

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
FACULDADE DE TECNOLOGIA DA PRAIA GRANDE CURSO
SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

RELATÓRIO TÉCNICO DE CONCLUSÃO DE CURSO
ADOÇÃO ONLINE EM PRAIA GRANDE-SP: MÃOS QUE SALVAM

MARIA LUIZA DA SILVA AUGUSTO

PRAIA GRANDE

2024

MARIA LUIZA DA SILVA AUGUSTO

RELATÓRIO TÉCNICO DE CONCLUSÃO DE CURSO
ADOÇÃO ONLINE EM PRAIA GRANDE-SP: MÃOS QUE SALVAM

Trabalho de conclusão de curso apresentado à Faculdade de Tecnologia de Praia Grande, como parte dos requisitos para a obtenção do título de Superior em Análise e Desenvolvimento de Sistemas.

PRAIA GRANDE

2024

“A grandeza de uma nação pode ser julgada pelo modo que seus animais são tratados.”

Mahatma Gandhi (1869 – 1948)

RESUMO

O presente estudo tem como finalidade primordial o desenvolvimento e construção de um website abrangente e interativo para a Organização Não Governamental (ONG) "Mãos que Salvam" em Praia Grande - SP. O estudo supracitado visa preencher uma lacuna existente na região, a escassez de órgãos gratuitos para o acolhimento de animais abandonados ou em situação de rua. Visando o suprimento da eminente demanda, será desenvolvida uma plataforma online com o intuito de facilitar a adoção desses animais, onde após contato com a instituição, será feito um anúncio com todas as informações. O website será desenvolvido utilizando as seguintes tecnologias, HTML, JavaScript e CSS para o frontend, e tendo o backend formulado com Java, utilizando o framework Spring, e banco de dados H2 para armazenar as informações relevantes.

No website será implementado um sistema de login seguro para administradores, tendo como intuito, o acesso exclusivo de pessoas autorizadas para realizar anúncios de animais disponíveis para adoção. Além disso, o banco de dados H2 será usado para armazenar informações detalhadas do animal, incluindo raça, idade, tamanho, descrição e informações de contato do anunciante.

O site contará com uma página inicial contendo todos os animais disponíveis para adoção, assim como suas respectivas páginas individuais, com foto, descrição detalhada de sua personalidade, necessidades especiais, informações de idade, raça e nome (caso ele já possua), juntamente às informações do anunciante.

Palavras-chave: adoção online, abrigo de animais, Praia Grande-SP, website interativo, frontend, backend, tecnologia, segurança, banco de dados, saúde pública.

ABSTRACT

This report aims to develop a comprehensive and interactive website for the animal shelter 'Helping Hands' in Praia Grande, SP. The proposal seeks to address a gap in the region, which is the lack of a free organization for the sheltering of abandoned animals or those in search of a new home. To achieve this, an online platform will be developed with the purpose of facilitating the adoption process for these animals, where, upon contacting the institution, an announcement will be created with all the necessary information. The website will be developed using the following technologies: HTML, JavaScript, and CSS for the frontend, and Java with the Spring framework for the backend, utilizing the H2 database to store relevant information.

The website will implement a secure login system for administrators, with the aim of providing exclusive access to authorized individuals to post announcements for animals available for adoption. Additionally, the H2 database will be used to store detailed information about the animals, including breed, age, size, description, and advertiser contact information.

The site will feature an initial page containing all the animals available for adoption, along with their respective individual pages, which will provide a detailed description of their personality, special needs, age, breed, and name (if already given).

Keywords: online adoption, animal shelter, Praia Grande-SP, interactive website, frontend, backend, technology, security, database.

LISTA DE ILUSTRAÇÕES

Figura 1 - Diagrama de classe	24
Figura 2 - Diagrama de Entidade Relacional.....	25
Figura 3 - Diagrama de Caso de Uso.....	26
Figura 4 - Index	41
Figura 5 - Descrição do animal.....	43

SUMÁRIO

RESUMO.....	4
ABSTRACT	5
LISTA DE ILUSTRAÇÕES	6
1. INTRODUÇÃO	9
1.1. OBJETIVO GERAL	10
1.2. OBJETIVOS ESPECÍFICOS	10
2. RECURSOS.....	10
2.1. RECURSOS HUMANOS	10
2.2. RECURSOS DE INFRAESTRUTURA	11
3. PLANEJAMENTO	12
3.1. DEFINIÇÃO DO ESCOPO DO PROJETO.....	13
3.2. LEVANTAMENTO DE REQUISITOS.....	14
3.3. ESPECIFICAÇÃO FUNCIONAL	14
3.4. ARQUITETURA DO SOFTWARE.....	15
3.5. DESENVOLVIMENTO DO SOFTWARE.....	16
3.5.1. Backend (Spring Boot).....	16
1) Configuração do Banco de Dados H2.....	16
3.5.2. Frontend (HTML, CSS, JavaScript)	16
3.6. TESTES DO SOFTWARE	17
3.6.1. Importância dos Testes de Software.....	17
3.6.2. Abordagem e Estratégia de Testes.....	17
3.6.3. Testes no Backend	18
3.6.4. Testes no Frontend.....	19
3.7. IMPLANTAÇÃO DO SOFTWARE.....	20
3.7.1. Estratégia de Implantação	20
3.7.2. Implantação em Ambiente Local.....	20

3.7.3.	Implantação em Ambiente na Nuvem (Azure)	21
3.7.4.	Processo de Teste Pós-Implantação	21
3.8.	CRONOGRAMA TÉCNICO	21
4.	DESENVOLVIMENTO SISTEMA / SOFTWARE / APLICATIVO	22
4.1.	LEVANTAMENTO DE REQUISITOS	22
4.2.	DIAGRAMA DE CLASSE	23
4.3.	DIAGRAMA DE ENTIDADE RELACIONAL	24
4.3.1.	Relações entre as Entidades	25
4.4.	DIAGRAMA DE CASO DE USO	26
4.5.	PROJETO TÉCNICO	34
4.5.1.	Java	34
4.5.2.	Spring Framework.....	35
4.5.3.	H2 Database	36
4.5.4.	Eclipse IDE	37
4.5.6.	Azure – Plataforma de Nuvem	38
4.6.	DESCRIÇÃO DAS INTERFACES	39
5.	CONSIDERAÇÕES FINAIS	44
5.1.	DESENVOLVIMENTOS FUTUROS.....	44
5.1.1.	Sugestões de Futuras Tecnologias.....	46

1. INTRODUÇÃO

No presente momento, animais domésticos abandonados em vias públicas são uma questão abrangente em grandes parcelas dos municípios do país. Embora o controle de zoonoses seja um dever dos órgãos públicos, representados pelas prefeituras, grande parte dos municípios brasileiros não dispõe de um sistema de infraestrutura adequada para deter o controle de animais em situação de abandono. Prefeituras que não enfrentam a problemática supracitada de maneira ativa são assoladas com transtornos multissetoriais, tais como: saúde pública, saúde animal e meio ambiente (Vargas et al., 2014).

Segundo Andrade (2011), diante da necessidade de implementação efetiva por parte do poder público de políticas de prevenção do abandono de animais, a Organização Mundial de Saúde (OMS) preconiza que o tratamento dado à questão deve ser embasado pelos seguintes aspectos: ser eficiente no sentido de modificar condutas e prevenir o abandono futuro dos animais; ser humanitário e justo, pois os animais são vítimas da falta de responsabilidade das pessoas; ser de responsabilidade de todos: autoridades, profissionais da saúde, educadores, especialistas em bem-estar animal, ONGs em geral (OMS, 2005).

Suprereferido, a sociedade de Praia Grande, litoral paulista, enfrenta aflições crescentes no que tange à problemática de abandono e vulnerabilidade dos animais. Como método que visa sanar ou diminuir essa parcela de problemas, existem ONGs e abrigos de animais, tais organizações foram criadas com o intento de proteger e abrigar animais em situação de rua. Todavia, mesmo com a criação e implementação dessas diversas instituições não governamentais ainda há pouca assistência para a população animal na região (Neto, 2020).

Visando o preenchimento de tal lacuna, foi desenvolvido um website, fornecendo assim, uma plataforma virtual e intuitiva, que facilite o processo de rastreio e adoção dos animais em situação de abandono. Não obstante, não se possui ainda uma estrutura física, o grupo de pessoas responsáveis pelo website trabalharão como intermediadores para adoção e bem-estar dos animais (Machado, 2018).

O sistema de adoção online viabiliza que pessoas interessadas conheçam de forma simples e desburocratizada os animais cadastrados e disponíveis para adoção,

tendo acesso às informações do pet como: foto, descrição detalhada de sua personalidade, idade estimada, porte físico, necessidades especiais, raça, juntamente com as informações de localização e contato do anunciante (Carvalho, 2017).

