o público alvo da aplicação, os dados desejados precisaram ser traduzidos para o português;
- Redução de Complexidade: caso os dados fossem consumidos diretamente da API, seria necessário implementar um mecanismo de tradução eficiente e consistente, o

qual seria consumido a cada consulta realizada, o que aumentaria a complexidade do sistema e afetaria seu desempenho, e

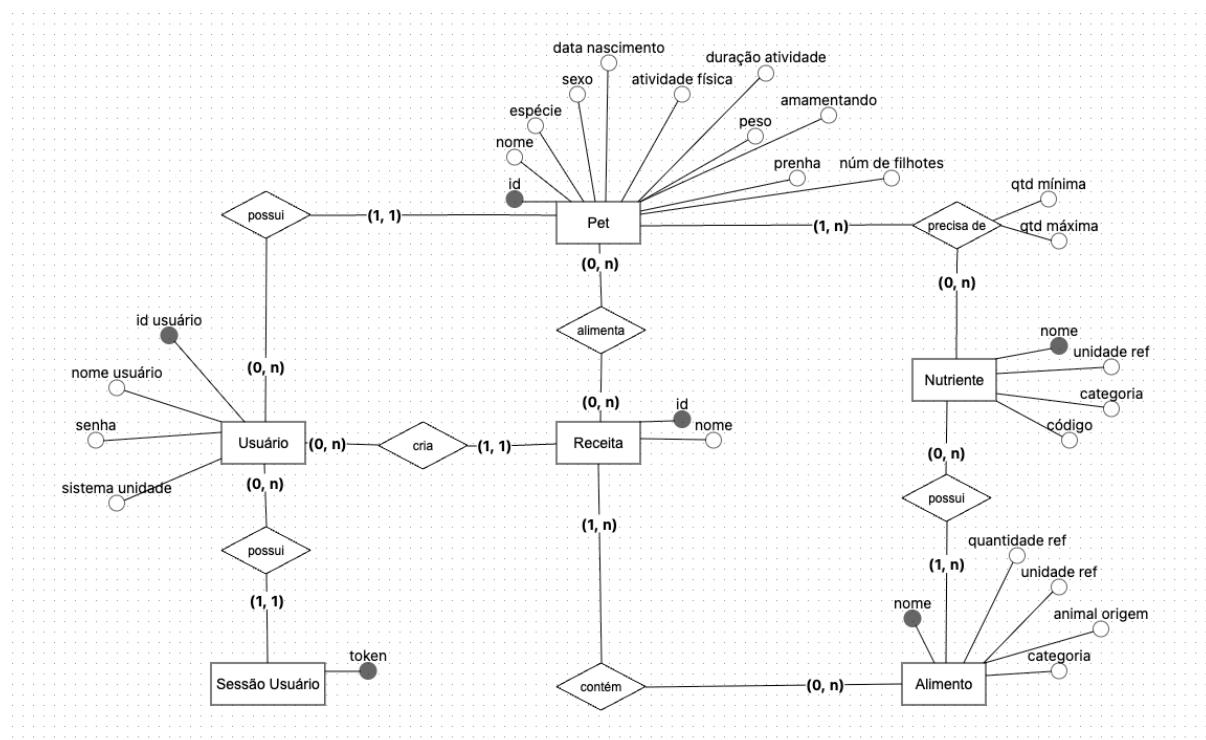
- Formato dos Dados: o armazenamento dos dados permitiu que estes fossem organizados e formatados de modo a facilitar seu consumo pelo sistema.

5.6.2 Visão de Dados

Com relação à Visão de Banco de Dados, esta compreende diagramas associados à estrutura interna e à organização do Banco de Dados do sistema. Como a plataforma NattyPet utiliza um banco de dados relacional, foram adotados modelos orientados a esse tipo de banco de dados.

Diagramas de Banco de Dados Relacionais têm como objetivo modelar a estrutura, a qual deve seguir as diretrizes de um modelo relacional, para armazenamento dos dados. O Diagrama Entidade-Relacionamento (DER) é apresentado na Figura 18, e pode ser conceituado como uma representação gráfica, sendo o primeiro nível de modelagem para visualização dos relacionamentos entre as diferentes entidades da aplicação.

Figura 18 – Diagrama Entidade-Relacionamento do NattyPet

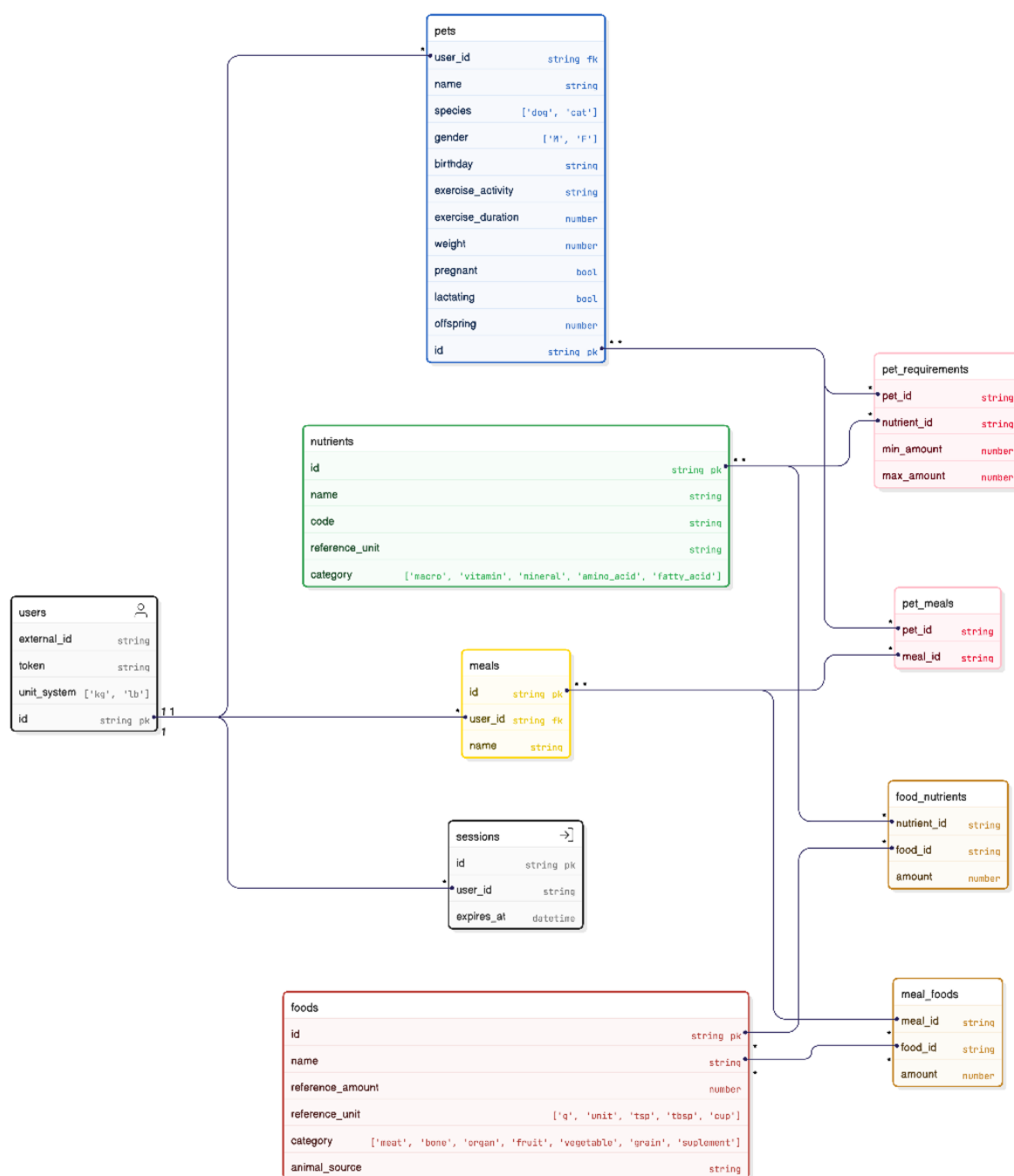


Fonte: Autores

Com a finalidade de uma melhor visualização do Diagrama Entidade-Relacionamento, este artefato também pode ser visualizado em maior detalhes no endereço <<https://app.brmodeloweb.com/#!/publicview/678bc38fdd16ef425b7da974>> com uma melhor qualidade de imagem, e mais espaço para apresentação.

Já o Diagrama Lógico de Dados (DLD), ilustrado na Figura 19, descreve como os dados são armazenados nas estruturas físicas do Sistema de Gerenciamento de Banco de Dados, e como estes se relacionam. O DLD também pode ser visualizado em maior detalhes no endereço: <<https://app.eraser.io/workspace/qKkH8Qc26wiUJ0KGBZZH>>.

Figura 19 – Diagrama Lógico de Dados do NattyPet

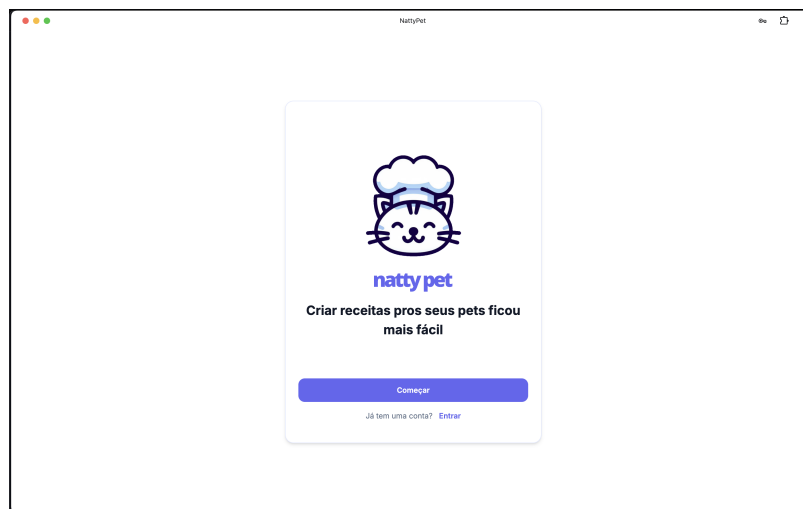


Fonte: Autores

5.7 Funcionalidades

Esta seção demonstra o fluxo de interação do usuário com a aplicação NattyPet, apresentando as telas que compõem a sua interface. O objetivo é ilustrar como o design da aplicação guia o tutor de forma intuitiva desde o primeiro acesso até a criação de uma dieta balanceada para seu animal de estimação. A jornada começa na tela de acesso, conforme visto na Figura 20.

Figura 20 – Tela de Acesso (Login)



Fonte: Autores

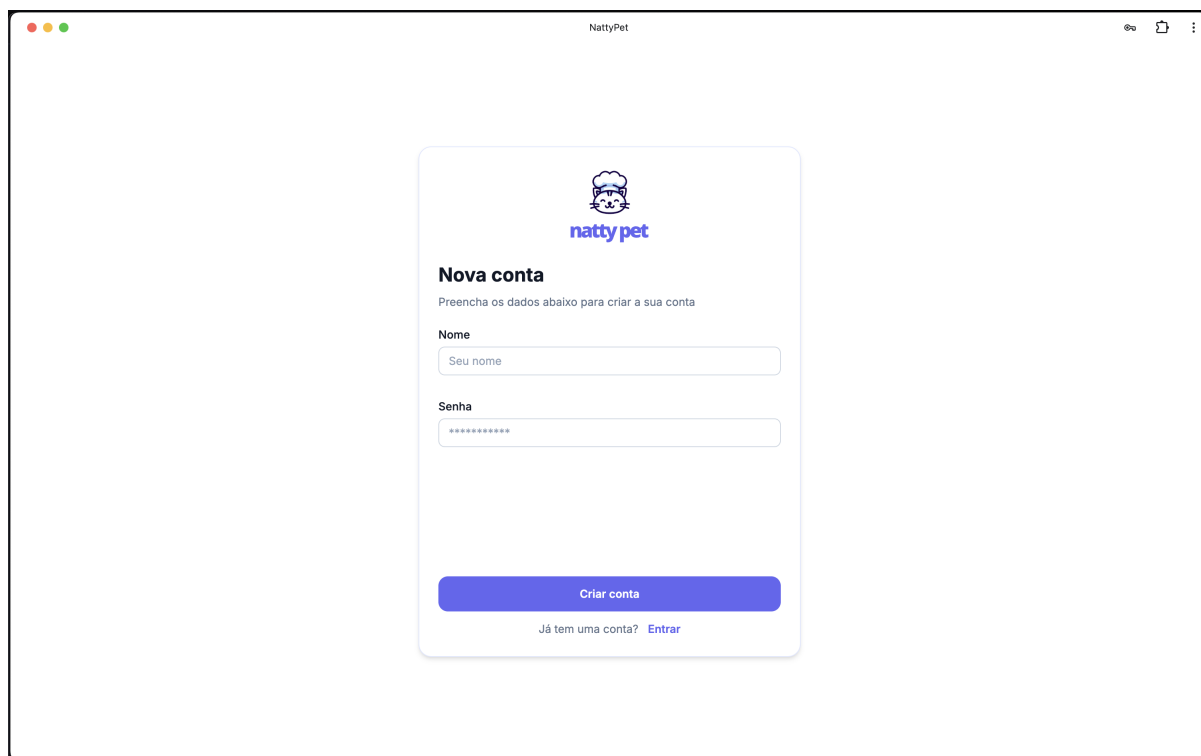
5.7.1 Cadastro e Autenticação de Usuário

O primeiro passo para utilizar a plataforma é o cadastro de uma conta de usuário, um processo simplificado para coletar as informações básicas necessárias para a criação do perfil, conforme ilustrado na Figura 21. A autenticação é gerenciada de forma segura, permitindo que o usuário acesse suas informações, *pets* e receitas salvas em qualquer dispositivo.

