

SISTEMA WEB DE ADOÇÃO: QUATRO PATAS

Rodrigo Reis Pereira de Santana, Samara Vieira Brandão Nunes

**Instituto Federal de Educação Ciência e Tecnologia da Bahia(IFBA) - Campus
Barreiras; Técnico em Informática**

rodrigo.reis5667@gmail.com, samarabrandao17@gmail.com

Resumo. *No cenário atual, onde a tecnologia permeia nossas vidas, a adoção de animais se torna uma ação nobre, mas desafiadora. Nesse contexto, a criação do sistema web 4 Patas surge como uma resposta eficaz e ética para combater os desafios enfrentados por abrigos e animais em busca de um lar. Este sistema centraliza informações, facilitando a conexão entre adotantes e abrigos. Portanto, exploraremos o desenvolvimento e a aplicação desse sistema, destacando como ele simplifica o processo de adoção e promove uma cultura de responsabilidade e empatia.*

Abstract. *In the current scenario, where technology permeates our lives, adopting animals becomes a noble but challenging action. In this context, the creation of the 4 Patas web system appears as an effective and ethical response to combat the challenges faced by shelters and animals looking for a home. This system centralizes information, facilitating the connection between adopters and shelters. Therefore, we will explore the development and application of this system, highlighting how it simplifies the adoption process and promotes a culture of responsibility and empathy.*

1. Introdução

O abandono de animais é uma questão crescente em muitas regiões do mundo. Segundo dados recentes da Organização Mundial da Saúde (OMS), só no Brasil, estima-se que mais de 30 milhões de animais, entre cães e gatos, estejam em situação de abandono (SEMAD, 2020). Segundo Winkler, esta situação pode ser provocada por alguns fatores, tais como a facilidade de reprodução dos animais, abandono por parte dos responsáveis ou criação de animais para a reprodução e comercialização (WINKLER, 2019). Esse número reflete uma realidade alarmante, onde a superlotação de abrigos e a falta de adoção de animais são problemas persistentes. Os abrigos, que têm a missão de cuidar

temporariamente desses animais, frequentemente não possuem recursos suficientes, tanto em termos de espaço quanto de orçamento, o que agrava ainda mais a situação. Neste contexto, a adoção de animais torna-se uma solução essencial, pois oferece a oportunidade de proporcionar lares para esses animais, além de aliviar a pressão nos abrigos.

O uso das redes sociais como meio de divulgação foi extremamente importante para as organizações de abrigo e canil conseguirem adotantes dos animais abandonados (COSTA, 2021). Por meio das redes sociais é possível construir uma enorme rede de apoio, utilizando grupos no Facebook, perfis no Instagram e Twitter. No entanto, as mídias sociais disponíveis não abordam alguns pontos importantes como facilidade na criação de filtros para busca dos animais, acesso fácil e intuitivo, acompanhamento do processo de adoção e facilidade no cadastro de novos animais pelos usuários da aplicação (COSTA, 2021).

Desta forma, o processo de adoção enfrenta uma série de desafios que dificultam a conexão entre adotantes e animais disponíveis. A falta de uma plataforma centralizada que permita visualizar os animais disponíveis para adoção em diferentes abrigos, a ausência de informações e atualizações sobre os animais, e a dificuldade de encontrar um processo ágil de comunicação entre abrigos e adotantes são alguns dos principais obstáculos. Potenciais adotantes, que muitas vezes estão dispostos a oferecer um lar para esses animais, acabam desestimulados pela falta de organização e pela complexidade em obter informações sobre o seu possível pet.

A tecnologia pode oferecer soluções eficazes para mitigar essas barreiras. O desenvolvimento de um sistema web de adoção de animais surge como uma solução para centralizar e organizar as informações, facilitando a busca e visualização de animais disponíveis para adoção. Ao conectar adotantes e abrigos de forma mais direta, esse tipo de sistema pode reduzir significativamente os obstáculos enfrentados no processo de adoção, proporcionando um ambiente acessível e eficiente. Além disso, ferramentas de busca e filtros específicos permitem que os adotantes encontrem animais que correspondam às suas preferências e estilos de vida, como idade, porte, detalhes do adotado e outras condições específicas.

O sistema "4 Patas" é uma proposta tecnológica que visa conectar abrigos e adotantes de maneira eficiente. O sistema web oferece ferramentas de busca intuitivas, filtros avançados, e informações detalhadas sobre os animais. Dessa forma, o "4 Patas"

tem como objetivo não apenas facilitar o processo de adoção, mas também incentivar a prática de adoção responsável, oferecendo uma interface clara e acessível tanto para os abrigos quanto para os adotantes. O sistema visa, assim, contribuir para a solução de um problema social importante, promovendo o bem-estar animal e a redução do número de animais abandonados.

1.1. Objetivos

O objetivo principal deste trabalho é desenvolver um sistema web que facilite o processo de adoção de animais domésticos, conectando de maneira eficiente canis, abrigos especializados e a população em geral. A plataforma visa centralizar a divulgação de animais disponíveis para adoção, tornando o processo mais ágil e acessível tanto para os abrigos quanto para os potenciais adotantes, ao mesmo tempo que reduz a burocracia e os obstáculos tradicionais encontrados nesse processo. Além disso, o sistema visa melhorar a experiência do usuário ao oferecer uma interface intuitiva e amigável, incentivando a adoção responsável e a correspondência adequada entre adotantes e animais.

Entre os objetivos específicos, destaca-se o aumento da visibilidade dos animais por meio de uma plataforma centralizada e acessível, permitindo que mais adotantes conheçam e interajam com abrigos e canis. O sistema facilitará a comunicação entre adotantes e abrigos, oferecendo canais de contato diretos que diminuam as barreiras de interação.

