Patinhas da Serra: Solução Web para Animais Perdidos, Abandonados e para Adoção[1]

Dhiulia Caroline Antunes da Silva[2], Fabieli de Conti[3]

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – Campus Farroupilha – 95.174-274 – Farroupilha – RS – Brazil

dhiuliaantunes@gmail.com, fabieli.conti@farroupilha.ifrs.edu.br

Abstract. *Animal abandonment is a concerning issue that affects communities and the environment, requiring actions that promote awareness and implement solutions. This article examines the context of animal abandonment and loss in Serra Gaúcha, highlighting the importance of technological tools to mitigate the problem. It is proposed to develop a web application composed of a front-end created with the React framework, a REST (Representational State Transfer) API (Application Programming Interface) developed in C# for the back-end, and a SQL (Structured Query Language) database, using the Microsoft SQL Server management system, for recording and managing publications and administrators. The application aims to optimize the dissemination of animals for adoption, lost or found pets, offering advanced features such as filters and ordering of occurrences, in addition to facilitating interaction between the community and local NGOs (Non-Governmental Organizations).*

Resumo. *O abandono de animais é uma questão preocupante que afeta comunidades e o meio ambiente, demandando ações que promovam a conscientização e soluções práticas. Este artigo aborda o contexto do abandono e perda de animais na Serra Gaúcha, destacando a importância de ferramentas tecnológicas para mitigar o problema. Propõe-se o desenvolvimento de uma aplicação web composta por um front-end criado com o framework React, uma API (Application Programming Interface) que segue o estilo de arquitetura REST (Representational State Transfer) desenvolvida em C# para o back-end, e um banco de dados SQL (Structured Query Language), utilizando o sistema de gerenciamento Microsoft SQL Server, para o registro e gerenciamento das publicações e dos administradores. A aplicação visa otimizar a divulgação de animais para adoção, pets perdidos ou encontrados, oferecendo recursos avançados como filtros e ordenação de ocorrências, além de facilitar a interação entre a comunidade e as ONGs* (Organizações não governamentais) *locais.*

1. Introdução

O abandono de animais é uma problemática crescente no Brasil, com impactos diretos na sociedade, no meio ambiente e na saúde pública. A estimativa de milhões de animais abandonados no país, divulgada pela OMS (Organização Mundial de Saúde), reflete a

[1] Artigo científico referente ao Trabalho de Conclusão de Curso do curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal do Rio Grande do Sul - Campus Farroupilha.

[2] Aluna matriculada no Trabalho de Conclusão de Curso.

[3] Professora orientadora do Trabalho de Conclusão de Curso.

dimensão desse desafio, que envolve não apenas a falta de políticas públicas eficazes, mas também a ausência de conscientização sobre posse responsável. Nesse cenário, iniciativas comunitárias e tecnológicas têm se mostrado ferramentas importantes para mitigar os efeitos do abandono e promover soluções sustentáveis.

Nesse contexto, três moradoras da cidade de Farroupilha - RS criaram um perfil na rede social Instagram @patinhasdaserraa, com o intuito de divulgar animais para adoção, animais perdidos ou que estejam abandonados nas cidades da região, buscando reduzir a quantidade de animais de rua e conscientizar a população. Quando um animal é encontrado aparentemente perdido, a informação é compartilhada a fim de localizar o dono ou encontrar um lar temporário, também servindo como sinalização para as ONGs (Organização Não Governamental) que realizam resgates de animais para levá-lo a um abrigo. Diante disso, pessoas que encontram um pet (animal de estimação) perdido, procuram um animal desaparecido ou desejam doar um animal podem entrar em contato para solicitar ajuda na divulgação. No entanto, como as três administradoras da conta possuem outras ocupações em seu dia a dia, nem sempre conseguem responder rapidamente ou solicitar informações adicionais. Para otimizar esse processo, surgiu a ideia de automatizá-lo parcialmente por meio de um formulário on-line.

A digitalização de processos, como a criação de plataformas online para registro e divulgação de ocorrências relacionadas aos pets surge como uma resposta eficiente. Essa ferramenta permite uma gestão mais ágil e organizada das informações, facilitando o contato entre tutores, adotantes e ONGs. Atualmente, aplicações web são amplamente utilizadas em diversas áreas, como tecnologia, saúde e educação, oferecendo soluções acessíveis e versáteis. Esses softwares, hospedados em servidores e acessíveis via navegador, permitem execução em computadores e dispositivos móveis, bastando somente conexão com a internet. Para o desenvolvimento do projeto foram utilizadas as tecnologias HTML (Hypertext Markup Language), JavaScript e PHP (Hypertext Preprocessor), as aplicações web variam de simples páginas a sistemas complexos, exemplificando a evolução da web para atender demandas mais sofisticadas e interativas (WK Technology, 2024). O trabalho apresentado neste artigo, intitulado “Patinhas da Serra”, pode ser classificado como uma aplicação simples.

Este artigo explora o contexto do abandono de animais na Serra Gaúcha e propõe uma solução tecnológica para aprimorar a comunicação e a busca por lares para pets abandonados ou perdidos. A análise baseia-se em dados sobre a realidade do abandono de animais no Brasil e no papel de iniciativas locais, como a criação de um perfil em redes sociais e um *website*, que busca integrar a comunidade na resolução desse problema. Atualmente, o Instagram do projeto Patinhas da Serra conta com mais de 340 ocorrências, a grande maioria na cidade de Farroupilha, ao longo 4.5 anos de atividade, demonstrando a procura da comunidade em divulgar animais para adoção, resgatados ou desaparecidos. Assim, o presente artigo visa abordar um referencial teórico que fundamenta as ações propostas e destaca a importância de utilizar tecnologias acessíveis e eficientes para reduzir o impacto desse cenário alarmante.

O restante deste trabalho está organizado em seções que visam apresentar, de forma clara e estruturada, o desenvolvimento e os resultados obtidos ao longo do

projeto. Inicialmente, é apresentado um referencial teórico que contextualiza a problemática do abandono de animais, explorando dados relevantes e trabalhos relacionados que fundamentam a proposta. Em seguida, a seção de metodologia descreve as tecnologias, ferramentas e procedimentos adotados no desenvolvimento da aplicação. A próxima seção detalha os aspectos do desenvolvimento, incluindo requisitos, modelagem e as decisões técnicas que nortearam o projeto. Posteriormente, são apresentados os resultados obtidos, evidenciando as funcionalidades implementadas e a forma como a solução proposta contribui para a problemática estudada. Por fim, são expostas as considerações finais, seguidas pelas referências bibliográficas utilizadas ao longo do artigo.

2. Referencial Teórico

Esta seção apresenta os fundamentos teóricos essenciais para o desenvolvimento do artigo, abordando questões relacionadas ao abandono e perda de animais, à divulgação de pets na Serra Gaúcha e ao uso de aplicações web como ferramenta de apoio.

2.1. O abandono de animais

Segundo Carvalho (2024), o abandono de animais é um problema crescente que afeta não apenas os animais, mas também as comunidades e o meio ambiente. De acordo com a OMS, foi estimado que, durante o ano de 2022, existiam aproximadamente 30 milhões de animais abandonados no Brasil, sendo 10 milhões de gatos e 20 milhões de cachorros.

Conforme Silva (2024), o relatório do Índice de Abandono Animal no Brasil, liderado pela Mars Petcare (líder mundial em pet food) em parceria com especialistas e organizações, busca entender o número de cães e gatos sem acesso a cuidados adequados. O objetivo é promover ações direcionadas para reduzir o abandono e melhorar a qualidade de vida dos animais. O estudo revela que o Brasil abriga aproximadamente 121,3 milhões de cães e gatos, sendo 82,1 milhões de cães e 39,2 milhões de gatos. Desses, cerca de 25% estão em situação de abandono. Essa pesquisa também pontuou que, dentre os 497 tutores de cães e 439 tutores de gatos entrevistados, 11% consideraram renunciar a seus cães e 13% a seus gatos nos próximos 12 meses. Os principais motivos apontados foram: incapacidade física para cuidar, mudança de residência, falta de tempo, alergia de um familiar e comportamento inadequado do animal.

Queiroz et al. (2019) enfatiza que o abandono de animais também é ocasionado pela compra impulsiva, seja para si ou como presente, sem a devida preparação para um compromisso que pode durar cerca de 15 anos. Antes de doar um animal, é essencial verificar se o novo tutor tem condições financeiras e disponibilidade para cuidar dele. A falta dessas condições pode levar o pet a sofrer fome, frio e solidão, ou até mesmo a fugir em busca de um ambiente melhor.

Esse cenário reflete questões sociais profundas, como a falta de conscientização sobre a posse responsável e a ausência de políticas públicas eficazes. Além disso, o abandono contribui para o aumento de animais em situação de rua, podendo gerar impactos na saúde pública.

Além disso, é importante destacar que esses problemas têm um impacto direto na sociedade. A falta de controle sobre a população de cães e gatos em situação de rua pode transformá-los em transmissores de diversas doenças, como raiva, cólera, toxoplasmose, giardíase, enterocolite, infecções bacterianas causadas por mordidas ou arranhões, leptospirose e outras zoonoses (Queiroz et al., 2019).

