

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

RAFAEL SEDOR OLIVEIRA DEDA

**UTFPETS: UMA APLICAÇÃO WEB PARA GERENCIAMENTO DE REFEIÇÕES
E CONTROLE NUTRICIONAL DE PETS**

GUARAPUAVA

2025

RAFAEL SEDOR OLIVEIRA DEDA

**UTFPETS: UMA APLICAÇÃO WEB PARA GERENCIAMENTO DE REFEIÇÕES
E CONTROLE NUTRICIONAL DE PETS**

**UTFPets: A Web Application for Meal Management and Nutritional Control of
Pets**

Projeto de Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em Tecnologia em Sistemas para Internet do Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Roni Fabio Banaszewski

GUARAPUAVA

2025



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

RESUMO

A alimentação adequada de animais de estimação é essencial para garantir saúde e bem-estar, especialmente em um contexto de crescimento do número de animais domésticos. Entretanto, diversos tutores enfrentam dificuldades ao gerenciar uma dieta equilibrada, em razão de rotinas agitadas e de lacunas no conhecimento técnico, o que pode acarretar problemas de saúde como obesidade e outras doenças crônicas. O presente trabalho descreve o desenvolvimento de um aplicativo denominado UTFPets, que permite registrar dados sobre a saúde e alimentação dos animais, receber lembretes de atividades e compartilhar informações com outros cuidadores ou estabelecimentos especializados. Entre os principais recursos, destacam-se o cadastro de usuário e do perfil do animal, lembretes para atividades essenciais, controle alimentar e seções educativas sobre segurança alimentar e nutrição. Além disso, a proposta inclui o envio de notificações para garantir a entrega eficaz de lembretes e informações. Prevê-se que o UTFPets simplifique o controle nutricional e promova a saúde dos animais, oferecendo aos tutores mais conscientização sobre os cuidados necessários e melhorando a qualidade de vida dos pets e a rotina dos cuidadores.

Palavras-chave: gestão de pets; nutrição animal; controle alimentar; aplicativo para pets; tecnologia em saúde animal.

ABSTRACT

Proper nutrition for pets is essential to ensure their health and well-being, especially in the context of an increasing number of domestic animals. However, many pet owners face challenges in managing a balanced diet due to busy routines and a lack of technical knowledge, resulting in health issues such as obesity and other chronic diseases. This project aims to develop an application called UTFPets. UTFPets will allow pet owners to record important health and nutrition data about their pets, receive activity reminders, and share information with other caregivers or establishments, such as pet hotels. Among its main features, the app will include user registration, animal profiles, reminders for essential activities, dietary control, and educational information about food safety and nutrition. The proposal also involves the use of technologies like Firebase Cloud Messaging (FCM), enabling push notifications to ensure that users receive relevant reminders and information conveniently. UTFPets is expected to simplify nutritional management and promote pet health by providing users with greater control and awareness of pet care, thereby improving the quality of life for pets and the daily routines of their caregivers.

Keywords: pet management; animal nutrition; dietary control; pet app; health technology for pets.

LISTA DE FIGURAS

Figura 1 – Página inicial do site do aplicativo Petzillas	13
Figura 2 – Página inicial do site do aplicativo FuncionalPet	14
Figura 3 – Página inicial do site do aplicativo PetDesk	14
Figura 4 – Página inicial do site do aplicativo Pet Diet Designer	14
Figura 5 – Diagrama do Banco de Dados do UTFPets (Fonte: Autoria Própria) . . .	23
Figura 6 – Wireframe de Cadastro de Pets	24
Figura 7 – Estrutura da tabela <code>users</code> (Fonte: Autoria Própria)	32
Figura 8 – Estrutura da tabela <code>pets</code> (Fonte: Autoria Própria)	33
Figura 9 – Estrutura da tabela <code>feedings</code> (Fonte: Autoria Própria)	34
Figura 10 – Estrutura da tabela <code>reminders</code> (Fonte: Autoria Própria)	35
Figura 11 – Estrutura da tabela <code>shared_pets</code> (Fonte: Autoria Própria)	36
Figura 12 – Estrutura da tabela <code>notifications</code> (Fonte: Autoria Própria)	37

LISTA DE ABREVIATURAS E SIGLAS

Siglas

API	Application Programming Interface
E2E	End-to-End
IDE	Integrated Development Environment
MVP	Minimum Viable Product
NPM	Node Package Manager
SGBD	Sistema Gerenciador de Banco de Dados
SPA	Single Page Application
UI	User Interface
UX	User Experience

SUMÁRIO

1	INTRODUÇÃO	7
1.1	Objetivo Geral	7
1.2	Objetivos Específicos	8
1.3	Estrutura do Trabalho	8
2	REFERENCIAL TEÓRICO	10
2.1	Nutrição e Saúde de Animais de Estimação	10
2.1.1	Obesidade e Doenças Relacionadas	10
2.1.2	Dieta Equilibrada e Fatores de Risco	10
2.2	Prevenção de Doenças e Rotina de Cuidados	11
2.2.1	Vacinação e Check-ups Veterinários	11
2.2.2	Colaboração entre Múltiplos Cuidadores	11
2.3	Aplicativos de Gerenciamento Alimentar e Sanitário	11
2.3.1	Benefícios e Desafios	11
2.4	Considerações	12
3	TRABALHOS RELACIONADOS	13
3.1	Aplicativos para Organização de Cuidados, Nutrição e Saúde de Pets . .	13
3.1.1	Petzillas	13
3.1.2	FuncionalPet	13
3.1.3	PetDesk	14
3.1.4	Pet Diet Designer	14
3.2	Considerações	15
4	MATERIAIS E MÉTODOS	16
4.1	Materiais	16
4.1.1	Escalabilidade em Aplicativos Web	16
4.1.2	Ambiente de Desenvolvimento	17
4.1.3	Ferramentas Auxiliares	17
4.2	Métodos	18
4.2.1	Metodologia Ágil: Scrum e Kanban	18
4.2.2	Estrutura das Sprints	18
4.2.3	Histórias de Usuário	19

4.2.4	Testes Automatizados	19
4.2.5	Ambientes de Desenvolvimento e Testes	20
4.3	Considerações	20
5	ANÁLISE E PROJETO DO SISTEMA	21
5.1	Implementação do Método MoSCoW	21
5.2	Estrutura do Aplicativo	21
5.3	Funcionalidades Prioritárias (Must Have)	22
5.3.1	Tutor	22
5.3.2	Cuidador Secundário	22
5.3.3	Administrador	23
5.4	Modelagem do Banco de Dados	23
5.5	Wireframes das Telas	24
5.6	Considerações	25
6	CONCLUSÃO	26
	REFERÊNCIAS	27
	APÊNDICE A FUNCIONALIDADES FUTURAS E EXPANSÃO DO UTF-	
	PETS	29
	A.1 Funcionalidades Importantes (Should Have)	29
	A.2 Funcionalidades Desejáveis (Could Have)	29
	A.3 Funcionalidades Futuras (Won't Have)	30
	A.4 Conclusão	30
	APÊNDICE B DETALHES DAS TABELAS DO BANCO DE DADOS . . .	32
	B.1 Tabela users	32
	B.2 Tabela pets	33
	B.3 Tabela feedings	34
	B.4 Tabela reminders	35
	B.5 Tabela shared_pets	36
	B.6 Tabela notifications	37

1 INTRODUÇÃO

A alimentação adequada é essencial para a saúde e o bem-estar de animais de estimação, especialmente em um momento em que cada vez mais pessoas estão acolhendo pets em suas casas e os tratando como parte da família. Uma nutrição balanceada não apenas previne doenças crônicas, como obesidade e diabetes, mas também contribui para que os animais tenham uma vida mais longa e saudável (FASCETTI; DELANEY, 2012). Entretanto, a obesidade é uma das condições nutricionais mais comuns entre cães e gatos, resultando em complicações que podem incluir problemas metabólicos e aumento nos custos com cuidados veterinários (LAFLAMME; FLAMMER; HANSEN, 2008; GERMAN, 2022).