5.7.2 Tela Home

Após a autenticação, o usuário é direcionado para a tela *Home*, apresentada na Figura 22. Esta tela serve como um painel de controle central, oferecendo acesso rápido às principais seções da aplicação: gerenciamento de *pets*, visualização de receitas, acesso ao criador de receitas e ao perfil do usuário. A interface foi projetada para ser limpa e funcional, com uma navegação clara que guia o tutor através das funcionalidades do NattyPet.

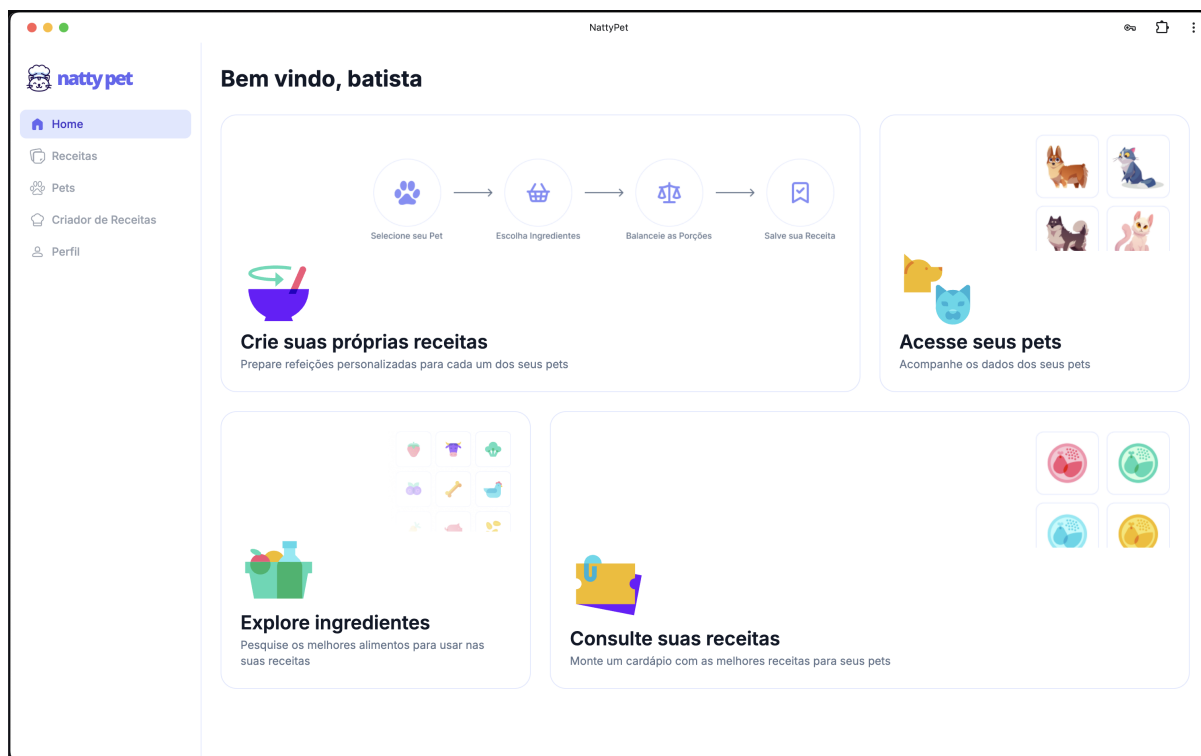
Figura 21 – Tela de Cadastro de Usuário



A interface de usuário para o cadastro no NattyPet. No topo, há uma barra de navegação com o nome 'NattyPet' e ícones de busca, favoritos e menu. O formulário centralizado contém o logo 'natty pet' com um cachorro estilizado. Abaixo, o título 'Nova conta' é seguido por uma instrução: 'Preencha os dados abaixo para criar a sua conta'. Há dois campos de entrada: 'Nome' com o placeholder 'Seu nome' e 'Senha' com caracteres ocultos por pontos. Um botão azul 'Criar conta' está na base do formulário, com um link 'Entrar' para quem já tem uma conta.

Fonte: Autores

Figura 22 – Tela Home Após Login

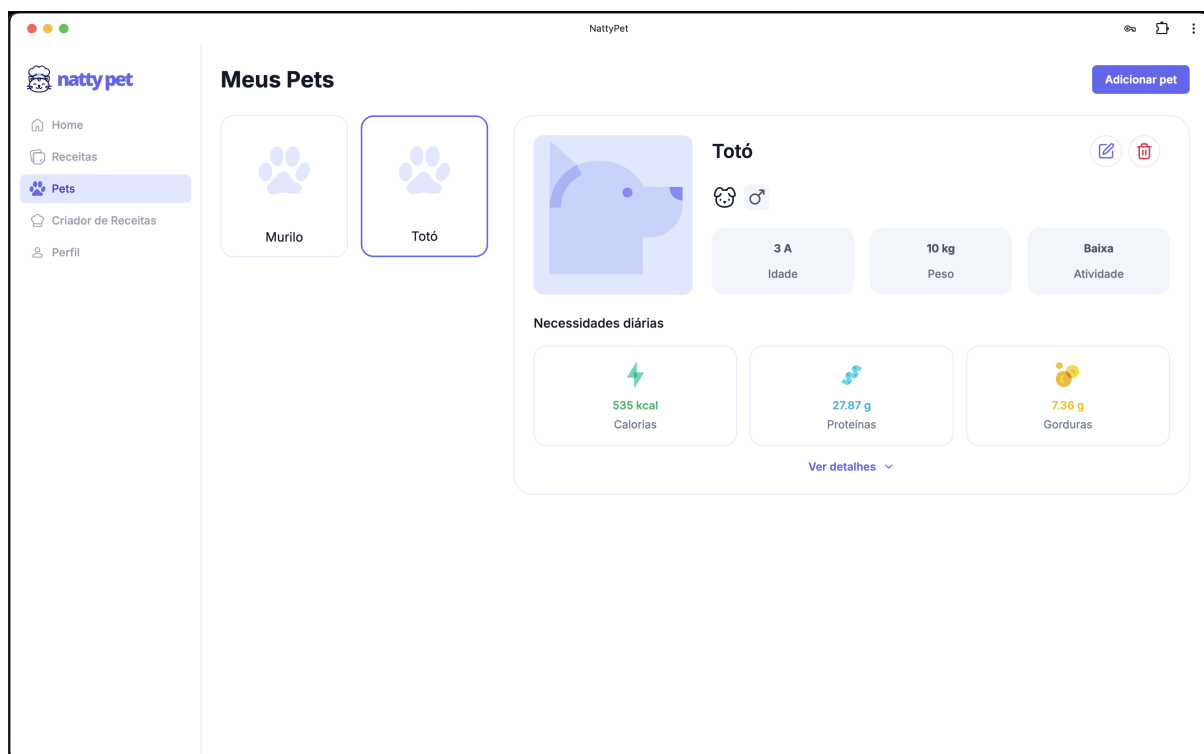


Fonte: Autores

5.7.3 Gerenciamento de Pets

O item de gerenciamento de *pets* permite ao usuário cadastrar um ou mais animais de estimação. Para cada *pet*, são solicitados dados relevantes como espécie (cão ou gato), idade, peso, nível de atividade e estado fisiológico (ex. castrado, gestante, lactante), como detalhado na Figura 23. Com base nessas informações, o módulo **calculator** do **balancing-service** determina as necessidades energéticas e nutricionais diárias do animal, que servem de base para o balanceamento das receitas.

Figura 23 – Tela de Gerenciamento de Pets



Fonte: Autores

5.7.4 Gerenciamento de Receitas

O gerenciamento de receitas é a funcionalidade central do NattyPet. O processo começa na tela de criação, ilustrada na Figura 24, onde o usuário pode selecionar ingredientes de uma vasta base de dados. À medida que os ingredientes são adicionados, o sistema interage com o **balancing-service** para:

- **Resolver o Problema de Otimização:** Utilizando Programação Linear, o serviço calcula as quantidades ideais de cada ingrediente para atender às restrições nutricionais do *pet* selecionado;

- **Fornecer Feedback Visual:** A interface exibe em tempo real o quão perto a receita está de atingir as metas nutricionais, com barras de progresso para os principais nutrientes, e
- **Salvar a Receita:** Uma vez que a receita esteja balanceada, o usuário pode salvá-la em seu perfil para consulta futura.

Figura 24 – Tela de Criação de Receita

Criador de Receitas

Nova receita para Totó

Cancelar Salvar receita

Minha Receita

Requisitos de Totó 465.9 / 535 kcal

Proteínas 69.7/27.87g Gorduras 10.3/7.36g

Vitaminas Minerais

Ver detalhes

Ingredientes

Balancear ingredientes

Peito, frango, sem osso e sem pele, cru 100 g X

Iogurte grego natural desnatado 100 g X

Fígado, pato, cru 100 g X

Rim, bovino, cru 100 g X

Escolha Ingredientes

Selecione ingredientes para adicioná-los na sua receita

mirtilo Categoria Animal

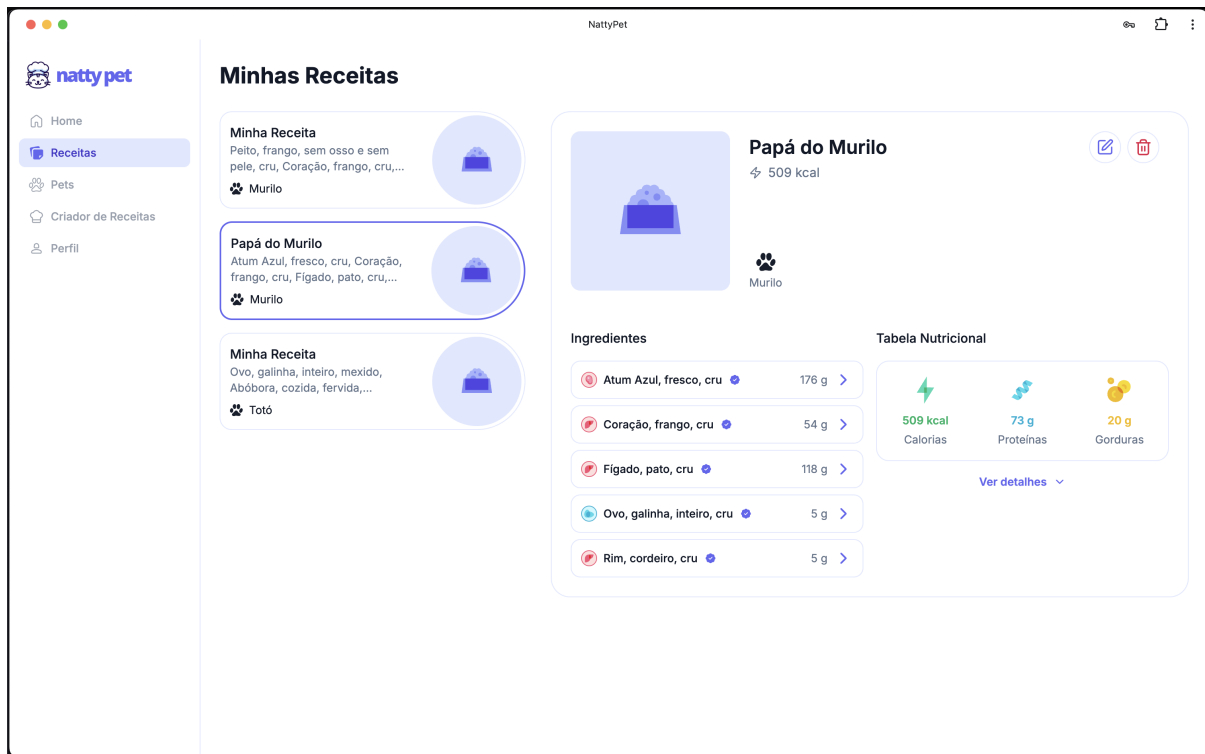
Peito, frango, sem osso e... X Iogurte grego natural des... X Fígado, pato, cru X Rim, bovino, cru X Mirtilo, cru X

Mirtilo, cru 100 g

Fonte: Autores

Após salvas, as receitas podem ser facilmente acessadas na tela de gerenciamento, ilustrada pela Figura 25. Nesta seção, o usuário tem uma visão geral de todas as suas criações, podendo visualizá-las, editá-las ou excluí-las, mantendo seu livro de receitas sempre organizado.

Figura 25 – Tela de Visualização das Receitas Salvas



Fonte: Autores

5.8 Algoritmo de Balanceamento de Receitas

Conforme justificado no [Capítulo 2](#), o núcleo da funcionalidade de balanceamento de receitas do NattyPet é um problema de otimização resolvido através de Programação Linear. Esta seção detalha como o problema foi modelado e implementado no microserviço de balanceamento (`balancing-service`), desenvolvido em Go.

