



Centro Universitário Senac

Danielle Rodrigues Pereira

Eduardo Jonson Bartikoski

Priscila Albuquerque Cardoso

Rafael Rodrigues dos Santos

Raphaely Silva Carneiro

Projeto Integrador – 2ª etapa – 1º semestre de 2025:

Desenvolvimento de sistemas orientado a dispositivos móveis e baseados na web:

Aplicativo Web e Móvel de Serviços de *Pet Sitters*

São Paulo

2025



Danielle Rodrigues Pereira
Eduardo Jonson Bartikoski
Priscila Albuquerque Cardoso
Rafael Rodrigues dos Santos
Raphaely Silva Carneiro

Projeto Integrador – 2ª etapa – 1º semestre de 2025:
Desenvolvimento de sistemas orientado a dispositivos móveis e baseados na web:
Aplicativo Web e Móvel de Serviços de Pet Sitters

Trabalho de conclusão de curso
apresentado ao Centro Universitário
Senac – como exigência parcial para
obtenção do grau em Análise e
Desenvolvimento de Sistemas.

Professor orientador: Carlos William
Ferreira de Lima

Professora tutora: Ruth Garcia

São Paulo

2025



SUMÁRIO

INTRODUÇÃO	4
1. REVISITA À PRIMEIRA ETAPA DO PROJETO INTEGRADOR.....	5
1.1 Definição da prova de conceito	5
2. PREPARAÇÃO DO AMBIENTE DE DESENVOLVIMENTO DO PROJETO	6
2.1 Seleção de tecnologia de front-end	6
2.2 Seleção de tecnologia de back-end	6
2.3 Seleção de banco de dados (BD)	7
3. DESENVOLVIMENTO DO FRONT-END DO PROJETO.....	8
4. DESENVOLVIMENTO DO BACK-END DO PROJETO	10
5. APRESENTAÇÃO DO PROJETO CONCLUÍDO	11
6. LINK DO PROJETO NO GITHUB	12
6.1 Código fonte do projeto.....	12
CONCLUSÃO.....	13
REFERÊNCIAS BIBLIOGRÁFICAS	14

INTRODUÇÃO

Nos dias de hoje, a convivência com animais de estimação se tornou uma parte fundamental da vida de muitas pessoas. Com a correria do cotidiano, a demanda por serviços que auxiliem os tutores a cuidar de seus pets aumentou significativamente. Nesse cenário, que já apresentamos na primeira parte deste projeto, o serviço de *Pet-Sitter* surgiu como uma solução prática e eficiente, proporcionando não apenas o bem-estar aos animais, mas também a tranquilidade e segurança aos seus donos.

Como já esclarecido anteriormente, um *pet-sitter* é um cuidador especializado que se dedica a administrar as necessidades dos pets em momentos que seus tutores não podem estar presentes, seja em períodos de férias, viagens a trabalho ou até mesmo em situações do dia a dia, quando os tutores não estão disponíveis ou estão impossibilitados de tais cuidados. Assim, o mercado de *pet-sitting* tem se expandido rapidamente, por causa da crescente demanda por esse tipo de comodidade, oferecendo oportunidades para profissionais dedicados e, ao mesmo tempo, criando valor para os tutores que buscam garantir a saúde e bem-estar de seus animais.

Neste projeto integrador, propomos o desenvolvimento de um sistema web que conecta tutores de pets a cuidadores qualificados, facilitando a busca e a contratação dos serviços de forma rápida, simples e segura. O sistema permitirá que os usuários encontrem cuidadores disponíveis em sua região, visualizem avaliações e experiências de outros tutores, e realizem reservas de forma prática.

Além disso, irá ressaltar a importância da personalização no cuidado com os animais, permitindo que os tutores escolham cuidadores que atendam a requisitos específicos, como experiência com determinadas raças ou necessidades especiais. Acreditamos que, por meio dessa plataforma, poderemos não apenas atender a uma demanda crescente, mas também contribuir para a construção de uma comunidade de amantes e responsáveis por animais, promovendo o bem-estar e a segurança dos pets em um ambiente de confiança.