Apesar da boa intenção, diversos tutores encontram dificuldades para oferecer uma dieta adequada aos pets. A rotina corrida e a falta de conhecimento técnico sobre nutrição animal tornam o manejo alimentar desafiador. Afinal, cuidar da nutrição de um pet vai muito além de escolher a ração: é necessário considerar a quantidade de nutrientes, a frequência das refeições e as necessidades específicas de cada animal, que podem variar segundo idade, porte, raça e nível de atividade física.

Além disso, a ausência de uma abordagem colaborativa entre os membros da família pode tornar a rotina de cuidados ainda mais complexa. Em lares onde mais de uma pessoa é responsável pela alimentação do animal, a falta de comunicação pode levar a uma alimentação descontrolada ou ao excesso de porções. Essas falhas acabam comprometendo a saúde do pet e evidenciam que, apesar da dedicação, ainda há obstáculos a serem superados (KOGAN; HELLYER, 2023).

Diante desse cenário, a tecnologia surge como uma aliada para facilitar o dia a dia dos tutores. Um aplicativo que centralize e organize as informações sobre a nutrição e os cuidados pode ser uma ferramenta útil para gerenciar a rotina alimentar, assegurar que as refeições sejam balanceadas e oferecer lembretes para atividades essenciais. A proposta deste trabalho é desenvolver um aplicativo que permita registrar dados relevantes sobre a saúde dos animais, receber lembretes de atividades rotineiras e compartilhar informações com outros cuidadores ou estabelecimentos, como hotéis para pets. Dessa forma, espera-se que a solução auxilie na organização da rotina dos pets em relação à alimentação e, assim, contribua para melhorar sua saúde e bem-estar.

1.1 Objetivo Geral

Desenvolver um aplicativo móvel que centralize e facilite a gestão dos cuidados e informações de animais de estimação, possibilitando o registro de dados importantes, o envio de lembretes para atividades rotineiras e o compartilhamento de informações com outros cuidadores ou estabelecimentos especializados, como hotéis para pets.

1.2 Objetivos Específicos

Para alcançar o objetivo geral, foram estabelecidos os seguintes objetivos específicos:

- Implementar um sistema de cadastro de usuário e perfil dos pets, permitindo o registro de informações detalhadas de cada animal e a criação de uma base personalizada para acompanhamento de saúde e bem-estar;
- Desenvolver funcionalidades de lembretes e notificações, de modo a auxiliar os tutores na manutenção de uma rotina de cuidados consistente, incluindo horários de alimentação, medicação e consultas;
- Criar um módulo de controle alimentar, no qual seja possível monitorar a dieta dos pets, controlar porções e frequência de refeições, promovendo uma nutrição equilibrada e individualizada;
- Integrar uma seção educativa voltada à segurança e alimentação dos pets, fornecendo orientações práticas sobre boas práticas alimentares e cuidados de saúde, a fim de capacitar os tutores na gestão nutricional;
- Implementar uma funcionalidade de compartilhamento de informações, viabilizando a colaboração entre diferentes cuidadores e garantindo que múltiplos responsáveis possam seguir a mesma rotina de cuidados e alimentação.

1.3 Estrutura do Trabalho

Este trabalho está organizado em seis capítulos:

- No Capítulo 1, é apresentada a introdução do projeto, destacando a motivação, os objetivos e a justificativa para o desenvolvimento do UTFPets, bem como a estrutura deste trabalho.
- No Capítulo 2, é apresentado o embasamento teórico que sustenta o projeto, descrevendo, em linhas gerais, conceitos e estudos relacionados a nutrição animal, prevenção de doenças e tecnologias para gerenciamento de cuidados com pets.
- No Capítulo 3, são analisados sistemas similares existentes, realizando-se uma comparação entre as soluções disponíveis, suas funcionalidades, vantagens e limitações.
- No Capítulo 4, são descritos os materiais e métodos empregados, incluindo as ferramentas, tecnologias e o processo metodológico utilizado no desenvolvimento do aplicativo.

- No Capítulo 5, são apresentados os artefatos de análise e projeto do sistema, tais como funcionalidades prioritárias, prototipação de telas e modelagem do banco de dados.
- Por fim, no Capítulo 6, apresenta-se a conclusão do trabalho, juntamente com sugestões para evoluções futuras.

2 REFERENCIAL TEÓRICO

O presente capítulo aborda aspectos fundamentais relacionados à nutrição e à saúde de animais de estimação, com ênfase na importância de controlar a dieta e manter o acompanhamento veterinário. Estudos recentes apontam a obesidade como uma das principais doenças nutricionais em cães e gatos, o que reforça a necessidade de soluções que auxiliem tutores a organizar melhor a rotina de alimentação e cuidados gerais.

2.1 Nutrição e Saúde de Animais de Estimação

Uma alimentação equilibrada é determinante para a qualidade de vida e a longevidade dos animais de estimação. De acordo com German (2022), o manejo adequado do peso, aliado ao fornecimento correto de nutrientes, é essencial para prevenir doenças crônicas como diabetes, dislipidemias e problemas ortopédicos. Além disso, há evidências de que o balanceamento nutricional, somado ao acompanhamento veterinário, reduz custos com tratamentos a longo prazo (CHANDLER, 2022).

2.1.1 Obesidade e Doenças Relacionadas

A obesidade figura entre as enfermidades nutricionais mais diagnosticadas em cães e gatos, caracterizando-se pela ingestão calórica acima das necessidades diárias e pela escassez de atividades físicas (GERMAN, 2022). Esse quadro, segundo o autor, acarreta aumento dos riscos de outras complicações, incluindo doenças hepáticas e cardiovasculares. Já Chandler (2022) ressalta que a conscientização dos tutores sobre a importância da alimentação balanceada pode amenizar esses problemas, diminuindo a incidência de sobrepeso e a necessidade de intervenções médicas posteriores.

2.1.2 Dieta Equilibrada e Fatores de Risco

Diferentes fases de vida — filhote, adulto ou idoso — demandam variações na proporção de proteínas, carboidratos e vitaminas. Fatores como raça e condição fisiológica (gestação, lactação) também exigem adaptações nutricionais (GERMAN, 2022). A falta de um controle preciso das porções ou o fornecimento de alimentos inapropriados gera riscos de deficiências e excessos, comprometendo não apenas a composição corporal, mas também a imunidade e a saúde oral (CHANDLER, 2022).

2.2 Prevenção de Doenças e Rotina de Cuidados

Para além da nutrição, a manutenção de um calendário de vacinas, a desparasitação regular e o controle de consultas veterinárias integram as práticas essenciais ao bem-estar animal. Embora a maioria dessas medidas preventivas seja relativamente simples, o esquecimento ou a falta de organização podem resultar em falhas na rotina de cuidados (KOGAN; HELLYER, 2023).

2.2.1 Vacinação e Check-ups Veterinários

O agendamento periódico de vacinas previne enfermidades infectocontagiosas, enquanto os check-ups de rotina permitem identificar precocemente alterações de saúde. Isso inclui tanto avaliações clínicas quanto a análise de parâmetros sanguíneos ou testes complementares (KOGAN; HELLYER, 2023). Ferramentas que fornecem lembretes dessas datas podem reduzir significativamente a taxa de faltas ou atrasos em procedimentos, favorecendo a prevenção de complicações futuras.