5.8.1 Modelagem do Problema

A tradução do desafio nutricional para um modelo matemático de Programação Linear envolve a definição precisa de variáveis, da função objetivo e, prioritariamente, das restrições que garantem uma dieta segura e balanceada.

5.8.1.1 Variáveis de Decisão e Função Objetivo

As **variáveis de decisão**, x_i , representam a quantidade em gramas de cada ingrediente i selecionado para a receita. A **função objetivo** do sistema foi definida para minimizar a soma total das quantidades dos ingredientes:

$$\text{Minimizar } Z = \sum_{i=1}^n x_i$$

Esta abordagem incentiva o uso eficiente dos ingredientes, evitando receitas com porções excessivamente pequenas e numerosas de múltiplos alimentos. Uma alternativa, como a minimização de custos, foi considerada para trabalhos futuros.

5.8.1.2 Restrições Nutricionais e de Formulação

As restrições são os principais insumos do modelo, garantindo que a receita final seja nutricionalmente adequada. Na formulação do modelo algorítmico, são considerados vários fatores como: requisitos mínimos e máximos de nutrientes, distribuição calórica, dentre outros. No entanto, durante a fase de testes, observou-se que a introdução de um grande número de restrições, especialmente aquelas relacionadas às quantidades mínimas de todos os micronutrientes, levava a uma queda drástica na taxa de sucesso do algoritmo, resultando frequentemente em problemas inviáveis. Por essa razão, a modelagem priorizou o atendimento dos macronutrientes (calorias, proteínas, gorduras), que são os principais responsáveis por garantir o equilíbrio energético da dieta e são mais gerenciáveis dentro das equações de otimização.

Além disso, diferentemente dos macronutrientes, a satisfação de micronutrientes como vitaminas e minerais está muito mais associada à seleção dos ingredientes do que à eficiência do algoritmo em si. Por exemplo, se uma receita for composta por cinco alimentos que contêm quantidades insignificantes de determinado nutriente, não será possível atingir o mínimo exigido para esse nutriente sem aumentar desproporcionalmente as quantidades desses alimentos, o que por sua vez levaria a um excesso calórico indesejado. Assim, o atendimento completo dos micronutrientes exige uma curadoria prévia de ingredientes nutricionalmente ricos e diversificados.

Dito isso, mesmo com foco no cumprimento das metas de macronutrientes, verificou-se que o algoritmo frequentemente atinge uma boa parcela dos micronutrientes, conforme demonstrado na seção de [Resultados](#).

As restrições foram implementadas no pacote `recipe-balancer` do serviço da seguinte forma:

- **Restrições de Macronutrientes Mínimos:** O modelo assegura que os níveis mínimos de calorias, proteínas e gorduras sejam atingidos. Implementado na função `addMacrosConstraints`, gera inequações na forma:

$$\sum_{i=1}^n (\text{proteína}_i \cdot x_i) \geq \text{MetaProteína}$$

;

- **Restrições de Limites de Nutrientes:** Para micronutrientes que possuem um teto de segurança (e.g., certas vitaminas e minerais), o modelo impõe um limite

máximo, conforme os dados dos arquivos JSON de referência. Esta lógica é aplicada pela função `addNutrientLimitConstraints`;

- **Restrições de Distribuição Calórica:** Para garantir um equilíbrio saudável entre as fontes de energia, restrições mais complexas foram adicionadas na função `addCaloricDistributionConstraints`. Por exemplo, o modelo garante que a energia vinda das proteínas represente no mínimo 40% do total calórico e que a energia das gorduras se mantenha entre 25% e 35%. A recomendação mínima de 40% de energia proveniente de proteínas é consistente com dietas hiperproteicas seguras e eficazes, conforme observado por [Laflamme \(2005\)](#). Quanto às gorduras, o intervalo de 25% a 35% da energia total respeita os limites recomendados pela AAFCO, apontados em [Karr-Lilienthal \(2009\)](#);
- **Restrições de Dominância de Ingredientes:** Para promover a diversidade nutricional e evitar que um único ingrediente domine a receita, a função de restrição `addCaloricConstraints` implementa uma regra onde nenhum ingrediente pode contribuir com mais de 50% do total de calorias da refeição, e
- **Restrições de Quantidade Mínima:** Assim como a restrição anterior, esta visa garantir a relevância de todos os alimentos selecionados na receita e impedir que o algoritmo zere a quantidade de um deles. Para isso, foi estabelecido um limite inferior de 5 gramas por ingrediente selecionado, uma vez que quantidades menores seriam operacionalmente inviáveis na preparação prática das refeições pelos tutores.

5.8.2 Implementação com Go e *Solver* HiGHS

Para resolver o problema de Programação Linear modelado, foi desenvolvido um microserviço em Go, que orquestra os cálculos e utiliza um *solver* de otimização externo de alto desempenho.

5.8.2.1 Visão Geral da Arquitetura do Serviço

A escolha da linguagem Go foi motivada por seu alto desempenho, tipagem estática e excelente suporte à concorrência. O núcleo do serviço é o *solver* HiGHS (*High-performance software for linear optimization*), uma biblioteca de código aberto reconhecida por sua eficiência na resolução de problemas de otimização linear em larga escala.

5.8.2.2 Interface com HiGHS via CGO

Como a biblioteca HiGHS é escrita em C++, a comunicação entre o código Go e o *solver* foi realizada utilizando CGO. Foram criados *wrappers* em C++ (`highs_wrapper.cpp`

e `highs_wrapper.h`) que expõem as funcionalidades essenciais do HiGHS (criar um modelo, passar as matrizes de restrições, executar o *solver* e obter a solução) para funções em Go, conforme visto no pacote `internal/highs`.

5.8.2.3 Estrutura de Dados e Fluxo de Execução

O fluxo de balanceamento segue as seguintes etapas:

1. O cliente envia a lista de ingredientes desejados e o perfil do *pet* via HTTP;
2. O pacote `calculator` determina os requisitos nutricionais diários do animal com base em seu perfil (idade, peso, dentre outros), utilizando os dados base dos arquivos JSON em `internal/utis/nutrients`;
3. O pacote `recipe-balancer` constrói a matriz de restrições e os vetores de limites (inferiores e superiores) a partir dos dados dos ingredientes e dos requisitos calculados;
4. Para otimização de memória, a matriz de restrições é convertida para o formato CSC (*Compressed Sparse Column*), implementado em `csc_matrix.go`;
5. O modelo de Programação Linear, agora no formato adequado, é passado para o *solver* HiGHS através da camada CGO;
6. O HiGHS executa o algoritmo Simplex (ou outro de seus métodos internos) e retorna a solução ótima, que é um vetor com as quantidades em gramas de cada ingrediente, e
7. A solução é formatada em JSON e retornada ao cliente.

Esta abordagem modular e o uso de ferramentas especializadas garantem que o balanceamento de receitas seja não apenas nutricionalmente correto, mas também computacionalmente rápido e escalável.

5.8.2.4 Resultados e Desempenho

Para avaliação da acurácia e desempenho do algoritmo, foram formulados 11 perfis diferentes de animais, sendo 7 perfis de cães e 5 perfis de gatos. Estes perfis foram montados visando explorar os diferentes fatores que influenciam os requisitos nutricionais em cães e gatos respectivamente. Tais fatores foram abordados na seção de [Fatores de Influência](#). O algoritmo de balanceamento foi avaliado por meio de uma série de testes onde o algoritmo foi utilizado para balancear 6 receitas para cada um dos perfis de cães avaliados e 5 receitas para cada um dos perfis de gatos, compreendendo um total de 67

casos de teste. A Tabela 13 apresenta os resultados obtidos, onde cada linha representa a média dos resultados dos casos de teste de cada perfil.

Os testes tiveram como objetivo verificar o atendimento aos requisitos nutricionais, incluindo macro e micronutrientes, bem como mensurar o desempenho computacional do algoritmo. Dos 67 casos de teste, apenas em um (uma das receitas de gatos para o perfil de Gato Filhote) não foi possível encontrar uma solução válida, isto é, que cumprisse todas as regras exigidas no modelo como explicado na seção de [Restrições Nutricionais e de Formulação](#). Para os perfis caninos, o tempo médio de execução foi de aproximadamente 0,8537ms, com um percentual médio de 60,49% de micronutrientes satisfeitos. Já para os perfis felinos, o tempo médio foi ligeiramente inferior, 0,7128ms, com 61,27% dos micronutrientes atendidos em média. Considerando todos os perfis de forma consolidada, o tempo médio de execução geral foi de 0,7982ms, e o percentual médio de micronutrientes satisfeitos foi de 60,8%. Esses resultados indicam que o algoritmo foi capaz de gerar soluções nutricionalmente razoáveis em um tempo de execução bastante reduzido, mesmo sob múltiplos perfis e variações de receitas.

Tabela 13 – Médias de Micronutrientes e Tempo de Execução por Perfil de *Pet*

Perfil do <i>Pet</i>	Macronutrientes Satisfeitos	% Médio de Micronutrientes Satisfeitos	Tempo Médio de Execução (ms)
Cão Filhote	SIM	55,47%	1,3915
Cão Jovem Adulto	SIM	62,50%	0,8933
Cão Adulto	SIM	62,67%	0,8500
Cão Adulto Atv Alta	SIM	62,67%	0,7395
Cão Senior	SIM	61,23%	0,7599
Cadela Prenha	SIM	59,65%	0,6704
Cadela Amamentando	SIM	59,23%	0,6713
Gato Filhote	PARCIAL*	60,87%	0,7311
Gato Adulto Inativo	SIM	63,11%	0,7702
Gato Adulto Ativo	SIM	62,61%	0,6752
Gata Prenha	SIM	59,17%	0,7106
Gata Amamentando	SIM	59,57%	0,6769

*Não foi encontrada solução para uma das receitas para Gato Filhote e esta foi desconsiderada da média.

Fonte: Autores

5.9 Resumo do Capítulo

Este capítulo adentrou nas especificidades da aplicação NattyPet. De início, foi apresentada a [Contextualização](#) a respeito da relevância atual dos *pets* na sociedade, da ascensão de dietas como a Alimentação Natural e da complexidade de sua correta implementação. Na sequência, foi introduzida a proposta de solução para a problemática

retratada, detalhando o projeto [NattyPet](#), [Público-Alvo](#), [Análises de SWOT](#) e de [Análise de Risco](#). O escopo do produto foi definido através do [Backlog](#), e sua [Identidade Visual](#) foi estabelecida orientando-se pelo Material 3. Adicionalmente, foi apresentada a [Arquitetura da Solução](#), as Funcionalidades da aplicação, e a implementação do [Algoritmo de balanceamento](#) baseado em Programação Linear.

6 Análise dos Resultados

O presente capítulo tem o intuito de apresentar os resultados deste trabalho, bem como descrever as atividades de análise aplicadas sobre a ferramenta NattyPet, cuja proposta foi detalhada no [Capítulo 5](#). A estrutura deste capítulo foi organizada em ciclos de Pesquisa-Ação, uma abordagem detalhada no [Capítulo 4](#), para proporcionar uma visão clara e cadenciada da evolução do projeto.

Inicialmente, a seção [Ciclo 1: Validação do Protótipo Inicial](#) aborda a primeira rodada de testes de usabilidade, focada em validar os fluxos essenciais da aplicação. Em seguida, a seção [Ciclo 2: Refinamento e Otimização da Usabilidade](#) detalha os resultados obtidos após a implementação de melhorias, demonstrando um avanço significativo na experiência do usuário. A seção [Ciclo 3: Validação da Aplicação Final](#) apresenta a análise da versão funcional da ferramenta com base no *feedback* direto dos usuários.

Adicionalmente, a seção [Análise de Qualidade de Código e Testes](#) discute a verificação da qualidade do código-fonte por meio da ferramenta SonarQube e os resultados dos testes de cobertura para o *frontend* e o *backend*. Por fim, o [Resumo do Capítulo](#) consolida as principais conclusões e aprendizados obtidos ao longo dos ciclos de avaliação.

6.1 Ciclo 1: Validação do Protótipo Inicial

O primeiro ciclo da Pesquisa-Ação foi dedicado à validação do protótipo de alta fidelidade, utilizando a ferramenta Maze, conforme descrito no [Capítulo 3](#). O objetivo central era avaliar a intuitividade e a clareza dos fluxos de interação antes do início do desenvolvimento da aplicação funcional.

6.1.1 Coleta e Análise dos Dados

A coleta de dados foi realizada com 5 (cinco) *stakeholders*, que interagiram com o protótipo na plataforma Maze sem treinamento prévio. Foram propostas 7 (sete) missões essenciais que simulam a jornada do usuário na aplicação:

1. **Criação de Conta:** Realizar o cadastro inicial na plataforma;
2. **Cadastro de Pet:** Adicionar um novo animal de estimação ao perfil;
3. **Visualizar Pet:** Consultar as informações de um *pet* cadastrado;
4. **Editar Pet:** Modificar os dados de um *pet* existente;

5. **Gerar Receita Automática:** Utilizar a funcionalidade de criação de receita com balanceamento automático;
6. **Criação Manual de Receita:** Elaborar uma receita de forma manual, ajustando os ingredientes, e
7. **Visualizar Receita:** Consultar os detalhes de uma receita salva.