Além disso, será implementada uma funcionalidade de filtragem avançada, permitindo que os adotantes busquem animais com base em critérios específicos como tipo de animal, idade, porte, sexo e status de castração. Outro objetivo importante é o armazenamento seguro e eficiente das informações no banco de dados, garantindo que todo o histórico de adoções, perfis de animais e dados dos abrigos estejam organizados e facilmente acessíveis para consultas futuras.

O sistema também irá registrar o histórico de interações entre adotantes e abrigos, mantendo um controle detalhado sobre as solicitações de adoção, aprovadas ou rejeitadas, além de permitir a consulta ao histórico de animais adotados ou ainda disponíveis. Dessa forma, promove-se a transparência e a segurança no processo de adoção, além de fornecer uma ferramenta poderosa para o gerenciamento de informações por parte dos abrigos e adotantes.

2. Referencial Teórico

Atualmente, a tecnologia está cada vez mais presente no nosso cotidiano, integrando-se de forma natural às atividades diárias e tornando-se uma grande aliada em diversas tarefas. Ela não apenas facilita processos habituais, mas também transforma a maneira como interagimos e resolvemos problemas. Nesse sentido, os sistemas web têm ganhado cada vez mais espaço, pois permitem a simplificação e automação de diversos serviços, tornando-os mais acessíveis, rápidos e eficazes.

Quando falamos em sistema web, estamos nos referindo a uma estrutura tecnológica que opera a partir de servidores e é acessada por meio de navegadores da internet, sem a necessidade de instalação de software adicional no dispositivo do usuário. Segundo Sommerville (2015, p.517), "os sistemas web são aplicações distribuídas, onde diferentes componentes de software são executados em computadores diferentes, mas cooperam para fornecer uma solução integrada", o que reforça a flexibilidade e a escalabilidade dessas plataformas, já que podem ser utilizados em qualquer dispositivo com acesso à internet, como smartphones, tablets ou computadores.

De acordo com Pressman (2011), os sistemas web são compostos por diversas camadas, sendo as mais fundamentais o servidor web, responsável por responder às requisições do cliente; o banco de dados, que organiza e armazena os dados; o back-end, que gerencia a lógica do sistema e a conexão com o banco de dados; e o front-end, que representa a interface gráfica que o usuário acessa por meio do navegador. Portanto, compreender esses elementos e como eles interagem entre si é crucial para o desenvolvimento e entendimento de qualquer sistema web, pois cada parte desempenha um papel essencial na eficiência, segurança e funcionalidade da aplicação.

O servidor web, muitas vezes considerado o coração do sistema, é o responsável por processar todas as solicitações feitas pelo usuário, garantindo que os dados sejam entregues de maneira eficiente e segura. Ele atua como uma ponte, conectando os dados armazenados no banco ao front-end, que é a interface visual com a qual os usuários interagem. Essa interação não apenas permite que a aplicação web funcione de maneira contínua e estável, mas também transforma a experiência do usuário em algo intuitivo e acessível.

Quando um usuário faz uma solicitação, como o carregamento de uma página ou o envio de um formulário, essa requisição é enviada ao servidor, que a processa e devolve a resposta apropriada ao navegador, tudo em questão de milissegundos. Essa comunicação entre cliente e servidor segue o modelo de arquitetura cliente-servidor, onde, segundo Kurose (2013), o cliente, representado pelo navegador, envia requisições ao servidor através de protocolos como HTTP ou HTTPS, e ele responde com o conteúdo solicitado, possibilitando uma troca de informações rápida e eficaz.

O papel do servidor se torna ainda mais relevante quando consideramos sua interação com o banco de dados, que é responsável por armazenar, gerenciar e fornecer acesso às informações cruciais para o funcionamento da aplicação. Ao organizar os dados de maneira lógica e estruturada, ele facilita tanto a recuperação quanto a manipulação das informações. Como apontam Date e Darwen (2016, p. 34), "um banco de dados é um sistema que permite a gestão de informações, proporcionando mecanismos para armazená-las de maneira estruturada e acessível". Dessa forma, o banco de dados atua como um repositório central, mantendo todas as informações relevantes do sistema disponíveis de forma organizada.

É válido ressaltar que a escolha do modelo de banco de dados é determinante para a eficiência do sistema. Bancos de dados relacionais, como MySQL e PostgreSQL, organizam os dados em tabelas, o que possibilita a realização de operações complexas por meio de consultas SQL. Por outro lado, bancos de dados NoSQL, como MongoDB, foram concebidos para lidar com grandes volumes de dados não estruturados, oferecendo maior flexibilidade e escalabilidade. Essa variedade de modelos permite aos desenvolvedores optar pela solução que melhor atende às necessidades específicas da aplicação.

Desse modo, a integração entre o banco de dados e o restante da aplicação é crucial para garantir a integridade e a segurança dos dados. Nesse cenário, o back-end tem um papel essencial, pois ele gerencia a lógica da aplicação e estabelece a conexão com o banco de dados. Cada operação realizada pelo usuário, como a criação de uma conta, login ou a atualização de informações, não apenas demanda o armazenamento e a recuperação eficiente dos dados, mas também envolve a aplicação de regras e validações que asseguram a qualidade e a conformidade das informações.

De acordo com Pereira (2019, p. 45), o back-end "é responsável por garantir que os dados fluam corretamente entre o usuário e o banco de dados, permitindo que as

informações sejam armazenadas, recuperadas e processadas de maneira eficiente e segura”. Essa definição destaca a função central do back-end na estrutura de um sistema web, já que ele lida diretamente com a lógica da aplicação e coordena todas as operações que ocorrem nos bastidores, invisíveis ao usuário final. Para realizar essas tarefas, o back-end utiliza diversas linguagens de programação que seguem diferentes paradigmas, como PHP, Python e Node.js.