1.1.OBJETIVO GERAL

O objetivo do presente projeto é o desenvolvimento de website que visa facilitar a adoção de animais em situação de abandono, conectando interessados aos grupos responsáveis pelo cuidado dos animais.

O website promoverá a conscientização da comunidade sobre a importância da adoção responsável e promover o cuidado e proteção dos animais.

Facilitar também o processo de rastreio de animais em situação de rua e abandono, promover um ambiente seguro e democratizado para anúncio e adoção desses animais.

1.2.OBJETIVOS ESPECÍFICOS

- Aplicar as técnicas de engenharia de software no desenvolvimento de website.
- Desenvolver sistema computacional utilizando linguagem Java com o framework Spring e banco de dados H2, explorando seus conceitos, vantagens e desvantagens.

2.RECURSOS

2.1.RECURSOS HUMANOS

O projeto é idealizado e desenvolvido por Maria Luiza da Silva Augusto, uma programadora backend com cinco anos de experiência na área de tecnologia. Com trajetória no desenvolvimento de sistemas web e APIs, Maria Luiza utiliza principalmente a linguagem Java em seus projetos, o que reflete seu domínio no desenvolvimento backend. Ao longo de sua carreira, adquiriu vasta experiência na

manipulação de dados e na utilização de ferramentas da Amazon Web Services (AWS), além de ter se especializado na criação de dashboards interativos e informativos utilizando o Tableau. Além de sua atuação profissional, Maria Luiza está atualmente ampliando seus conhecimentos ao cursar Análise e Desenvolvimento de Sistemas na FATEC Praia Grande, buscando sempre se atualizar nas novas tecnologias e melhores práticas do mercado.

2.2.RECURSOS DE INFRAESTRUTURA

O website utilizará a plataforma Microsoft AZURE para hospedagem em nuvem, e usufrui da máquina virtual A2 V2, capaz de suportar tráfego de até um milhão e duzentos mil acessos por ano, tendo um custo mensal de R\$315,00.

Para o desenvolvimento do presente projeto, foram utilizados dois computadores, sendo um desktop equipado com um processador Ryzen 5 5600 X, 16 gigabytes de memória ram ddr4 de 3600mhz, uma gpu dedicada RX 5600XT de 6gb e dois ssds NVME m.2 de um terabyte de memória e uma velocidade de leitura de sete mil megabytes por segundo, utilizando o Windows 10 como sistema operacional (SO) e um notebook ACER Espire-e5 equipado com um processador Ryzen 5-2500U, 12 gigabytes de memória ram ddr4 2800mhz, gpu interna radeon veja 7 e um hard-disk de um terabyte utilizando também como S.O o windows.

Para navegação e desenvolvimento online foi utilizado como provedor de internet a empresa DESKTOP, com um pacote de quinhentos gigabits de velocidade.

Categoria	Descrição	Quantidade	Custo Unitário (R\$)	Custo (R\$)	Total
Hospedagem em Nuvem	Microsoft Azure (plano A2 V2, custo mensal)	12 meses	315,00	3.780,00	

Equipamento - Desktop	Montagem de um computador	1 unidade	Estimado: 9.000,00	9.000,00
Equipamento - Notebook	Acer Aspire E5	1 unidade	Estimado: 3.500,00	3.500,00
Provedor de Internet	DESKTOP (500 Mbps, custo mensal)	12 meses	150,00	1.800,00
Software - Windows 10	Licença do sistema operacional Windows 10	2 licenças	1.200,00	2.400,00

3. PLANEJAMENTO

O desenvolvimento do site será conduzido por meio de um cronograma bem estruturado, com etapas e prazos claramente definidos para garantir a execução eficiente e sem contratempos. O projeto será desenvolvido utilizando tecnologias modernas, que foram escolhidas cuidadosamente para atender aos requisitos de escalabilidade, segurança e desempenho. A arquitetura adotada visa criar um sistema robusto e preparado para futuros crescimentos, permitindo que novos recursos possam ser adicionados de forma simples e sem comprometer a performance.

A comunicação entre o frontend e o backend será realizada de maneira direta, sem a utilização de APIs externas, o que elimina a necessidade de intermediários e simplifica o fluxo de dados. Essa abordagem assegura uma integração mais rápida e reduz a latência, já que as requisições são processadas e as respostas enviadas diretamente do servidor backend para o cliente, sem etapas adicionais de comunicação. O backend será o responsável por lidar com todas as requisições dos usuários, processar as informações de forma eficiente e renderizar as páginas de maneira dinâmica.

O uso do Spring Boot para o backend proporciona uma estrutura sólida e altamente eficiente, permitindo que a aplicação seja desenvolvida de forma rápida e com alta performance. O Spring Boot oferece diversas funcionalidades out-of-the-box, como o gerenciamento de dependências e a configuração simplificada do servidor, garantindo que o sistema tenha uma base robusta desde o início. Além disso, o Thymeleaf será utilizado para a renderização das páginas HTML, permitindo que o conteúdo dinâmico seja gerado diretamente pelo backend e exibido ao usuário sem a necessidade de requisições adicionais ou sobrecarga de recursos.

Essa arquitetura oferece uma experiência de usuário mais fluida, já que as informações são processadas e exibidas em tempo real. A renderização dinâmica das páginas, facilitada pelo Thymeleaf, permite que o conteúdo da página seja atualizado de forma instantânea sem recarregar o site, proporcionando uma navegação mais ágil e intuitiva. Isso contribui para a criação de uma plataforma responsiva e interativa, onde o usuário consegue acessar informações rapidamente e de forma consistente, independentemente de seu dispositivo.

A escolha de não utilizar APIs externas também traz benefícios significativos em termos de segurança. Ao evitar a comunicação com serviços externos, o sistema reduz as vulnerabilidades que poderiam ser exploradas por terceiros e garante que todos os dados trafeguem dentro de um ambiente controlado. O backend, sendo responsável por toda a lógica de processamento e manipulação de dados, assegura que as informações sensíveis sejam tratadas de forma segura, sem o risco de exposição durante as interações com a interface do usuário.

Essa solução não apenas contribui para uma performance otimizada e maior controle sobre a segurança, mas também proporciona uma manutenção mais fácil e eficiente do sistema. Com o controle centralizado no backend, é possível realizar ajustes e atualizações diretamente no servidor, sem a necessidade de modificar múltiplos pontos de comunicação ou fazer ajustes complexos em integrações externas. Isso torna o sistema mais flexível e preparado para alterações e expansões futuras.

3.1.DEFINIÇÃO DO ESCOPO DO PROJETO

Nesta etapa inicial, o objetivo e os limites do sistema serão definidos. Será elaborado um planejamento técnico que descreva a integração direta entre o backend, utilizando Java e Spring Boot, e as páginas HTML renderizadas dinamicamente.

Produto gerado: Documento de escopo de Projeto, incluindo um diagrama geral da arquitetura com foco no backend processando e servindo páginas completas.

3.2.LEVANTAMENTO DE REQUISITOS

Será realizado o levantamento técnico das funcionalidades do sistema, como autenticação de administradores, cadastro de animais e exibição de dados diretamente nas páginas HTML renderizadas. Para isso, serão realizadas reuniões com possíveis usuários para identificar requisitos funcionais e não funcionais.

Produto Gerado: Documento de Requisitos, detalhando requisitos como:

- Sistema de autenticação usando Spring Security diretamente nas páginas.
- Integração do backend com HTML usando Thymeleaf para renderização dinâmica.
- Uso do H2 Database para armazenar informações de animais e administradores.
- Design responsivo com HTML e CSS.

3.3.ESPECIFICAÇÃO FUNCIONAL

Os requisitos levantados serão traduzidos em funcionalidades específicas. Cada funcionalidade será documentada com os fluxos de navegação, interações entre as páginas e o backend, além dos modelos de dados.

Produto Gerado: Documento de Especificação Funcional, incluindo:

- Estrutura de navegação entre as páginas (ex.: página inicial, cadastro de animal, formulário de adoção).

- Descrição dos controladores Spring Boot que gerenciam as requisições.
- Design de formulários dinâmicos com Thymeleaf.