Também cabe ressaltar que, além de configurar-se como um ato de crueldade, o abandono de animais domésticos é crime previsto na Lei Federal nº 9.605/98, com pena de prisão e multa, agravada se o ato levar à morte do animal. A Lei nº 14.064/20, sancionada em 2020, aumentou a pena máxima de detenção de um para cinco anos, transferindo o julgamento para a vara criminal. Abandono também inclui negligenciar cuidados básicos, como alimentação, higiene e saúde, ou manter o animal acorrentado e isolado (CRMVPA, 2024).

Ainda conforme Queiroz et al. (2019), o crescimento das ONGs dedicadas à defesa dos animais tem sido cada vez mais significativo. Essas organizações, atuando de forma independente, preenchem a lacuna deixada pelo poder público, que muitas vezes não consegue agir de maneira eficaz. Elas se empenham em proteger a vida e o bem-estar de animais vulneráveis, onde a assistência do Estado é insuficiente.

2.2. Trabalhos relacionados

Como parte do processo de desenvolvimento do sistema proposto, foi realizada uma pesquisa exploratória sobre sites e aplicativos que compartilham objetivos semelhantes, voltados à causa animal, especialmente no que diz respeito à adoção de pets, resgate de animais em situação de rua e auxílio a tutores em busca de seus animais perdidos. O intuito dessa análise comparativa foi identificar funcionalidades comuns, boas práticas, limitações e diferenciais das plataformas existentes, a fim de alinhar o projeto às necessidades reais dos usuários e oferecer uma solução mais acessível, colaborativa e inclusiva. Essa investigação permitiu compreender o papel das grandes ONGs, plataformas de diretórios de adoção e redes de apoio à causa, além de destacar oportunidades ainda pouco exploradas, como o engajamento direto de voluntários e cidadãos comuns por meio de uma interface intuitiva e colaborativa.

Como resultado da pesquisa, foram encontrados os sites: Adote um Focinho; Ampara Animal e Petfinder, que serão detalhados nos tópicos abaixo**.**

2.2.1. Adote um Focinho

A Associação Casa da Passagem São Lázaro é uma organização sem fins lucrativos fundada oficialmente em maio de 2006, em São Paulo, com o objetivo de resgatar, cuidar e promover a adoção de animais em situação de abandono. Seu trabalho, no entanto, já era realizado de forma independente por seus fundadores muito antes da formalização. Movidos pela indignação frente ao descaso com os animais de rua, os idealizadores da associação uniram esforços para garantir abrigo, cuidados veterinários e um lar definitivo para cães resgatados. Atualmente, a instituição mantém um sítio com aproximadamente 150 animais, operando no limite de sua capacidade física e contando apenas com recursos próprios e doações de apoiadores. Todos os cães são vacinados e castrados antes da adoção. (Adote um Focinho, 2023)

Ainda conforme o site desse projeto, a trajetória da associação é marcada por desafios, como a perda de seu primeiro abrigo improvisado e a necessidade urgente de realocação dos animais. Graças à solidariedade de voluntários, doações e apoio profissional, a Casa da Passagem São Lázaro conseguiu construir um espaço seguro e adequado para continuar sua missão de proteção e respeito aos animais.

O site disponibiliza informações sobre os animais disponíveis para adoção e os interessados devem preencher um formulário e aguardar o contato da associação ou utilizar meios de comunicação como WhatsApp dos fundadores. A Figura 1 ilustra a interface em que os pets são apresentados.

Figura 1. Pets aguardando adoção

Fonte: Adote um Focinho, 2025

2.2.2. Petfinder

A Petfinder é uma plataforma online dedicada à promoção da adoção de animais, funcionando como um banco de dados pesquisável de pets que precisam de um lar e como um diretório de abrigos e organizações de adoção. Cada instituição parceira possui sua própria página inicial e banco de dados com os animais disponíveis. (PetFinder, 2025)

A missão da Petfinder é utilizar os recursos da internet para aumentar a conscientização sobre a adoção de animais e melhorar a eficácia dos programas de adoção em todo o Brasil. Atuando como uma iniciativa digital inovadora, a plataforma busca aplicar os avanços tecnológicos para enfrentar os principais desafios do resgate animal. Além disso, realiza campanhas e programas com foco na defesa e no bem-estar dos animais. Também cabe destacar que a Petfinder não realiza resgates nem mantém abrigos físicos, atuando exclusivamente online para dar visibilidade aos animais que

aguardam adoção. Atualmente, a plataforma contabiliza centenas de cães e gatos disponíveis e adotados por meio de sua rede de apoio. (PetFinder, 2025)

A Figura 2 demonstra a página em que os pets que procuram lar são exibidos.

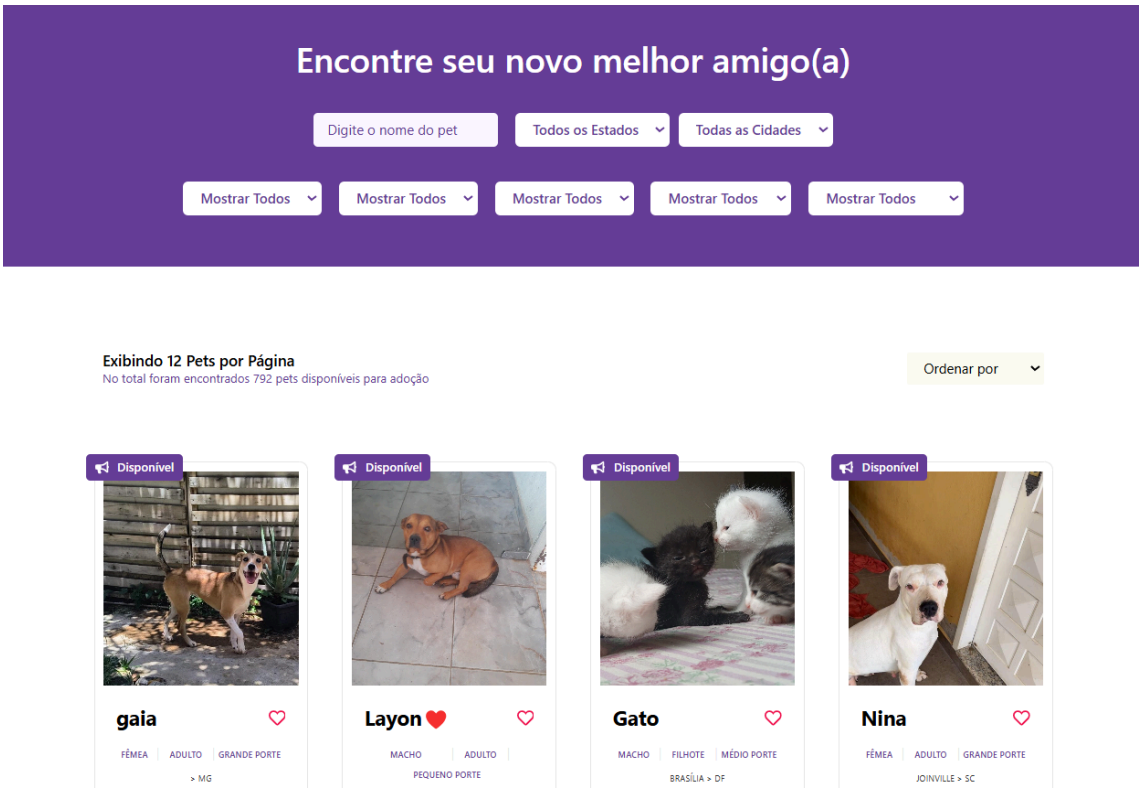

Figura 2. Pets disponíveis para adoção

Fonte: PetFinder, 2025

2.2.3. Ampara Animal

O Instituto Ampara Animal é uma organização não-governamental sem fins lucrativos, 100% brasileira, fundada e liderada por mulheres. Criado em 2010 com o propósito de apoiar cães e gatos em situação de vulnerabilidade, o Instituto tem como foco principal a conscientização e a prevenção. Em 2013, recebeu do Ministério da Justiça o título de OSCIP (Organização da Sociedade Civil de Interesse Público), reforçando sua credibilidade e compromisso com a causa animal. Em 2015, a Ampara tornou-se a maior e mais relevante entidade de proteção e defesa animal do Brasil, sendo reconhecida em 2019 como uma das 100 melhores ONGs do país, com destaque na região Sudeste. Com a expansão de suas atividades, em 2016 foi criada a Ampara Silvestre, uma vertente voltada ao bem-estar de animais silvestres em cativeiro e à conservação da fauna brasileira. (Ampara, 2025)

O compromisso com a preservação ambiental também se concretizou em 2021, quando o Instituto passou a proteger diretamente uma área de 100 hectares no Pantanal

Mato-grossense. A missão da organização é transformar a sociedade por meio de ações preventivas e educativas voltadas à defesa dos direitos dos animais. Além da adoção, o Instituto oferece a possibilidade de apadrinhamento de animais resgatados, muitos dos quais são idosos, doentes ou necessitam de cuidados especiais. Por meio de contribuições mensais, padrinhos e madrinhas ajudam a garantir a saúde e o bem-estar desses animais, recebendo regularmente notícias e fotos de seus afilhados (Ampara, 2025).

Na Figura 3 é possível observar os animais que podem ser adotados a partir dessa plataforma.

Figura 3. Animais para adoção

Fonte: Ampara, 2025

2.2.4. Comparativo entre os sistemas

A partir da pesquisa referente aos projetos semelhantes, citados acima, foi possível montar uma tabela comparativa (Tabela 1), para compreender os possíveis diferenciais que o presente projeto oferece.