Entre as linguagens mais usadas no desenvolvimento de sistemas web, o PHP merece destaque por ser uma das mais populares para back-end. Sua sintaxe simples e sua integração direta com servidores web fazem dele uma escolha eficiente para a criação de sistemas dinâmicos. Embora outras linguagens, como Python e Node.js, também sejam amplamente utilizadas, o PHP é conhecido por ser especialmente vantajoso quando se busca desenvolver sistemas baseados na programação orientada a objetos. Isso porque ele oferece suporte nativo a esse paradigma, permitindo a criação de classes, herança e encapsulamento de maneira clara e acessível.

Ao falar em paradigmas de programação, estamos nos referindo a forma como as soluções são construídas. Um exemplo é a programação estruturada, que organiza o código de forma linear e sequencial. Entretanto, para sistemas mais complexos, como as aplicações web, o paradigma de programação orientada a objetos (OO) se destaca. Isso porque, esse modelo facilita a manutenção, reutilização e a expansão do código ao longo do tempo, o que é fundamental em projetos de longo prazo.

A programação orientada a objetos organiza o software em "objetos", que são como pequenas unidades que combinam dados e ações. Esses objetos são criados a partir de "classes", que funcionam como moldes que definem as características e os comportamentos das entidades do sistema. Isso torna o código mais fácil de entender e ajustar, além de reduzir a repetição. Ao encapsular dados e funções dentro de objetos, fica mais simples criar, manter e melhorar funcionalidades como autenticação de usuários ou controle de acesso de forma organizada e eficiente.

O front-end, por sua vez, desempenha uma função igualmente vital, porém visível, no funcionamento de um sistema web, pois é a camada com a qual o usuário final interage diretamente. Essa camada é responsável por criar a interface visual e garantir que os dados fornecidos pelo back-end sejam apresentados de forma clara, acessível e intuitiva. Para isso, o front-end utiliza três principais linguagens de

programação: HTML, CSS e JavaScript, que se complementam para criar páginas interativas e visualmente atraentes.

O HTML (HyperText Markup Language) funciona como a base de qualquer página web, sendo responsável pela organização e disposição dos elementos. Isso porque, ele define a estrutura inicial, usando tags específicas para criar e organizar cabeçalhos, parágrafos, links e imagens de maneira hierárquica e lógica, o que facilita a navegação e a acessibilidade. Já o CSS (Cascading Style Sheets), por sua vez, é responsável pela definição do estilo e do layout da página. Através dele que cores, fontes, espaçamentos e layouts são determinados, garantindo que a interface não seja apenas funcional, mas também chamativa e bonita.

Por fim, o JavaScript, sendo a linguagem de programação que adiciona interatividade, permite que os usuários realizem ações dinâmicas na página, como cliques em botões, rolagem automática, preenchimento de formulários, ou mesmo animações complexas, melhorando significativamente a interatividade da interface. Portanto, essas três linguagens trabalham de maneira integrada, com o HTML fornecendo a estrutura, o CSS o estilo, e o JavaScript a interação, criando um ambiente rico e interativo.

No entanto, vale ressaltar que a eficácia do front-end, vai além de sua função estética, sendo crucial para a usabilidade e o desempenho do sistema como um todo. Uma interface bem projetada é intuitiva, fácil de navegar e proporciona uma interação sem frustrações para o usuário. Como observa Silva (2020, p.102), “uma boa interface de usuário é aquela que consegue guiar o usuário sem que ele perceba, proporcionando uma experiência fluida e agradável”. Isso destaca a importância de um design centrado no usuário, que não apenas atende às suas expectativas, mas também melhora sua experiência geral com o sistema, facilitando a execução de tarefas. Assim, o front-end, aliado a boas práticas de design, contribui diretamente para o sucesso de uma aplicação web.

Logo, a essência dos sistemas web reside na harmonia entre suas diversas camadas, sendo que cada uma desempenha um papel vital na criação de experiências enriquecedoras e funcionais. Levando essa reflexão para o contexto específico de um sistema web de adoção de animais, a tecnologia oferece um meio eficaz de conectar potenciais adotantes com abrigos e animais. A utilização de um servidor, aliado a um banco de dados eficiente e a uma interface de fácil navegação, permite que os usuários

acessem informações detalhadas sobre os animais disponíveis para adoção, facilitando a busca por um novo companheiro.

Além disso, a automação de processos, como o gerenciamento de dados e a comunicação entre abrigos e adotantes, torna o sistema mais ágil e acessível, promovendo a adoção responsável. Dessa forma, um sistema web voltado para adoção de animais vai além de simplificar tarefas; ele desempenha um papel crucial ao conectar pessoas com causas importantes, promovendo o bem-estar animal e ampliando as chances de adoções bem-sucedidas.

3. Proposta

O sistema web “4 Patas” foi desenvolvido para otimizar o processo de adoção de animais em abrigos, conectando adotantes e instituições de forma intuitiva e acessível. Seu objetivo principal é proporcionar uma experiência amigável tanto para os abrigos, que gerenciam seus animais e adoções, quanto para os usuários que desejam adotar. O sistema visa aumentar a visibilidade dos animais disponíveis, simplificar a comunicação entre adotantes e abrigos, e promover uma adoção responsável, analisando cuidadosamente o perfil do adotante antes da aprovação.

A primeira funcionalidade essencial é o cadastro de animais pelos abrigos, como ilustrado na Figura 1. Os responsáveis podem registrar perfis detalhados, incluindo nome, idade, sexo, porte, status de castração e fotos, permitindo que os adotantes tenham uma visão clara de cada animal. A atualização e remoção de perfis são integradas, garantindo que as informações estejam sempre atualizadas e melhorando a gestão interna.