3.4.ARQUITETURA DO SOFTWARE

O sistema foi planejado utilizando a arquitetura **MVC (Model-View-Controller)**, garantindo a separação de responsabilidades para melhor organização do projeto e facilidade de manutenção. O objetivo principal do sistema é proporcionar uma plataforma intuitiva e funcional para facilitar a adoção de animais.

a) Modelo (Model)

- Representa os dados do sistema, como animais disponíveis para adoção, usuários (adotantes e administradores), e processos de adoção.
- As classes de modelo foram criadas com base em **JPA (Java Persistence API)** para mapeamento objeto-relacional, utilizando anotações apropriadas para gerenciar as relações entre tabelas.

b) Visão (View)

- Interface para interação com o sistema. Desenvolvida com **HTML5** para estruturação, **CSS3** para estilos, e **JavaScript** para interatividade.
- **Thymeleaf** foi utilizado para renderizar páginas dinâmicas, exibindo informações como listagem de animais disponíveis para adoção e detalhes de cada animal.

c) Controlador (Controller)

- Responsável por gerenciar as interações entre a visão e o modelo. Os controladores processam requisições HTTP e gerenciam os dados exibidos nas páginas.

d) Banco de Dados (H2)

- Configurado para armazenar informações relacionadas aos animais, usuários e adoções.

- Relações foram mapeadas, por exemplo, entre usuários e animais adotados, permitindo a representação precisa dos dados.

3.5.DESENVOLVIMENTO DO SOFTWARE

O desenvolvimento foi dividido em backend e frontend, integrados para criar uma experiência de usuário fluida.

3.5.1.Backend (Spring Boot)

1) Configuração do Banco de Dados H2

- O banco de dados foi configurado para persistência durante o desenvolvimento, assegurando o armazenamento das informações de maneira eficiente.

2) Controladores Spring:

- Controladores foram desenvolvidos para gerenciar funcionalidades específicas, como o cadastro, listagem e gerenciamento de animais, usuários e processos de adoção.

3) Autenticação com Spring Security:

- Foi implementado um sistema de autenticação para proteger páginas restritas, como o gerenciamento de dados administrativos e de animais.

3.5.2.Frontend (HTML, CSS, JavaScript)

a) Páginas dinâmicas:

- Criadas para exibir informações de maneira intuitiva, como a listagem de animais disponíveis para adoção, detalhamento de cada animal, e formulários para adoção e cadastro.

b) Formulários:

- Desenvolvidos para o cadastro de animais, processo de adoção, e login de usuários, garantindo consistência e integridade das informações.

c) Scripts JavaScript:

- Utilizados para validar formulários e proporcionar uma experiência interativa e amigável ao usuário.

3.6. TESTES DO SOFTWARE

Os testes de software são uma etapa essencial no ciclo de desenvolvimento, pois asseguram que o sistema funciona conforme o planejado, atende aos requisitos estabelecidos e proporciona uma experiência de usuário satisfatória. No caso do site de adoção de animais, os testes foram planejados e realizados com foco na confiabilidade, segurança e usabilidade da aplicação.

3.6.1. Importância dos Testes de Software

Os testes desempenham um papel crucial no sucesso do projeto, garantindo:

- **Qualidade do Produto:** Certificam que o sistema está livre de defeitos e atende aos padrões de qualidade.
- **Redução de Riscos:** Identificam e corrigem falhas antes da implantação, reduzindo o impacto de problemas no ambiente de produção.
- **Satisfação do Usuário:** Asseguram que a experiência do usuário final seja intuitiva, funcional e livre de erros.
- **Manutenção Simplificada:** Um software bem testado é mais fácil de manter, já que as funcionalidades principais estão documentadas e validadas.

3.6.2. Abordagem e Estratégia de Testes

Os testes do sistema foram organizados em dois níveis principais: backend e frontend, com cada nível focando em aspectos específicos da aplicação.

3.6.3. Testes no Backend

Os testes backend têm como objetivo garantir que a lógica de negócios e a interação com o banco de dados funcionem corretamente.

a) Testes Unitários

Os testes unitários verificam o funcionamento de métodos individuais nos serviços e repositórios do backend.

- Ferramenta Utilizada: JUnit, um framework amplamente usado para testes unitários em Java.
- Objetivo: Validar que cada unidade de código (métodos ou classes) funcione isoladamente.
- Exemplo: Testar a lógica para verificar a disponibilidade de um animal para adoção.

b) Testes de Integração

Os testes de integração verificam a interação entre diferentes componentes do sistema, como controladores, serviços e o banco de dados.

- Ferramenta Utilizada: Spring Test, um módulo específico para testes em aplicações Spring Boot.
- Objetivo: Assegurar que as diferentes camadas do sistema (Model, View, Controller) se integrem de maneira correta.
- Exemplo: Testar o fluxo completo de uma requisição de adoção, desde a submissão do formulário até o armazenamento dos dados no banco de dados H2.

c) Testes de Segurança

Como o sistema utiliza Spring Security, foram realizados testes específicos para assegurar que:

- Apenas usuários autenticados possam acessar áreas restritas.
- Os dados sensíveis estejam protegidos contra ataques, como injeção de SQL ou acessos não autorizados.

3.6.4. Testes no Frontend

Os testes frontend garantem que as páginas HTML sejam renderizadas corretamente e que a interação do usuário com o sistema seja fluida e intuitiva.

a) Testes de Usabilidade

Os testes de usabilidade verificaram a facilidade de navegação e o entendimento da interface pelo usuário.

- Método: Foram realizados testes com voluntários simulando fluxos comuns, como navegar pela listagem de animais ou preencher um formulário de adoção.
- Objetivo: Identificar problemas de navegação, como botões confusos ou informações difíceis de encontrar.

b) Testes de Funcionalidade

Verificaram a funcionalidade das páginas e formulários integrados ao backend.

- Método: Testes manuais e automatizados para simular diferentes cenários, como a submissão de formulários com dados válidos e inválidos.
- Objetivo: Garantir que o sistema reage adequadamente a entradas do usuário e exibe mensagens de erro claras quando necessário.

c) Testes de Responsividade

Como o site deve ser acessado de diferentes dispositivos, foram realizados testes para garantir que as páginas funcionem bem em resoluções variadas.

- Ferramentas Utilizadas: Google Chrome DevTools e outros simuladores de dispositivos móveis.
- Objetivo: Assegurar que o layout e as funcionalidades se adaptem

corretamente a telas de tamanhos diferentes.

d) Testes de Interatividade

Os testes avaliaram os scripts em JavaScript, como validação de formulários e elementos interativos.

- Exemplo: Testar se o botão "adotar" redireciona para a página correta e se validações em campos obrigatórios são exibidas antes do envio do formulário.

3.7.IMPLANTAÇÃO DO SOFTWARE

A implantação do sistema é uma etapa crucial para disponibilizar a plataforma de adoção de animais aos usuários finais. Nesta fase, o software desenvolvido é configurado e transferido para um ambiente de produção, onde ficará acessível para administradores e adotantes. A implantação foi planejada para garantir acessibilidade, desempenho, e segurança, considerando cenários de uso local e em nuvem.

3.7.1.Estratégia de Implantação

A implantação do sistema foi planejada para ocorrer em duas etapas principais:

- **Ambiente Local:** Um servidor local para testes e validações internas antes da publicação oficial.
- **Ambiente na Nuvem:** Publicação em um provedor de nuvem para garantir acessibilidade remota e escalabilidade.

3.7.2.Implantação em Ambiente Local

No ambiente local, o sistema foi configurado para testes internos, garantindo que todas as funcionalidades estivessem operando conforme o esperado antes da disponibilização em ambiente de produção.

- **Servidor de Aplicação:** O sistema foi configurado para rodar no Tomcat

embutido do Spring Boot, evitando a necessidade de configuração de servidores externos.

- **Banco de Dados:** O H2 Database foi mantido em modo persistente, permitindo armazenar dados mesmo após o reinício do servidor, o que facilita os testes.
- **Acesso ao Sistema:** O sistema foi configurado para ser acessível por meio do endereço `http://localhost:8080`, utilizado por desenvolvedores para validações e ajustes finais.

3.7.3. Implantação em Ambiente na Nuvem (Azure)

Após a validação no ambiente local, o sistema foi configurado para ser publicado na nuvem, utilizando o **Microsoft Azure**, um provedor de nuvem amplamente utilizado por sua robustez e escalabilidade.

3.7.4. Processo de Teste Pós-Implantação

Para garantir o sucesso da implantação, foi realizado um ciclo de testes em ambiente de produção:

1. **Testes de Funcionalidade:** Validação de fluxos completos, como cadastro de usuários, listagem de animais e submissão de formulários.
2. **Testes de Performance:** Simulação de múltiplos acessos simultâneos para garantir que o sistema suporte a carga prevista.
3. **Testes de Segurança:** Verificação do funcionamento do SSL, proteção contra acessos não autorizados e conformidade com as práticas de segurança da informação.