2.2.2 Colaboração entre Múltiplos Cuidadores

Em muitos domicílios, mais de uma pessoa é responsável pelos cuidados diários do animal, o que potencializa erros como a duplicação de porções ou o esquecimento de medicamentos (KOGAN; HELLYER, 2023). Assim, soluções digitais que centralizem informações e permitam o registro de cada tarefa tendem a melhorar a comunicação entre tutores e familiares, viabilizando uma rotina mais estruturada.

2.3 Aplicativos de Gerenciamento Alimentar e Sanitário

O avanço de tecnologias móveis e web tem proporcionado o surgimento de aplicações voltadas ao gerenciamento de rotinas de cuidados com pets. Segundo DataReportal (2023), o aumento do acesso à internet e a popularização de smartphones estimulam a procura por aplicativos que ofereçam lembretes de alimentação, monitoramento de peso e agendamento de procedimentos veterinários. Essas soluções, quando bem utilizadas, podem melhorar a adesão a planos nutricionais e a eficácia das medidas preventivas (KOGAN; HELLYER, 2023).

2.3.1 Benefícios e Desafios

Entre os principais benefícios das aplicações de gerenciamento alimentar e sanitário, destacam-se a automatização de lembretes, o registro histórico de dados e a possibilidade

de acompanhamento remoto (GERMAN, 2022). Entretanto, aspectos como a necessidade de conexão estável e a curva de aprendizado para usuários menos habituados a tecnologias ainda representam desafios (KOGAN; HELLYER, 2023). Além disso, há limitações na coleta de dados precisos, pois o êxito depende, em grande parte, da disciplina dos tutores ao informar peso, porções, horários e eventuais sintomas ou sinais clínicos (CHANDLER, 2022).

2.4 Considerações

A revisão da literatura mostra a relevância de se manter um controle nutricional adequado, reforçar a prevenção de doenças e promover a colaboração entre múltiplos cuidadores para garantir o bem-estar de animais de estimação. Nesse contexto, ferramentas web e móveis mostram-se promissoras ao oferecer funcionalidades como lembretes de refeição, monitoramento de peso e registro de procedimentos veterinários.

No próximo capítulo, serão discutidos trabalhos relacionados que já se propõem a gerenciar rotinas de cuidado e alimentação para pets, analisando suas funcionalidades, pontos fortes e limitações.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta alguns aplicativos existentes no mercado que oferecem funcionalidades para o gerenciamento da saúde e da alimentação de animais de estimação. O objetivo é avaliar em que medida tais soluções atendem às demandas de tutores e como o aplicativo proposto (UTFPets) se diferencia em termos de recursos e abrangência.

3.1 Aplicativos para Organização de Cuidados, Nutrição e Saúde de Pets

Nos últimos anos, surgiram diversas soluções tecnológicas para auxiliar no controle de vacinas, agendamento de consultas, planejamento nutricional e outras práticas relacionadas ao bem-estar dos pets. Entre elas, destacam-se Petzillas, FuncionalPet, PetDesk e Pet Diet Designer, descritos a seguir.

3.1.1 Petzillas

O Petzillas, ilustrado na Figura 1, é um aplicativo gratuito disponível para *Android* e *iOS*, focado em fornecer aos tutores recursos para acompanhamento de vacinas, medicamentos, vermífugos, antipulgas, consultas veterinárias, controle de peso e higiene. Um de seus pontos fortes é a possibilidade de compartilhar dados com outros cuidadores, facilitando a colaboração em lares com múltiplas pessoas responsáveis pelo mesmo animal.



Figura 1 – Página inicial do site do aplicativo Petzillas

3.1.2 FuncionalPet

O FuncionalPet, apresentado na Figura 2, é um software voltado principalmente para clínicas veterinárias, com ênfase na elaboração de dietas personalizadas para cães e gatos. Profissionais podem usar a ferramenta para gerar planos nutricionais e acompanhar a evolução do animal. Por outro lado, a versão atual não oferece recursos de lembretes diários ou de compartilhamento de tarefas entre diversos cuidadores.

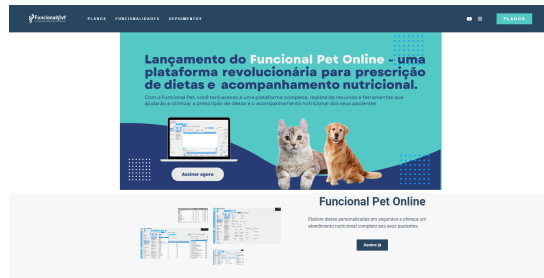


Figura 2 – Página inicial do site do aplicativo FuncionalPet

3.1.3 PetDesk

O PetDesk, ilustrado na Figura 3, destaca-se por auxiliar no gerenciamento de compromissos dos animais, como consultas veterinárias, lembretes de medicação e controle de vacinas. É uma ferramenta útil para quem necessita de mais organização no acompanhamento da saúde do pet. Entretanto, não contempla um módulo nutricional ou o controle de porções diárias, limitando-se ao gerenciamento de eventos de saúde.

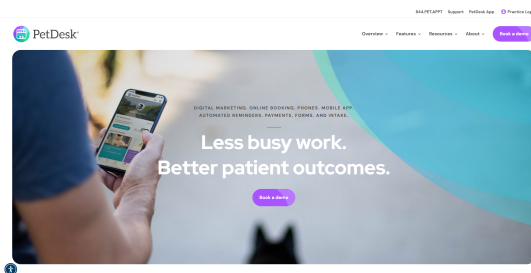


Figura 3 – Página inicial do site do aplicativo PetDesk

3.1.4 Pet Diet Designer

O Pet Diet Designer, ilustrado na Figura 4, é focado na alimentação de pets, com planejamento de dietas, porções e monitoramento do consumo de água. Embora ofereça informações nutricionais e dicas para manter a saúde dos animais, não prevê o registro de cuidados médicos ou colaboração entre diferentes cuidadores.

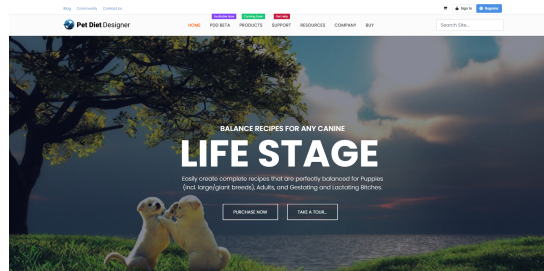


Figura 4 – Página inicial do site do aplicativo Pet Diet Designer

3.2 Considerações

A análise desses aplicativos mostra que há diversas soluções voltadas a aspectos pontuais do cuidado de animais, como planejamento nutricional ou gerenciamento de vacinas. No entanto, poucos integram todas essas funcionalidades em uma única aplicação, e a maioria não conta com recursos robustos para colaboração entre vários cuidadores.

O UTFPets foi idealizado para suprir essa lacuna ao unir, em um só aplicativo, o registro de refeições, o controle de porções, lembretes de atividades e consultas veterinárias, além de uma abordagem colaborativa que permite a participação de diferentes responsáveis pelo mesmo pet. Dessa forma, a proposta do UTFPets combina e expande os pontos positivos das soluções existentes, englobando tanto o controle nutricional quanto o gerenciamento geral de saúde, de maneira integrada e centrada na cooperação entre usuários.

4 MATERIAIS E MÉTODOS

O presente capítulo descreve o conjunto de recursos tecnológicos e metodológicos empregados no desenvolvimento do UTFPets, bem como o processo adotado para organização das tarefas e validação do sistema. Serão apresentados, inicialmente, os *materiais* selecionados (ferramentas, linguagens e ambiente de desenvolvimento). Em seguida, serão detalhados os *métodos* aplicados ao longo do projeto, incluindo as metodologias ágeis (Scrum e Kanban), a abordagem de containerização com Docker e o uso de testes automatizados com Selenium.