A imagem mostra a interface de usuário para o cadastro de um animal no sistema "4 PATAS". No topo, há uma barra de navegação com o logotipo "4 PATAS" e links para "Minha Conta", "Pedidos de Adoção", "Histórico", "Meus Animais" e "Sair". O formulário principal, intitulado "Cadastrar Animal", contém os seguintes campos: "Nome do Animal" (campo de texto), "dd/mm/aaaa" (campo de data com ícone de calendário), "Sexo" (menu suspenso), "Castrado/Não" (menu suspenso), "Porte" (menu suspenso) e "Fazer upload" (botão para upload de fotos). Abaixo desses campos, há uma área para "Descrição detalhada sobre o animal" com um campo de texto grande. No final do formulário, há um botão laranja "CADASTRAR". Uma seta amarela aponta para o link "Meus Animais" na barra de navegação.

Figura 1. Página de Cadastro de animal.

Para os adotantes, o sistema oferece uma busca eficiente de animais com filtros como espécie, idade, porte e sexo (apontados na Figura 2). Esses filtros facilitam a navegação e ajudam o usuário a encontrar rapidamente o animal que mais se adequa às suas preferências. A galeria de fotos interativa proporciona uma experiência visual agradável, incentivando o interesse pela adoção.

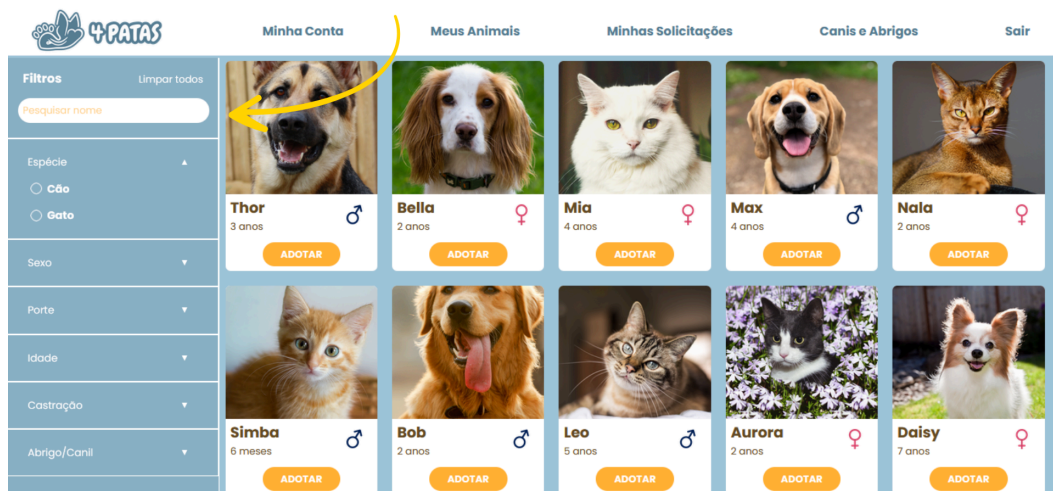


Figura 2. Galeria de animais com filtro de busca .

Além disso, há uma página de detalhes do animal, onde as informações são apresentadas de forma clara. Nela, os adotantes podem visualizar dados específicos e solicitar a adoção através de um botão dedicado. O processo de solicitação é simplificado, com um formulário que coleta as informações necessárias para que o abrigo avalie cada pedido.

A integração entre abrigos e adotantes é prática e segura. Os abrigos podem aprovar ou rejeitar as solicitações com base no perfil do adotante, garantindo adoções bem-sucedidas, que é um dos principais objetivos do “4 Patas”.

O sistema “4 Patas” se destaca pela preocupação com o armazenamento seguro de informações. Todas as interações e transações são registradas, garantindo que o histórico completo de adoções, perfis de animais e dados dos abrigos estejam organizados e facilmente acessíveis para consultas. Esse controle inclui o registro detalhado das solicitações de adoção, sejam elas aprovadas ou rejeitadas, oferecendo uma visão clara do processo para ambos os lados. Os adotantes podem acessar o histórico de animais que adotaram ou ainda estão disponíveis, o que promove transparência em todas as etapas.

Além disso, o sistema facilita a comunicação entre adotantes e abrigos, proporcionando um espaço onde todas as interações são centralizadas e registradas. Através de um painel de administração, os abrigos podem visualizar e gerenciar as solicitações de adoção, enquanto os adotantes têm acesso a atualizações sobre o andamento de seus pedidos. Esse nível de controle e organização permite uma gestão eficiente, além de fornecer uma ferramenta poderosa para abrigos administrarem suas adoções e interações de maneira segura e transparente.

Assim, o sistema atende plenamente aos objetivos propostos, oferecendo uma plataforma que facilita a gestão de animais pelos abrigos, melhora a visibilidade dos animais disponíveis e promove uma adoção responsável e segura, beneficiando tanto as instituições de canis e abrigos quanto os adotantes.

4. Metodologia

O desenvolvimento do sistema web de adoção "4 Patas" começou com um processo cuidadoso de levantamento de requisitos, que foi essencial para garantir que o sistema atendesse às necessidades tanto dos usuários quanto dos canis e abrigos. Antes de iniciar a implementação, foi importante definir com clareza as funcionalidades que o sistema deveria oferecer. A primeira necessidade identificada foi a criação de um sistema de cadastro de usuários, que permitisse aos futuros adotantes se registrarem fornecendo informações básicas como nome, e-mail, senha e outros dados relevantes. Isso facilitaria a autenticação posterior, essencial para garantir que apenas usuários cadastrados pudessem acessar funcionalidades específicas, como a solicitação de adoção.

Além do cadastro de usuários, o sistema deveria ser capaz de autenticar tanto canis e abrigos, quanto os próprios usuários. A autenticação de canis e abrigos era vital para que esses estabelecimentos pudessem acessar áreas restritas, onde seriam responsáveis por gerenciar perfis de animais e aprovar solicitações de adoção. Assim, estabeleceu-se que o sistema deveria permitir que os responsáveis pelos canis e abrigos entrassem com suas credenciais (e-mail e senha), garantindo que somente eles pudessem modificar as informações dos animais sob seus cuidados.