3.8. CRONOGRAMA TÉCNICO

Etapa	Prazo Estimado	Entrega
Definição do Escopo	1ª Semana	Documento de Escopo e Arquitetura
Levantamento de Requisitos	2ª Semana	Documento de Requisitos
Especificação Funcional	3ª Semana	Documento de Especificação Funcional
Arquitetura do Software	4ª Semana	Estrutura inicial do projeto
Desenvolvimento do Software	5ª a 8ª Semana	Protótipo funcional
Testes do Software	9ª Semana	Relatório de Testes
Implantação do Software	10ª Semana	Sistema publicado

4. DESENVOLVIMENTO SISTEMA / SOFTWARE / APLICATIVO

4.1. LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos foi executado com base em análises de plataformas similares, pesquisa documental e modelagem de usuários. A definição das funcionalidades priorizou as necessidades da organização “**Mãos que Salvam**” e dos potenciais usuários do sistema.

Tipo de Requisito	Descrição
Requisitos Funcionais	
Consulta de Animais Disponíveis	Usuários poderão visualizar uma lista de animais com nome, idade, raça, fotos e descrições.
Exibição Detalhada de	Cada animal terá uma página exclusiva com

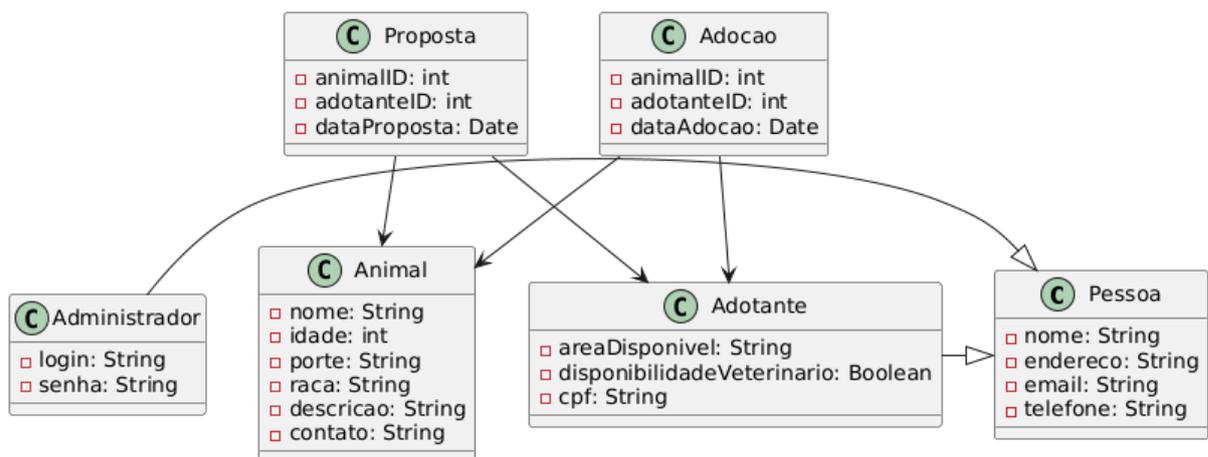
Animais	informações detalhadas e galeria de fotos.
Formulário de Interesse	Usuários preencherão um formulário com dados pessoais e informações sobre o ambiente de acolhimento.
Gerenciamento de Anúncios	Administradores poderão adicionar, editar e excluir anúncios de animais.
Gerenciamento de Contatos	Administradores acessarão os dados dos interessados para triagem das solicitações.
Confirmação de Adoção	Administradores poderão marcar animais como adotados e notificar outros interessados.
Gerenciamento Geral do Site	Ferramentas administrativas para atualizar conteúdos e controlar a página inicial.
Requisitos Não Funcionais	
Usabilidade	Navegação simples e intuitiva para todos os usuários.
Responsividade	Layout adaptado para desktop, tablet e smartphone.
Desempenho	Tempo de resposta inferior a 10 segundos.
Segurança	Sistema de autenticação para administradores e proteção de dados pessoais.
Confiabilidade	Disponibilidade de 99,9%, com no máximo 43,2 minutos de inatividade mensal.
Acessibilidade	Adequação às diretrizes WCAG para garantir acesso universal.

4.2.DIAGRAMA DE CLASSE

O sistema foi modelado utilizando um diagrama de classe baseado na abordagem de orientação a objetos. A classe **Pessoa** serve como base para os atores do sistema, armazenando atributos genéricos, como nome, endereço, e-mail e telefone. A classe **Administrador** herda de Pessoa e adiciona os atributos login e senha, garantindo acesso restrito às funcionalidades administrativas. A classe **Adotante**, também derivada de Pessoa, possui informações específicas, como CPF, área disponível para o animal e disponibilidade de veterinário.

A classe **Animal** centraliza os dados dos pets cadastrados, contendo atributos como nome, idade, porte, raça, descrição e contato do responsável. Já a classe **Proposta** gerencia solicitações de adoção, relacionando um adotante e um animal, com a data da proposta. Por fim, a classe **Adoção** registra a efetivação da adoção, também associando o adotante e o animal, mas com a data final da transação.

Figura 1 - Diagrama de classe



Fonte: Desenvolvida pelo autor

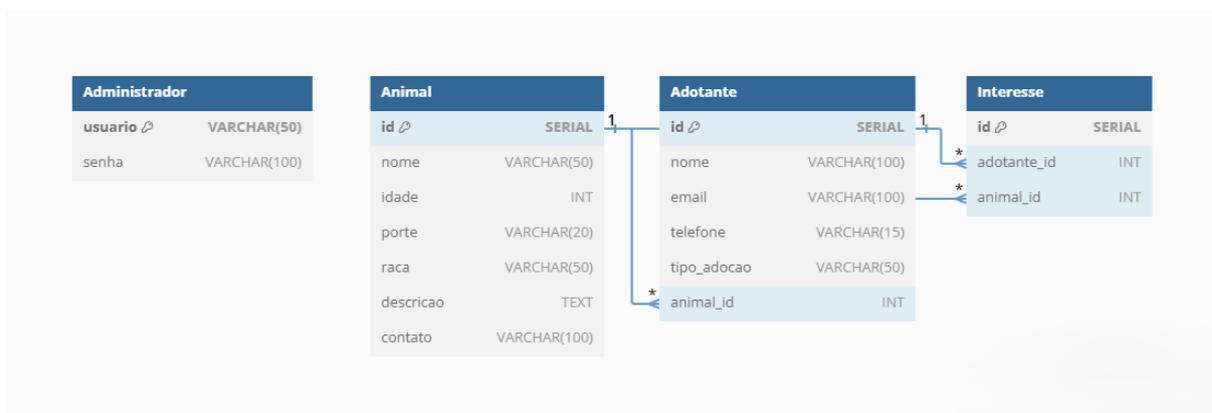
4.3. DIAGRAMA DE ENTIDADE RELACIONAL

O Diagrama de Entidade Relacional (DER) do sistema de adoção de animais foi projetado para ilustrar as relações entre as principais entidades envolvidas. O diagrama compreende quatro entidades principais: **Administrador**, **Animal**, **Adotante**, e **Interesse**, que são relacionadas entre si para viabilizar o gerenciamento

e a adoção de animais.

- a) **Administrador:** Representa a entidade responsável pela administração do sistema. Cada administrador tem um usuário e senha associados para autenticação e controle de acesso ao sistema.
- b) **Animal:** Contém informações sobre os animais disponíveis para adoção, como nome, idade, porte, raça, descrição e contato. Esta entidade é crucial para armazenar todos os dados dos animais cadastrados.
- c) **Adotante:** Representa a pessoa interessada em adotar um animal. Contém dados pessoais do adotante, como nome, e-mail, telefone e tipo de adoção (temporária ou permanente). O adotante pode estar vinculado a um ou mais animais.
- d) **Interesse:** Representa a interação entre o adotante e o animal. Através dessa entidade, é possível saber quais adotantes demonstraram interesse por quais animais.