Assim, nosso projeto se apresenta não apenas uma ferramenta tecnológica, mas uma resposta à necessidade de cuidados qualificados e humanos que garantam a felicidade de nossos companheiros de quatro patas.

1. REVISITA À PRIMEIRA ETAPA DO PROJETO INTEGRADOR

Em revisita à primeira etapa do nosso projeto integrador, continuamos com a proposta em desenvolver a prova de conceito de um sistema web voltado para serviços de *pet-sitting*, que deverá ter como base teórica a jornada de uso de uma das personas que criamos na etapa apresentada anteriormente. O foco será a jornada da persona Maria de Castro, 35 anos, professora universitária, moradora de uma grande cidade, que tem a tutela de um cachorro SRD e um gato da raça persa. Ela busca cuidados para seus pets durante suas longas jornadas de trabalho e eventuais viagens.

Num sistema web voltado para esse tipo de prestação de serviços ou para uma empresa deste ramo, como já descrevemos anteriormente, acreditamos que funcionalidades que permitam ao usuário criar e editar seu perfil, fazer seu login, cadastrar senha, cadastrar seus pets e fazer alterações conforme a necessidade, procurar por profissionais *petsitters* disponíveis, assim como visualizar seus perfis e avaliações recebidas, enviar solicitações de agendamentos e fazer pagamentos, sejam utilidades essenciais para uma boa execução da tecnologia.

Assim, nos mantemos no objetivo exposto na primeira etapa deste projeto, que é apresentar um sistema web de serviços de babá de *pets* que tem por principal objetivo proporcionar uma solução prática, confiável, segura e eficiente que atenda às necessidades tanto dos tutores quanto dos profissionais cuidadores de animais.

1.1. Definição da prova de conceito

Entendendo prova de conceito como sendo a apresentação de uma versão básica do sistema web em desenvolvimento. A seguir mostraremos uma versão que permitirá ao usuário:

- 1) Criar o perfil, com seus dados pessoais;
- 2) Logar-se com inserção de senha de acesso;
- 3) Cadastrar seu(s) pet(s);
- 4) Enviar solicitação de agendamentos e
- 5) Listar os agendamentos.

O sistema deve permitir que o usuário comece e termine esse processo de forma simples, intuitiva, rápida e eficiente.

2. PREPARAÇÃO DO AMBIENTE DE DESENVOLVIMENTO DO PROJETO

2.1. Seleção de tecnologia de front-end

Para desenvolvermos a prova de conceito com as funcionalidades apresentadas no item anterior, para a escolha da tecnologia de *front-end* tentamos levar em conta fatores cruciais como: facilidade de desenvolvimento, escalabilidade, suporte a dispositivos móveis e integração com *back-end*, assim, optamos pelo React.js, que é uma biblioteca JavaScript desenvolvida pelo Facebook em 2011. Essa biblioteca é extremamente popular para a construção de interfaces interativas de usuário.

Ela tem como vantagens o alto desempenho e eficiência através do uso de um DOM virtual; tem ampla comunidade e rico ecossistema de bibliotecas; é flexível para integração com outras bibliotecas e frameworks; seus componentes reutilizáveis facilitam a manutenção e escalabilidade do aplicativo; por tudo isso, é ideal para uma aplicação que pode vir a se expandir no futuro, pois permite uma estrutura modular e reutilizável.

2.2. Seleção de tecnologia de back-end

Para o *back-end* do nosso sistema web, cuja interface do *front-end*, como já expusemos, pretendemos construir com o React.js, escolhemos a tecnologia Node.js com Express.js, pois o Node.js é uma plataforma baseada em JavaScript que permite executar códigos no servidor, e o Express.js é um framework minimalista para Node.js, característica que facilita a criação de aplicações.