Outro ponto essencial foi a criação de perfis detalhados dos animais disponíveis para adoção. Cada perfil de animal deveria conter informações importantes como nome, idade, sexo, porte, histórico e fotos. Isso ajudaria a criar uma apresentação atraente dos

animais, aumentando as chances de adoção. Além disso, foi fundamental planejar uma funcionalidade que permitisse aos usuários solicitarem a adoção desses animais diretamente pela plataforma. A solicitação de adoção envolvia o preenchimento de um formulário onde o usuário forneceria informações sobre si mesmo, o que permitiria uma avaliação mais criteriosa da adequação do adotante.

A capacidade de aprovar ou rejeitar solicitações de adoção também foi considerada uma funcionalidade crucial. Os canis e abrigos precisariam revisar essas solicitações com base em critérios pré-definidos, como o ambiente doméstico do adotante e sua disponibilidade para cuidar do animal. Essa função foi pensada para garantir que as adoções fossem realizadas de maneira responsável e que os animais encontrassem lares adequados.

Além dessas funcionalidades centrais, o planejamento incluiu um sistema de gerenciamento de perfis de animais que permitisse aos canis e abrigos adicionar, editar ou remover perfis conforme necessário. Dessa forma, seria possível manter os perfis de animais sempre atualizados, facilitando o gerenciamento interno das informações.

Outro requisito importante identificado foi a necessidade de um sistema de busca e filtros que possibilitasse aos usuários encontrar animais de acordo com critérios específicos, como espécie, idade, porte e sexo. O objetivo era facilitar a navegação e melhorar a experiência do usuário, garantindo que ele pudesse localizar rapidamente o animal que mais se adequasse às suas preferências. Para complementar essa funcionalidade, uma galeria de fotos atrativa foi planejada, permitindo uma visualização rápida e eficiente dos animais disponíveis para adoção. Isso ajudaria a engajar mais os usuários, tornando a experiência de navegação mais visual e interativa.

A página de detalhes do animal também foi planejada com a mesma atenção. Nessa página, todas as informações relevantes sobre o animal seriam exibidas, proporcionando aos usuários uma visão completa antes de tomarem a decisão de solicitar a adoção. Nesse contexto, foi incluído um botão de "Adotar", facilitando a interação direta entre o usuário e o sistema.

Por fim, durante o levantamento dos requisitos, foi decidido que o sistema manteria um registro do histórico de adoções realizadas. Essa funcionalidade seria útil tanto para os adotantes da plataforma quanto para os canis e abrigos, permitindo que todas as adoções fossem documentadas de forma detalhada, com informações sobre o animal adotado, o adotante e a data da adoção.

Com os requisitos definidos, o próximo passo foi o planejamento das páginas que atenderiam a essas funcionalidades. Cada página do sistema foi pensada para desempenhar um papel específico. A página de cadastro de usuários, por exemplo, foi planejada para coletar informações de forma eficiente, enquanto a página de gerenciamento de animais foi feita para que os responsáveis pelos abrigos pudessem adicionar, editar e remover perfis de animais com facilidade. A página de busca de animais, juntamente com seus filtros, foi desenhada de forma a garantir que os usuários pudessem navegar com rapidez entre os perfis disponíveis e a página de detalhes do animal, a qual ofereceria todas as informações necessárias para que os adotantes pudessem tomar decisões sensatas e responsáveis.

Durante essa fase de planejamento, utilizamos a ferramenta Canva para criar esboços visuais das páginas. O uso dessa ferramenta permitiu que tivéssemos uma visão clara de como o design do sistema deveria ser, ajustando elementos de usabilidade e design antes de iniciar a implementação. Esse processo ajudou a evitar retrabalhos e garantiu que as páginas fossem visuais e intuitivas desde o início.

Após concluir o planejamento do sistema "4 Patas", voltamos nossa atenção para a estruturação do banco de dados, que desempenharia um papel central no armazenamento e gerenciamento das informações. Decidimos utilizar o MySQL, em conjunto com a ferramenta MySQL Workbench, que nos forneceu um ambiente visual intuitivo para criar, modelar e gerenciar o banco de dados de forma eficiente.

O banco de dados foi implementado à medida que o desenvolvimento das páginas do sistema avançava. Adotamos uma abordagem iterativa, onde cada tabela e suas respectivas relações eram criadas conforme surgiam novas funcionalidades no sistema. Dessa forma, o banco de dados evoluiu paralelamente à estruturação do frontend e backend, garantindo que os dados necessários estivessem sempre disponíveis para suportar as funcionalidades planejadas.

Iniciamos a construção do banco de dados com as tabelas fundamentais para o funcionamento do sistema: usuário e abrigo. Essas duas tabelas foram a base do sistema de cadastro e autenticação. A tabela usuario armazenava informações dos usuários do sistema, como nome, e-mail, senha, telefone, CPF e endereço, além de dados relevantes para identificação e segurança, como a data de nascimento e o timestamp de cadastro. Já a tabela abrigo foi desenhada para armazenar os dados dos canis e abrigos cadastrados, incluindo CNPJ ou CPF, endereço completo, informações de contato e, opcionalmente,

o site da instituição. Além disso, ambas as tabelas possuíam chaves primárias e restrições de unicidade em campos como e-mail e documentos (CPF ou CNPJ), assegurando a integridade dos dados.