4.1 Materiais

A definição das tecnologias utilizadas no UTFPets baseou-se em critérios como robustez, escalabilidade e facilidade de manutenção. A Tabela 1 apresenta um resumo das principais ferramentas e linguagens adotadas, juntamente com suas finalidades.

Tabela 1 – Principais tecnologias adotadas no UTFPets

Tecnologia	Finalidade
Angular (GOOGLE, 2022)	Framework para construção de aplicações <i>web</i> dinâmicas, adotado no desenvolvimento do <i>frontend</i> .
Laravel (OTWELL, 2022)	Framework em PHP para criação de Application Programming Interface (API) escaláveis, com suporte a autenticação e mapeamento objeto-relacional.
PostgreSQL	Sistema Gerenciador de Banco de Dados, conhecido por confiabilidade e suporte a consultas complexas.
Cloudinary (LTD., 2022)	Armazenamento e otimização de imagens, permitindo redução de tamanho e adaptação para múltiplos dispositivos.
Docker (MERKEL, 2014)	Containerização do ambiente, garantindo padronização e eliminando conflitos de dependências entre máquinas.
Selenium	Automação de testes de interface (End-to-End (E2E)), simulando cliques, preenchimentos de formulário e navegação.

4.1.1 Escalabilidade em Aplicativos Web

No contexto do UTFPets, a escalabilidade refere-se à capacidade de o aplicativo lidar com um número crescente de usuários e requisições sem perda significativa de desempenho. Conforme Merkel (2014), o uso de contêineres com Docker possibilita a replicação de serviços em múltiplas instâncias, distribuídas em servidores distintos. Além disso, serviços em nuvem como a Render (RENDER, 2022) permitem expandir recursos de processamento e armazena-

mento de forma dinâmica, garantindo que o sistema mantenha a disponibilidade mesmo em períodos de pico de acessos.

4.1.2 Ambiente de Desenvolvimento

Para assegurar a consistência entre as máquinas dos colaboradores e facilitar a implantação, emprega-se Docker como tecnologia de containerização. Conforme Merkel (2014), o Docker cria ambientes isolados, chamados *containers*, nos quais as aplicações rodam de forma independente, eliminando problemas de incompatibilidade de bibliotecas e versões de sistemas operacionais. Dessa forma, o UTFPets é composto por contêineres específicos para cada serviço:

- **Frontend:** Implementado em Angular, responsável pela interface e interação do usuário via navegador, seguindo o conceito de Single Page Application (SPA).
- **Backend:** Desenvolvido em Laravel, onde são processadas as requisições da API e realizada a comunicação com o banco de dados.
- **Banco de Dados:** Utilizando PostgreSQL, garantindo confiabilidade e escalabilidade no armazenamento de dados.
- **Armazenamento de Mídia:** Empregando Cloudinary, que realiza compressão e adaptação das imagens dos pets.

Além disso, o projeto segue uma abordagem **Monorepo**, em que todo o código-fonte (frontend, backend e scripts) se encontra em um único repositório, facilitando a gestão de dependências e a integração contínua.

4.1.3 Ferramentas Auxiliares

Diversas outras ferramentas contribuem para o desenvolvimento eficiente do UTFPets:

- **Visual Studio Code** (MICROSOFT, 2022): Ambiente de Desenvolvimento Integrado (Integrated Development Environment (IDE)) para edição de código, com extensões específicas para Angular, Laravel e Docker.
- **Node Package Manager (NPM)** (NPM, 2022) e **Composer** (COMMUNITY, 2022): Gerenciadores de pacotes para o *frontend* (Angular) e *backend* (Laravel), respectivamente.
- **Git** (CONTRIBUTORS, 2022) e **GitFlow**: Sistema de controle de versão distribuído, com a estratégia GitFlow para organização de ramificações (*branches*) e implantação de novas funcionalidades.

- **pgAdmin**: Interface gráfica para administração do banco de dados PostgreSQL.
- **Render** (RENDER, 2022): Serviço de hospedagem em nuvem para o *backend*, facilitando o deploy e escalabilidade.

4.2 Métodos

Para conduzir o desenvolvimento do UTFPets de forma organizada e iterativa, foram adotadas metodologias ágeis, com destaque para Scrum e Kanban. Esta seção descreve como tais métodos foram aplicados ao projeto, além de explicar o uso de testes automatizados e a estrutura dos ambientes de desenvolvimento e testes.

4.2.1 Metodologia Ágil: Scrum e Kanban

A adoção de metodologias ágeis busca fornecer flexibilidade e eficiência na entrega de funcionalidades (SCHWABER; SUTHERLAND, 2017; ANDERSON, 2010). O **Scrum** foi escolhido como base, visto que organiza o trabalho em ciclos curtos, denominados *sprints*, nos quais ocorrem planejamento, execução, revisão e retrospectiva. Cada *sprint* tem duração de duas semanas, permitindo entregas incrementais do sistema e maior rapidez na assimilação de feedbacks dos usuários.

- **Product Owner e Scrum Master**: No contexto do UTFPets, essas funções são exercidas pelo professor orientador, que prioriza requisitos e garante a correta aplicação do Scrum, removendo impedimentos quando necessário.
- **Time de Desenvolvimento**: Constituído pelo autor do trabalho, responsável por implementar as funcionalidades conforme as prioridades e critérios de aceitação definidos no *backlog*.

Paralelamente, emprega-se o **Kanban** para uma visualização clara do fluxo de trabalho. De acordo com Anderson (2010), o uso de colunas como *To Do*, *Doing*, *Testing* e *Done* possibilita acompanhar o progresso das tarefas em um quadro visual. Essa abordagem fornece transparência e auxilia na identificação de eventuais gargalos durante as *sprints*.

4.2.2 Estrutura das Sprints

Ao início de cada *sprint*, é realizada uma reunião de planejamento na qual são definidas as funcionalidades prioritárias e estimadas as tarefas a serem concluídas no período. Durante a *sprint*, cada tarefa circula pelas colunas do quadro Kanban, refletindo o estágio atual de desenvolvimento:

- **To Do:** Tarefas ainda não iniciadas.
- **Doing:** Tarefas em andamento.
- **Testing:** Tarefas em fase de testes antes de serem disponibilizadas.
- **Done:** Tarefas concluídas e integradas ao sistema.

No final da *sprint*, ocorre a revisão e a **retrospectiva**, na qual o time avalia o trabalho realizado, discute dificuldades encontradas e propõe melhorias para o próximo ciclo.

4.2.3 Histórias de Usuário

Para descrever e priorizar as funcionalidades, são utilizadas **Histórias de Usuário** no formato:

“Como [tipo de usuário], quero [funcionalidade] para [benefício].”

Esse padrão possibilita que as necessidades dos usuários sejam claras e de fácil compreensão. Exemplo:

“Como tutor de pet, quero cadastrar um novo pet para acompanhar as refeições, garantindo um melhor controle nutricional.”

4.2.4 Testes Automatizados

A garantia de qualidade do UTFPets está baseada em ****testes automatizados**** que abrangem diferentes níveis de verificação:

- **Testes de Interface (E2E):** Conduzidos com Selenium (PROJECT, 2022), para simular interações reais do usuário na aplicação. Clicam em botões, preenchem formulários e verificam fluxos críticos (por exemplo, cadastro de pet, registro de refeição).
- **Testes Unitários:** Aplicados ao **backend** com PHPUnit e ao **frontend** com Jest. Esses testes avaliam blocos de código específicos (funções, métodos ou componentes) para assegurar que comportamentos esperados sejam mantidos.

Conforme apontado por Myers, Sandler e Badgett (2011), a adoção de testes automatizados reduz falhas em produção, melhora a confiabilidade do sistema e agiliza a correção de erros no processo de desenvolvimento.