Tabela 1. Comparativo entre o projeto e os sistemas pesquisados

Critério	Patinhas da Serra	Adote um Focinho	Petfinder Brasil	Ampara Animal
Objetivo Principal	Plataforma colaborativa para adoção, pets perdidos e resgatados	Adoção direta de cães resgatados pela ONG	Diretório de animais para adoção listados por ONGs e abrigos	Defesa animal, advocacy e ações educativas
Tipo de Contribuinte	Aberto a qualquer pessoa (usuários comuns, tutores que precisam de ajuda e voluntários)	Fechado à atuação da própria ONG	Principalmente ONGs e protetores parceiros	Atuação interna da organização
Publicação de Ocorrências	Usuários podem criar postagens de pets perdidos, achados ou para adoção	Apenas animais sob cuidado da ONG são publicados	ONGs cadastradas publicam seus animais	Adoção de animais, entre outras informações relacionadas à preservação ambiental
Gestão de Administradores	Sistema completo de cadastro, edição e exclusão de admins	Não aplicável	Descentralizado por ONG	Não aplicável
Usuários anônimos/voluntários podem contribuir?	Sim. Esse é um dos principais diferenciais.	Não (apenas equipe da ONG)	Não diretamente – apenas ONGs listadas	Não
Interface adaptada a dispositivos móveis	Sim, como requisito não funcional	Sim	Sim	Sim

Fonte: Autoria Própria

Comparativamente, o sistema “Patinhas da Serra” destaca-se por oferecer uma abordagem colaborativa e inclusiva, permitindo que qualquer pessoa, mesmo sem

vínculo com ONGs ou abrigos, possa contribuir ativamente com a causa animal. Diferente das plataformas existentes, que em sua maioria são voltadas exclusivamente para organizações formalizadas ou mantêm controle centralizado sobre as postagens, o sistema aqui desenvolvido abre espaço para a participação da comunidade em geral, facilitando o registro de ocorrências de pets perdidos, resgatados ou disponíveis para adoção. Além disso, ao adotar um processo de moderação por administradores autenticados, garante-se a segurança e a veracidade das informações publicadas, promovendo maior confiabilidade ao usuário final.

A atenção à usabilidade e à responsividade também é importante, tornando a ferramenta acessível em diferentes dispositivos e facilitando o engajamento de voluntários em qualquer contexto. Dessa forma, o projeto não apenas complementa as iniciativas já existentes, mas amplia seu alcance social, fortalecendo a rede de proteção animal por meio da tecnologia e da colaboração comunitária.

3. Metodologia

O sistema proposto neste trabalho corresponde, portanto, a uma aplicação web, que utiliza JavaScript no front-end - isto é, na interface acessada pelo usuário - e conta com uma API (Application Programming Interface) C# como back-end, esta por sua vez, seguindo os princípios REST (Representational State Transfer).

Para Souza (2019), o front-end é a parte do site visível e interativa para o usuário, responsável por elementos estéticos e interações, enquanto o back-end é a parte que o usuário não vê, mas que garante a funcionalidade do site, processando ações e resolvendo problemas técnicos. Ambos são essenciais para a experiência e o funcionamento do site.

De acordo com Lima (2020), REST é um modelo de arquitetura que define diretrizes para comunicação em sistemas distribuídos utilizando protocolos da web, especialmente HTTP (Hypertext Transfer Protocol), sem a necessidade de protocolos complexos como SOAP (Simple Object Access Protocol). Sua arquitetura segue o princípio de *statelessness* (sem estado), onde cliente e servidor são independentes, comunicando-se por meio de requisições e respostas. O cliente solicita serviços, enquanto o servidor processa e retorna as informações. Métodos HTTP, como GET (solicitação recurso/leitura), POST (suporte a formulários/inserção), PUT (atualização) e DELETE (exclusão), são usados para operações específicas, e respostas incluem códigos de status como 200 (sucesso) ou 404 (recurso não encontrado). REST é amplamente utilizado por sua simplicidade e eficiência na troca de dados entre cliente e servidor.

A API desenvolvida para back-end da aplicação web foi implementada utilizando a linguagem de programação orientada a objetos C#. Através dessa API, o front-end fará as requisições necessárias para inserção de novas postagens na base de dados, consulta de postagens pendentes e aprovadas, aprovação de postagens, login, entre outros recursos do sistema.

Para o front-end, optou-se pelo uso de uma biblioteca do JavaScript, chamada React. Segundo Lopes (2023), o React é amplamente utilizado por sua facilidade de aprendizado e uso, pois se baseia em linguagens como JavaScript e HTML, que são

familiares para iniciantes. Além disso, permite a criação de interfaces de forma ágil com o uso de componentes reutilizáveis, que podem ser aplicados em diferentes projetos sem a necessidade de escrever código do zero.

Em relação ao armazenamento de dados, foi utilizado o banco de dados relacional Microsoft SQL Server, sendo as principais informações a serem gerenciadas: administrador e postagens.

A Figura 4 representa as duas tabelas do sistema: a primeira, referente às postagens realizadas no sistema e a segunda relacionada aos administradores, responsáveis pela aprovação dessas postagens.

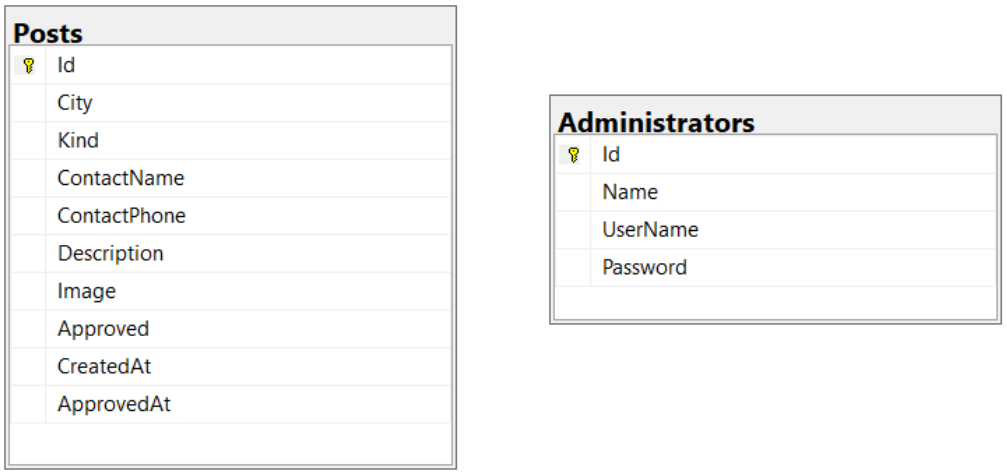

Figura 4. Representação das tabelas

Fonte: Autoria Própria

Para o projeto Patinhas da Serra, o sistema de gerenciamento de dados escolhido foi o Microsoft SQL Server. A escolha do Microsoft SQL Server neste trabalho, mesmo com apenas duas tabelas simples e sem relacionamentos, é justificada por suas funcionalidades robustas e pela flexibilidade na realização de consultas, como destacado por Taylor (2025). Segundo a autora, uma das principais vantagens do SQL Server é sua capacidade de permitir consultas flexíveis e eficientes, apoiando a análise de dados por meio de operações como *joins*, subconsultas e instruções condicionais, o que garante extração de informações específicas mesmo em grandes volumes de dados. Assim, optar pelo SQL Server proporciona uma base sólida, confiável e escalável, adequada não apenas para o cenário atual, mas também para possíveis evoluções futuras do projeto.

4. Desenvolvimento

4.1. Análise

Segundo Sommerville (2011), os requisitos de um sistema descrevem o que ele deve fazer, os serviços que oferece e as restrições ao seu funcionamento, refletindo as necessidades dos clientes. O processo de identificar, analisar, documentar e validar esses

requisitos é conhecido como engenharia de requisitos. Eles são geralmente classificados em dois tipos: requisitos funcionais, que definem os serviços que o sistema deve fornecer e como deve reagir a certas situações; e requisitos não funcionais, que impõem restrições ao sistema, como desempenho, normas e processos de desenvolvimento, aplicando-se frequentemente ao sistema como um todo.

Rosa (2021) destaca que, para compreensão do funcionamento e do que deve ser desenvolvido para o sistema, listam-se seus determinados requisitos e regras de negócio, garantindo que o programa cumpra seu objetivo e indicando a forma pela qual esse objetivo será alcançado. Esses requisitos sustentam o desenvolvimento de software, desde o design até a implementação, facilitando a comunicação entre stakeholders e assegurando um produto adequado ao público e contexto.

Quanto aos requisitos funcionais, pode-se dizer que o sistema será projetado para atender tanto usuários gerais quanto administradores, com funcionalidades específicas para cada perfil.

Os usuários gerais terão a capacidade de acessar o banco de dados de pets, visualizando informações detalhadas e imagens de animais perdidos, disponíveis para adoção ou resgatados. Também poderão filtrar as ocorrências por tipo (perdido, adoção ou resgatado) e criar novas postagens por meio de um formulário. Após preencherem os dados, as postagens serão enviadas para aprovação antes de serem exibidas no sistema.