Os resultados, consolidados na [Tabela 13](#), revelaram que, embora todas as tarefas tenham alcançado uma taxa de sucesso de 100%, indicando que os fluxos eram funcionalmente compreensíveis, as métricas de **Erros de Clique** e **Média de Duração** apontaram desafios significativos. As missões de **Gerar Receita Automática** e **Editar Pet** apresentaram altas taxas de erro (57,8% e 46,4%, respectivamente), sugerindo que a interface não estava suficientemente clara para guiar os usuários de forma eficiente. A satisfação, embora positiva, demonstrou uma leve queda na tarefa de **Criação Manual de Receita**, reforçando a necessidade de otimizações.

Tabela 14 – Resultados Consolidados do Ciclo 1 de Testes de Usabilidade no Protótipo.

Fluxo	Total de Testadores	Escala de Satisfação dos Usuários	Taxa de Sucesso	Erros de Clique	Média de Duração
Criação de Conta	5	100% Muito Fácil	100%	41.9%	33.2s
Cadastro de Pet	5	100% Muito Fácil	100%	27.8%	113.7s
Visualizar Pet	5	100% Muito Fácil	100%	16.7%	5.6s
Editar Pet	5	100% Muito Fácil	100%	46.4%	20.4s
Gerar Receita Automática	5	80% Muito Fácil 20% Fácil	100%	57.8%	141.5s
Criação Manual de Receita	5	60% Muito Fácil 20% Fácil 20% Neutro	100%	29.8%	101.8s
Visualizar Receita	5	100% Muito Fácil	100%	23.1%	12.1s

Fonte: Autores

6.1.2 Plano de Ação e Divulgação dos Resultados

A análise dos dados do Ciclo 1 resultou em um plano de ação focado em otimizar a interface do protótipo. As principais ações definidas foram:

- Redesenhar componentes interativos nos fluxos de **Gerar Receita Automática** e **Editar Pet** para torná-los mais intuitivos;
- Melhorar a sinalização visual e o *feedback* em ações críticas para diminuir a ocorrência de cliques incorretos, e
- Simplificar o formulário de **Criação Manual de Receita** para melhorar a percepção de facilidade.

As melhorias foram implementadas no protótipo, e os resultados deste ciclo foram coletados, validando a transição para o próximo ciclo de testes.

6.2 Ciclo 2: Refinamento e Otimização da Usabilidade

O segundo ciclo teve como foco avaliar a eficácia das melhorias implementadas no protótipo refinado. O objetivo era validar se as alterações haviam solucionado os problemas de usabilidade identificados no Ciclo 1, e se o protótipo estava aderente o suficiente para guiar o desenvolvimento.

6.2.1 Coleta e Análise dos Dados

A coleta de dados seguiu a mesma metodologia do ciclo anterior, utilizando a plataforma Maze com os mesmos 5 *stakeholders*. Os resultados, apresentados na [Tabela 14](#), demonstraram um impacto positivo e significativo das melhorias.

Houve uma redução drástica na taxa de erros de clique em missões críticas, como **Gerar Receita Automática** (de 57,8% para 7,4%) e **Criação de Conta** (de 41,9% para 0%). A média de duração também diminuiu consideravelmente na maioria das tarefas, indicando maior fluidez e eficiência na interação. Esses resultados evidenciaram o sucesso das otimizações realizadas.

Tabela 15 – Resultados Consolidados do Ciclo 2 de Testes de Usabilidade no Protótipo.

Fluxo	Total de Testadores	Escala de Satisfação dos Usuários	Taxa de Sucesso	Erros de Clique	Média de Duração
Criação de Conta	5	100% Muito Fácil	100%	0%	10.3s
Cadastro de Pet	5	100% Muito Fácil	100%	13.3%	44.3s
Visualizar Pet	5	100% Muito Fácil	100%	28.6%	3.6s
Editar Pet	5	100% Muito Fácil	100%	40.0%	13.8s
Gerar Receita Automática	5	80% Muito Fácil 20% Fácil	100%	7.4%	26.8s
Criação Manual de Receita	5	80% Muito Fácil 20% Fácil	100%	49.2%	142.4s
Visualizar Receita	5	100% Muito Fácil	100%	9.1%	7.7s

Fonte: Autores

6.2.2 Plano de Ação e Divulgação dos Resultados

Os resultados positivos do Ciclo 2 validaram as melhorias de *design* e usabilidade. Com isso, o plano de ação subsequente foi aprovar o protótipo refinado como base para o desenvolvimento da aplicação funcional. O ciclo de refinamento do protótipo foi, então, concluído com sucesso, e seus resultados foram coletados para dar início à fase de implementação, que seria avaliada no Ciclo 3.

6.3 Ciclo 3: Validação da Aplicação Final

O terceiro e último ciclo de Pesquisa-Ação validou a aplicação final desenvolvida, garantindo que a experiência do usuário se manteve consistente, eficaz e satisfatória após a fase de implementação. Os testes foram conduzidos na versão funcional do *software*, com foco em missões essenciais para a jornada do tutor.

6.3.1 Coleta e Análise dos Dados

A validação contou com a participação de 4 (quatro) *stakeholders*, que já haviam participado dos testes realizados nos dois primeiros ciclos, que foram instruídos a realizar 6 (seis) missões na aplicação real:

1. **Missão 1:** Criar sua conta na aplicação;
2. **Missão 2:** Cadastrar um novo *pet*;
3. **Missão 3:** Visualizar os dados do *pet* cadastrado;
4. **Missão 4:** Editar as informações do *pet*;
5. **Missão 5:** Realizar a criação manual de uma receita, e
6. **Missão 6:** Visualizar a receita criada.

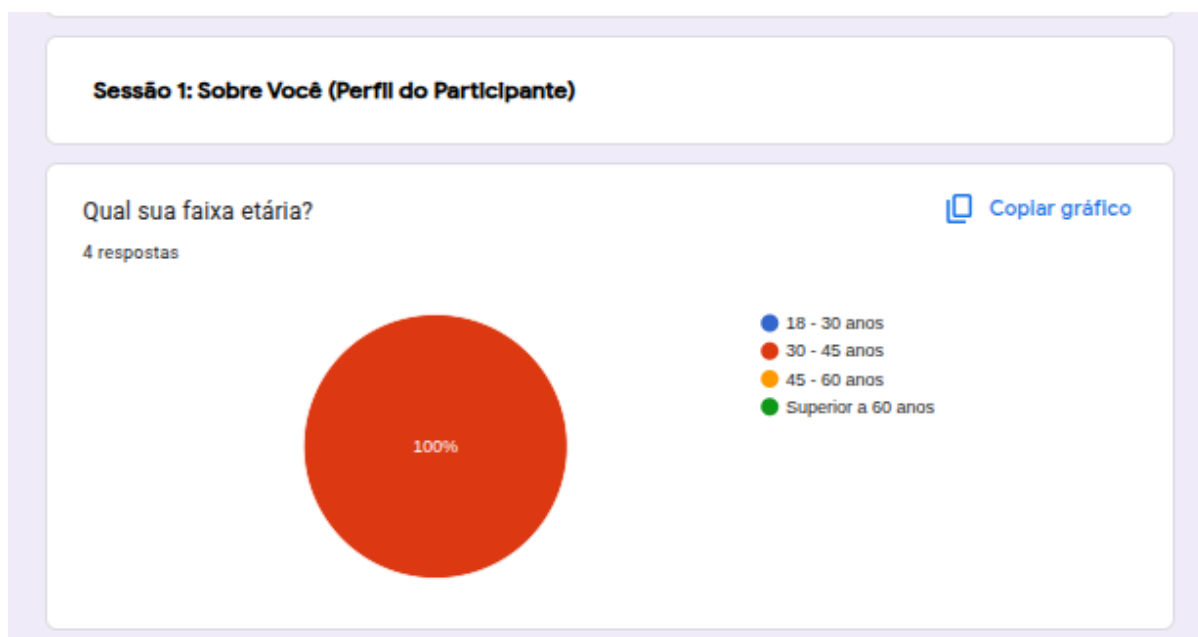
Após a execução das tarefas, os participantes preencheram um formulário de avaliação de usabilidade construído com o Google Forms. O questionário buscou coletar tanto dados demográficos; quanto a percepção sobre a usabilidade da ferramenta.

Primeiramente, analisou-se o perfil dos participantes. Conforme a Figura 26, todos (100%) pertencem à faixa etária dos 30 aos 45 anos. As Figuras 27 e 28 mostram, respectivamente, que 75% dos testadores já se consideravam familiarizados com o domínio de alimentação natural para *pets*, e 50% já tinham o costume de preparar receitas de alimentação natural com frequência. Baseado nesses dados podemos estabelecer uma correlação entre os usuários entrevistados e as personas Raquel e Amanda definidas no Capítulo 5.

A Figura 29 apresenta o nível de familiaridade dos participantes com aplicações similares. Notou-se que metade dos usuários não possuía familiaridade alguma; enquanto os demais se dividiam entre "Pouco familiar" e "Extremamente familiar", indicando um público diversificado.

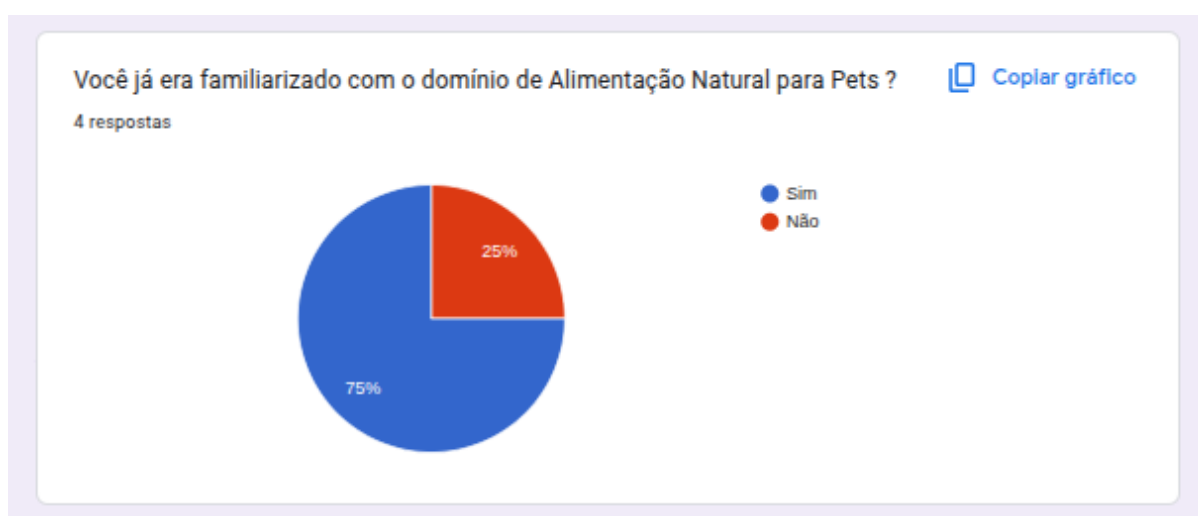
As Figuras 30 e 31 consolidam a avaliação da facilidade de execução das tarefas propostas. Todos os participantes classificaram as missões de "Criar conta", "Cadastrar Pet", "Visualizar Pet" e "Editar Pet" como "Muito Fáceis". Na criação de receita, a percepção variou entre "Fácil" e "Muito Fácil", e a tarefa de encontrar e visualizar uma receita foi majoritariamente avaliada como "Muito Fácil", com uma resposta indicando "Fácil".

Figura 26 – Gráfico da Faixa Etária dos Participantes.



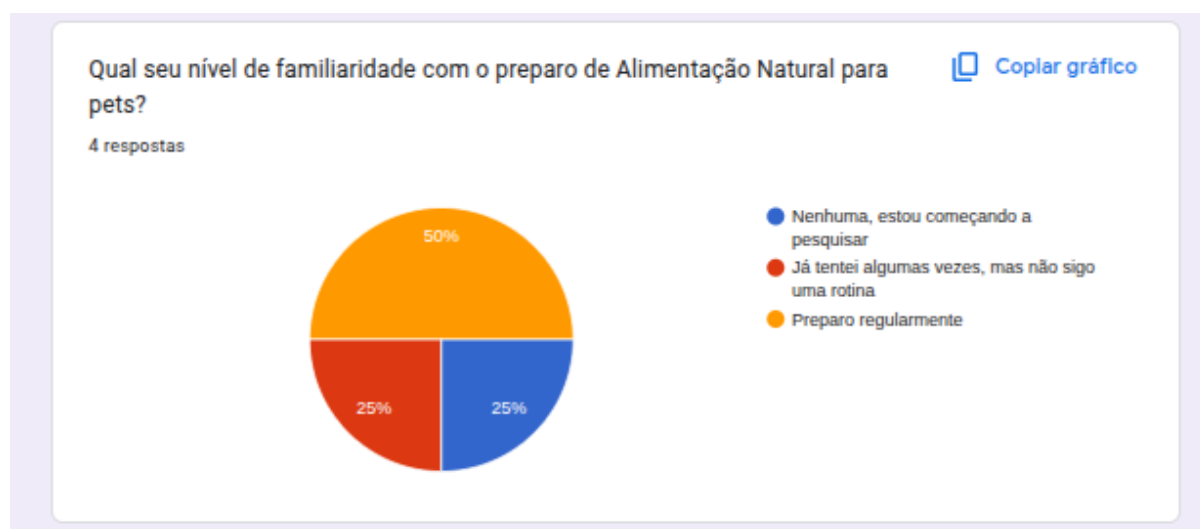
Fonte: Autores

Figura 27 – Gráfico de Familiaridade dos Usuários com o Domínio de Alimentação Natural.