Essa opção tem como vantagens a consistência, pois usaremos o JavaScript no *front-end*, e ao utilizar Node.js no *back-end* será permitido que tenhamos uma única linguagem em ambos os lados; há também a questão do desempenho, pois o Node.js é conhecido por sua alta performance, especialmente em aplicações que requerem I/O (*input/output*) muito ativos, como chamadas de API; outra vantagem é a existência de uma vasta quantidade de pacotes disponíveis no NPM (*Node Package Manager*), o que facilita a implementação de funcionalidades como autenticação, notificações e integração de pagamentos.

Assim, acreditamos que a combinação entre o React.js e o Node.js com Express.js, permitirá flexibilidade, integração, desenvolvimento ágil e eficiente, além de uma grande base de conhecimento acessível para qualquer problema que possamos encontrar ao longo do processo de desenvolvimento, e também é muito benéfico se pretendermos escalar o aplicativo no futuro.

2.3. Seleção de banco de dados (BD)

No sistema web que estamos desenvolvendo, no qual pretendemos utilizar o React.js no *front-end* e Node.js com Express.js no *back-end*, a escolha do banco de dados é muito importante para garantir desempenho, escalabilidade, confiabilidade e segurança. Optamos pelo MySQL devido a suas diversas vantagens que se alinham perfeitamente às necessidades deste projeto.

Temos a vantagem da compatibilidade e integração, pois o MySQL é amplamente compatível com Node.js, permitindo uma integração eficiente por meio de bibliotecas como Mysql2 e Sequelize. Isso facilita a comunicação entre o back-end e o banco de dados, garantindo consultas rápidas e seguras.

Também temos as questões de desempenho e eficiência, pois o MySQL é conhecido por sua alta performance em operações de leitura e escrita, sendo otimizado para aplicações web que exigem respostas rápidas. Como o *PetSittingApp* lida com informações de usuários, pets e serviços, um banco de dados eficiente é essencial para garantir uma experiência fluida.

Por ser um banco de dados relacional, o MySQL permite a organização estruturada dos dados, garantindo integridade referencial e evitando inconsistências. Isso é fundamental para um sistema que gerencia múltiplas entidades, como clientes, cuidadores e agendamentos.

O MySQL é altamente escalável, permitindo que o *PetSittingApp* cresça sem comprometer o desempenho. Ele suporta particionamento, replicação e otimizações que garantem estabilidade mesmo com um grande número de usuários ativos.

Por fim, temos como vantagem os recursos avançados de controle de acesso, criptografia e autenticação. O MySQL protege os dados dos usuários contra acessos não autorizados. Isso é essencial para garantir a privacidade e segurança das informações no aplicativo.

3. DESENVOLVIMENTO DO FRONT-END DO PROJETO

Imagem 1 – Árvore de diretório do front-end

The screenshot displays a file explorer interface for the 'petsitting-frontent' directory. The left sidebar shows a tree view of the project structure, including folders like 'node_modules', 'react-dom', and 'cjs'. The main area on the right shows a commit history table for the 'petsitting-frontent' branch, which is 4 commits ahead of and 1 commit behind the 'main' branch. The table lists the following files and their last commit messages and dates:

Name	Last commit message	Last commit date
..		
node_modules	Mensagem do commit	1 hour ago
src	Removendo .env do controle de versão	yesterday
package-lock.json	Mensagem do commit	1 hour ago
package.json	Mensagem do commit	1 hour ago

Below this table, there is another identical table showing the same commit history for the 'petsitting-frontent' branch. The interface also includes a search bar at the top left and a 'Contribute' button at the top right.

The image displays two screenshots of a Git web interface, likely GitHub, showing commit history and file structure.