Já os administradores terão acesso a funcionalidades exclusivas, como realizar login no sistema para acessar a área administrativa. Eles poderão aprovar ou recusar postagens de usuários referentes a qualquer tipo de ocorrência. Além disso, terão ferramentas para consultar a lista de postagens pendentes, cadastrar novos administradores, bem como editar ou remover administradores já existentes.

Em relação aos requisitos não funcionais da aplicação, cabe mencionar que o sistema deve garantir alto desempenho, oferecendo tempos de resposta rápidos ao buscar e exibir informações dos pets. A usabilidade é outro ponto crucial: tanto usuários gerais quanto administradores devem encontrar uma interface intuitiva e amigável, e o formulário para criação de postagens deve ser acessível em dispositivos móveis. Em termos de segurança, todas as funcionalidades administrativas estarão protegidas por autenticação obrigatória, permitindo acesso apenas a administradores autenticados. Para maior confiabilidade, o sistema deverá estar disponível 24/7, assegurando que os usuários possam consultar informações sobre os pets a qualquer momento.

Para Clemente (2024), regras de negócio são diretrizes que definem como um software deve operar para atender às necessidades de uma organização ou mercado. Elas influenciam o comportamento, operações e restrições do sistema, moldando o processamento de dados e execução de funcionalidades. Tais regras refletem políticas, procedimentos e condições essenciais para garantir consistência, eficiência e conformidade nas operações organizacionais.

O sistema adota como regras de negócio: todas as postagens de pets perdidos, para adoção ou resgatados passarão por aprovação de um administrador antes de serem publicadas, garantindo controle do conteúdo; postagens aprovadas serão exibidas no sistema, enquanto as recusadas serão deletadas; o gerenciamento de administradores é

restrito: somente administradores autenticados poderão cadastrar, editar ou remover outros administradores; apenas administradores terão autorização para aprovar ou recusar postagens e acessar funcionalidades administrativas. O sistema bloqueará qualquer tentativa de acesso não autorizado às áreas administrativas por meio de autenticação obrigatória.

Com o objetivo de demonstrar o funcionamento do sistema, esboçou-se o diagrama de caso de uso, representado na Figura 5. O diagrama conta com dois atores: usuário e administrador. O usuário é o ator que utiliza do sistema para enviar as postagens para publicação e realizar consultas, enquanto o administrador as aprova.

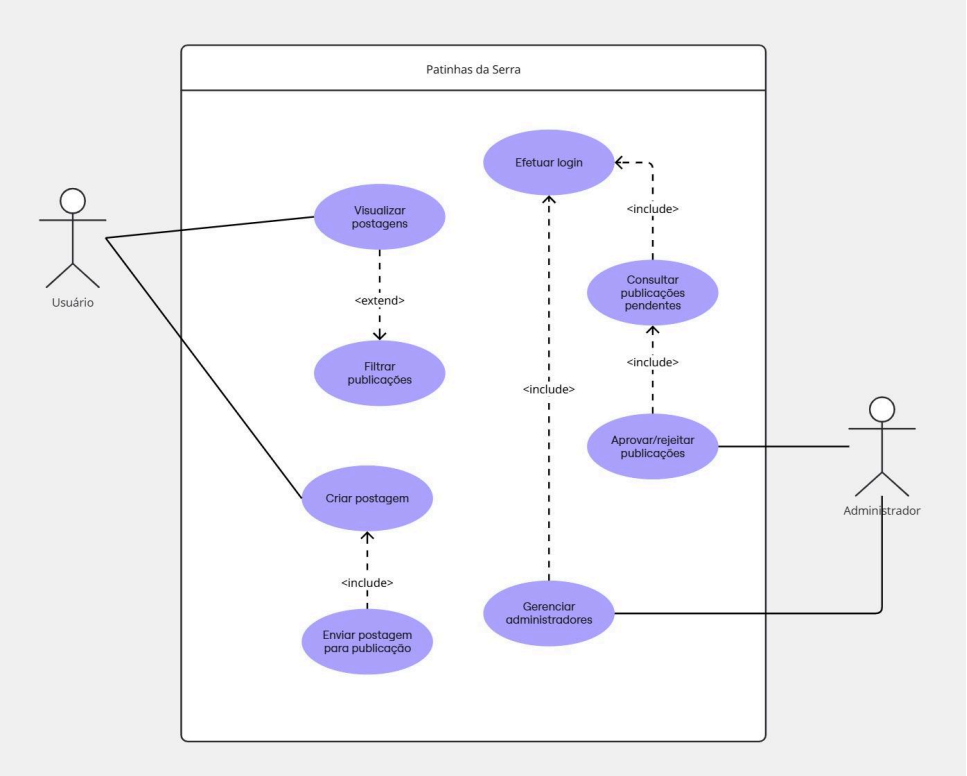

Figura 5. Diagrama de Caso de Uso

Fonte: Autoria Própria

O diagrama de caso de uso evidencia as funcionalidades principais do sistema Patinhas da Serra, destacando as ações realizadas por usuários e administradores. Usuários podem visualizar e filtrar postagens, além de criar e enviar publicações para aprovação. Já os administradores têm permissões adicionais, como aprovar ou rejeitar publicações, consultar aprovações pendentes e gerenciar outros administradores. O login é obrigatório para realizar ações restritas, e o uso de relacionamentos como *include* e *extend* ajuda a organizar as dependências entre os casos de uso.

4.2. Implementação

4.2.1. Autenticação JWT (JSON Web Token)

Segundo Lima (2021), a autenticação é essencial na maioria das aplicações, pois assegura que apenas usuários registrados e autorizados possam acessar suas informações. Em aplicações web, é comum realizar esse processo por meio de formulários de login, armazenando os dados de autenticação na sessão do navegador.

Ribas (2023) descreve JWT como um padrão amplamente utilizado para autenticação e autorização em aplicações web e mobile, permitindo a troca segura de informações entre partes. Seu funcionamento envolve três etapas principais: primeiro, o servidor gera um token assinado com uma chave secreta após validar as credenciais do usuário; em seguida, esse token é enviado ao cliente, geralmente pelo cabeçalho "Authorization"; por fim, nas requisições seguintes, o servidor verifica a validade do token recebido, conferindo sua assinatura e extraindo os dados necessários para garantir que o usuário tem permissão de acesso. Por ser compacto, seguro e independente de plataforma, o JWT é ideal para aplicações modernas e distribuídas. A figura 6 exemplifica o funcionamento do JWT, na prática.

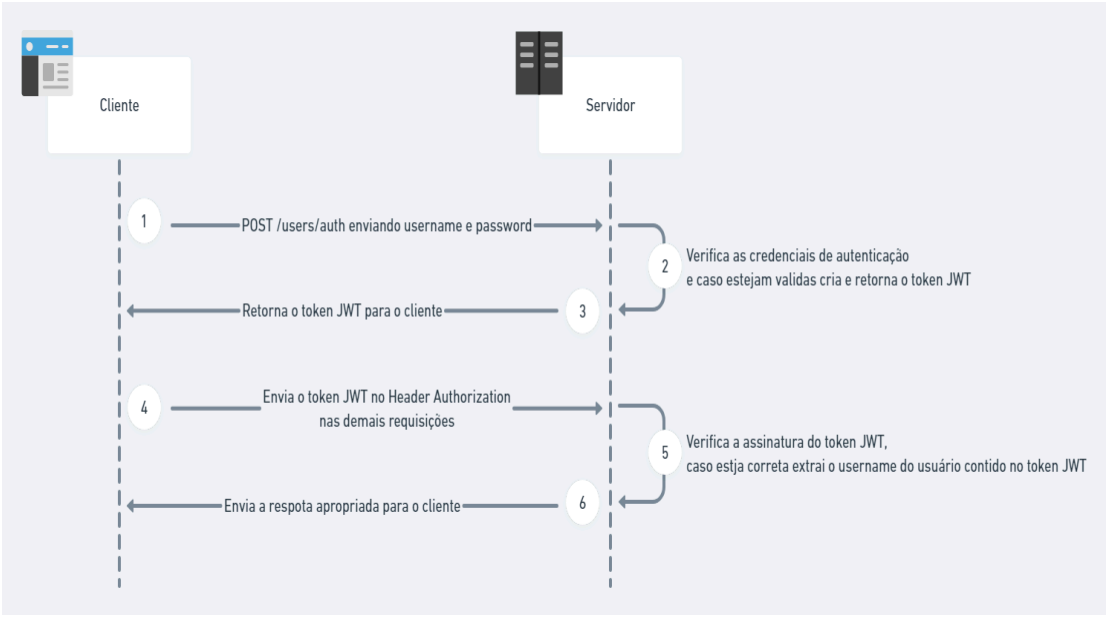

Figura 6. Funcionamento JWT

Fonte: Lima, 2021

Nesse sistema, a funcionalidade de autenticação é iniciada por meio de um *endpoint* do tipo “HttpPost”, representado na Figura 7. Petit (2024) define que um *endpoint* de API é um endereço específico (Uniform Resource Locator ou URL) que indica onde um recurso ou um conjunto de recursos pode ser acessado dentro de uma API. Ele serve como o local onde as requisições são enviadas e por onde o cliente interage com a API.