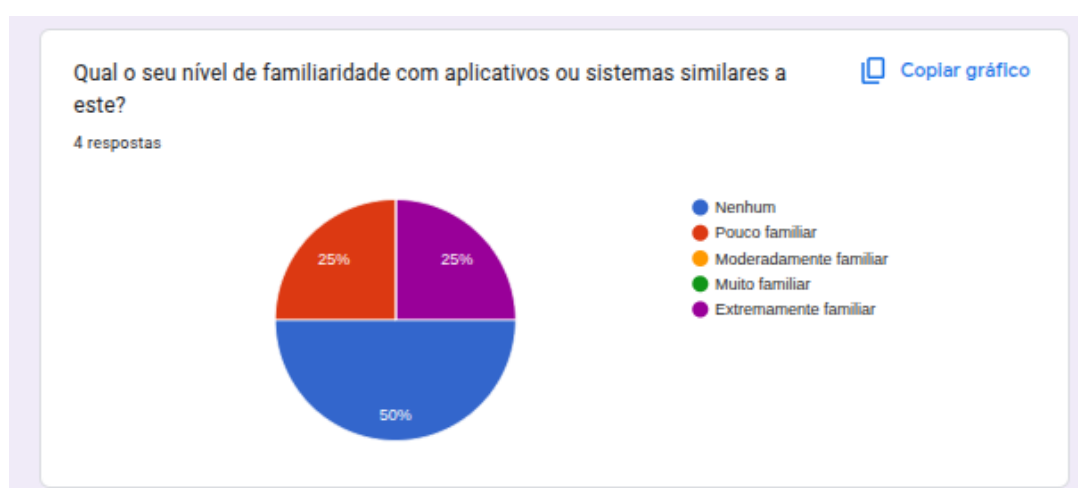
Fonte: Autores

Figura 28 – Gráfico de Familiaridade dos Usuários com o Preparo de Rações.



Fonte: Autores

Figura 29 – Gráfico de Familiaridade dos Usuários com Aplicações Similares.



Fonte: Autores

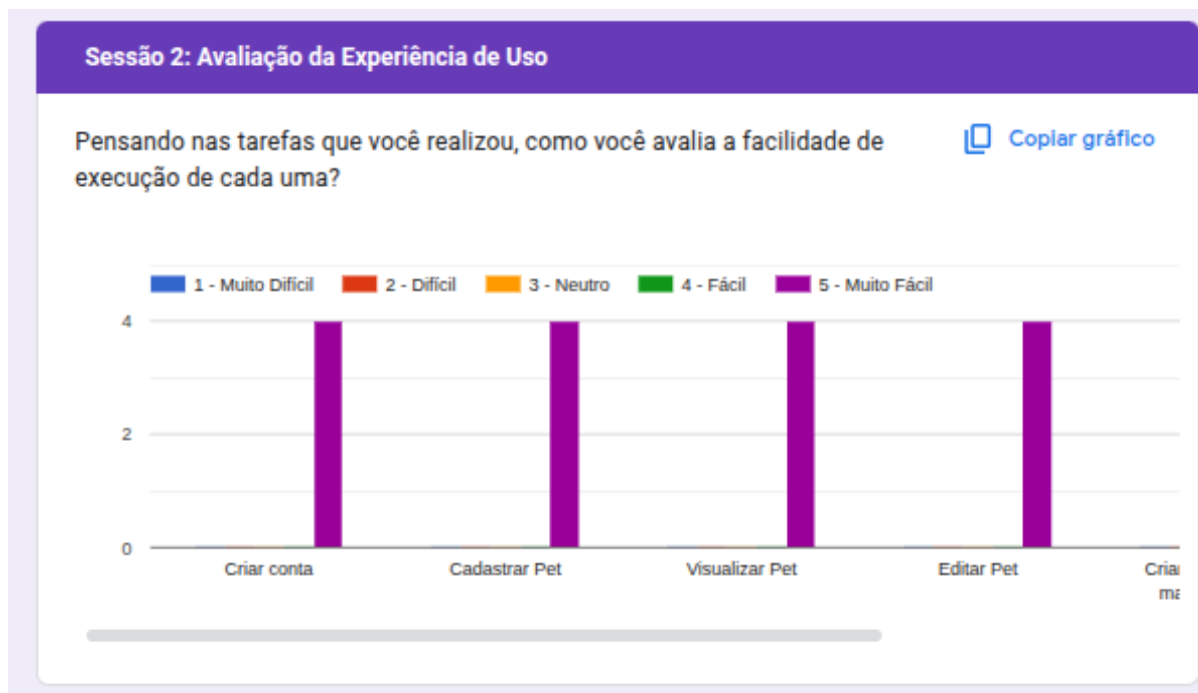
De modo geral, a facilidade de uso do aplicativo foi unanimemente classificada como "5 - Muito Fácil", como mostra a Figura 32, validando a usabilidade da interface.

Finalmente, a Figura 33 demonstra que 100% dos participantes indicaram a probabilidade máxima (10 de 10) de recomendar o NattyPet a um amigo ou colega, sendo um forte indicador de satisfação e aceitação do produto.

6.3.2 Plano de Ação Pós-Ciclo 3

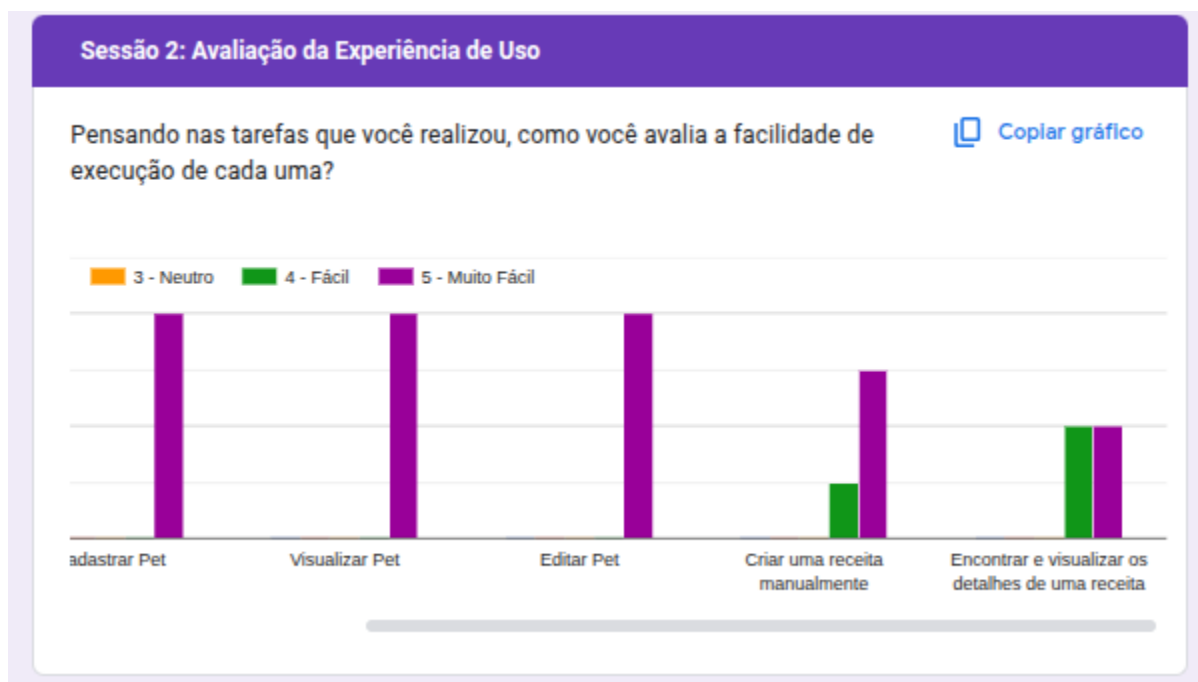
A análise dos resultados coletados pelo formulário permitiu identificar pontos de melhoria e *bugs* pontuais na aplicação. Com base nesse *feedback*, foi elaborado e executado um plano de ação final, que incluiu a correção dos problemas reportados e o refinamento de componentes da interface para aprimorar a experiência do usuário. Com a implementação

Figura 30 – Avaliação da Experiência de Uso (Parte 1).



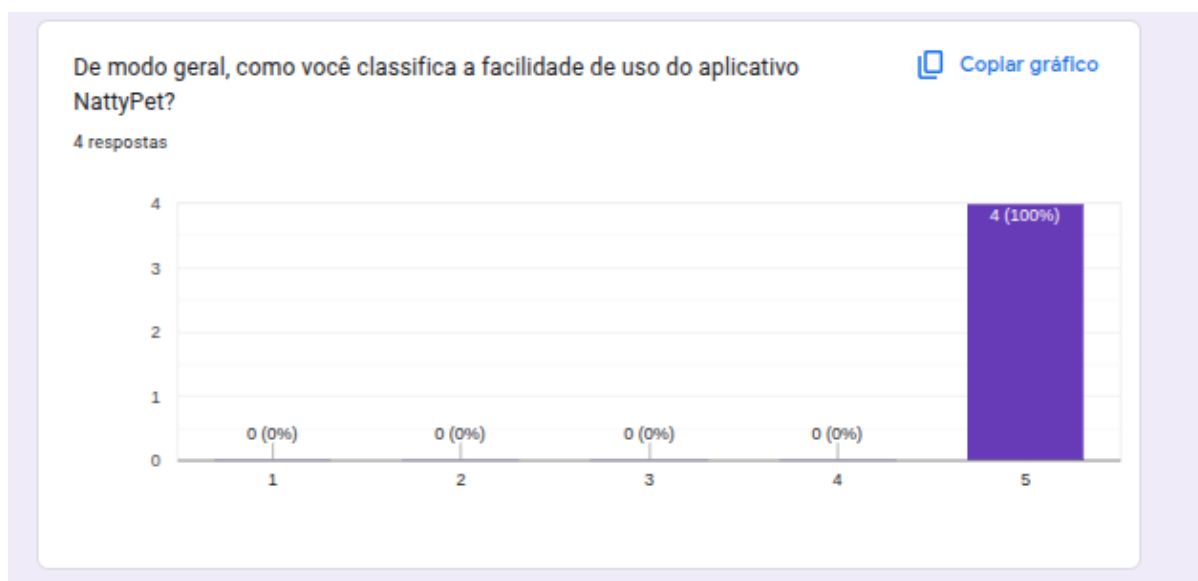
Fonte: Autores

Figura 31 – Avaliação da Experiência de Uso (Parte 2).



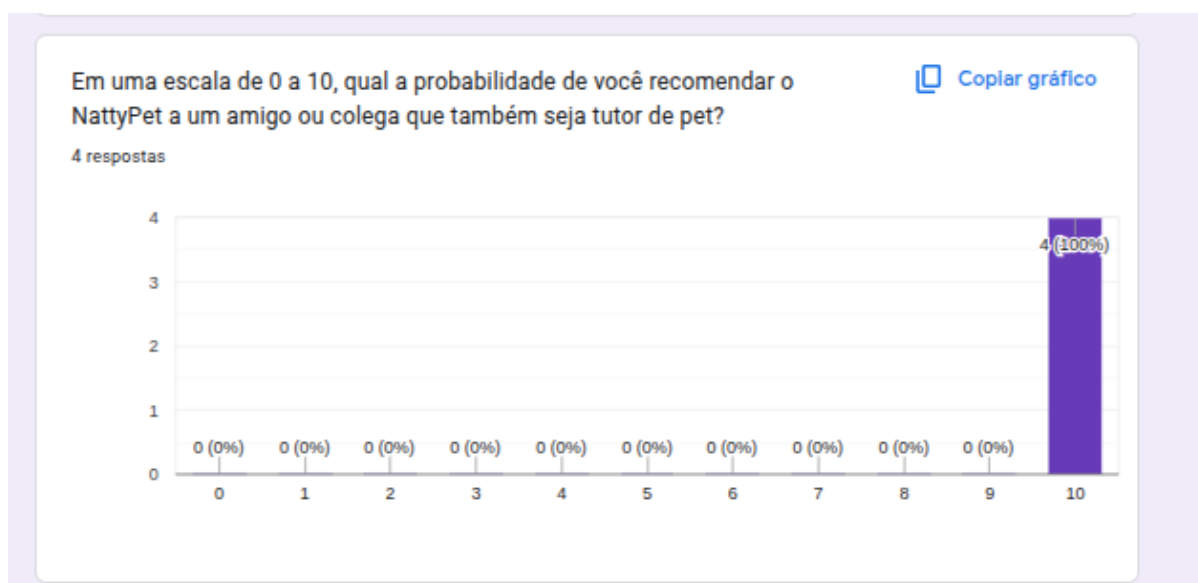
Fonte: Autores

Figura 32 – Classificação Geral da Facilidade de Uso do Aplicativo.



Fonte: Autores

Figura 33 – Probabilidade de Recomendação do Aplicativo.



Fonte: Autores

dessas melhorias, o ciclo de desenvolvimento e validação da ferramenta foi concluído.