Top Screenshot:

- Header:** This branch is 4 commits ahead of, 1 commit behind main.
- Table:**

Name	Last commit message	Last commit date
..		
node_modules	Mensagem do commit	1 hour ago
src	Removendo .env do controle de versão	yesterday
package-lock.json	Mensagem do commit	1 hour ago
package.json	Mensagem do commit	1 hour ago
- File Structure (Left Panel):**
 - index.js
 - jsx-dev-runtime.js
 - jsx-dev-runtime.react-server.js
 - jsx-runtime.js
 - jsx-runtime.react-server.js
 - package.json
 - react.react-server.js
 - scheduler
 - scheduler-unstable_mock.d...
 - scheduler-unstable_mock.pr...
 - scheduler-unstable_post_tas...
 - scheduler-unstable_post_tas...
 - scheduler.development.js
 - scheduler.native.developme...
 - scheduler.native.production.js
 - scheduler.production.js
 - LICENSE
 - README.md
 - index.js
 - index.native.js
 - package.json
 - unstable_mock.js
 - unstable_post_tasks.js
 - package-lock.json

Bottom Screenshot:

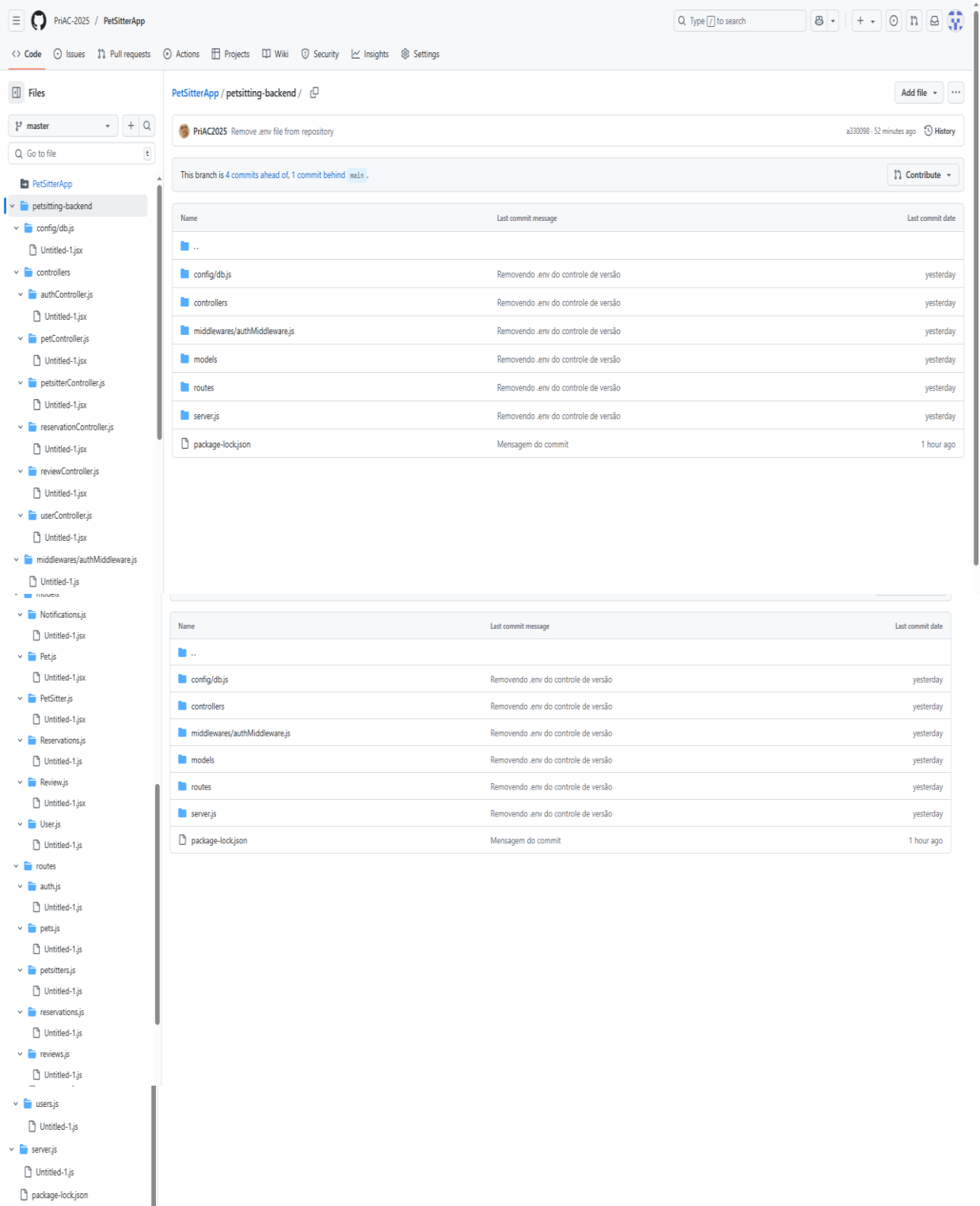
- Header:** This branch is 4 commits ahead of, 1 commit behind main.
- Table:**