Utilizando a notação AllowAnonymous, conforme demonstrado na figura abaixo, esse endpoint permite que usuários não autenticados acessem a rota. Através do corpo da requisição, são recebidos o nome de usuário e a senha, utilizados para realizar a autenticação. Internamente, o *endpoint* invoca um método responsável por verificar se o administrador correspondente às informações fornecidas existe na base de dados. Caso as credenciais estejam incorretas, uma exceção é lançada, interrompendo o fluxo e retornando uma resposta de usuário não encontrado.

```
[HttpPost]
[AllowAnonymous]
0 referências
public async Task<ActionResult<dynamic>> AuthenticateAsync([FromBody] LoginRequest loginRequest)
{
    var admin = await _administratorBusiness.FindOneAsync(loginRequest.UserName);
    if (admin == null || admin.Password != loginRequest.Password)
        throw new NotFoundException();

    var token = _tokenService.GenerateToken(admin);

    return new
    {
        loginRequest.UserName,
        token
    };
}
```

Figura 7. Endpoint de login

Fonte: Autoria Própria

Quando o nome de usuário e a senha informados estão corretos, o sistema cria um "token", que funciona como uma identidade digital temporária. Esse token é o que permite que a pessoa continue navegando no sistema sem precisar digitar a senha a todo momento.

A criação desse token é feita por um serviço especial, chamado TokenService. Esse serviço usa informações secretas guardadas com segurança (em um arquivo de configurações do sistema) para garantir que o token seja confiável e não possa ser falsificado, fazendo com que o sistema mantenha a segurança dos dados.

O método GenerateToken (Figura 8) tem como principal função gerar um token de acesso, que funciona como uma identificação digital temporária do usuário autenticado. Esse token é criado utilizando a biblioteca JwtSecurityTokenHandler, uma ferramenta amplamente usada para a criação de tokens no padrão JWT, compatível com aplicações modernas baseadas em APIs.

O código-fonte responsável pela geração do token está representado na Figura 8.

```
public class TokenService(IOptionsMonitor<Settings> settingsMonitor)
{
    private readonly Settings _settings = settingsMonitor.CurrentValue;

    1 referência
    public string GenerateToken(Administrator admin)
    {
        var tokenHandler = new JwtSecurityTokenHandler();
        byte[] key = Encoding.ASCII.GetBytes(_settings.Secret);

        var tokenDescriptor = new SecurityTokenDescriptor
        {
            Subject = new ClaimsIdentity(
            [
                new Claim(ClaimTypes.Name, admin.UserName),
                new Claim(ClaimTypes.Role, "Admin")
            ]),
            Expires = DateTime.UtcNow.AddHours(1),
            SigningCredentials = new SigningCredentials(
                new SymmetricSecurityKey(key),
                SecurityAlgorithms.HmacSha256Signature)
        };
        var token = tokenHandler.CreateToken(tokenDescriptor);
        return tokenHandler.WriteToken(token);
    }
}
```

Figura 8. Código-fonte referente a geração do token

Fonte: Autoria Própria

A primeira etapa do processo consiste em obter uma chave secreta armazenada de forma segura nas configurações do sistema. Essa chave, convertida para um formato apropriado, é utilizada para assinar digitalmente o token, garantindo que ele não seja alterado ou falsificado durante sua transmissão. A assinatura serve como uma proteção contra acessos indevidos.

Em seguida, são definidas as informações que estarão dentro do token, chamadas de claims. Nesse caso, são incluídos o nome do usuário e o papel que ele desempenha no sistema, como "Admin". Essas informações são importantes, pois podem ser utilizadas futuramente para verificar se o usuário tem permissão para acessar determinadas áreas ou funcionalidades da aplicação.

Outro aspecto importante é o tempo de validade do token, configurado para expirar em uma hora. Essa limitação ajuda a manter a segurança, impedindo que o mesmo token seja reutilizado indefinidamente. Após a criação, o token é convertido em uma *string* codificada, ou seja, um texto legível pronto para ser enviado ao cliente.

Por fim, esse token é retornado ao lado do nome de usuário como resposta ao processo de autenticação. A partir desse momento, o cliente deve armazenar esse token e enviá-lo automaticamente no cabeçalho das próximas requisições, evitando a necessidade de reenviar o nome de usuário e senha a cada operação. Esse modelo promove maior segurança, desempenho e compatibilidade com arquiteturas modernas de software.

Com isso, torna-se possível configurar rotas na API que só podem ser acessadas mediante a apresentação do token, como é o caso dos endpoints restritos a administradores. Para isso, utiliza-se a anotação Authorize, conforme demonstrado na Figura 9, que ilustra as rotas de gerenciamento de cadastro de administradores.

```
[HttpGet]
[Authorize]
0 referências
public async Task<ActionResult<IEnumerable<Administrator>>> FindAllAsync()
{
    return Ok(await _business.FindAllAsync());
}


[HttpPost]
[Authorize]
0 referências
public async Task<ActionResult<Administrator>> CreateAsync(CreateAdministratorRequest request)
{
    return Ok(await _business.CreateAsync(request));
}

[HttpPut("{id}")]
[Authorize]
0 referências
public async Task<ActionResult<Administrator>> UpdateAsync([FromBody] UpdateAdministratorRequest request)
{
    return Ok(await _business.UpdateAsync(request));
}
```

Figura 9. Endpoints de gerenciamento de administradores

Fonte: Autoria Própria

A aplicação front-end armazena esse token durante o processo de login e o envia, quando necessário, no cabeçalho da requisição à API, garantindo acesso às rotas protegidas.

4.2.2. Entity Framework (C#)

ORM (Mapeamento Objeto-Relacional) é uma técnica usada no desenvolvimento de software para facilitar a integração entre o código da aplicação e bancos de dados relacionais. Em vez de escrever comandos SQL diretamente, o desenvolvedor trabalha com objetos na linguagem de programação utilizada e o ORM se encarrega de traduzir essas operações para o banco de dados. Isso torna o código mais limpo, intuitivo e fácil de manter. Além disso, muitos ORMs oferecem funcionalidades adicionais, como controle de transações, *cache* de dados e gerenciamento de relacionamentos entre tabelas, o que agiliza o desenvolvimento de sistemas mais complexos (J., 2023).

Para o *website* Patinhas da Serra, optou-se por utilizar o ORM Entity Framework. O Entity Framework é uma ferramenta da Microsoft que facilita o desenvolvimento de aplicações ao permitir que os programadores trabalhem com modelos de domínio — representações conceituais de entidades e suas relações — em vez de lidar diretamente com comandos SQL ou estruturas físicas de banco de dados. Ele atua como uma ponte entre o código da aplicação e o banco de dados, traduzindo automaticamente as operações feitas sobre os objetos do modelo para comandos específicos da base de dados. Com isso, o Entity Framework elimina a necessidade de

dependência direta do banco e torna o processo de desenvolvimento mais simples, organizado e desacoplado da estrutura física do armazenamento de dados (Microsoft, 2023).

A partir disso, pode-se dizer que o Entity Framework simplifica significativamente o trabalho com bancos de dados em aplicações .NET. Uma das utilizações dele no projeto foi para realizar o mapeamento das classes C# para tabelas no banco SQL, evitando a necessidade de escrever SQL manualmente. A figura 10 demonstra a implementação que realiza esse mapeamento das entidades na base de dados configurada no projeto.

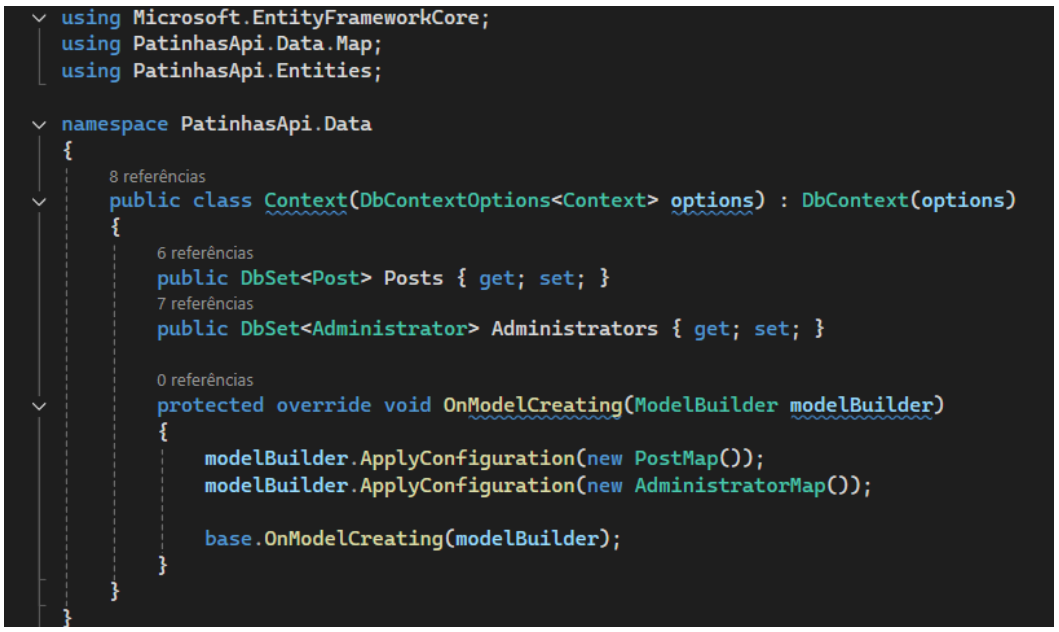

Figura 10. Definição do contexto

Fonte: Autoria Própria

Na classe Context, são definidos os conjuntos de entidades (DbSet<Post> e DbSet<Administrator>) que representam as tabelas no banco de dados (tabela de postagem e tabela de administrador, respectivamente). Além disso, o mapeamento das entidades é configurado através do método OnModelCreating, utilizando classes de configuração para definir como cada entidade será representada na base, determinando quais campos são de preenchimento obrigatório e campos de auto incremento, a exemplo do identificador.