Figura 2 - Diagrama de Entidade Relacional



Fonte: Desenvolvida pelo autor

4.3.1. Relações entre as Entidades

- **Administrador - Animal:** O administrador é responsável por cadastrar, editar e excluir informações dos animais disponíveis para adoção.
- **Adotante - Animal:** O adotante expressa interesse por um animal, criando uma relação entre essas entidades.
- **Interesse:** Esta tabela intermediária liga adotantes a animais, representando as ações de interesse por adoção. Cada registro de interesse relaciona um

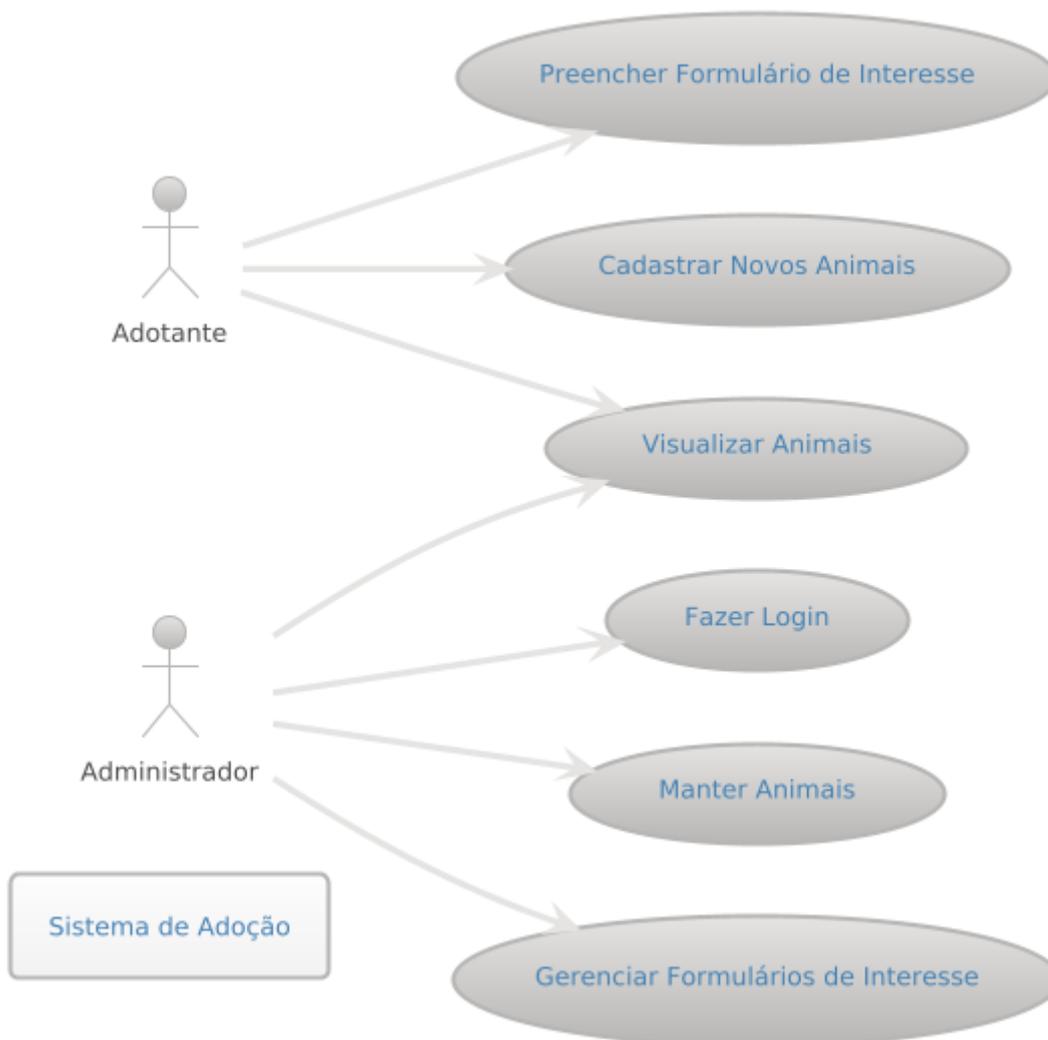
adotante a um animal específico.

Essas entidades e suas interações permitem o funcionamento do sistema de adoção, facilitando o gerenciamento dos animais e a conexão com os adotantes, além de garantir o controle e a segurança das informações.

4.4. DIAGRAMA DE CASO DE USO

O diagrama de caso de uso reflete as interações funcionais entre os atores do sistema e suas operações.

Figura 3 - Diagrama de Caso de Uso



a) Caso de Uso: Manter Animais

Ator: Administrador

Pré-condição: Administrador autenticado e com permissões adequadas para gerenciar os animais.

Fluxo principal:

1. Escolher a opção de manutenção de animais:

- O Administrador acessa o menu principal do sistema, que contém várias opções de gestão.
- O Administrador clica na opção "Manter Animais", que está localizada em uma área específica do painel administrativo.
- O sistema valida que o Administrador está autenticado e possui permissões para essa ação.
- Se o Administrador não tiver permissão, o sistema exibe uma mensagem de erro: "Acesso não autorizado".

2. Exibir lista de animais:

- O sistema consulta o banco de dados e exibe uma lista de todos os animais disponíveis para adoção.
- A lista pode incluir fotos e informações detalhadas dos animais, como: nome, idade, raça, porte físico, e uma breve descrição do animal.
- A lista é apresentada em formato de grade ou tabela, dependendo da preferência do Administrador. O sistema exibe também uma barra de rolagem ou paginação caso haja muitos animais cadastrados.

- Se não houver animais cadastrados, o sistema exibe uma mensagem: "Nenhum animal cadastrado".

3. Aplicar filtros de busca:

- O Administrador deseja filtrar os animais com base em certos critérios (raça, porte, idade, localização). O sistema fornece campos de filtro como dropdowns ou caixas de texto para preenchimento.
- O Administrador preenche os filtros desejados e clica no botão "Filtrar".
- O sistema processa os filtros e atualiza a lista de animais com base nos critérios especificados.
- Caso nenhum animal corresponda aos filtros, o sistema exibe uma mensagem: "Nenhum animal encontrado com os critérios fornecidos".

4. Exibir lista filtrada:

- O sistema apresenta a lista de animais que atendem aos filtros inseridos, com todas as informações visíveis e bem-organizadas.
- O Administrador pode clicar em um animal para visualizar mais detalhes, ou então realizar uma nova filtragem ou buscar um animal específico.

5. Cadastrar novo animal:

- O Administrador deseja cadastrar um novo animal para adoção. Para isso, clica no botão "Novo", disponível na interface de manutenção.
- O sistema valida se o Administrador está autenticado e com permissão para realizar a criação.
- O sistema redireciona o Administrador para um formulário de criação, que inclui campos como: nome do animal, idade, raça,

porte, sexo, descrição e foto.

- O sistema valida que todos os campos obrigatórios sejam preenchidos corretamente (exemplo: nome, foto e descrição são obrigatórios).
- Se o Administrador não preencher todos os campos obrigatórios, o sistema exibe uma mensagem de erro, como: "Por favor, preencha todos os campos obrigatórios".

6. Editar informações de um animal existente:

- O Administrador escolhe um animal da lista e clica no botão "Alterar" ao lado desse animal.
- O sistema exibe um formulário com os dados do animal já preenchidos, permitindo que o Administrador edite as informações.
- O Administrador altera os campos necessários (nome, raça, porte, localização.) e, ao finalizar, clica no botão "Salvar".
- O sistema valida os dados inseridos (exemplo: texto com mais de 3 caracteres, foto no formato correto) e, se tudo estiver correto, salva as alterações e atualiza a lista.
- Caso o Administrador cometa algum erro ao preencher, o sistema exibe mensagens de erro para cada campo inválido, como: "Nome inválido" ou "Foto não suportada".

7. Excluir um animal do sistema:

- O Administrador seleciona um animal da lista e clica no botão "Excluir".
- O sistema exibe uma janela de confirmação, perguntando se o Administrador realmente deseja excluir o animal.
- O Administrador confirma clicando no botão "Confirmar".
- O sistema remove permanentemente o animal do banco de dados e atualiza a lista de animais.

- Se a exclusão for bem-sucedida, o sistema exibe a mensagem: "Animal excluído com sucesso". Caso contrário, o sistema exibe um erro: "Erro ao excluir o animal. Tente novamente."

Fluxos alternativos:

- **Fluxo alternativo Criar:**

1. O Administrador clica no botão "Novo".
2. O sistema exibe um formulário de cadastro para o Administrador preencher com as informações do animal, como nome, idade, raça, porte, descrição e foto.
3. O Administrador preenche todos os campos obrigatórios e opcionalmente pode adicionar uma foto do animal. Após revisar as informações inseridas, o Administrador clica em "Salvar" para concluir o cadastro.
4. O sistema salva os dados do animal e retorna à lista de animais, agora incluindo o novo animal cadastrado.

- **Fluxo alternativo Alterar:**

1. O Administrador seleciona um animal da lista e clica no botão "Alterar".
2. O sistema exibe um formulário de edição com os dados atuais do animal selecionado.
3. O Administrador altera os dados conforme necessário (como modificar a descrição, atualizar a foto ou ajustar as informações de porte e idade).
4. Após editar as informações, o Administrador clica no botão "Salvar" para confirmar as alterações.

5. O sistema atualiza os dados do animal e retorna à lista com as informações atualizadas.

- **Fluxo alternativo Excluir:**

1. O Administrador seleciona um animal da lista e clica no botão "Excluir".
2. O sistema exibe uma janela de confirmação solicitando que o Administrador confirme se deseja realmente excluir o animal.
3. O Administrador confirma a exclusão clicando no botão "Confirmar".
4. O sistema remove o animal do banco de dados e atualiza a lista de animais, sem incluir o animal excluído.

b) Caso de Uso: Visualizar Animais

Ator: Administrador

Pré-condição: Acesso ao sistema, ou seja, o Adotante deve estar em uma página pública ou autenticado para visualizar os animais..