Após garantir o funcionamento do sistema de usuários e abrigos, avançamos para a criação da tabela animal, que seria essencial para o registro dos animais disponíveis para adoção. Cada animal cadastrado pelos abrigos era identificado com atributos como nome, tipo (cão ou gato), data de nascimento, sexo, status de castração, porte e uma descrição detalhada que ajudaria a potencializar as chances de adoção. A tabela também incluía uma referência direta à tabela abrigo, por meio de uma chave estrangeira (`abrigo_id`), que permitia vincular cada animal ao seu abrigo de origem. Essa relação foi fundamental para o gerenciamento adequado dos animais dentro do sistema, facilitando operações como a atualização e remoção dos perfis.

Na etapa seguinte, com a funcionalidade de solicitação de adoção ganhando corpo, foi necessário implementar a tabela `solicitacao_adocao`. Essa tabela foi projetada para armazenar as informações relativas às adoções solicitadas pelos usuários. A tabela capturava dados cruciais, como o ID do usuário e do animal envolvidos no processo, o status da solicitação (em análise, aprovada ou recusada), além de armazenar o formulário preenchido pelo usuário durante a solicitação. A inclusão de chaves estrangeiras para os IDs de `usuario` e `animal` garantiu que as solicitações estivessem sempre vinculadas aos seus registros correspondentes, assegurando a integridade referencial e permitindo uma navegação eficiente entre as informações.

Após a estruturação do banco de dados, voltamos nossa atenção para o desenvolvimento do back-end, implementado utilizando PHP. Optamos por uma abordagem orientada a objetos (OO), que nos permitiu organizar de maneira mais clara e eficiente o código do sistema, além de possibilitar uma melhor reutilização das funcionalidades. A orientação a objetos se mostrou ideal para a modelagem das entidades do sistema, uma vez que nos permitiu encapsular as responsabilidades e atributos de cada entidade em classes específicas, facilitando tanto a manutenção quanto a extensão futura do sistema.

Começamos criando classes que representavam as principais entidades do sistema: `Usuario`, `Abrigo`, `Animal` e `SolicitacaoAdocao`. Cada uma dessas classes encapsulava todos os atributos e métodos necessários para manipular os dados dessas entidades e interagir diretamente com o banco de dados. Essa abordagem não apenas

promoveu uma separação clara de responsabilidades, mas também nos proporcionou uma maneira consistente e segura de lidar com operações envolvendo dados.

Cada classe era responsável por seus próprios métodos de CRUD (Create, Read, Update, Delete), que são as operações essenciais para o gerenciamento de dados no banco. Por exemplo, na classe `Usuario`, tínhamos métodos como `create()`, que lidava com o cadastro de novos usuários; `readOne()`, que buscava as informações de um usuário específico no banco de dados; `update()`, que permitia a atualização dos dados de um usuário existente; e `delete()`, responsável por remover um usuário quando necessário. O mesmo padrão foi aplicado às classes `Abrigo`, `Animal` e `SolicitacaoAdocao`.

Além dessas classes, criamos um arquivo `config.php`, que era responsável por estabelecer a conexão com o banco de dados. Esse arquivo centralizava a lógica de conexão, facilitando sua reutilização em diferentes partes do sistema. Utilizamos a função `include_once` para importar o arquivo de configuração e outros componentes necessários em cada página, o que garantiu que o código fosse modular e eficiente.

Ao desenvolver as páginas do sistema, seguimos uma abordagem onde o PHP e o HTML coexistiam na mesma página. Isso significa que a lógica do PHP, que controlava a funcionalidade da aplicação (como a inserção e atualização de dados no banco), era processada e logo em seguida o HTML era renderizado para exibir o resultado ao usuário. Essa integração direta do PHP com o HTML permitiu um fluxo simples e direto no desenvolvimento das páginas, sem a necessidade de separar o front-end e o back-end em arquivos distintos.

No entanto, mesmo com essa integração entre PHP e HTML, entendemos que a organização visual do sistema era fundamental para proporcionar uma boa experiência de usuário. Por isso, todo o CSS foi organizado em arquivos externos, que eram importados nas páginas. Isso garantiu não apenas a separação da lógica de estilo, mas também uma maior eficiência, já que o CSS poderia ser reutilizado em diferentes partes do sistema e facilmente atualizado conforme a necessidade.

Quanto ao JavaScript, embora cada página individualmente tivesse uma quantidade relativamente pequena de código, ele desempenhou um papel fundamental no funcionamento dinâmico e interativo do sistema como um todo. Utilizamos JavaScript extensivamente para diversas funcionalidades cruciais, como exibição de mensagens de erro, feedbacks visuais, como alertas e notificações que informavam

sobre o preenchimento incorreto de formulários ou o sucesso de determinadas operações, interatividade com botões, busca automática do CEP, formatação de CPF, CNPJ e telefone, manipulação de divs ocultas, e implementação de filtros dinâmicos, o que tornou a experiência do usuário mais fluida e responsiva.

Contudo, apesar do uso intensivo de JavaScript, decidimos não modularizar o código em arquivos externos para scripts individuais. Em vez disso, mantivemos o código diretamente dentro das páginas, no final da tag `<body>`. Essa decisão foi tomada por questões de praticidade e simplicidade, uma vez que o código em si não era excessivamente extenso e a implementação dentro das próprias páginas facilitava a manutenção em pequenos projetos. Mesmo assim, o JavaScript desempenhou um papel importante no funcionamento do sistema, proporcionando interatividade e dinâmica a várias partes essenciais da plataforma.

Tanto para o desenvolvimento do front-end quanto do back-end, utilizamos o Visual Studio Code como ambiente de programação. Sua interface intuitiva, ampla variedade de extensões e ferramentas de depuração foram fundamentais para garantir a produtividade durante todas as etapas do projeto. A escolha dessa ferramenta nos permitiu integrar diferentes linguagens e tecnologias de maneira eficiente, tornando o processo de desenvolvimento mais ágil e organizado.