4.2.5 Ambientes de Desenvolvimento e Testes

Para separar o código em evolução de eventuais liberações estáveis, o UTFPets conta com dois ambientes principais:

- **Ambiente de Desenvolvimento:** Utilizado para implementação contínua e experimentação de novas funcionalidades. Aqui, o time de desenvolvimento executa testes de integração e E2E, garantindo que não ocorram regressões no código.
- **Ambiente de Testes:** Recebe as versões concluídas ao final de cada *sprint*. Nesse ambiente, ocorrem testes mais extensivos e validações finais antes de qualquer disponibilização oficial.

Esse fluxo estruturado fortalece a qualidade do software e permite entregas iterativas e incrementais, alinhadas aos princípios ágeis.

4.3 Considerações

O conjunto de materiais e métodos aqui apresentados fornece a base para o desenvolvimento do UTFPets. A combinação de metodologias ágeis (Scrum e Kanban), containerização com Docker e testes automatizados com Selenium forma um ecossistema de alto controle de qualidade e adaptabilidade. No capítulo seguinte, serão expostas a análise e o projeto do sistema, incluindo a modelagem do banco de dados e a definição das principais funcionalidades da aplicação.

5 ANÁLISE E PROJETO DO SISTEMA

Este capítulo apresenta a concepção do UTFPets de forma estruturada, abordando a priorização de requisitos por meio do método MoSCoW, a arquitetura conceitual (usuários, locais e animais), as funcionalidades essenciais, a modelagem do banco de dados e os protótipos de interface (*wireframes*). Ao final, faz-se referência ao Apêndice A, que detalha as funcionalidades futuras planejadas para versões subsequentes.

5.1 Implementação do Método MoSCoW

Para priorizar as funcionalidades do UTFPets, foi aplicado o método **MoSCoW**, que classifica os requisitos em quatro categorias:

- **Must Have (Essencial)**: Funcionalidades obrigatórias para o funcionamento do Minimum Viable Product (MVP), garantindo que a aplicação seja viável para o primeiro lançamento.
- **Should Have (Importante)**: Funcionalidades que melhoram a experiência do usuário ou adicionam valor ao sistema, mas não são imprescindíveis na versão inicial.
- **Could Have (Desejável)**: Recursos adicionais que podem ser incluídos no futuro, ampliando as capacidades do sistema.
- **Won't Have (Não serão implementadas agora)**: Funcionalidades que, embora relevantes, não estão no escopo atual e podem ser consideradas em atualizações posteriores.

Este capítulo foca nas funcionalidades **Must Have**, descritas ao longo das seções seguintes, de modo a viabilizar o lançamento inicial do UTFPets. As demais categorias (*Should Have*, *Could Have* e *Won't Have*) estão descritas no **Apêndice A**, que consolida o *roadmap* de evolução planejado para o projeto.

5.2 Estrutura do Aplicativo

O UTFPets organiza os dados de maneira hierárquica para facilitar o gerenciamento colaborativo. A estrutura conceitual é composta por:

- **Usuário**: Cada usuário pode cadastrar e gerenciar múltiplos pets, definindo permissões para cuidadores ou estabelecimentos.
- **Local**: Representa o ambiente onde os pets estão localizados (ex.: residência, clínica veterinária). Um usuário pode administrar vários locais.

- **Pet:** Cada animal possui um perfil individual (nome, espécie, raça, idade, peso, histórico alimentar).
- **Informações:** Incluem registros de alimentação, lembretes médicos e atividades, proporcionando um acompanhamento diário do pet.

Essa organização possibilita ao tutor manejar diversos pets e conceder níveis de acesso diferenciados, garantindo uma experiência colaborativa no cuidado dos animais.

5.3 Funcionalidades Prioritárias (Must Have)

As funcionalidades **Must Have** correspondem ao núcleo do UTFPets, assegurando que o sistema seja útil aos tutores desde o primeiro lançamento.

5.3.1 Tutor

- **HU001:** Como tutor, cadastrar um perfil detalhado do pet (nome, idade, raça, porte, restrições alimentares) para personalizar o acompanhamento nutricional.
- **HU002:** Registrar e monitorar as refeições do pet, anotando horários e quantidades, assegurando uma dieta equilibrada.
- **HU003:** Receber lembretes automáticos sobre alimentação e medicação, garantindo uma rotina consistente.
- **HU004:** Compartilhar informações do pet com cuidadores autorizados, facilitando a gestão colaborativa.
- **HU005:** Gerenciar permissões de acesso, controlando quais dados podem ser vistos ou editados por cuidadores secundários.

5.3.2 Cuidador Secundário

- **HU006:** Visualizar o perfil do pet, incluindo horários e detalhes de alimentação, mantendo o cuidado padronizado.
- **HU007:** Registrar tarefas concluídas, possibilitando ao tutor acompanhar as atividades já executadas.

5.3.3 Administrador

- **HU008:** Garantir a segurança de dados (autenticação, criptografia) e a privacidade das informações dos usuários.
- **HU009:** Gerenciar o cadastro de tutores e pets, preservando a integridade da base de dados.
- **HU010:** Configurar lembretes automáticos e notificações globais, garantindo que os usuários recebam avisos importantes.

Essas funções essenciais viabilizam o controle de alimentação e saúde de pets, permitindo a participação de diferentes cuidadores e a administração segura do sistema.

5.4 Modelagem do Banco de Dados

O banco de dados do UTFPets foi projetado para armazenar informações essenciais de usuários, pets, lembretes e histórico de alimentação. Optou-se pelo **PostgreSQL** como Sistema Gerenciador de Banco de Dados (SGBD), devido à sua confiabilidade e capacidade de lidar com consultas complexas.

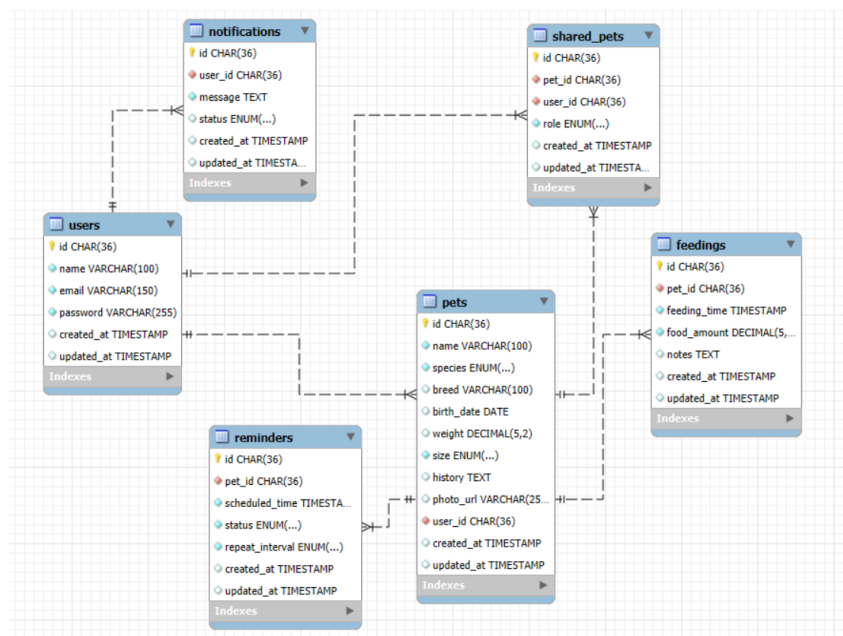


Figura 5 – Diagrama do Banco de Dados do UTFPets (Fonte: Autoria Própria)