Fluxo principal:

1. Acessar a lista de animais disponíveis para adoção:

- O Adotante visualiza a interface principal do sistema e encontra um link ou botão para visualizar os animais disponíveis para adoção.
- Ao clicar nesse link, o sistema consulta o banco de dados e retorna uma lista com todos os animais disponíveis, exibindo as informações principais de cada animal (foto, nome, idade, raça e porte).
- O sistema pode organizar essa lista de maneira ordenada (por

nome, idade ou tipo de raça) ou fornece filtros para o Adotante refinar a busca.

2. Exibição detalhada das informações dos animais:

- Cada animal na lista tem uma foto e informações detalhadas, incluindo:
 1. Nome do animal
 2. Idade
 3. Raça
 4. Porte físico
 5. Descrição do animal, que inclui comportamentos e histórico
- Se o Adotante desejar saber mais sobre um animal específico, ele pode clicar no nome ou na foto, o que o direcionará a uma página de detalhes.
- A página de detalhes pode incluir informações adicionais, como vídeos ou depoimentos de outros adotantes sobre o animal.
- Caso o Adotante não encontre nenhum animal de seu interesse, o sistema pode sugerir outros animais similares ou exibir uma mensagem de "Nenhum animal encontrado no momento".

c) Caso de Uso: Preencher Formulário

Ator: Administrador

Pré-condição: Acesso ao sistema, ou seja, o Adotante deve estar em uma página pública ou autenticado para visualizar os animais.

Fluxo principal: Selecionar um animal para adoção:

- O Adotante visualiza a lista de animais e clica no animal de seu interesse para obter mais detalhes sobre ele.
- O sistema redireciona o Adotante para uma página com informações mais detalhadas sobre o animal selecionado.
- Na parte inferior da página, o Adotante encontra um botão ou link para "Preencher Formulário de Interesse".

1. Exibição do formulário de interesse:

- Ao clicar no botão, o sistema exibe um formulário solicitando informações pessoais do Adotante (nome, endereço, telefone, e-mail) e informações sobre o ambiente que o animal ocupará (exemplo: tipo de residência, se há outros animais etc.).
- O formulário contém campos obrigatórios e opcionalmente um campo para mensagens ou perguntas adicionais.

2. Preencher e confirmar o envio:

- O Adotante preenche todos os campos do formulário, que incluem informações pessoais e sobre o ambiente.
- Após preencher o formulário, o Adotante clica em "Enviar Formulário".
- O sistema valida se os campos obrigatórios foram preenchidos corretamente (por exemplo, um número de telefone válido ou um e-mail correto).
- Se algum campo estiver incompleto ou incorreto, o sistema exibe mensagens de erro, como: "Por favor, insira um número de telefone válido".

3. Processamento do formulário:

- Após a validação bem-sucedida, o sistema registra os dados no banco de dados.
- O sistema envia uma mensagem de confirmação ao Adotante,

informando que o formulário foi recebido e que o processo de adoção está sendo analisado.

- O Administrador recebe uma notificação informando sobre o envio do formulário e pode acessar os dados preenchidos para revisar a solicitação de adoção.

4.5.PROJETO TÉCNICO

Nesta seção, serão apresentadas as tecnologias utilizadas no desenvolvimento do projeto *Mãos que salvam*, com foco nas ferramentas Java, Spring, H2, Eclipse e Azure. Será discutido o histórico, os conceitos, as vantagens, as desvantagens e as justificativas para a escolha de cada uma dessas tecnologias para o projeto.

4.5.1.Java

Java é uma linguagem de programação orientada a objetos, desenvolvida pela Sun Microsystems na década de 1990 e atualmente mantida pela Oracle Corporation. Sua principal característica é a portabilidade, possibilitada pela Java Virtual Machine (JVM), que permite a execução do código em diferentes plataformas sem modificações (GOSLING et al., 2000).

Vantagens:

- Portabilidade: O código Java, uma vez compilado, pode ser executado em várias plataformas sem a necessidade de alterações, devido à JVM.
- Facilidade de aprendizado: A sintaxe simples e similar a outras linguagens populares torna o aprendizado mais acessível.

- Segurança: Java proporciona um ambiente seguro, com gerenciamento de exceções e controle de acesso (GOSLING et al., 2000).

Desvantagens:

- Desempenho: A execução na JVM pode ser mais lenta do que o código nativo em alguns casos.
- Consumo de memória: Java pode consumir mais memória, especialmente em sistemas com muitos dados.

A escolha do Java para o projeto *Mãos que salvam* foi motivada pela robustez, portabilidade e flexibilidade que a linguagem oferece. Sua compatibilidade com bancos de dados e a segurança proporcionada são essenciais para o desenvolvimento de sistemas corporativos escaláveis e confiáveis.

4.5.2.Spring Framework

O Spring Framework é um framework de desenvolvimento para a linguagem Java, lançado em 2003 por Rod Johnson, com o objetivo de simplificar a criação de aplicativos corporativos. Ele é baseado em conceitos como Inversão de Controle (IoC) e Injeção de Dependência (DI), que promovem o desacoplamento de componentes e facilitam o desenvolvimento de sistemas modulares e de fácil manutenção (JOHNSON, 2003).

Vantagens:

- Modularidade: O Spring oferece uma arquitetura modular, permitindo que módulos independentes sejam utilizados de forma combinada ou isolada (JOHNSON, 2003).
- Injeção de Dependência: O Spring facilita o desenvolvimento de

sistemas com baixo acoplamento e fácil testabilidade.

- Facilidade de integração: O Spring é compatível com diversos outros frameworks, como Hibernate e Thymeleaf, facilitando a integração (PIVOTAL, 2021).

Desvantagens:

- Curva de aprendizado: A flexibilidade e os recursos avançados do Spring podem exigir um tempo de aprendizagem considerável para iniciantes.
- Complexidade de configuração: Embora o Spring Boot tenha simplificado a configuração, sistemas mais complexos ainda podem demandar uma configuração inicial mais detalhada (PIVOTAL, 2021).

A escolha do Spring Framework se deve à sua flexibilidade e capacidade de criar sistemas escaláveis e integrados. Sua arquitetura permite que o projeto *Mãos que salvam* seja modular e facilmente expandido no futuro.

4.5.3.H2 Database

O H2 é um banco de dados relacional em memória, desenvolvido em Java e amplamente utilizado para projetos acadêmicos e protótipos. Ele suporta transações, é compatível com JDBC e SQL, e pode ser utilizado em modos embutido ou servidor (H2 DATABASE, 2021).

Vantagens:

- Desempenho em memória: O H2 oferece alto desempenho por operar diretamente na memória, o que é ideal para sistemas temporários e protótipos (H2 DATABASE, 2021).

- Fácil configuração: A configuração do H2 é simples e não exige grandes ajustes para começar a ser utilizado.
- Compatibilidade com SQL: O H2 é compatível com SQL, permitindo a utilização de consultas SQL padrão para manipulação de dados.

Desvantagens:

- Persistência limitada: Como banco de dados em memória, o H2 pode não ser ideal para sistemas de produção com grandes volumes de dados ou alta demanda de persistência.

A escolha do H2 para o projeto *Mãos que salvam* foi motivada pela sua simplicidade, desempenho rápido e compatibilidade com SQL, tornando-o ideal para o desenvolvimento e testes rápidos do sistema.

4.5.4.Eclipse IDE

O Eclipse é uma plataforma de desenvolvimento integrada (IDE) de código aberto, amplamente utilizada para o desenvolvimento em Java. Criada pela IBM em 2001, o Eclipse oferece suporte extensivo ao desenvolvimento Java, incluindo ferramentas de depuração, integração com sistemas de controle de versão e suporte a várias linguagens de programação (ECLIPSE FOUNDATION, 2021).

Vantagens:

- Suporte completo a Java: Eclipse oferece recursos completos para o desenvolvimento, depuração e integração de projetos Java.
- Extensibilidade: A plataforma permite a adição de plugins para suportar diversas tecnologias e funcionalidades (ECLIPSE FOUNDATION, 2021).

- Comunidade ativa: O Eclipse tem uma comunidade global de desenvolvedores, proporcionando ampla documentação e recursos de apoio.

Desvantagens:

- Desempenho: O Eclipse pode consumir muitos recursos em sistemas com projetos grandes, o que pode afetar o desempenho.
- Curva de aprendizado: Novos usuários podem achar a interface do Eclipse complexa devido à quantidade de funcionalidades disponíveis.

O Eclipse foi escolhido para o projeto devido ao seu suporte completo ao desenvolvimento em Java e à sua grande comunidade de suporte. A integração com o H2 Database e as ferramentas de depuração tornam o Eclipse uma ferramenta essencial para o desenvolvimento do sistema.