Name	Last commit message	Last commit date
..		
node_modules	Mensagem do commit	1 hour ago
src	Removendo .env do controle de versão	yesterday
package-lock.json	Mensagem do commit	1 hour ago
package.json	Mensagem do commit	1 hour ago
- File Structure (Left Panel):**
 - src
 - App.js
 - Untitled-1.jsx
 - components
 - Auth
 - Login.js
 - Untitled-1.jsx
 - Register.js
 - Untitled-1.jsx
 - Pet/PetForm.js
 - Untitled-1.jsx
 - PetSitters/PetSitterList.js
 - Untitled-1.jsx
 - Profile
 - PetSitterProfile.js
 - Untitled-1.jsx
 - UserProfile.js
 - Untitled-1.jsx
 - Reservation/ReservationForm.js
 - Untitled-1.jsx
 - index.js
 - Untitled-1.jsx
 - Untitled-2.jsx
 - navigation
 - navigation
 - AppNavigator.js
 - Untitled-1.jsx
 - MobileNavigator.js
 - Untitled-1.jsx
 - screens
 - LoginScreen.js
 - Untitled-1.jsx
 - PetProfileScreen.js
 - Untitled-1.jsx
 - PetsitterDetailScreen.js
 - Untitled-1.jsx
 - PetsitterListScreen.js
 - Untitled-1.jsx
 - RatingScreen.js
 - Untitled-1.jsx
 - SignupScreen.js
 - Untitled-1.jsx
 - services/api.js
 - Untitled-1.js
 - package-lock.json
 - package.json