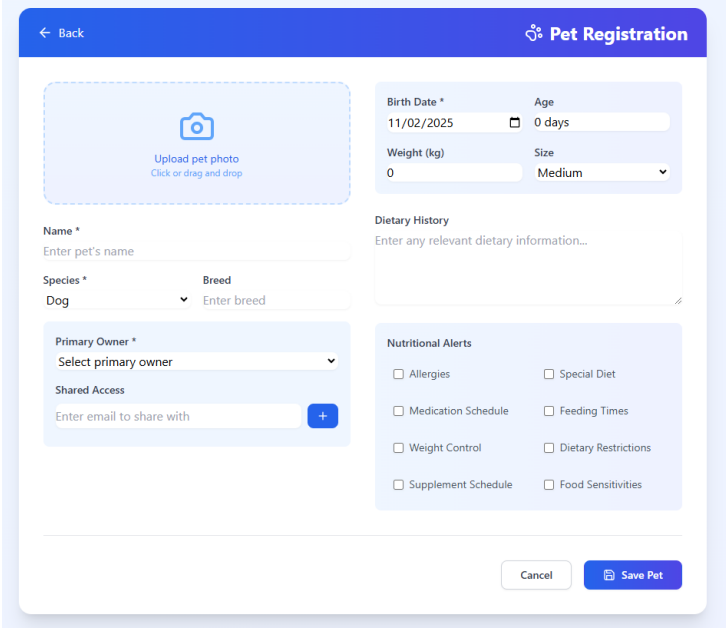
Como ilustra a Figura 5, há diversas tabelas interligadas, abrangendo o relacionamento entre usuários, pets, lembretes, notificações e registros de alimentação. No **Apêndice B**, encontram-se as descrições completas (campos, tipos, chaves primárias e estrangeiras). Abaixo, apresenta-se um resumo das tabelas principais:

- **users:** Contém informações sobre os usuários do sistema, incluindo nome, e-mail e senha protegida por *hash*.
- **pets:** Armazena os dados dos pets cadastrados, como nome, espécie, raça, idade, peso e histórico alimentar.
- **shared_pets:** Define os vínculos entre usuários e pets, permitindo o compartilhamento de informações entre cuidadores autorizados.
- **feedings:** Registra as refeições do pet, armazenando horários e quantidade de alimento administrado, garantindo um histórico alimentar detalhado.
- **reminders:** Controla lembretes programados para eventos como alimentação e administração de medicamentos, alertando os usuários no momento adequado.
- **notifications:** Mantém um histórico de mensagens enviadas ao usuário, informando sobre atualizações e atividades no sistema.

Essa modelagem permite que o sistema seja altamente funcional e seguro, provendo fluxo eficiente de informações e possibilitando um gerenciamento preciso dos cuidados com pets.

5.5 Wireframes das Telas

Para validar a usabilidade do UTFPets, foram criados *wireframes* das telas principais. A Figura 6 ilustra o protótipo de cadastro de pets:



The wireframe shows a 'Pet Registration' form with a blue header bar containing a back arrow and the title. The form is organized into several sections: a photo upload area with a camera icon and instructions; a 'Name' field; a 'Species' dropdown menu with a 'Breed' field; a 'Primary Owner' dropdown; a 'Shared Access' section with an email input and a plus button; a 'Birth Date' and 'Age' section; a 'Weight (kg)' and 'Size' section; a 'Dietary History' text area; and a 'Nutritional Alerts' section with checkboxes for Allergies, Special Diet, Medication Schedule, Feeding Times, Weight Control, Dietary Restrictions, Supplement Schedule, and Food Sensitivities. At the bottom right are 'Cancel' and 'Save Pet' buttons.

Figura 6 – Wireframe de Cadastro de Pets
(Fonte: Autoria Própria)

Elementos da Tela:

- **Foto do Pet:** Possibilita inserir uma imagem para identificação visual.
- **Dados Básicos:** Inclui campos para nome, raça, espécie, idade, peso e porte.
- **Histórico Alimentar:** Espaço para restrições e hábitos alimentares.
- **Lembretes Nutricionais:** Configuração de alertas para alimentação, suplementação e medicação.
- **Compartilhamento:** Permite conceder permissões a cuidadores secundários.

O design segue princípios de User Experience (UX)/User Interface (UI), focando em acessibilidade e usabilidade. O uso de cores suaves e a fonte **Roboto** confere um aspecto moderno, enquanto o layout responsivo atende a diferentes dispositivos.

5.6 Considerações

A análise e o projeto do UTFPets abordaram o método **MoSCoW** para priorizar as funcionalidades, estabelecendo as características fundamentais (*Must Have*) que compõem a primeira versão do sistema. Também foi definida a arquitetura hierárquica (usuário, local, animal), o modelo de banco de dados em PostgreSQL e a concepção das telas principais por meio de *wireframes*.

Visando à contínua evolução do UTFPets, funcionalidades categorizadas em *Should Have*, *Could Have* e *Won't Have* estão descritas no **Apêndice A**. Esse plano de expansão assegura que o sistema permaneça escalável e apto a incorporar novos recursos, conforme as demandas dos usuários e o surgimento de tecnologias relacionadas à saúde e bem-estar de animais de estimação.

6 CONCLUSÃO

O desenvolvimento do UTFPets se apresenta como uma solução inovadora para auxiliar tutores no gerenciamento da alimentação e cuidados de seus animais de estimação. A análise de trabalhos relacionados demonstrou que existem aplicativos que oferecem funcionalidades específicas, como controle de vacinas, planejamento de dietas e lembretes de consultas. No entanto, a integração dessas funcionalidades em um único aplicativo, que proporcione gestão completa e colaborativa, ainda é uma lacuna no mercado.

Durante a concepção do UTFPets, foram estabelecidos objetivos claros para garantir o desenvolvimento de um aplicativo intuitivo e eficiente. A solução permitirá o registro detalhado de informações dos pets, a criação de lembretes automáticos, a gestão alimentar e a possibilidade de compartilhamento de informações entre tutores e cuidadores. A escolha de tecnologias modernas, como Angular, Laravel, PostgreSQL, Docker e Cloudinary, garantirá escalabilidade, segurança e eficiência na execução do sistema.

A abordagem metodológica adotada, baseada em metodologias ágeis, permitirá que o desenvolvimento do sistema ocorra de maneira iterativa e adaptável às necessidades identificadas ao longo do processo. A utilização do método MoSCoW possibilitará a priorização das funcionalidades, assegurando que a versão inicial do aplicativo atenda às necessidades essenciais dos tutores.

Com a implementação do UTFPets, espera-se simplificar a rotina de cuidados dos tutores, promovendo a saúde e o bem-estar dos animais de estimação. O aplicativo fornecerá uma experiência mais organizada e colaborativa, permitindo que os tutores monitorem e administrem os cuidados de seus pets de forma eficiente e confiável.

No futuro, a evolução do UTFPets incluirá a expansão de funcionalidades, como a implementação de um histórico avançado de saúde, recomendações baseadas em inteligência artificial para ajustes na alimentação e integração com serviços veterinários. Dessa forma, o UTFPets busca se consolidar como uma ferramenta essencial no cuidado de animais de estimação, promovendo uma gestão eficiente e inovadora para os tutores.