4.5.6. Azure – Plataforma de Nuvem

Azure é a plataforma de nuvem da Microsoft, lançada em 2010, que oferece uma ampla gama de serviços, incluindo computação, armazenamento e banco de dados. A plataforma permite que aplicativos sejam escalados conforme a demanda, sem a necessidade de manutenção de infraestrutura física (MICROSOFT, 2021).

Vantagens:

- Escalabilidade: Azure permite que a infraestrutura do aplicativo seja dimensionada de forma eficiente, conforme as necessidades do projeto.

- Segurança: Azure oferece diversos recursos de segurança e conformidade, garantindo a proteção dos dados e aplicações (MICROSOFT, 2021).
- Integração com ferramentas Microsoft: Azure integra-se facilmente com ferramentas como Visual Studio, simplificando o processo de desenvolvimento e gerenciamento.

Desvantagens:

- Curva de aprendizado: A plataforma possui uma vasta gama de serviços, o que pode tornar o seu aprendizado desafiador para iniciantes.
- Custo: Dependendo da utilização, os custos de uso de serviços em nuvem podem ser altos.

A escolha do Azure se deu pela sua escalabilidade, segurança e integração com ferramentas de desenvolvimento. A plataforma é ideal para hospedar o *Mãos que salvam*, permitindo expansão e confiabilidade para o sistema.

4.6.DESCRICÃO DAS INTERFACES

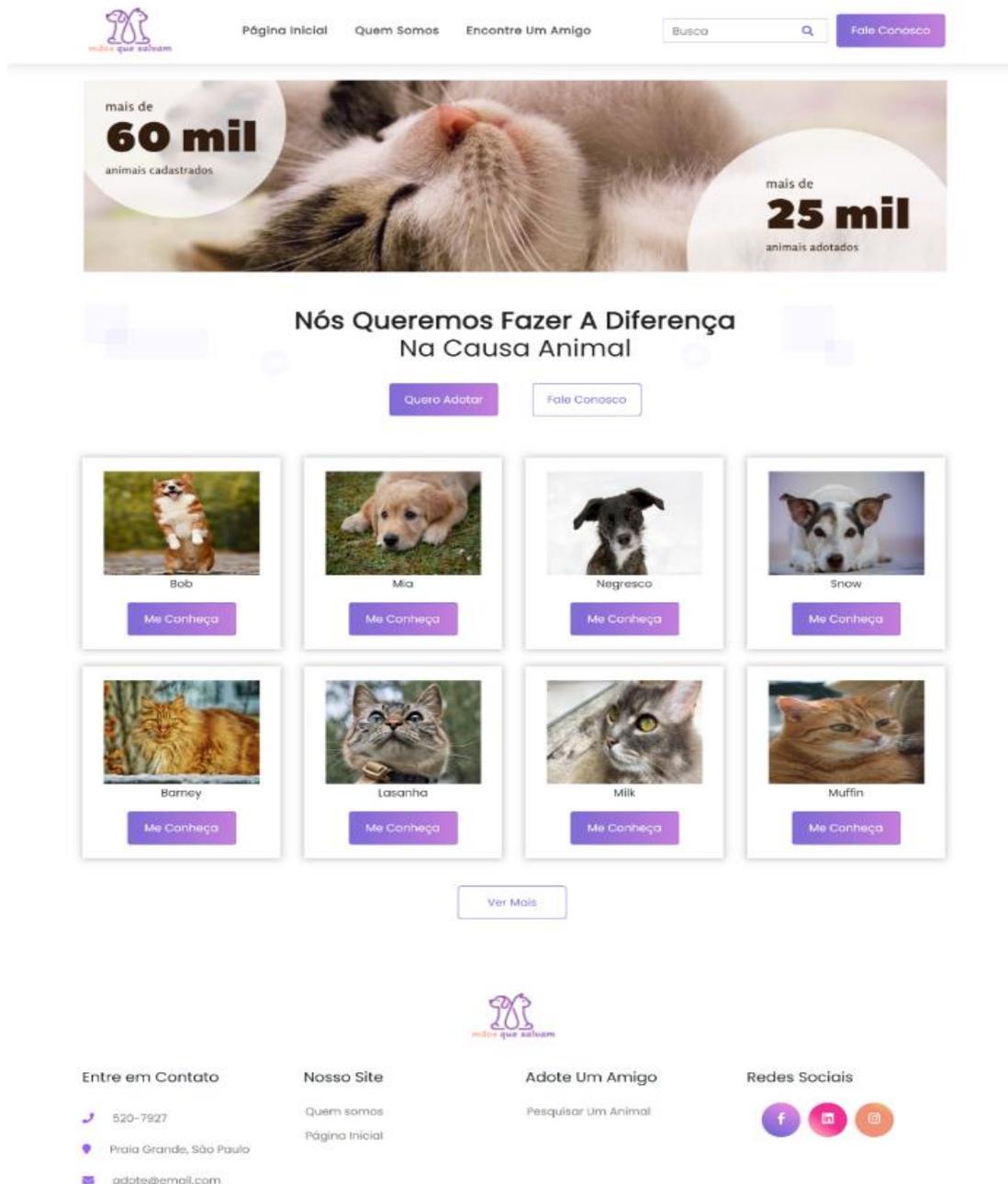
As interfaces do sistema *Mãos que salvam* foram projetadas para oferecer uma experiência de usuário intuitiva e funcional, atendendo às necessidades de ambos os usuários finais e administradores. A seguir, descrevem-se as telas principais do sistema, destacando suas funcionalidades e a interação esperada.

a) **Tela 1:** Página Inicial (Index)

A tela inicial, ou *index*, apresenta os animais mais recentemente cadastrados, além de fornecer informações sobre a iniciativa. Nessa interface,

o usuário pode realizar uma busca por animais utilizando o campo de pesquisa, facilitando a localização de animais por nome, raça ou outros critérios. A tela foi projetada com o objetivo de ser acessível e clara, permitindo aos usuários navegarem de forma eficiente e direta (ALMEIDA, 2022).

Figura 4 - Index



Fonte: Desenvolvida pelo autor

b) Tela 2: Menu de Administração

Para o administrador, foi incorporado um menu específico intitulado “Gerenciar Animais”. Essa funcionalidade permite a administração completa dos dados dos animais, com as opções de adicionar, editar e excluir registros.

A interface do administrador foi desenhada para garantir a eficiência no gerenciamento, com botões de fácil acesso e feedback visual claro sobre cada ação executada (SANTOS, 2023).

- Adicionar novo animal: O administrador pode cadastrar um animal, preenchendo informações como nome, raça e descrição.
- Editar animal existente: Permite a atualização de dados de animais previamente cadastrados.
- Excluir animal: Caso necessário, o administrador pode remover um animal da lista de adoção.

Essas ações são essenciais para a manutenção atualizada da base de dados de animais disponíveis para adoção, e foram organizadas de forma que o administrador possa realizar as operações com agilidade e sem erros.

c) **Tela 3:** Descrição do Animal

Na interface de detalhes, o sistema oferece uma visão aprofundada das informações sobre cada animal, incluindo fotos e descrições detalhadas. Essa tela foi projetada para oferecer clareza e facilitar a decisão dos usuários sobre a adoção dos animais, alinhando-se aos princípios de transparência e acessibilidade nas informações (GOMES, 2021).

Figura 5 - Descrição do animal

The image shows a web interface for an animal adoption service. At the top, there is a navigation bar with the logo 'maos que salvam' and menu items: 'Página Inicial', 'Quem Somos', and 'Encontre Um Amigo'. A search bar and a 'Fale Conosco' button are also present.

The main content area features a profile for a dog named 'Bob', who is 2 years old. The text describes Bob as a small, 2-year-old dog with a history of being abandoned and rescued. It mentions that Bob is a playful adolescent who likes to play with his toys and is currently in a temporary home. The text also states that Bob is a social animal who likes to share his space with other animals.

Below the text is a photograph of a brown and white dog sitting on a paved surface.

Below the profile is a light blue banner with the text 'Vamos seguir com a adoção?' and a subtext 'Preencha o formulário que logo entraremos em contato com você.' Below this banner is a white form with the following fields:

- Nome Completo: Ana Julia Terence
- Email: maosquesalvam@gmail.com
- Você quer oferecer um lar temporário ou fixo? (Dropdown menu)
- Tipo De Lar (Dropdown menu)
- Telefone: (11)99222-2222

At the bottom of the page, there is a footer with the logo 'maos que salvam' and four columns of contact information:

- Entre em Contato:** 520-7927, Praia Grande, São Paulo, adote@email.com
- Nosso Site:** Quem somos, Página Inicial
- Adote Um Amigo:** Pesquisar Um Animal
- Redes Sociais:** Facebook, Instagram, YouTube icons

Fonte: Desenvolvida pelo autor

As interfaces descritas foram projetadas com foco na usabilidade e eficiência. A tela inicial permite uma navegação rápida para os usuários, enquanto as funcionalidades de administração garantem ao gestor do sistema o controle completo sobre os cadastros de animais. A interação entre as telas é fluida e organizada, promovendo uma experiência eficiente tanto para os usuários quanto para os administradores.