Realizada a configuração do contexto e as entidades, é necessário gerar uma migração que representa as alterações necessárias no banco de dados. Isso é possível através do uso do comando "*dotnet ef migrations add InitialCreate*" em um console. Após a geração da migração, utiliza-se o comando "*dotnet ef database update*", esse, por sua vez, cria automaticamente as tabelas no banco de dados (caso ainda não existam), com base na configuração feita e na *string* de conexão definida no projeto.

Além de facilitar a criação da base de dados, o Entity Framework também torna

extremamente simples o gerenciamento de dados. Na Figura 11, é possível verificar que a classe AdministratorRepository utiliza o contexto para adicionar e remover registros da tabela Administrators. Tudo isso é feito sem a necessidade de escrever instruções SQL, bastando trabalhar diretamente com objetos C#.

```
2 referências
public class AdministratorRepository : IAdministratorRepository
{
    private readonly Context _context;

    0 referências
    public AdministratorRepository(Context context)
    {
        _context = context;
    }


    2 referências
    public async Task<Administrator> CreateAsync(Administrator entity)
    {
        await _context.Administrators.AddAsync(entity);
        await _context.SaveChangesAsync();

        return entity;
    }


    2 referências
    public async Task DeleteAsync(int id)
    {
        var administrator = await _context.Administrators.FindAsync(id) ?? throw new KeyNotFoundException();
        try
        {
            _context.Administrators.Remove(administrator);
            await _context.SaveChangesAsync();
        }
        catch (Exception)
        {
            throw;
        }
    }
```

Figura 11. Criação e deleção de administrador

Fonte: Autoria Própria

4.2.3. Design Responsivo

PM3 (2024) declara que o design responsivo é uma abordagem que adapta automaticamente o layout de um site para diferentes tamanhos e tipos de tela, garantindo que o conteúdo seja acessível e funcional tanto em desktops quanto em dispositivos móveis. Com o crescimento do uso de smartphones para navegação, essa prática se tornou um padrão essencial no desenvolvimento web, substituindo a ideia de ser apenas uma inovação. Ela assegura que todos os usuários tenham uma experiência consistente e satisfatória, independentemente do dispositivo utilizado.

Em relação à importância do design responsivo, Oliveira (2024) também destaca a contribuição significativa para a melhoria da experiência do usuário, proporcionando uma navegação fluida e intuitiva, independentemente do tamanho da tela. Além disso, favorece o SEO (Search Engine Optimization ou Otimização para Mecanismos de Busca), já que o Google prioriza sites responsivos nos resultados de busca, tornando essa abordagem essencial para a visibilidade online. Outro benefício é a redução de custos de manutenção, pois uma única versão do site adaptável elimina a necessidade de gerenciar múltiplas versões. Por fim, o design responsivo aumenta as taxas de conversão

ao facilitar a realização de ações como compras e preenchimento de formulários em qualquer tipo de dispositivo.

Com base nesses pontos destacados pelos autores mencionados acima, é evidente a importância de disponibilizar uma aplicação web que funcione adequadamente em diferentes dispositivos. Considerando que muitos usuários farão o registro de ocorrências envolvendo pets por meio de celulares, é essencial que o sistema permita a realização dessas postagens via dispositivos móveis. Para a equipe administrativa, essa adaptabilidade também é relevante, pois facilita a aprovação ou remoção de publicações de maneira prática e rápida pelo celular. Dessa forma, garante-se uma experiência completa e eficiente, permitindo que o site cumpra seu propósito independentemente do dispositivo utilizado para acessá-lo. A utilização do React no front-end permitiu desenvolver o site com foco em uma experiência responsiva, adaptando-se de forma eficiente a diferentes tamanhos de tela.

Um exemplo de aplicação de responsividade neste *website* é o design dos *cards*. Cada *card* corresponde às informações de uma postagem, contendo a foto do pet, como pode ser observado na Figura 14. A Figura 12 representa o código JavaScript que gera esse componente.

```
import stylesCard from '../styles/Card.module.css';

function PetCard({ id, imagem, cidade, tipo, descricao, contatoNome, contatoTelefone, onClick }) {
  return (
    <div className={stylesCard.card} onClick={() => onClick({ id, imagem, cidade, tipo, descricao, contatoNome, contatoTelefone })}>
      <img src={imagem} alt="Imagem do pet" />
      <h4>{tipo}</h4>
      <p>
        <span>Cidade:</span> {cidade}
      </p>
      <div>
        <button className={stylesCard.details_button}>
          Detalhes
        </button>
      </div>
    </div>
  );
}

export default PetCard;
```

Figura 12. Componente “PetCard”

Fonte: Autoria Própria

Segundo Corrêa (2023), o CSS, sigla para "Cascading Style Sheets" (em português, "Folhas de Estilo em Cascata"), é a tecnologia responsável pela estilização dos elementos HTML, permitindo o controle visual da página, como cores, fontes, espaçamentos e o posicionamento dos componentes. Essa capacidade torna-se fundamental no desenvolvimento de interfaces responsivas.

A responsividade no CSS desse componente é feita utilizando *media queries*, que são regras condicionais que aplicam estilos diferentes dependendo do tamanho da tela. Isso permite que a interface se adapte automaticamente para oferecer uma melhor experiência em diferentes dispositivos, como computadores, tablets e celulares.

A figura 13 ilustra o CSS correspondente a essas adaptações.

```css
/* Responsividade */
@media (max-width: 1024px) {
  .card {
    width: 280px; /* Mantém largura confortável em tablets */
  }
}

@media (max-width: 768px) {
  .card {
    width: 90%; /* Ocupa mais espaço da tela */
  }

  .card img {
    height: 180px;
  }
}

@media (max-width: 480px) {
  .card {
    width: 90%; /* Quase tela cheia */
  }

  .card img {
    height: 180px;
  }

  .card h4 {
    font-size: 16px;
  }

  .card p {
    font-size: 14px;
  }

  .card_actions a {
    font-size: 14px;
    padding: 8px;
  }
}
```

Figura 13. CSS do componente “PetCard”

Fonte: Autoria Própria

A primeira regra condicional demonstrada na Figura 13 define que, para interfaces com largura máxima de 1024 pixels — um intervalo frequentemente associado a tablets em modo paisagem — a largura do card é ajustada para 280 pixels. Esse valor reduzido visa proporcionar uma organização mais eficiente dos elementos na interface, respeitando os limites físicos do dispositivo.

À medida que a resolução diminui, novas regras são aplicadas. Para telas com até 768 pixels de largura, típicas de tablets em modo retrato ou dispositivos móveis com telas maiores, o card passa a ocupar 90% da largura da janela. Além disso, a altura das imagens é restringida a 180 pixels, evitando que elas dominem visualmente o componente e mantendo a harmonia da disposição dos elementos.

Em dispositivos com largura inferior ou igual a 480 pixels, como smartphones de menor porte, a responsividade é intensificada. A largura do card permanece em 90%, enquanto os tamanhos das fontes nos elementos “h4” e “p” são reduzidos para 16 pixels e 14 pixels, respectivamente. Itens de ação contidos na classe, como botões ou links, também são redimensionados, com redução de tamanho da fonte e do preenchimento interno (*padding*), garantindo sua usabilidade mesmo em telas mais compactas.

5. Resultados

Nesta seção, são apresentados os principais resultados obtidos a partir do desenvolvimento do sistema. São detalhadas funcionalidades implementadas do sistema, bem como a integração entre as tecnologias adotadas. Além disso, discute-se de que forma os objetivos estabelecidos foram alcançados e de que maneira a solução contribui para amenizar a problemática do abandono de animais. A tela inicial permite uma navegação intuitiva. Nela, é exibida a listagem das postagens, bem como opções de filtros, possibilitando que a busca seja feita a partir do tipo da postagem (disponível para adoção, animal encontrado/resgatado, procura-se animal). Também conta com um menu superior, que pode redirecionar o usuário ao formulário de postagem ou ao *login* para administradores. Conforme abordado na seção anterior, o *layout* do sistema prioriza a usabilidade, com elementos visuais claros e acessíveis, garantindo uma boa experiência tanto em dispositivos móveis quanto em desktops. A tela está representada na Figura 14.

Figura 14. Tela Inicial

Fonte: Autoria Própria

A figura 15 demonstra essa mesma tela inicial, na visão de dispositivos móveis.

Figura 15. Tela Inicial em dispositivos móveis

Fonte: Autoria Própria

Cada postagem corresponde a um *card* na tela e, clicando em "Detalhes" no *card*, são apresentadas as demais informações da postagem, conforme descrito na figura 16.