REFERÊNCIAS

- ANDERSON, D. J. **Kanban: Successful evolutionary change for your technology business**. Seattle, WA: Blue Hole Press, 2010.
- CHANDLER, M. Nutrition for weight management in cats and dogs. **UK Vet: Companion Animal**, v. 27, n. 5, p. 234–242, 2022.
- COMMUNITY, C. **Composer - Dependency Manager for PHP**. 2022. Disponível em: <https://getcomposer.org/>.
- CONTRIBUTORS, G. **Git - Distributed Version Control System**. 2022. Disponível em: <https://git-scm.com/>.
- DATAREPORTAL. **Digital 2023: Global Overview Report**. 2023. <https://datareportal.com/reports/digital-2023-global-overview-report>. Accessed: 2023-08-01.
- FASCETTI, A.; DELANEY, S. **Applied Veterinary Clinical Nutrition**. [S.l.]: Wiley-Blackwell, 2012.
- GERMAN, A. J. Updates on obesity management for dogs and cats. **Veterinary Clinics of North America: Small Animal Practice**, v. 52, n. 5, p. 1053–1070, 2022.
- GOOGLE. **Angular - The Modern Web Framework**. 2022. Disponível em: <https://angular.io/>.
- KOGAN, L. R.; HELLYER, P. W. Access to veterinary care in the us: A preliminary study of pet owner experiences. **Animals**, v. 13, n. 2, p. 340, 2023.
- LAFLAMME, D. P.; FLAMMER, S. A.; HANSEN, B. D. Obesity in dogs and cats: A metabolic and endocrine disorder. **Topics in Companion Animal Medicine**, v. 23, n. 3, p. 126–131, 2008.
- LTD., C. **Cloudinary - Image and Video API for Developers**. 2022. Disponível em: <https://cloudinary.com/>.
- MERKEL, D. Docker: lightweight linux containers for consistent development and deployment. **Linux Journal**, v. 2014, n. 239, p. 2, 2014.
- MICROSOFT. **Visual Studio Code - Code Editing. Redefined**. 2022. Disponível em: <https://code.visualstudio.com/>.
- MYERS, G. J.; SANDLER, C.; BADGETT, T. **The Art of Software Testing**. 3. ed. [S.l.]: Wiley, 2011.
- NPM, I. **NPM - Node Package Manager**. 2022. Disponível em: <https://www.npmjs.com/>.
- OTWELL, T. **Laravel - The PHP Framework for Web Artisans**. 2022. Disponível em: <https://laravel.com/>.
- PROJECT, S. **Selenium - Web Browser Automation**. 2022. Disponível em: <https://www.selenium.dev/>.
- RENDER. **Render - Cloud Hosting for Developers**. 2022. Disponível em: <https://render.com/>.
- SCHWABER, K.; SUTHERLAND, J. **The Scrum Guide**. 2017. <https://scrumguides.org/scrum-guide.html>. Accessed: 2023-08-01.

APÊNDICE A – Funcionalidades Futuras e Expansão do UTFPets

Este apêndice apresenta as funcionalidades classificadas como **Should Have**, **Could Have** e **Won't Have**, que podem ser implementadas em futuras versões do UTFPets. Essas funcionalidades foram definidas com base na priorização estabelecida pelo método Moscow, garantindo uma evolução estruturada da aplicação.

O UTFPets foi concebido como um sistema escalável, possibilitando a incorporação de novas funcionalidades ao longo do tempo. A seguir, são detalhadas as funcionalidades futuras, organizadas conforme seu nível de prioridade e impacto na experiência do usuário.

A.1 Funcionalidades Importantes (Should Have)

As funcionalidades listadas nesta seção são consideradas relevantes para aprimorar a experiência do usuário e tornar o sistema mais completo. No entanto, sua implementação não é essencial para a versão inicial do UTFPets. Elas poderão ser incluídas em futuras iterações do sistema à medida que o aplicativo evoluir.

- **HU011** - Como tutor, desejo acessar informações educativas sobre cuidados e nutrição animal, para aprimorar o manejo diário dos pets e garantir melhores condições de saúde.
- **HU012** - Como tutor, quero personalizar os lembretes de alimentação, medicação e atividades, para adaptar a rotina do aplicativo às necessidades específicas do meu pet.
- **HU015** - Como tutor, desejo visualizar um gráfico de acompanhamento do peso do meu pet, permitindo uma análise da evolução do seu estado físico ao longo do tempo.

Essas funcionalidades contribuirão significativamente para tornar o UTFPets uma ferramenta mais informativa e flexível, oferecendo maior autonomia ao usuário no gerenciamento das rotinas dos pets.

A.2 Funcionalidades Desejáveis (Could Have)

As funcionalidades listadas nesta seção são desejáveis e agregariam valor ao UTFPets, mas sua implementação não é prioritária no escopo atual. Elas poderão ser desenvolvidas futuramente, dependendo das necessidades identificadas pelos usuários e da viabilidade técnica.

- **HU013** - Como tutor, desejo acessar um histórico detalhado de refeições e atividades do meu pet, permitindo um monitoramento contínuo de sua alimentação e bem-estar.
- **HU014** - Como tutor, quero integrar os lembretes do UTFPets ao calendário do meu dispositivo móvel, para centralizar as notificações e facilitar o gerenciamento das tarefas diárias relacionadas ao pet.

- **HU016** - Como tutor, desejo receber sugestões automáticas de dieta com base na idade, porte e raça do meu pet, auxiliando na formulação de uma alimentação mais equilibrada.

A inclusão dessas funcionalidades expandirá as capacidades do UTFPets, tornando-o um aplicativo mais interativo e conectado ao cotidiano dos usuários.

A.3 Funcionalidades Futuras (Won't Have)

As funcionalidades listadas nesta seção foram identificadas como relevantes, mas não serão implementadas na versão inicial do UTFPets devido a restrições de tempo, recursos e complexidade técnica. Entretanto, elas poderão ser consideradas para versões futuras do sistema, à medida que o aplicativo amadurecer e novas demandas forem identificadas.

- **Dietas personalizadas:** Implementação de um módulo onde veterinários poderiam criar dietas específicas para cada pet diretamente dentro do aplicativo, considerando suas características e necessidades nutricionais individuais.
- **Integração com serviços veterinários:** Implementação de uma funcionalidade que permitiria consultas remotas e acompanhamento da saúde do pet em tempo real, facilitando a comunicação entre tutores e profissionais da área veterinária.
- **Sistema de recomendação de produtos:** Introdução de um sistema baseado em inteligência artificial para sugerir rações, suplementos e acessórios de acordo com o perfil do pet e seu histórico alimentar.
- **Monitoramento via dispositivos inteligentes:** Integração do UTFPets com dispositivos como coleiras inteligentes e rastreadores de atividade física para fornecer dados sobre comportamento, sono e exercícios dos pets.

Embora essas funcionalidades não estejam no escopo atual, elas representam possibilidades para a evolução do UTFPets, permitindo sua adaptação a novas tendências tecnológicas e demandas do mercado.

A.4 Conclusão

A definição de funcionalidades futuras e sua categorização conforme o método Moscow garantem um planejamento estratégico para o desenvolvimento do UTFPets. As funcionalidades listadas como **Must Have** foram implementadas na versão inicial, enquanto as demais (**Should Have**, **Could Have** e **Won't Have**) compõem um roadmap de evolução do sistema.

APÊNDICE B – Detalhes das Tabelas do Banco de Dados

Este apêndice reúne as descrições completas das tabelas que compõem o banco de dados do UTFPets, complementando a visão geral apresentada no Capítulo 5.

B.1 Tabela `users`

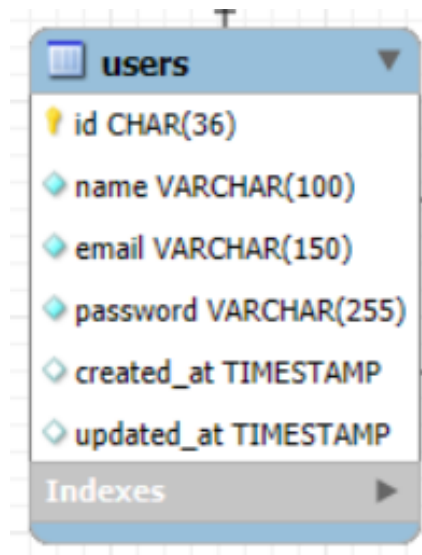


Figura 7 – Estrutura da tabela `users` (Fonte: Autoria Própria)

A tabela `users` armazena os dados de identificação e autenticação dos usuários. Os principais campos são:

- `id`: Identificador único do usuário (tipo `CHAR(36)`) gerado via UUID.
- `name`: Nome do usuário (`VARCHAR(100)`).
- `email`: Endereço de e-mail (`VARCHAR(150)`).
- `password`: Senha criptografada (`VARCHAR(255)`).
- `created_at`, `updated_at`: Registros de data/hora de criação e atualização, respectivamente.