5. CONSIDERAÇÕES FINAIS

O referido trabalho descreve os processos e atividades desenvolvidas no projeto acadêmico para demonstrar os conhecimentos adquiridos no curso tecnólogo de Análise e Desenvolvimento de Sistemas. O foco do projeto foi a criação de um website para facilitar a adoção de animais em situação de abandono, conectando pessoas dispostas a oferecer um lar a animais necessitados.

Foram aplicados conceitos de análise de requisitos, modelagem de diagramas, desenvolvimento de banco de dados e implementação de sistemas. As tecnologias escolhidas, como Java, Spring Framework e H2 Database, foram adequadas às necessidades do projeto, garantindo escalabilidade, segurança e facilidade de manutenção. A interface foi desenvolvida com foco na usabilidade, visando uma navegação intuitiva tanto para administradores quanto para usuários.

Este trabalho não apenas consolidou os conhecimentos adquiridos ao longo do curso, mas também contribuiu para a promoção da adoção responsável de animais, utilizando a tecnologia como ferramenta para solucionar um problema social relevante.

5.1. DESENVOLVIMENTOS FUTUROS

Embora o projeto tenha alcançado seus objetivos iniciais, há várias áreas que podem ser exploradas e aprimoradas em desenvolvimentos futuros para aumentar a eficácia e o alcance do sistema. A seguir, são apresentadas algumas sugestões de melhorias e expansões:

1. Integração com Redes Sociais

A integração do website com plataformas de redes sociais como

Facebook, Instagram e Twitter pode aumentar significativamente a visibilidade dos animais disponíveis para adoção. Isso pode ser feito através da implementação de APIs que permitam o compartilhamento automático de novos cadastros de animais nas redes sociais, aumentando as chances de adoção.

2. Sistema de Notificações

Implementar um sistema de notificações por e-mail ou SMS para informar os adotantes sobre novos animais disponíveis, atualizações no status de adoção e eventos relacionados à ONG. Isso pode melhorar o engajamento dos usuários e aumentar a taxa de adoção.

3. Aplicativo Mobile

Desenvolver um aplicativo mobile para Android e iOS pode tornar o sistema mais acessível e conveniente para os usuários. O aplicativo pode incluir funcionalidades como busca por animais, notificações push, e um sistema de geolocalização para encontrar animais próximos.

4. Análise de Dados e Relatórios

Implementar um módulo de análise de dados que permita aos administradores gerar relatórios sobre o número de adoções, tipos de animais mais adotados, tempo médio de adoção, entre outros. Isso pode ajudar a ONG a entender melhor o perfil dos adotantes e otimizar suas estratégias de adoção.

5. Sistema de Doações

Adicionar uma funcionalidade que permita aos usuários fazer doações diretamente pelo website. Isso pode incluir doações financeiras, bem como doações de itens necessários para o cuidado dos animais, como ração, medicamentos e brinquedos.

6. Parcerias com Clínicas Veterinárias

Estabelecer parcerias com clínicas veterinárias para oferecer serviços de

saúde animal a preços reduzidos para os adotantes. Isso pode ser integrado ao sistema, permitindo que os adotantes agendem consultas e acompanhem o histórico de saúde dos animais.

7. Sistema de Avaliação e Feedback

Implementar um sistema de avaliação e feedback onde os adotantes possam compartilhar suas experiências e avaliar o processo de adoção. Isso pode ajudar a ONG a identificar áreas de melhoria e aumentar a confiança dos futuros adotantes.

8. Expansão do Banco de Dados

Migrar do banco de dados H2 para um banco de dados mais robusto e escalável, como PostgreSQL ou MySQL, para suportar um maior volume de dados e acessos simultâneos, especialmente se o sistema for expandido para outras regiões ou estados.

9. Inteligência Artificial para Correspondência de Adoção

Utilizar algoritmos de inteligência artificial para sugerir animais aos adotantes com base em suas preferências e histórico de adoção. Isso pode aumentar a eficiência do processo de adoção e garantir que os animais sejam colocados em lares adequados.

10. Multilinguagem

Adicionar suporte a múltiplos idiomas no website para atender a uma audiência mais ampla, especialmente em regiões turísticas ou com uma população diversificada.

5.1.1. Sugestões de Futuras Tecnologias

1. Kubernetes para Orquestração de Contêineres

Kubernetes é uma plataforma de orquestração de contêineres que pode ser utilizada para gerenciar a escalabilidade e a disponibilidade do sistema. Utilizar Kubernetes pode garantir que o sistema seja altamente disponível e capaz de lidar com picos de tráfego, além de facilitar a implantação contínua e a gestão de microserviços.

Justificativa: Kubernetes permite a automação da implantação, escalabilidade e operações de contêineres de aplicativos, proporcionando alta disponibilidade e resiliência.

2. Elasticsearch para Busca Avançada

Elasticsearch é um mecanismo de busca e análise em tempo real que pode ser utilizado para melhorar a funcionalidade de busca no website. Integrar Elasticsearch pode permitir buscas mais rápidas e precisas, além de oferecer funcionalidades avançadas como autocompletar e sugestões de pesquisa.

Justificativa: Elasticsearch oferece uma busca rápida e escalável, melhorando a experiência do usuário ao procurar por animais específicos.

3. GraphQL para Consultas de Dados

GraphQL é uma linguagem de consulta para APIs que permite aos clientes solicitar exatamente os dados de que precisam. Utilizar GraphQL pode melhorar a eficiência das consultas de dados e reduzir a quantidade de dados transferidos entre o cliente e o servidor.

Justificativa: GraphQL permite consultas mais eficientes e flexíveis, melhorando o desempenho e a experiência do usuário.

4. Machine Learning para Previsão de Adoções

Implementar modelos de machine learning para prever quais animais têm maior probabilidade de serem adotados com base em dados históricos. Isso pode ajudar a ONG a focar seus esforços de marketing e aumentar a taxa de adoção.

Justificativa: Machine learning pode fornecer insights valiosos e previsões precisas, ajudando a otimizar o processo de adoção.

5. Blockchain para Transparência e Segurança

Utilizar blockchain para registrar todas as transações de adoção pode aumentar a transparência e a segurança do sistema. Isso pode incluir o registro de informações sobre os animais, adotantes e processos de adoção, garantindo a integridade dos dados.

Justificativa: Blockchain oferece um registro imutável e transparente, aumentando a confiança e a segurança no processo de adoção.

6.REFERÊNCIAS BIBLIOGRÁFICAS

GOSLING, J.; YELLIN, B.; BRAUN, G. **The Java Programming Language**. 4. ed. Boston: Addison-Wesley, 2000.

JOHNSON, R. **Expert One-on-One: J2EE Development without EJB**. 1. ed. Indianapolis: Wrox Press, 2003.

PIVOTAL. **Spring Framework Documentation**. 2021. Disponível em: <https://docs.spring.io/spring-framework/docs/current/reference/html/>. Acesso em: 24 nov. 2024.

H2 DATABASE. **H2 Database Engine Documentation**. 2021. Disponível em: <https://h2database.com/html/main.html>. Acesso em: 24 nov. 2024.

ECLIPSE FOUNDATION. **Eclipse IDE Documentation**. 2021. Disponível em: <https://www.eclipse.org/documentation/>. Acesso em: 24 nov. 2024.

MICROSOFT. **Azure Documentation**. 2021. Disponível em: <https://azure.microsoft.com/en-us/documentation/>. Acesso em: 24 nov. 2024.

ALMEIDA, João. **Desenvolvimento de interfaces para sistemas web**. São Paulo: Editora Acadêmica, 2022.

GOMES, Ana. **Acessibilidade em sistemas digitais**. Rio de Janeiro: Livros Técnicos, 2021.

SANTOS, Carlos. **Gestão de dados e interfaces de administração**. Curitiba: Editora Tecnologia, 2023

SILVA, João. **Análise e desenvolvimento de sistemas: conceitos e práticas**. São Paulo: Editora Acadêmica, 2021.

COSTA, Maria. **Arquitetura de sistemas web: princípios e tecnologias**. Rio de Janeiro: Editora Tecnologia, 2020.

SANTOS, Pedro. **Banco de dados relacionais e desenvolvimento de sistemas**. Curitiba: Editora Universitária, 2022.