Para implementar e testar todas as funcionalidades do nosso sistema, foi imprescindível a utilização de um servidor. Não há como concluir o desenvolvimento do banco de dados, do back-end e do front-end sem um ambiente adequado que suporte a execução das aplicações web. Diante disso, escolhemos o XAMPP no início do projeto, uma solução que se destacou por suas características e benefícios, tornando-se a escolha ideal para o nosso contexto.

O XAMPP é uma plataforma que combina diversas tecnologias essenciais para o desenvolvimento web, incluindo o servidor Apache, o sistema de gerenciamento de banco de dados MySQL e a linguagem de programação PHP. Essa combinação permitiu que, em um único pacote, tivéssemos acesso a todas as ferramentas necessárias para criar e gerenciar nosso sistema de forma eficiente.

A facilidade de instalação e configuração do XAMPP foi um dos principais fatores que nos levaram a escolhê-lo. Com um simples download, conseguimos criar um ambiente de desenvolvimento completo, sem as complicações geralmente associadas à configuração de servidores. Essa simplicidade foi crucial, pois o foco inicial do projeto

era desenvolver e testar rapidamente as funcionalidades, evitando retrabalho e perda de tempo com configurações complexas.

Além disso, a capacidade de simular um ambiente de produção foi uma grande vantagem. Embora estivéssemos utilizando o sistema localmente, era fundamental testar o projeto em condições que se aproximassem daquelas que encontraríamos quando o sistema fosse hospedado em um servidor real. Com o Apache, o MySQL e o PHP devidamente configurados no XAMPP, pudemos garantir que o comportamento do sistema seria consistente. Essa configuração nos permitiu replicar as interações do sistema de maneira fiel, evitando surpresas indesejadas durante a fase de implantação.

Portanto, ao final da análise, o XAMPP se destacou por ser uma solução completa, gratuita, fácil de instalar e que oferece ferramentas robustas para desenvolvimento web. Essas características fizeram com que ele fosse a escolha ideal para o desenvolvimento do "4 Patas", proporcionando um ambiente de trabalho eficiente e confiável durante todas as fases do projeto.

5. Resultados

Ao longo do desenvolvimento do sistema, realizamos uma série de testes minuciosos para assegurar que cada funcionalidade operasse conforme o esperado. Cada função implementada passou por uma bateria de testes rigorosos, que incluíam tanto a verificação da integridade dos dados quanto a validação das interações do usuário. Por exemplo, ao testar o cadastro de usuários e abrigos, garantimos que os campos obrigatórios fossem preenchidos corretamente (como ilustrado na Figura 3) e que as mensagens de erro aparecessem de maneira clara e informativa quando as informações estavam faltando ou eram inválidas (como ilustrado na Figura 4). Para cada função, como a solicitação de adoção e a exibição de animais, fizemos simulações com dados reais, analisando as respostas do sistema e a forma como ele lidava com diferentes cenários, sempre focando na precisão e na segurança das operações.

The screenshot shows a registration form titled "1 - Cadastro de Abrigos e Canis" on a light blue background decorated with paw prints. The form contains the following fields and values:

- Logo: 4 PATAS
- Field 1: Abriço Esperança
- Field 2: 00.000.000/0000-00
- Field 3: abrigoesperanca@gmail.com
- Field 4: +55 (00) 00000-0000
- Buttons: VOLTAR (blue) and PRÓXIMO (orange)

A yellow arrow points to the second field (CNPJ/CPF).

Figura 3. Página de Cadastro de Abrigo.

This screenshot shows the same registration form as Figure 3, but with an error message at the top: "O CNPJ/CPF não é válido. O telefone não é válido." (The CNPJ/CPF is not valid. The phone number is not valid.). The form fields contain the following values:

- Field 1: Abriço Esperança
- Field 2: 000.000.00
- Field 3: abrigoesperanca1@gmail.com
- Field 4: +55 (00) 00
- Buttons: VOLTAR (blue) and PRÓXIMO (orange)

A yellow arrow points to the error message.

Figura 4. Mensagem de erro ao preencher incorretamente os campos.

Além das funcionalidades, a usabilidade do sistema também foi um foco importante em nossos testes. Realizamos sessões de avaliação onde usuários em potencial interagiram com o sistema, proporcionando feedback direto sobre a interface e a experiência geral. Observamos como os usuários navegavam pelas diferentes seções, se encontravam facilmente as informações desejadas e se as instruções eram claras. A partir desse feedback, realizamos ajustes e refinamentos, como melhorias na disposição dos elementos na tela e na clareza das mensagens exibidas, visando criar uma experiência intuitiva e agradável. Essa fase de testes de usabilidade foi crucial para garantir que o sistema não apenas funcionasse, mas também fosse acessível e fácil de usar para todos os perfis de usuários.

Visando o futuro, estamos entusiasmados com as possibilidades de expandir o alcance do nosso sistema. Uma das principais direções que pretendemos seguir é a busca por parcerias com abrigos e organizações que atuem na adoção de animais. Estabelecer essas colaborações permitirá que testemos o sistema em ambientes reais, onde os usuários possam interagir diretamente com as funcionalidades oferecidas. Esses testes práticos proporcionarão insights valiosos sobre o desempenho do sistema em situações do dia a dia e a aceitação pelos usuários finais. O feedback obtido será fundamental para continuarmos aprimorando o sistema, garantindo que ele atenda não apenas às necessidades funcionais, mas também às expectativas dos usuários.

Além disso, ao firmar parcerias com abrigos, poderemos obter dados reais sobre as interações dos usuários e suas experiências ao utilizar a plataforma. Isso incluirá informações sobre a taxa de sucesso das solicitações de adoção, o nível de satisfação dos usuários com o processo e eventuais dificuldades encontradas. Essas informações não apenas ajudarão na identificação de áreas que precisam de melhorias, mas também servirão como base para futuras atualizações do sistema, assegurando que ele evolua de acordo com as necessidades dos usuários e do mercado.