B.2 Tabela `pets`

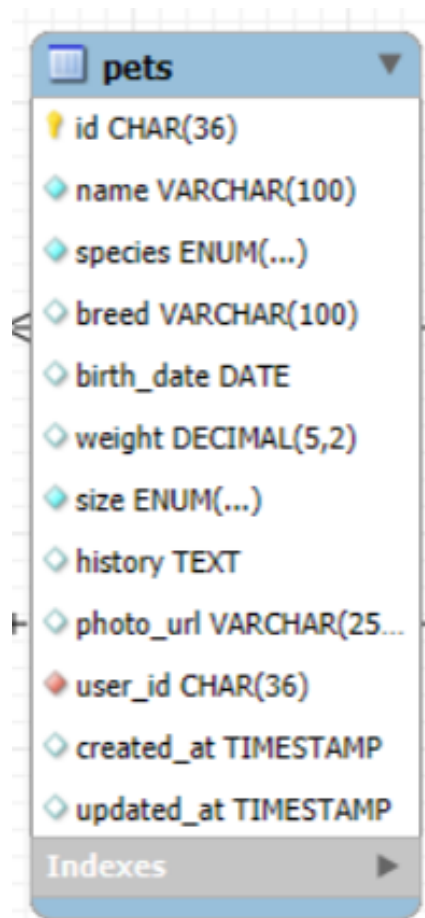


Figura 8 – Estrutura da tabela `pets` (Fonte: Autoria Própria)

A tabela `pets` guarda os dados de cada animal cadastrado no sistema:

- `id`: UUID único do pet (`CHAR(36)`).
- `name`: Nome do pet (`VARCHAR(100)`).
- `species`, `breed`, `size`: Campos do tipo `ENUM(...)` ou `VARCHAR`.
- `birth_date`: Data de nascimento (`DATE`).
- `weight`: Peso (`DECIMAL(5,2)`).
- `history`: Texto livre para anotações.
- `photo_url`: Caminho/URL da imagem no armazenamento.
- `user_id`: Chave estrangeira referenciando o dono principal (`users.id`).
- `created_at`, `updated_at`: Datas de criação e atualização.

B.3 Tabela feedings

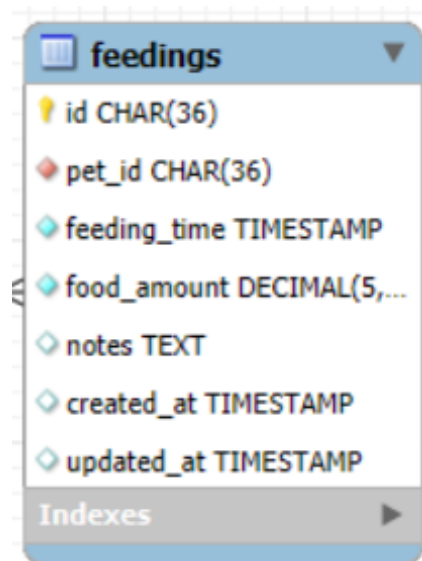


Figura 9 – Estrutura da tabela feedings (Fonte: Autoria Própria)

A tabela `feedings` registra as refeições de cada pet:

- `id`: UUID (CHAR(36)).
- `pet_id`: Chave estrangeira para `pets`.
- `feeding_time`: Data e hora (TIMESTAMP).
- `food_amount`: Quantidade de alimento (DECIMAL(5, x)).
- `notes`: Observações (p. ex., tipo de ração).
- `created_at`, `updated_at`: Marcas de tempo de criação e atualização.

B.4 Tabela reminders

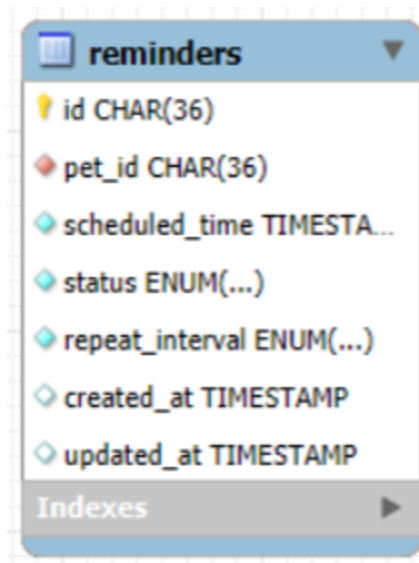


Figura 10 – Estrutura da tabela reminders (Fonte: Autoria Própria)

A tabela `reminders` armazena lembretes programados para eventos relacionados ao cuidado dos pets:

- `id`: UUID (CHAR(36)).
- `pet_id`: Chave estrangeira para `pets`.
- `scheduled_time`: Data/hora (TIMESTAMP).
- `status`: ENUM(...) indicando estado (ativo, concluído etc.).
- `repeat_interval`: Periodicidade (ENUM(...)).
- `created_at`, `updated_at`: Datas de criação e atualização.

B.5 Tabela `shared_pets`

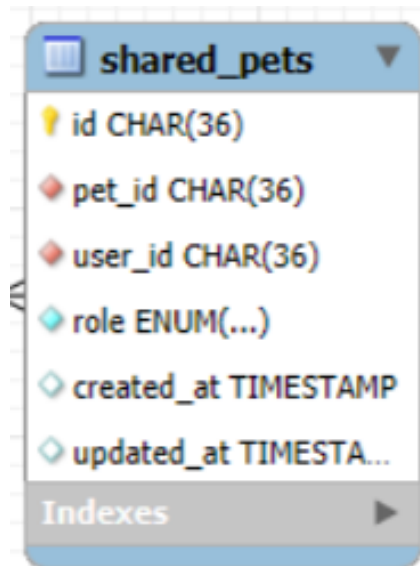


Figura 11 – Estrutura da tabela `shared_pets` (Fonte: Autoria Própria)

A tabela `shared_pets` controla o compartilhamento de um pet entre diferentes usuários:

- `id`: UUID (CHAR(36)).
- `pet_id`: Referência para `pets`.
- `user_id`: Referência para `users`, identificando o cuidador adicional.
- `role`: ENUM(...) definindo acesso ('viewer', 'editor' etc.).
- `created_at`, `updated_at`: Criação e atualização.

B.6 Tabela notifications

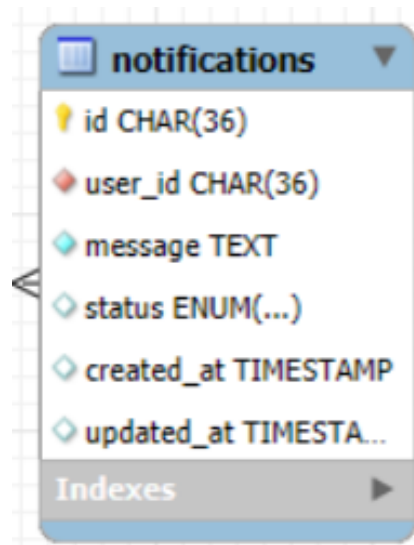


Figura 12 – Estrutura da tabela notifications (Fonte: Autoria Própria)

A tabela `notifications` mantém um histórico de avisos enviados ao usuário:

- `id`: UUID (CHAR(36)).
- `user_id`: Relaciona qual usuário recebeu a notificação.
- `message`: Conteúdo textual do aviso ou alteração.
- `status`: ENUM(...) (lida, pendente etc.).
- `created_at`, `updated_at`: Datas de criação e atualização.