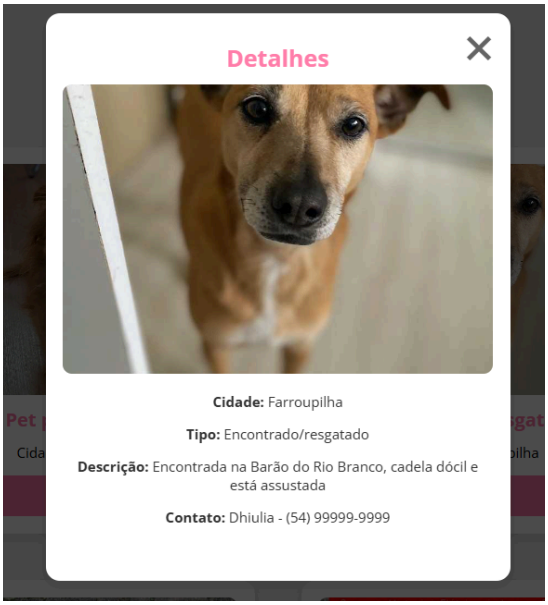

Figura 16. Detalhes da postagem

Fonte: Autoria Própria

O formulário disponibilizado no menu através da opção “Postar” permite que qualquer usuário informe uma ocorrência, submetendo a postagem para aprovação dos administradores. Todos os campos são de preenchimento obrigatório. A figura 17 ilustra o formulário no formato para dispositivos móveis.

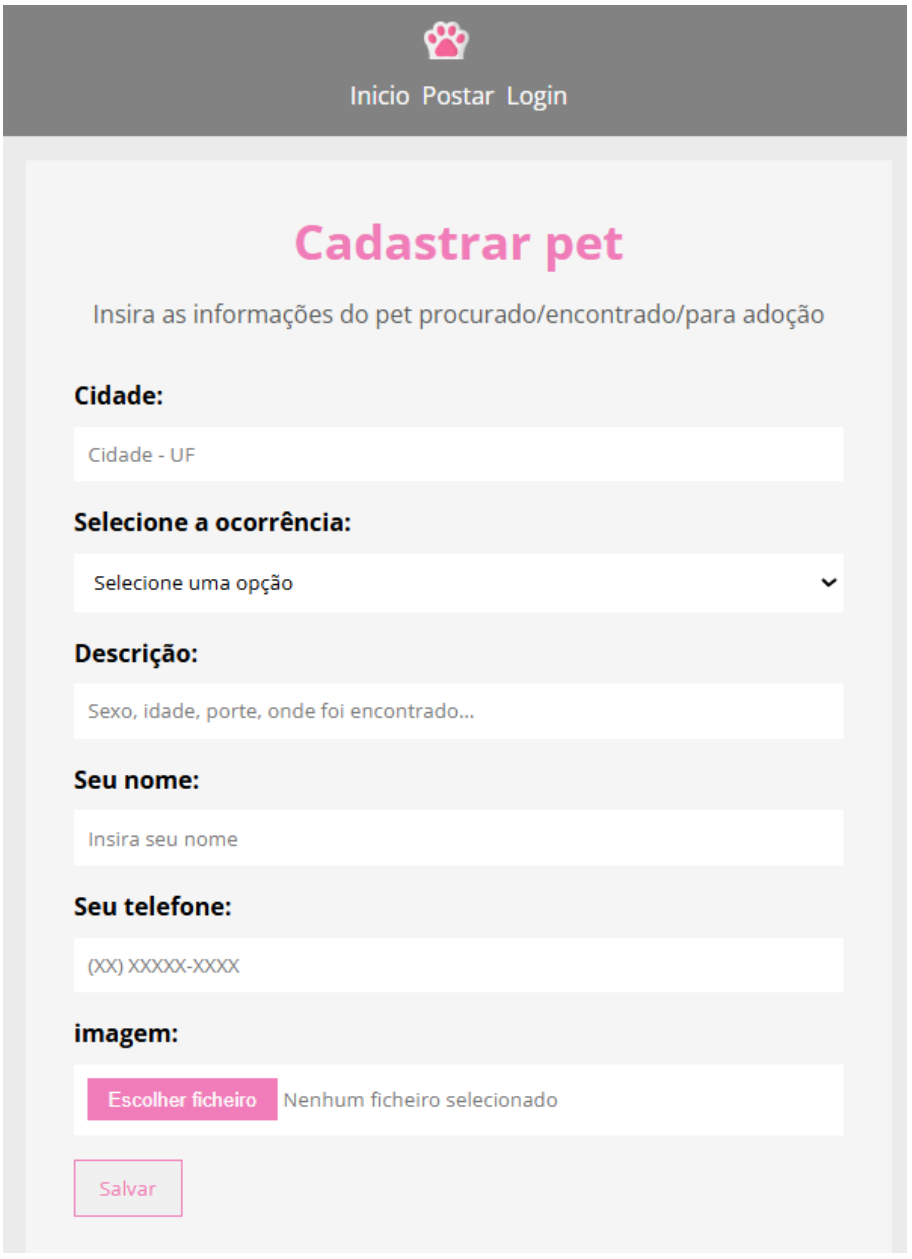

Figura 17. Formulário de postagem

Fonte: Autoria Própria

Já a figura 18 demonstra o mesmo formulário, no *layout* para *desktop*.

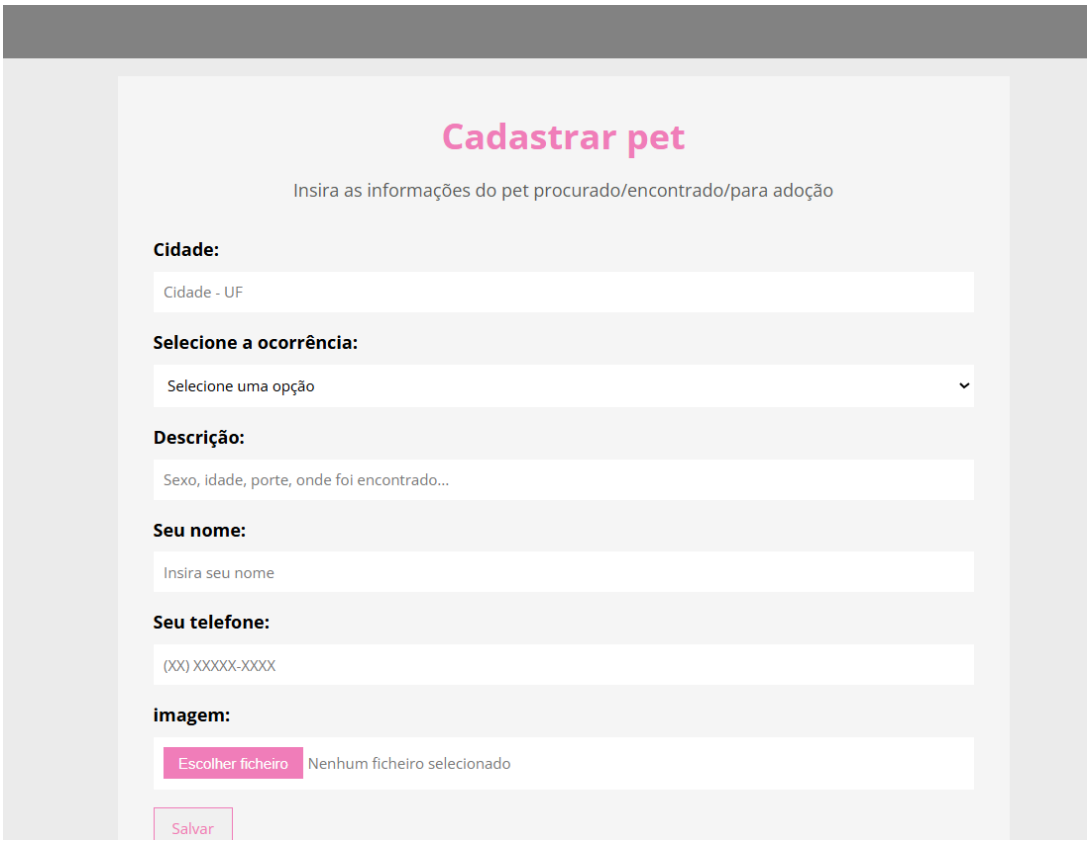

Figura 18. Formulário de postagem para *desktop*

Fonte: Autoria Própria

Ao realizar o login do administrador, novas opções são disponibilizadas no menu superior Figura 19, entre elas Pendentes, Administradores, Sair, permitindo que o administrador visualize as postagens pendentes de aprovação, realize a consulta e gerenciamento dos administradores cadastrados e efetue o logout do sistema, respectivamente.

Figura 19. Menu do administrador

Fonte: Autoria Própria

A tela de publicações pendentes, apresentada na Figura 20 é similar a tela inicial, contando com os mesmos filtros e *cards* de postagem. A diferença está dentro do *card*, que conta com os botões responsáveis por aprovar ou remover a solicitação de postagem.