Fonte: Elaboração própria

4. DESENVOLVIMENTO DO BACK-END DO PROJETO

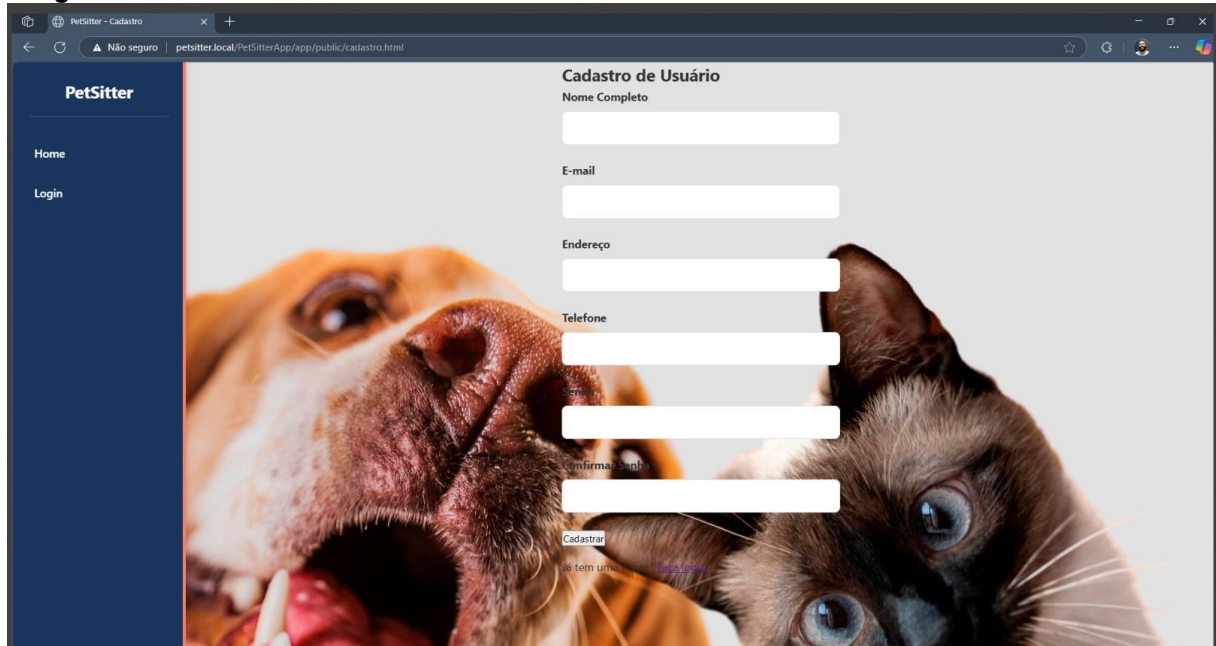
Imagem 2 – Árvore de diretório do back-end



Fonte: Elaboração própria

5. APRESENTAÇÃO DO PROJETO CONCLUÍDO

Imagem 3: Tela de cadastro de usuário



PetSitter

Home

Login

Cadastro de Usuário

Nome Completo

E-mail

Endereço

Telefone

Senha

Confirmação de senha

Cadastrar

Fonte: <https://br.freepik.com/fotos-gratis/colagem-de-animal-de-estimacao-bonito-isolada>

O link do vídeo de apresentação do projeto entra-se no endereço:
<https://youtu.be/5eZ4EhFStCA>

6. LINK DO PROJETO NO GITHUB

6.1. Código fonte do projeto

O código-fonte deste Projeto Integrador: *PetSitterApp*, está disponível no GitHub, sob o endereço: <https://github.com/PriAC-2025/PetSitterApp/tree/main>, o que permite o acesso ao processo de desenvolvimento deste sistema, onde está exposta toda a estrutura dos códigos que criamos, para alcançarmos o que foi proposto no item 1.1 deste projeto. E suas funcionalidades estão demonstradas com o vídeo da prova de conceito disponível em: <https://youtu.be/5eZ4EhFStCA>.

Enfim, os códigos-fontes que usamos na criação da página do *PetSitterApp*, traduzem rapidez e eficiência ao desenvolvedor, e uma página dinâmica e intuitiva ao usuário, com escalabilidade, que permite que o sistema cresça sem grandes dificuldades, e a reutilização de código que, conforme a necessidade, podem ser usados em diversas partes do sistema.

CONCLUSÃO

O desenvolvimento do *PetSitterApp* foi guiado pela preocupação genuína que temos com o bem-estar animal e pela necessidade de oferecer comodidade e confiança aos tutores que enfrentam desafios diários para conciliar suas rotinas com os cuidados de seus pets.

A construção desta fase do projeto foi fundamentada no *Design Thinking*, tomando como referência as características e necessidades das personas apresentadas na primeira etapa deste, que têm vidas ativas e extensas jornadas de trabalho, somadas a eventuais viagens, o que frequentemente os impedem de garantir atenção contínua aos seus animais de estimação. Essa abordagem nos permitiu estruturar um produto alinhado às necessidades reais de usuários em situações semelhantes.

Para a codificação do aplicativo, optamos por React.js no *front-end* devido à sua eficiência na criação de interfaces dinâmicas e responsivas. No *back-end*, utilizamos Node.js com Express.js, garantindo escalabilidade e um fluxo ágil de comunicação entre os serviços. Já para o banco de dados, escolhemos o MySQL, que garante desempenho, escalabilidade, confiabilidade e segurança.