6.4 Análise de Qualidade de Código e Testes

Além dos ciclos de teste de usabilidade, a qualidade do projeto foi monitorada por meio de análise estática de código e testes de cobertura.

6.4.1 Análise Estática de Código com SonarQube

Para garantir a qualidade interna do *software*, foi utilizada a ferramenta SonarQube, detalhada no [Capítulo 3](#). Integrada ao fluxo de desenvolvimento, a plataforma realizou análises contínuas do código-fonte para identificar *bugs*, vulnerabilidades e *code smells*. A Figura 34 exibe o painel do SonarQube para os três principais componentes do projeto: `nattypet-api`, `nattypet-balancing-service` e `nattypet-webapp`. Os resultados indicam um alto padrão de qualidade, com zero *bugs*, zero vulnerabilidades e uma classificação "A" em manutenibilidade para ambos os componentes. A análise do *SonarQube* permitiu que a equipe de desenvolvimento mantivesse a qualidade do código, corrigindo problemas de forma proativa e assegurando a robustez e a segurança da aplicação NattyPet.

Figura 34 – Resultados da Análise de Qualidade de Código no SonarQube.

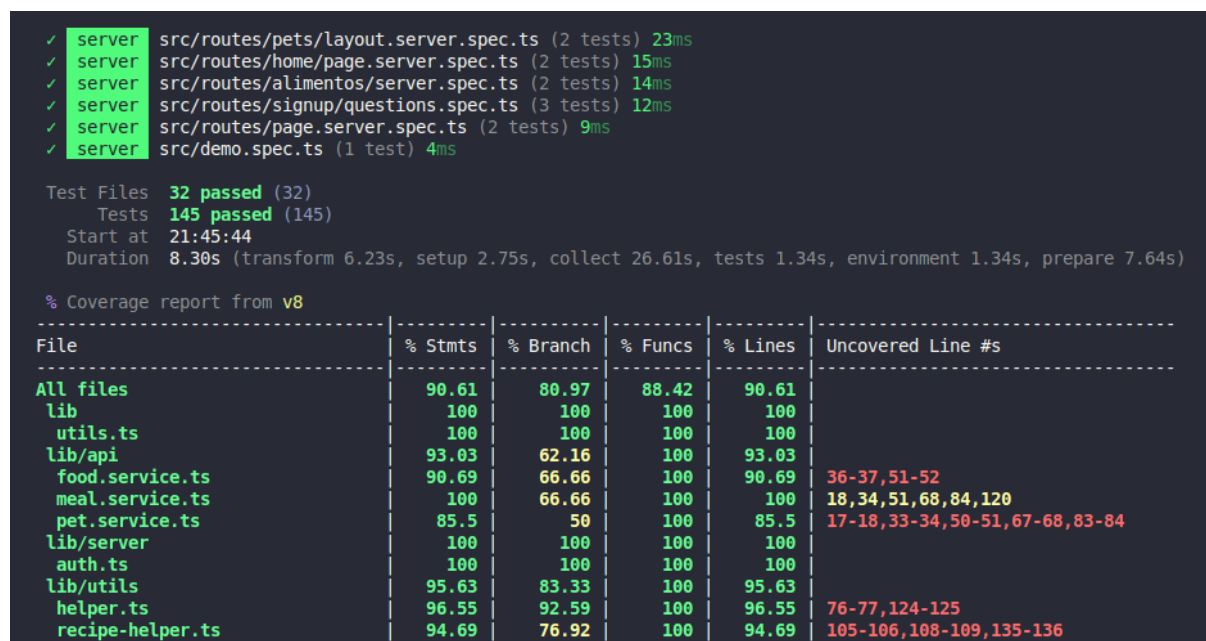


Fonte: Autores

6.4.2 Testes de Cobertura

Foram implementados testes unitários e de integração para o *backend* e o *frontend*, a fim de garantir o correto funcionamento das lógicas de negócio e dos componentes de interface. A cobertura de código foi utilizada como métrica para avaliar a abrangência dos testes. A Figura 35 apresenta o relatório de cobertura para o *frontend* (*nattypet-webapp*). Com 145 testes passando, alcançou-se uma cobertura geral de 90.61% das declarações (*statements*), demonstrando que a maior parte da base de código está validada por testes automatizados.

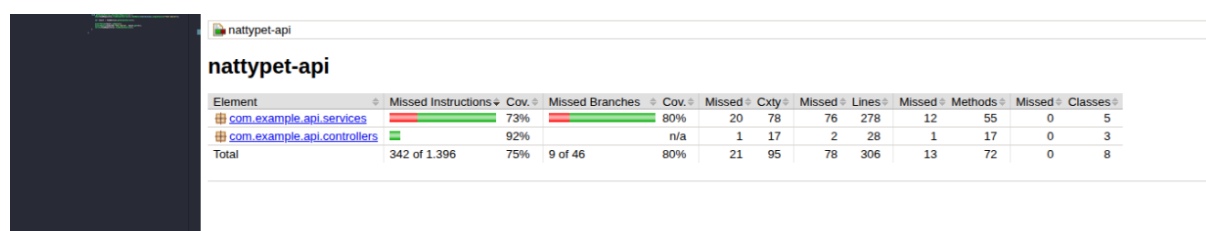
Figura 35 – Resultados dos Testes de Cobertura do *Frontend*.



Fonte: Autores

Para o *backend* (*nattypet-api*), o relatório de cobertura, exibido na Figura 36, mostra uma cobertura de 75% das instruções e 80% dos ramos (*branches*). Esses valores indicam uma boa cobertura dos fluxos lógicos e das funcionalidades da API, contribuindo para a estabilidade e confiabilidade do sistema.

Figura 36 – Resultados dos Testes de Cobertura do *Backend*.



Fonte: Autores

6.5 Resumo do Capítulo

Neste capítulo, foram apresentados os resultados obtidos ao longo de três ciclos de Pesquisa-Ação, que guiaram a evolução da ferramenta NattyPet desde o protótipo até a aplicação final. O [primeiro ciclo](#) validou a compreensibilidade dos fluxos, mas identificou pontos críticos de usabilidade. O [segundo ciclo](#) demonstrou o sucesso das melhorias implementadas, com uma otimização clara da experiência do usuário, o que permitiu aprovar o *design* para o desenvolvimento. O [terceiro ciclo](#) validou a versão funcional da aplicação, onde o *feedback* dos usuários confirmou a alta usabilidade e satisfação com o produto.

Adicionalmente, a [análise de qualidade](#), com o uso do *SonarQube* e a execução de testes de cobertura,

reforçou a robustez técnica do projeto. A metodologia de ciclos iterativos, combinando *feedback* de usuários e análise técnica, provou-se fundamental para construir uma solução sólida, intuitiva e alinhada aos objetivos propostos no [Capítulo 1](#).

7 Conclusão

Este capítulo tem por finalidade apresentar as considerações finais deste Trabalho de Conclusão de Curso. Inicialmente, na seção [Trabalho Realizado](#), é apresentado um resumo das atividades desempenhadas ao longo do projeto. Em seguida, a seção [Resultados Obtidos](#) discute o alcance dos objetivos e a resposta às questões de pesquisa que nortearam o desenvolvimento. Por fim, a seção [Trabalhos Futuros](#) aponta possíveis caminhos para a evolução e o aprimoramento da plataforma NattyPet.

7.1 Trabalho Realizado

O desenvolvimento do projeto NattyPet seguiu as etapas de planejamento e execução definidas no [Capítulo 4](#) deste trabalho. A Tabela 16 consolida o andamento das principais atividades, desde a concepção teórica até a implementação e análise da solução final, demonstrando a conclusão do ciclo de desenvolvimento proposto para este TCC.

Tabela 16 – Andamento Consolidado das Atividades do Projeto

Atividade	Andamento	Link Comprobatório
Definição do Tema e Escopo	Concluída	Capítulo 1
Levantamento Bibliográfico	Concluída	Capítulo 2
Estudo das Tecnologias de Suporte	Concluída	Capítulo 3
Definição da Metodologia	Concluída	Capítulo 4
Definição do Backlog do Produto	Concluída	Seção 5.1
Desenvolvimento do Protótipo de Alta Fidelidade	Concluída	Seção 5.2
Elaboração da Proposta (TCC1)	Concluída	Própria Monografia
Apresentação da Proposta (TCC1)	Concluída	Própria Monografia
Correções da Banca (TCC1)	Concluída	Própria Monografia
Desenvolvimento do Algoritmo de Balanceamento	Concluída	Seção 5.3.2
Desenvolvimento da Aplicação (MVP)	Concluída	Capítulo 5
Ciclos de Teste e Análise de Resultados	Concluída	Capítulo 6
Documentação Final (Monografia)	Concluída	Própria Monografia
Apresentação à Banca (TCC2)	A Fazer	-

Fonte: Autores

7.2 Resultados Obtidos

O desenvolvimento deste trabalho foi guiado por um objetivo geral e por questões de pesquisa específicas, cujos resultados são detalhados a seguir.

O **Objetivo Geral** de desenvolver um "*apoio computacional para tutores de pets que desejam prover uma Alimentação Natural para os seus pets, sendo essa balanceada e orientada de acordo com os principais guias de nutrição para animais domésticos*" foi plenamente alcançado. O resultado concreto é a plataforma NattyPet funcional, que oferece aos usuários as ferramentas necessárias para calcular as necessidades de seus animais e formular receitas que atendam a esses requisitos de forma segura e intuitiva.

As **Questões de Pesquisa e Desenvolvimento**, propostas no [Capítulo 1](#), foram respondidas da seguinte forma:

- **Como viabilizar um produto de software que auxilie tutores de pets a criarem receitas customizadas e nutricionalmente balanceadas?**

Resultado: A viabilização foi alcançada através do desenvolvimento de uma aplicação *full-stack*, composta por um *frontend* em SvelteKit e um microserviço de backend em Go. A solução integra o cálculo de necessidades nutricionais com um *solver* de otimização, provendo uma interface de usuário que abstrai a complexidade matemática para o tutor.

- **Como balancear receitas para pets de maneira automatizada e em atendimento aos principais requisitos nutricionais vigentes?**

Resultado: O problema foi modelado matematicamente como um problema de Programação Linear (PL), detalhado nos [Capítulos 2](#) e [5](#). A solução foi implementada utilizando a linguagem Go e o *solver* de otimização HiGHS, que calcula as quantidades ideais de cada ingrediente para satisfazer dezenas de restrições nutricionais simultaneamente, garantindo o balanceamento automático e preciso das receitas.

- **Como garantir que o produto tenha uma boa usabilidade para o público alvo, que não possui conhecimentos técnicos no domínio teórico da nutrição animal?**

Resultado: A usabilidade foi um pilar central do projeto, sendo garantida através de um processo de *design* centrado no usuário, que incluiu a criação de personas, a elaboração de um protótipo de alta fidelidade na ferramenta Figma e a condução de ciclos de testes de usabilidade com a ferramenta Maze, cujos resultados orientaram o desenvolvimento final da interface.

Os **Objetivos Específicos** também foram concluídos com sucesso:

1. **Levantamento do domínio de interesse do projeto:**

Resultado: O [Capítulo 2](#) consolida a pesquisa sobre Nutrição Animal, guias da FEDIAF, e os cálculos de necessidades energéticas.

2. Estudo quanto às particularidades dos cálculos e da solução algorítmica:

Resultado: A análise de abordagens e a escolha da Programação Linear como solução ótima foram documentadas nos Capítulos 2 e 5.

3. Estudo sobre *Progressive Web Apps*:

Resultado: A tecnologia foi estudada e sua aplicação justificada na arquitetura do projeto, descrita no Capítulo 5.

4. Desenvolvimento do produto de software:

Resultado: A plataforma NattyPet foi implementada, com seu *frontend*, *backend* e serviço de balanceamento, conforme detalhado no Capítulo 5.

5. Análise preliminar dos resultados obtidos:

Resultado: O Capítulo 6 apresenta os resultados dos ciclos de pesquisa-ação e dos testes de usabilidade, validando a eficácia e a aceitação da aplicação.

6. Documentação da solução:

Resultado: A presente monografia documenta todas as fases do projeto, desde a concepção teórica até a implementação e análise dos resultados.

7.3 Trabalhos Futuros

Com base no desenvolvimento do MVP (Produto Mínimo Viável) e nas funcionalidades mapeadas no *backlog* do produto, são apontadas diversas oportunidades para a evolução do NattyPet em trabalhos futuros:

- **Geração Automatizada de Receitas:** Implementar a funcionalidade da versão V2 do *backlog*, onde o sistema sugere uma receita completa de forma automática, selecionando os ingredientes de uma base de dados pré-definida para otimizar o balanceamento, sem que o usuário precise escolher cada item manualmente;
- **Ferramenta de Avaliação do Escore de Condição Corporal (ECC):** Desenvolver o módulo interativo previsto para a versão V3, que guiaria o tutor com imagens e perguntas para ajudá-lo a classificar o ECC do seu animal de forma mais precisa, refinando o cálculo das necessidades calóricas;
- **Inteligência de Compatibilidade de Receitas:** Adicionar a funcionalidade da versão V4, que, dada uma receita, indicaria para quais dos outros *pets* cadastrados pelo usuário aquela mesma receita seria nutricionalmente adequada, facilitando o manejo de múltiplos animais;

- **Expansão e Regionalização da Base de Ingredientes:** Atualmente, a plataforma utiliza a base de dados do USDA, que, embora robusta, carece de ingredientes específicos da culinária brasileira (e.g., cortes de carne, frutas e vegetais regionais). Uma evolução crucial seria a expansão e regionalização desta base, permitindo que usuários da plataforma cadastrarem novos ingredientes, enriquecendo a plataforma e aumentando significativamente sua aplicabilidade e precisão para o público brasileiro, e
- **Aprimoramentos de Usabilidade e Acesso:** Implementar funcionalidades da versão V5, como *login* social (Google, Apple) e aprimorar as capacidades *offline* da PWA para permitir a consulta de receitas salvas sem conexão com a internet.