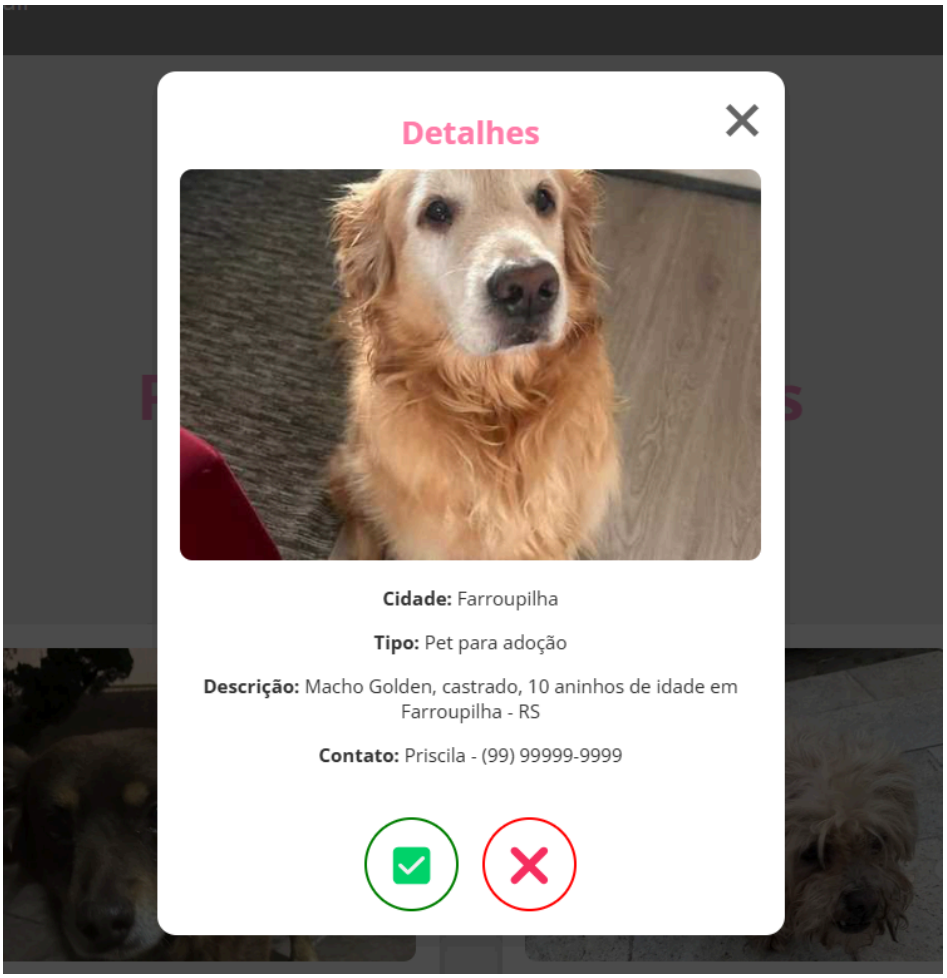

Figura 20. *Card* de publicação pendente

Fonte: Autoria Própria

A tela de administradores (Figura 21) conta com uma tabela informando os campos: nome e nome de usuário de cada administrador, permitindo editar e remover individualmente cada um. Também é possível criar um novo administrador.

A Figura 21 ilustra o formulário destinado ao cadastro.

Figura 21. Cadastro de administrador

Fonte: Autoria Própria

Dessa forma, observa-se que o sistema desenvolvido atende aos requisitos propostos, oferecendo uma solução funcional, intuitiva e responsiva. As funcionalidades implementadas contribuem significativamente para facilitar o processo de divulgação e busca por animais, auxiliando na redução dos impactos causados pelo abandono e promovendo a adoção responsável.

6. Considerações finais

O sistema desenvolvido cumpre com êxito todos os requisitos previamente definidos, oferecendo uma plataforma intuitiva e funcional voltada à divulgação de pets para adoção, encontrados ou desaparecidos em Farroupilha e região. Por meio do formulário, moradores podem registrar ocorrências diretamente, informando dados relevantes como o tipo de caso, descrição, cidade e um telefone para contato. Após o envio, as ocorrências passam por um processo de aprovação por parte dos administradores, garantindo a qualidade e confiabilidade das publicações.

Uma vez aprovadas, as ocorrências se tornam públicas no website, onde usuários podem consultá-las com facilidade. O sistema se destaca por oferecer filtros por tipo de ocorrência e ordenação por data, algo que redes sociais como o Instagram não contemplam de forma estruturada, o que torna a navegação mais eficiente e objetiva.

Como sugestões para melhorias futuras, destacam-se: implementar *hash* nas

senhas utilizando algoritmos específicos e seguros, reforçando a proteção de dados dos usuários; refinar o *design* do modal de detalhes e aplicar melhorias na hierarquia visual dos *cards*, mantendo a identidade da aplicação; adicionar paginação dinâmica com *scroll* infinito e melhoria nas mensagens de retorno ao usuário que atualmente estão genéricas.

Em suma, o sistema alcança seu propósito de forma eficaz, criando uma ponte entre a comunidade e a causa animal, e abre portas para evoluções contínuas tanto em aspectos técnicos quanto de usabilidade.

7. Referências

ADOTE UM FOCINHO. **Quem Somos?** 2023. Disponível em: http://www.adoteumfocinho.com.br/sobre-nos/. Acesso em: 30 abr. 2025.

CARVALHO, Joao Cesar Almeida. **30 milhões de animais estão nas ruas, segundo dados da Organização Mundial da Saúde.** 2024. Disponível em: https://labnoticias.jor.br/2024/01/26/30-milhoes-de-animais-estao-nas-ruas-segundo-dados-da-organizacao-mundial-da-saude-oms/. Acesso em: 26 nov. 2024.

CORRÊA, Amauri Blanco. **Guia da linguagem CSS.** 2023. Disponível em: https://www.treinaweb.com.br/blog/guia-da-linguagem-css. Acesso em: 12 mai. 2025.

CLEMENTE, Paulo. **Regras de Negócios: O que você precisa saber.** 2024. Disponível em: https://blog.rocketseat.com.br/regras-de-negocios-o-que-voce-precisa-saber/. Acesso em: 2 dez. 2024.

CRMVPA. **Abandono animal é crime.** 2024. Disponível em: https://crmvpa.org.br/abandono-animal-e-crime/. Acesso em: 26 nov. 2024.

INSTITUTO AMPARA ANIMAL. **Conheça o Instituto Ampara Animal.** 2025. Disponível em: https://institutoamparanimal.org.br/. Acesso em: 7 abr. 2025.

J., Gabriel. **O que é ORM e para o que serve?** 2023. Disponível em: https://dev.to/gabrielgcj/o-que-e-orm-e-para-o-que-serve-4dgl. Acesso em: 8 abr. 2025.

LIMA, Guilherme. **REST: Conceito e fundamentos.** 2020. Disponível em: https://www.alura.com.br/artigos/rest-conceito-e-fundamentos?srsltid=AfmBOorB38URoQEFFPvRzttYa3Bjcd80u4U0T0VAQPlc7ZvnntccsUwZ. Acesso em: 27 nov. 2024.

LIMA, Cleyson**. Fluxo de autenticação baseado em JWT.** 2021. Disponível em: https://www.treinaweb.com.br/blog/fluxo-de-autenticacao-baseado-em-jwt. Acesso em: 8 abr. 2025.

LOPES, Michele. **React: o que é e como funciona.** 2023. Disponível em: https://ebaconline.com.br/blog/react-o-que-e-como-funciona. Acesso em: 27 nov. 2024.

MICROSOFT. **Visão geral do Entity Framework**. 2023. Disponível em: https://learn.microsoft.com/pt-br/dotnet/framework/data/adonet/ef/overview. Acesso em: 8 abr. 2025.

OLIVEIRA, Valquiria Pires de**. Design responsivo: a importância para sites modernos**. 2024. Disponível em:

https://velx.com.br/insights/design-responsivo-a-importancia-para-sites-modernos/. Acesso em: 21 abr. 2025.

PETFINDER. **about.** Disponível em: https://www.petfinder.ong.br/about/. Acesso em: 7 abr. 2025.

PETIT, Alicia. **Um guia abrangente para endpoints de API.** 2024. Disponível em: https://www.getambassador.io/blog/guide-api-endpoints. Acesso em: 13 abr. 2025.

PM3. **Design responsivo: o que é, pilares e por que é tão importante?** 2024. Disponível em: https://pm3.com.br/blog/design-responsivo/. Acesso em: 21 abr. 2025.

QUEIROZ, Francisca Karolina do Nascimento et. al. **Abandono de animais no Brasil: consequências geradas à sociedade**. 2019. Disponível em: https://periodicos.ufam.edu.br/index.php/resbam/article/view/6615/6303. Acesso em: 26 nov. 2024.

RIBAS, João. **JWT – O que é? Como funciona**. DIO, 28 fev. 2023. Disponível em: https://www.dio.me/articles/jwt-o-que-e-como-funciona. Acesso em: 8 abr. 2025.

SILVA, Anita de Souza. **Índice de Abandono no Brasil.** 2024. Disponível em: https://institutomvc.org.br/site/index.php/2024/04/04/indice-de-abandono-no-brasil/. Acesso em: 26 nov. 2024.

SOMMERVILLE, Ian. **Engenharia de software**. 9. ed. São Paulo: Pearson Education do Brasil, 2011. p. 57-59.

SOUZA, Ivan de. **O que é front-end e back-end e quais as suas diferenças?** 2019. Disponível em: https://rockcontent.com/br/blog/front-end-e-back-end/. Acesso em: 27 nov. 2024.

TAYLOR, Eliza. **What are the Advantages of SQL?** 2025. Disponível em: https://www.theknowledgeacademy.com/blog/advantages-of-sql/. Acesso em: 09 jul. 2025.

WK Technology. **Aplicações Web – O que são e como funcionam?** 2024. Disponível em: https://wktechnology.com.br/aplicacoes-web-o-que-sao-e-como-funcionam/. Acesso em: 26 nov. 2024.