Por fim, a implementação de um ciclo contínuo de testes e feedback será vital para a longevidade e relevância do sistema. Com uma abordagem proativa, buscaremos constantemente a inovação e a melhoria, garantindo que nossa plataforma não apenas atenda, mas supere as expectativas de todos os envolvidos no processo de adoção. Com isso, esperamos não apenas facilitar a adoção de animais, mas também contribuir significativamente para a causa da proteção animal, criando um sistema que realmente faça a diferença.

6. Conclusão

O abandono de animais e a superlotação de abrigos representam problemas graves e recorrentes, que tornam o processo de adoção uma solução essencial, mas frequentemente subutilizada. A falta de sistemas centralizados e a dificuldade em acessar informações atualizadas sobre os animais disponíveis são barreiras que desestimulam os possíveis adotantes. A proposta do sistema "4 Patas" surge exatamente para enfrentar essas dificuldades, oferecendo uma solução prática e eficiente. Ao centralizar os dados dos abrigos e dos animais, o sistema visa facilitar o encontro entre adotantes e animais, promovendo um processo mais organizado.

Os objetivos traçados para este projeto foram amplamente cumpridos. O sistema melhora a visibilidade dos animais para adoção, facilita a comunicação entre abrigos e adotantes, e oferece funcionalidades de filtragem que tornam a busca por animais mais específica. Além disso, a segurança no armazenamento de informações, com o registro completo de todo o histórico de adoções e interações, assegura a transparência do processo e facilita o controle por parte dos abrigos. Esses avanços demonstram que o "4 Patas" tem um enorme potencial para impactar positivamente o cenário da adoção de animais, ajudando a reduzir o número de animais abandonados.

Referências

- PRESSMAN, Roger S. Engenharia de Software: Uma Abordagem Profissional. 7. ed. Porto Alegre: AMGH, 2011.
- SOMMERVILLE, Ian. Engenharia de Software. 10. ed. São Paulo: Pearson Education do Brasil, 2015.
- ATTRIX TECNOLOGIA. Sistema Web: o que é e por que vale a pena investir em um. Disponível em: <https://www.attri.com.br/blog/sistema-web-o-que/>. Acesso em: 27 set. 2024.
- TRYBE. Aplicações Web: entenda o que são e como funcionam. Blog da Trybe. Disponível em: <https://blog.betrybe.com/desenvolvimento-web/aplicacoes-web/>. Acesso em: 27 set. 2024.
- VALUEHOST. Como funciona um Servidor Web? 2023. Disponível em: <https://www.valuehost.com.br/blog/servidor-web/>. Acesso em: 29 set. 2024.
- SEMPREUPDATE. Tudo o que Você Precisa Saber Sobre os Servidores Web. 2022. Disponível em: <https://sempreupdate.com.br/linux/tutoriais/tudo-o-que-voce-precisa-saber-sobre-os-servidores-web/>. Acesso em: 29 set. 2024.
- KUROSE, James F.; ROSS, Keith W. Redes de Computadores. 6. ed. Porto Alegre: Bookman, 2013.
- DATE, C. J.; DARWEN, H. Fundamentals of Database Systems. 6. ed. São Paulo: Pearson, 2016.
- DEVMEDIA. PostgreSQL x MySQL. Qual Escolher?. DevMedia, 2020. Disponível em: <https://www.devmedia.com.br/postgresql-x-mysql-qual-escolher/3923>. Acesso em: 29 set. 2024.

SAPHIR. Conheça os principais bancos de dados NoSQL (não relacionais). Blog Saphir, 2020. Disponível em: <https://blog.saphir.com.br/conheca-os-principais-bancos-de-dados-nosql-nao-relacionais/>. Acesso em: 29 set. 2024.

PEREIRA, José da Silva. Desenvolvimento web moderno: princípios, práticas e tendências. São Paulo: Editora Técnica, 2019.

DIO. O que significa back-end, front-end e API. Disponível em: <https://www.dio.me/articles/o-que-significa-back-end-front-end-e-api>. Acesso em: 29 set. 2024.

Alura. O que é back-end e como ele funciona. Disponível em: <https://www.alura.com.br/artigos/backend?srsId=AfmBOoqayju1E8RLGKotee9LReIlytwia5IsJBCdiSO9AxkNKJnMZIEV>. Acesso em: 29 set. 2024.

SILVA, Maria. Interface do Usuário: Princípios e Práticas de Design. São Paulo: Editora XYZ, 2020.

DEVMEDIA. JavaScript e interatividade. Disponível em: <https://www.devmedia.com.br/javascript/>. Acesso em: 02 out. 2024.

MOZILLA DEVELOPER NETWORK. HTML: HyperText Markup Language. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTML>. Acesso em: 02 out. 2024.

W3SCHOOLS. CSS Introduction. Disponível em: https://www.w3schools.com/css/css_intro.asp. Acesso em: 02 out. 2024.

SEMAD. Mesmo sem transmitir o Coronavírus, cães e gatos têm sido alvo de abandono. 2020. Disponível em: (Acesso em: 04 out. 2024.).

WINKLER, V. O vínculo entre ser humano e os animais. 2019.

COSTA, M. S. d. S. A. Campanha para a sensibilização da adoção de animais de companhia e combate ao seu abandono. Tese (Doutorado) — Instituto Politécnico de Lisboa, Escola Superior de Comunicação Social, 2021.

MERCADO ONLINE DIGITAL. XAMPP: o que é e como instalar. Disponível em: <https://mercadoonlinedigital.com/blog/xampp/>. Acesso em: 4 out. 2024.

APACHE FRIENDS. XAMPP documentation. Disponível em: <https://www.apachefriends.org/docs/>. Acesso em: 4 out. 2024.