Essas sugestões representam um caminho claro para transformar o NattyPet em uma plataforma ainda mais completa para a comunidade de tutores que praticam a Alimentação Natural.

Referências

- AAFCO. *Official Publication*. Champaign, IL: Association of American Feed Control Officials, 2013. Citado 2 vezes nas páginas 22 e 56.
- ABC, B. *DogFoodCalc*. 2023. Apple Store. Disponível em: <<https://apps.apple.com/us/app/dogfoodcalc/id1452557840>>. Citado na página 116.
- AGUIAR, F.; CAROLI, P. *Product Backlog Building: um guia prático para criação e refinamento de backlog para produtos de sucesso*. São Paulo: Editora Caroli, 2020. Citado na página 37.
- AHMAD, M. O.; MARKKULA, J.; OIVO, M. Kanban in Software Development: A Systematic Literature Review. In: *2013 39th Euromicro Conference on Software Engineering and Advanced Applications*. [S.l.]: IEEE, 2013. p. 9–16. Citado na página 57.
- ALFLEN, N. C.; PRADO, E. P. V. Técnicas de elicitação de requisitos no desenvolvimento de software: uma revisão sistemática da literatura. *AtoZ : Novas Práticas em Informação e Conhecimento*, 2021. Citado na página 37.
- Amazon Web Services. *What is Java? - Amazon Web Services (AWS)*. 2024. Acessado em: 26 de novembro de 2024. Disponível em: <<https://aws.amazon.com/what-is/java/>>. Citado na página 48.
- ANAND, G.; KODALI, R. Benchmarking the benchmarking models. *Benchmarking: An International Journal*, Emerald Group Publishing Limited, 2008. Citado na página 116.
- ANDERSON, D. J. *Kanban: Successful Evolutionary Change for Your Technology Business*. [S.l.]: Blue Hole Press, 2010. Citado na página 57.
- ANDERSSON, R. *Inter Font Family*. 2023. A widely used open-source typeface optimized for user interfaces. Disponível em: <<https://github.com/rsms/inter>>. Citado na página 74.
- ASSOCIAÇÃO BRASILEIRA DA INDÚSTRIA DE PRODUTOS PARA ANIMAIS DE ESTIMAÇÃO. *Manual Pet Food: Referência em qualidade e segurança dos alimentos: Manual pet food*. São Paulo, 2021. Disponível em: <http://abinpet.org.br/wp-content/uploads/2020/05/manual_pet_food_ed10_completo_digital.pdf>. Citado 2 vezes nas páginas 17 e 65.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR ISO/IEC 25010: Engenharia de software e de sistemas — requisitos de qualidade e avaliação de software (square) — modelos de qualidade de sistema e de software*. Rio de Janeiro, 2011. Esta norma cancela e substitui a ABNT NBR ISO/IEC 9126-1:2003. Citado na página 37.
- BJORNVAD, C. et al. Evaluation of a nine-point body condition scoring system in physically inactive pet cats. *American Journal of Veterinary Research*, v. 72, n. 4, p. 433–437, 2011. [S.l.]: Adapted. Citado na página 30.

- BROWN, S. *Animal Diet Formulator*. 2024. Disponível em: <<https://animaldietformulator.com/>>. Citado na página 116.
- BUFF, P. R. et al. Natural pet food: A review of natural diets and their impact on canine and feline physiology. *Journal of Animal Science*, v. 92, p. 3781–3791, 2014. Citado na página 22.
- BURKHOLDER, W. Use of body condition scores in clinical assessment of the provision of optimal nutrition. *Journal of the American Veterinary Medical Association*, v. 217, n. 5, p. 650–654, 2000. [S.l.]: Adapted. Citado na página 27.
- Byte Magazine. *The bible of all fundamental algorithms and the work that taught many of today's software developers most of what they know about computer programming*. 1995. [S.l.]: Adapted. Citado na página 42.
- CARCIOFI, A. C. *Apostila de Nutrição de Cães e Gatos*. 2019. Disponível em: <<https://www.fcav.unesp.br/Home/departamentos/clinicacv/AULUSCAVALIERICARCIOFI/apostila-nutricao-caes-e-gatos-2019.pdf>>. Citado 3 vezes nas páginas 27, 30 e 33.
- CAROLI, P. *Lean Inception: Como alinhar pessoas e construir o produto certo*. 1. ed.. ed. [S.l.]: Caroli, 2018. Citado na página 72.
- COLLIARD, L. et al. Risk factors for obesity in dogs in france. *Journal of Nutrition*, v. 136, n. 7, p. 1951S–1954S, 2006. Citado na página 27.
- DOBENECKER, B.; KÖSTLIN, R.; MATIS, U. Mal-and overnutrition in puppies with or without clinical disorders of skeletal development. *Journal of Animal Physiology and Animal Nutrition*, v. 80, n. 1-5, p. 76–81, 1998. [S.l.]: Adapted. Citado na página 33.
- Docker, Inc. *Docker Overview - Get Started with Docker*. 2024. Acessado em: 26 de novembro de 2024. Disponível em: <<https://docs.docker.com/get-started/docker-overview/>>. Citado na página 49.
- Eraser, Inc. *Eraser - Remove dados com segurança e eficácia*. 2024. Acessado em: 26 de novembro de 2024. Disponível em: <<https://www.eraser.io/?r=0>>. Citado na página 50.
- FEDIAF. *Code of Good Labelling Practice for Pet Food*. 2018. <https://www.fediaf.org/images/FEDIAF_labeling_code_2019_onlineOctober2019.pdf>. Citado na página 22.
- FEDIAF. *Nutritional Guidelines For Complete and Complementary Pet Food for Cats and Dogs*. 2024. <https://europeanpetfood.org/wp-content/uploads/2024/09/FEDIAF-Nutritional-Guidelines_2024.pdf>. Citado 14 vezes nas páginas 21, 22, 23, 25, 26, 27, 28, 30, 31, 32, 33, 34, 35 e 56.
- FEEDREAL Calculator. 2023. Disponível em: <<https://www.feedreal.com/calculator>>. Citado na página 116.
- FERREIRA, B. et al. Technique for representing requirements using personas: a controlled experiment. *IET Software*, v. 12, n. 3, p. 280–290, 2018. Disponível em: <<https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-sen.2017.0313>>. Citado na página 66.

Figma, Inc. *Figma Design - A melhor ferramenta de design para equipes*. 2024. Acessado em: 26 de novembro de 2024. Disponível em: <<https://www.figma.com/pt-br/design/>>. Citado na página 49.

FINKE, M. Energy requirements of adult female beagles. *Journal of Nutrition*, v. 124, n. suppl_12, p. 2604S–2608S, 1994. [S.l.]: Adapted. Citado na página 33.

FRANÇA, J. *Alimentos convencionais versus naturais para cães adultos*. 93 p. Tese (Tese (Doutorado em Zootecnia)) — Universidade Federal de Lavras, Lavras, MG, 2009. Citado na página 17.

GERHARDT, T. E.; SILVEIRA, D. T. *Métodos de Pesquisa*. [S.l.]: Plageder, 2009. Citado 4 vezes nas páginas 53, 54, 55 e 59.

GERMAN, A. et al. A simple, reliable tool for owners to assess the body condition of their dog or cat. *Journal of Nutrition*, v. 136, n. 7, p. 2031S–2033S, 2006. [S.l.]: Adapted. Citado na página 31.

GIL, A. C. et al. *Como Elaborar Projetos de Pesquisa*. São Paulo: Atlas, 2002. v. 4. Citado na página 55.

Git Community. *Git - Distributed Version Control System*. 2024. Acessado em: 26 de novembro de 2024. Disponível em: <<https://git-scm.com/>>. Citado na página 46.

GitHub Docs. *Sobre o GitHub e o Git - Documentação do GitHub*. 2024. Acessado em: 26 de novembro de 2024. Disponível em: <<https://docs.github.com/pt/get-started/start-your-journey/about-github-and-git>>. Citado na página 47.

Go Developers. *Go Documentation - An Overview of Go Programming Language*. 2024. Acessado em: 26 de novembro de 2024. Disponível em: <<https://go.dev/doc/>>. Citado na página 48.

Google, Inc. *Google Forms - Crie e analise pesquisas gratuitamente*. 2024. Disponível em: <<https://www.google.com/forms/about/>>. Citado na página 50.

HEINZE, C. R.; GOMEZ, F. C.; FREEMAN, L. M. Assessment of commercial diets and recipes for home-prepared diets recommended for dogs with cancer. *Journal of the American Veterinary Medical Association*, v. 241, n. 11, p. 1453–1460, 2012. Citado na página 18.

Hetzner, Inc. *Hetzner*. 2025. Disponível em: <<https://www.hetzner.com/cloud>>. Citado na página 51.

HILLIER, F. S.; LIEBERMAN, G. J. *Introduction to Operations Research*. 10th. ed. New York, NY: McGraw-Hill Education, 2015. Citado na página 42.

HUDAIB, A. et al. Requirements prioritization techniques comparison. *Computer Science Department, the World Islamic Science and Education University, Amman, Jordan*, Computer Science Department, the World Islamic Science and Education University, Amman, Jordan, 2018. Disponível em: <<https://doi.org/10.5539/mas.v12n2p62>>. Citado na página 37.

- KARR-LILIENTHAL, L. K. *Pet Foods: How to Read Labels*. 2009. Accessed via University of Nebraska–Lincoln Extension. Disponível em: <<https://extensionpubs.unl.edu/publication/g1960/na/html/view#target4>>. Citado na página 89.
- KEALY, R. et al. Effects of limited food consumption on the incidence of hip dysplasia in growing dogs. *Journal of Veterinary Medicine Series A*, v. 201, p. 857–857, 1992. [S.l.]: Adapted. Citado na página 30.
- KOERICH, M. S. et al. Pesquisa-ação: ferramenta metodológica para a pesquisa qualitativa. *Revista Eletrônica de Enfermagem*, v. 11, n. 3, 2009. Citado na página 59.
- LAFLAMME, D. P. Nutrition for aging cats and dogs and the importance of body condition. *Veterinary Clinics: Small Animal Practice*, Elsevier, v. 35, n. 3, p. 713–742, 2005. Disponível em: <https://pawsoflife-org.k9handleracademy.com/Library/Health/Laflamme_2005.pdf>. Citado na página 89.
- LARSEN, J. A. et al. Evaluation of recipes for home-prepared diets for dogs and cats with chronic kidney disease. *Journal of the American Veterinary Medical Association*, v. 240, n. 5, p. 532–538, 2012. Citado na página 18.
- LUND, E. M. Prevalence and risk factors for obesity in adult cats from private us veterinary practices. *International Journal of Applied Research in Veterinary Medicine*, v. 3, p. 88–96, 2005. Citado na página 27.
- LUTOWSKI, R. *Software requirements: encapsulation, quality, and reuse*. [S.l.]: Auerbach Publications, 2005. [S.l.]: Adapted. Citado na página 36.
- MACHADO, F. N. R. *Análise e Gestão de requisitos de software: onde nascem os sistemas*. [S.l.]: Editora Érica, 2011. [S.l.]: Adapted. Citado na página 36.
- Maze, Inc. *Maze - Teste de usabilidade e prototipagem de alta fidelidade*. 2024. Disponível em: <<https://maze.design/>>. Citado na página 50.
- MAZON, M.; MOURA, W. Cachorros e humanos: mercado de rações pet em perspectiva sociológica. *Civitas-Revista de Ciências Sociais*, Pontifícia Universidade Católica do Rio Grande do Sul, v. 17, p. 138–158, 2017. ISSN 1519-6089. Citado 2 vezes nas páginas 17 e 65.
- MDN Web Docs. *Service Worker API*. 2024. Acessado em: 26 de novembro de 2024. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API>. Citado na página 42.
- MDN Web Docs. *What is a progressive web app?* 2024. Acessado em: 26 de novembro de 2024. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Guides/What_is_a_progressive_web_app>. Citado 3 vezes nas páginas 39, 40 e 41.
- MEYER, H.; ZENTEK, J. Energie und nährstoffe-stoffwechsel und bedarf. In: *Ernährung des Hundes*. 5th. ed. [S.l.]: P. Parey Verlag, 2005. p. 49–96. [S.l.]: Adapted. Citado 2 vezes nas páginas 31 e 33.

Microsoft. *Visual Studio Code - Code Editing. Redefined*. 2024. Acessado em: 26 de novembro de 2024. Disponível em: <<https://code.visualstudio.com>>. Citado na página 46.

MIRANDA, E. Moscow rules: A quantitative exposé. In: *International Conference on Agile Software Development*. [S.l.]: Springer, 2022. p. 19–34. Citado na página 72.