Além das escolhas tecnológicas, o uso de ferramentas como Miro, para organização e visualização de ideias, na primeira fase do projeto, posteriormente, o uso do Visual Studio Code, para desenvolvimento eficiente, e do GitHub, para versionamento e colaboração, trouxeram aprendizados valiosos e aprimoraram nossa experiência ao longo do projeto.

Por fim, a imersão na vasta bibliografia especializada não apenas fortaleceu a fundamentação teórica do nosso trabalho, mas também ampliou significativamente nosso conhecimento dentro das áreas aplicadas na Análise e Desenvolvimento de Sistemas.

Esse projeto representa um avanço concreto na busca por soluções tecnológicas que promovam o bem-estar animal, ao mesmo tempo em que oferecem praticidade aos tutores que enfrentam desafios cotidianos no cuidado com seus pets, pois essas soluções têm desempenhado um papel crucial tanto no mercado de trabalho como na prestação de serviços, impulsionando a eficiência, a automação e inovação para os que a empregam em seu dia a dia.

REFERÊNCIAS BIBLIOGRÁFICAS

ALGAR TELECOM. Tecnologias do dia a dia: benefícios e aplicações práticas. 22 de agosto de 2024. **Algar Telecom**. Disponível em: <https://blog.algartelecom.com.br/tecnologias-do-dia-a-dia-beneficios/> Acesso em: 04 de maio de 2025.

APETI. Tecnologia no mercado de trabalho: vantagens e desvantagens de aplicá-la. 21 de novembro de 2023. **APETI**. Disponível em: <https://www.apeti.org.br/tecnologia-no-mercado-de-trabalho-vantagens-e-desvantagens-de-aplica-la> Acesso em: 04 de maio de 2025.

AWARI. Aprenda a Utilizar o React Js com Mysql: guia completo para desenvolvedores. 24 de agosto de 2023. **AWARI**. Disponível em: [Aprenda a Utilizar o React Js com Mysql: Guia Completo para Desenvolvedores](#) Acesso em: 18 de maio de 2025.

AZEVEDO, Marcelo Teixeira de. **Sistemas distribuídos** – São Paulo: Editora Senac São Paulo, 2021. (Série Universitária)

DUARTE, Luiz. Como usar Node.js + MySQL. 07 de outubro de 2020. **Luiztools**. Disponível em: [Como usar Node.js + MySQL – LuizTools](#) Acesso em: 18 de maio de 2025.

IBRAHIM, [Muhammed Salih Seyed](#). ReactJS – advantages and disadvantages. 12 de novembro de 2023. **Medium**. <https://medium.com/@muhammedsalihseyedibrahim/reactjs-advantages-disadvantages-16f479b3aa47> Acesso em 01 de maio de 2025.

LEITE, Leonardo Alexandre Ferreira. **Banco de dados: bases conceituais e pragmáticas** – São Paulo: Editora Senac São Paulo, 2020. (Série Universitária)

MASIERO, Andrey Araujo. **Arquitetura de software** – São Paulo: Editora Senac São Paulo, 2021. (Série Universitária)

MDNWEBDOCS. Express/Node Introduction. **mdn web docs**. Disponível em: https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Server-side/Express_Nodejs/Introduction Acesso em: 04 de maio de 2025.

NEVES, Vinícios. React: o que é, como funciona e um guia dessa popular ferramenta JS. 17 de janeiro de 2023. **Alura**. <https://www.alura.com.br/artigos/react-js> Acesso em 01 de maio de 2025.

REACT. Documentação do React Nativo. **React**. Disponível em: <https://reactnative.dev/docs/getting-started> Acesso em: 02 de maio de 2025.

VARELLA, Walter Augusto. **Infraestrutura de TI** – São Paulo: Editora Senac São Paulo, 2019. (Série Universitária)