MORAES, L. F. *Funcional Pet*. 2024. Disponível em: <<https://funcionalpet.com.br/>>. Citado na página 116.

National Research Council (NRC). *Nutrient Requirements of Dogs and Cats*. Washington, DC: National Academies Press, 2006. [S.l.]: Adapted. Citado 2 vezes nas páginas 22 e 30.

NORMAN, G. Likert scales, levels of measurement and the “laws” of statistics. *Advances in Health Sciences Education*, 2010. Citado 2 vezes nas páginas 39 e 60.

NUTROVET. *NutroVET*. 2024. Disponível em: <<https://nutrovet.com.br/>>. Citado na página 116.

OLIVEIRA, L. D.; FISCHER, M. M. *Alimentação Natural para Cães*. São Paulo: [s.n.], 2022. Citado 7 vezes nas páginas 18, 19, 21, 22, 27, 56 e 65.

OLIVEIRA, M. C. C. et al. Evaluation of the owner’s perception in the use of homemade diets for the nutritional management of dogs. *Journal of Nutritional Science*, v. 3, p. e23, 1–5, 2014. Citado na página 18.

PEDRINELLI, V. et al. Concentrations of macronutrients, minerals and heavy metals in home-prepared diets for adult dogs and cats. *Scientific Reports*, Nature Publishing Group, v. 9, p. 13058, 2019. Citado na página 18.

PostgreSQL Global Development Group. *About PostgreSQL - Open Source Database*. 2024. Acessado em: 26 de novembro de 2024. Disponível em: <<https://www.postgresql.org/about/>>. Citado na página 49.

REGULATION of the European parliament and of the council on the placing on the market and use of feed. *Regulation (EC) No 767*, European Union, v. 2009, 2009. Adapted. Disponível em: <<https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32009R0767>>. Citado na página 22.

RILLICH, J. *The BARF App*. 2024. Apple Store. Disponível em: <<https://apps.apple.com/us/app/barf-raw-feeding/id1530130136>>. Citado na página 116.

RUDD, J.; STERN, K.; ISENSEE, S. Low vs. high-fidelity prototyping debate. *Interactions*, ACM New York, NY, USA, v. 3, n. 1, p. 76–85, 1996. Citado na página 76.

SANTOS, G. A. et al. Uso de programação linear para formulação de rações de custo mínimo para frangos de corte. *Revista de Ciências Agroveterinárias*, v. 17, n. 3, p. 394–401, 2018. Citado na página 43.

SAURO, J.; LEWIS, J. R. *Quantifying the User Experience: Practical Statistics for User Research*. [S.l.: s.n.]: Morgan Kaufmann, 2012. Citado 2 vezes nas páginas 38 e 39.

SCHWABER, K.; BEEDLE, M. *Agile Software Development with Scrum*. Upper Saddle River, NJ: Prentice Hall, 2002. v. 1. Citado na página 57.

SEDANO, T.; RALPH, P.; PÉRAIRE, C. The product backlog. In: *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. [S.l.: s.n.], 2019. p. 200–211. Citado na página 37.

SILVA, L. da et al. MERCADO DA ALIMENTAÇÃO NATURAL PET NO BRASIL: Um estudo sobre a oferta e demanda por produtos naturais no mercado brasileiro. In: *Anais do Fórum Internacional On-line de Empreendedorismo e Inovação no Agro 2021*. Piracicaba, SP, Brasil: Galoá, 2021. Disponível em: <<https://proceedings.science/finovagro/finovagro-2021/trabalhos/mercado-da-alimentacao-natural-pet-no-brasi-um-estudo-sobre-a-oferta-e-demanda-p?lang=pt-br>>. Citado 4 vezes nas páginas 17, 21, 56 e 65.

SILVA, R. N. d. Avaliação das medidas morfométricas na determinação da condição corporal de gatos domésticos de diferentes escores de condição corporal. *Universidade Federal da Paraíba*, 2021. Disponível em: <<https://repositorio.ufpb.br/jspui/handle/123456789/21683>>. Citado 2 vezes nas páginas 28 e 30.

SLATER, M. et al. Diet and exercise patterns in pet dogs. *Journal of the American Veterinary Medical Association*, v. 207, n. 2, p. 186–190, 1995. [S.l.]: Adapted. Citado na página 31.

Sociedade Brasileira de Zootecnia. Alimentação natural para cães e gatos. *Revista Brasileira de Zootecnia*, v. 39, 2010. Suplemento Especial. Disponível em: <<https://doi.org/10.1590/S1516-35982010001300007>>. Citado 3 vezes nas páginas 17, 56 e 65.

SOMMERVILLE, I. *Software Engineering*. Pearson, 2016. (Always Learning). ISBN 9780133943030. Disponível em: <<https://books.google.com.br/books?id=tW4VngEACAAJ>>. Citado 2 vezes nas páginas 36 e 57.

SonarSource. *SonarQube - Code Quality and Security*. 2024. Disponível em: <<https://www.sonarsource.com/products/sonarqube/>>. Citado na página 47.

Spring Team. *Why Spring? - Spring Framework*. 2024. Acessado em: 26 de novembro de 2024. Disponível em: <<https://spring.io/why-spring>>. Citado na página 48.

SUPERCAC. *SUPERCAC PET*. 2024. Disponível em: <<https://www.agropecuaria.inf.br/supercac-pet>>. Citado na página 116.

SUTHERLAND, J. *Jeff Sutherland's Scrum Handbook*. [S.l.]: [S.l.: s.n.], 2010. Citado na página 72.

Svelte Team. *Svelte Documentation - Overview*. 2024. Acessado em: 26 de novembro de 2024. Disponível em: <<https://svelte.dev/docs/svelte/overview>>. Citado na página 47.

Svelte Team. *SvelteKit Documentation - Introduction*. 2024. Acessado em: 26 de novembro de 2024. Disponível em: <<https://svelte.dev/docs/kit/introduction>>. Citado na página 48.

TEAM, G. M. D. *Material Design 3 (Material You)*. 2023. A modern design system focused on adaptability and dynamic theming. Disponível em: <<https://m3.material.io/>>. Citado na página 74.

WEB.DEV. What are progressive web apps (pwAs)? In: *Progressive Web Apps*. [S.l.]: Google Developers, 2024. Disponível em: <<https://web.dev/articles/what-are-pwas?hl=pt-br>>, acessado em 21 de novembro de 2024. Citado 2 vezes nas páginas 39 e 40.

Apêndices

APÊNDICE A – Benchmarking dos Recursos das Aplicações

O *benchmarking* pode ser descrito como uma análise contínua de estratégias, funções, produtos ou serviços, comparados as melhores organizações de cunho similar. Estas informações são coletadas por meio de pesquisas, visitas, entrevistas, questionários, entre outros métodos, e são utilizadas para realizar o auto-aperfeiçoamento. Com isso, o *benchmarking*, é uma técnica relevante para a implementação de mudanças para escalar ou exceder esses padrões ([ANAND; KODALI, 2008](#)).

Com isso, foi desenvolvido um *benchmarking* com o objetivo de analisar e comparar os recursos das principais aplicações de cunho similar ao da natureza do proposto projeto. Para isto, foram analisados os recursos das aplicações: The BARF App na versão 1.7.6 ([RILLICH, 2024](#)), DogFoodCalc na versão 1.19.0 ([ABC, 2023](#)), FeedReal Calculator ([FEEDREAL... , 2023](#)), Animal Diet Formulator ([BROWN, 2024](#)), NutroVET ([NUTRO-VET, 2024](#)), SUPERCRAC PET na versão 1.3 ([SUPERCRAC, 2024](#)) e Funcional Pet ([MORAES, 2024](#)). De acordo com a Figura 37, procurou-se analisar os seguintes recursos e demonstrar os diferenciais do projeto proposto:

- Restrição de dispositivo/Sistema Operacional: software amplamente disponível;
- Destinado para tutores: software que tem como público-alvo tutores, que não possuem conhecimento técnico no domínio de nutrição natural;
- Geração automatizada de receitas: capacidade do software gerar receitas personalizadas de maneira automática;
- Balanceamento de receitas com ingredientes pré selecionados: redistribuição das quantidades de ingredientes a fim de balancear os nutrientes a serem consumidos com os requisitos nutricionais da dieta proposta;
- Acesso a bases de dados de alimentos: acesso a base de dados de alimentos disponibilizados pelos órgãos regulamentadores;
- Permite armazenamento de receitas: capacidade do software de salvar receitas desejadas;
- Fornece teste de Escore de Condição Corporal (ECC): capacidade do software oferecer testes de ECC para uma melhor análise da receita a ser gerada. Tal Conceito está melhor explicado e contextualizado no [Capítulo 2 - Referencial Teórico](#);

- Destinado para uso com cães: capacidade do software de lidar com requisitos de cães;
- Destinado para uso com gatos: capacidade do software de lidar com requisitos de gatos;
- Cálculo de Requisitos Nutricionais: capacidade de calcular os requisitos nutricionais diários individuais de cada *pet*, e
- Permite adicionar ingredientes: capacidade do software de receber novos ingredientes a serem utilizados em novas receitas.

Figura 37 – Benchmarking com Aplicações Existentes

		RECURSOS											
		Amplamente disponível	Destinado para tutores	Utiliza guias nutricionais	Geração de receitas	Balancete de receitas	Utiliza BD de alimentos	Salva receitas	Fornecer teste ECC	Suporte para cães	Suporte para gatos	Calcula reqs. nutricionais	Permite adicionar ingredientes
APLICAÇÕES	NattyPet (Aplicação Proposta)												
	The BARF App												
	Dog Food Calc												
	Feed Real Calculator												
	Animal Diet Formulator												
	Nutro VET												
	SuperCrac Pet												
	Funcional Pet												

Legenda

 Possui recurso

 Não possui recurso

Fonte: Autores

APÊNDICE B – Termo de Consentimento Livre e Esclarecido

A seguir tem-se o *template* de Termo de Consentimento Livre e Esclarecido que será disponibilizado para os usuários durante a fase de Validação da Usabilidade do Protótipo de Alta Fidelidade. Um termo similar também será disponibilizado para os profissionais da área de Nutrição Animal que forem consultados para validação do funcionamento do sistema NattyPet.

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Título do Projeto de pesquisa: **NattyPet: Um WebApp para Geração de Receitas Balanceadas Nutricionalmente para Pets**

Pesquisadores Responsáveis: **Lucas Felipe Soares e João Victor Teixeira Batista**

Nome do participante:

Data de nascimento:

Você está sendo convidado (a) para ser participante do Projeto de pesquisa intitulado “**NattyPet: Um WebApp para Geração de Receitas Balanceadas Nutricionalmente para Pets**” de responsabilidade dos pesquisadores **Lucas Felipe Soares e João Victor Teixeira Batista**

Leia cuidadosamente o que se segue e pergunte sobre qualquer dúvida que você tiver. Caso se sinta esclarecido (a) sobre as informações que estão neste Termo e aceite fazer parte do estudo, peço que assine ao final deste documento, em duas vias, sendo uma via sua e a outra da pesquisadora responsável pela pesquisa. Saiba que você tem total direito de não querer participar.

1. O trabalho tem por objetivo realizar um levantamento de dados sobre usabilidade e acessibilidade do aplicativo NattyPet.
2. A participação nesta pesquisa consistirá na utilização do aplicativo NattyPet com o objetivo de testar algumas interações específicas do aplicativo. A partir do teste de usabilidade será passado um formulário para a realização da avaliação de aspectos relacionados a usabilidade e acessibilidade do aplicativo.
3. O teste de usabilidade será gravado a fim de documentar o tempo de realização de cada atividade solicitada e manter o rastro das percepções e comentários emitidos pelos participantes, além de servir de garantia de que os dados posteriormente relatados na pesquisa não serão alterados indevidamente. As gravações serão acessadas somente pela pesquisadora.
4. Os benefícios com a participação nesta pesquisa serão o auxílio na evolução e melhoria de um aplicativo gratuito que tem como objetivo ajudar a população mais carente.
5. Os participantes não terão nenhuma despesa ao participar da pesquisa e poderão retirar sua concordância na continuidade da pesquisa a qualquer momento.
6. Não há nenhum valor econômico a receber ou a pagar aos voluntários pela participação, uma vez que a participação é voluntária, e o participante pode deixar de contribuir no momento que desejar, bastando comunicar aos pesquisadores.

Rubrica do pesquisador
dora: _____. Rubrica do participante: _____.



7. O nome dos participantes será mantido em sigilo, assegurando assim a sua privacidade, e se desejarem terão livre acesso a todas as informações e esclarecimentos adicionais sobre o estudo e suas consequências, enfim, tudo o que queiram saber antes, durante e depois da sua participação.

8. Os dados coletados serão utilizados única e exclusivamente para fins desta pesquisa, e os resultados poderão ser publicados.

Qualquer dúvida, pedimos a gentileza de entrar em contato com Lucas Felipe Soares, pesquisador responsável pela pesquisa, telefone: (61) 999956808, e-mail: *lucasfs1007@gmail.com*

Eu, _____, RG nº _____ declaro ter sido informado e concordo em ser participante do Projeto de pesquisa acima descrito.

____ de _____ de 2025.

Assinatura do participante

Nome e assinatura do responsável por obter o consentimento

Rubrica do pesquisador
dora: _____. Rubrica do participante: _____.