



Associação Propagadora Esdeva
Centro Universitário Academia – UniAcademia
Curso de Engenharia de Software
Grupo de Estudos

- DESENVOLVIMENTO ÁGIL DE APLICAÇÕES COM REACT E JAVA - FOSTER PET

Matheus de Oliveira Toledo¹

Centro Universitário Academia, Juiz de Fora, MG

Gabriel Marques França²

Centro Universitário Academia, Juiz de Fora, MG

Marco Antonio Pereira Araújo³

Centro Universitário Academia, Juiz de Fora, MG

RESUMO

Este trabalho apresenta uma proposta de uma aplicação web que sirva como uma plataforma que permita a realização de adoções e doações de animais de forma online, com intuito de mitigar os números de animais em condição de vulnerabilidade, e, também, fornecer um ambiente virtual propício e seguro para o processo de adoção e doação de animais.

É dissertado, também, como foi planejado o processo de desenvolvimento, desde o levantamento de requisitos, validação dos requisitos, criação de trilhas de treinamento, plano de desenvolvimento e quais tecnologias foram empregadas na criação da plataforma.

Palavras-chave: Sistema de adoção de animais. React. Java. Aplicação Web.

¹ Discente do Curso de Engenharia de Software, do Centro Universitário Academia.

² Discente do Curso de Engenharia de Software, do Centro Universitário Academia.

³ Docente do Curso de Engenharia de Software, do Centro Universitário Academia. Orientador.

ABSTRACT

This work presents a proposal for a web application that serves as a platform that allows the adoption and donation of animals online, in order to mitigate the numbers of animals in vulnerable conditions, and also to provide a favorable and secure virtual environment for the animal adoption and donation process.

It is also discussed how the development process was planned, from requirements gathering, requirements validation, creation of training trails, development plan itself and which technologies were used in the development of the platform.

Keywords: Animal Adoption System. React. Java. Web application.

1 INTRODUÇÃO

É sabido que a relação interespecie entre humanos e animais é uma antiga união que acompanhou o processo civilizatório humano, proporcionando inúmeros e variados benefícios (FARACO, 2008). Um dos benefícios buscados é a companhia que os animais nos proporcionam, sendo geralmente testemunhado pelas nossas relações com cães, gatos e outros animais.

Sendo tradicional no Brasil a companhia de cães ou gatos, esse vínculo se tornou tão forte que até mesmo, atualmente, pessoas se consideram “mães” ou “pais” de pets; uma maneira mais popular para o termo família multiespecie, que significa família constituída por pessoas e animais (FARACO, 2008; ANDRADE *et al.*, 2020).

O Brasil, de acordo com o Instituto Pet Brasil (2019), conta com uma população de 140 milhões de pets, sendo que, destes, 3,9 milhões são animais em condição de vulnerabilidade (ACV), ou seja, que vivem sob a amparo de famílias classificadas como abaixo da linha da pobreza, ou que vivem nas ruas, mas recebem qualquer tipo de atenção. Animais sob os cuidados de ONG⁴s de proteção animal ou abandonados às ruas não estão incluídos na contagem de ACV.

O abandono desses animais ocasiona uma superpopulação nas ruas, trazendo inúmeros transtornos e, dentre eles, ameaças à saúde pública (SOUZA, 2014). Mesmo defronte tal cenário, de acordo a pesquisa de Alberes Cavalcanti (2021), as

⁴ Organização não governamental.

Organizações sociais se permitem confiar na existência de um cenário que favoreça a transformação desse paradigma. Essas organizações têm reconhecida utilidade, mas lutam contra a crônica falta de recursos, o que torna precioso qualquer tipo de subsídio, principalmente aqueles mais especializados (ANDRADE *et al.*, 2020).

Dessa forma, este trabalho propõe amenizar esses números viabilizando a doação e adoção de animais por meio de uma plataforma online, onde todo o processo será documentado e legitimado com um contrato ao final.

A plataforma de doação e adoção de animais Foster Pet permitirá que pessoas interessadas tanto em adotar, quanto em doar, possam fazê-lo na plataforma online, promovendo um espaço virtual tal como os espaços físicos onde feiras de adoção são realizadas. A plataforma permitirá que ONGs de proteção animal possam anunciar os animais abrigados, para que estes tenham visibilidade e maior oportunidade de serem adotados.

Ao final do processo, é desejável a emissão de um documento contratual que legitime o processo de doação/adoção entre as partes envolvidas.

2 REFERENCIAL TEÓRICO

2.1 Ciclo de vida de software

A NBR ISSO/IEC 12207:1998 define o ciclo de vida como uma “Estrutura contendo processos, atividades e tarefas envolvidas no desenvolvimento, operação e manutenção de um produto de software, abrangendo a vida do sistema, desde a definição de seus requisitos até o término de seu uso.” (DEVMEDIA, acesso em 2021).

Os modelos de ciclo de vida mais comuns são os modelos: Cascata, Evolutivo, Incremental, Prototípico e Espiral. Foi escolhido o modelo em Espiral como ciclo de vida do software em desenvolvimento no projeto por conta de sua abordagem cíclica, que dá a oportunidade, a cada iteração do ciclo, de correções e aprimoramentos no software e documentação.

2.2 Reutilização de Software

Reutilização de Software é uma ideia que visa a reutilização de produtos, conceitos ou soluções previamente elaboradas durante a realização de um software,

com intuito de melhorar a qualidade e a manufatura do software em questão (DEVMEDIA, acesso em 2021).

Peter Freeman (1980) define a reutilização de software como a “utilização de produtos de software, construídos ao longo do processo de desenvolvimento, em uma situação diferente daquela para a qual foram originalmente produzidos”

Tanto no desenvolvimento de nossa plataforma quanto no desenvolvimento geral de softwares, a reutilização de partes funcionais de código tem um papel importante na manutenção da qualidade do software, diminuindo o volume de código *boilerplate*⁵ como, por exemplo, na utilização de funções ou na criação de bibliotecas que são frequentemente utilizadas.

2.3 Bibliotecas

Bibliotecas de software são uma coleção de funções, componentes, rotinas e similares; podendo ser importadas em outros projetos a fim aprimorar o rendimento no desenvolvimento de software, evitando a reescrita de código e soluções.

Maurice Wilkes, David Wheeler e Stanley Gill já citavam o conceito de bibliotecas em 1951, no livro *A Preparação de Programas para um Computador Eletrônico Digital*, com a chamada “biblioteca de sub-rotinas” contendo:

(...) sub-rotinas de diferentes tipos. Algumas serão sub-rotinas de utilidade geral e estarão em constante uso pelos programadores. (...) Essas sub-rotinas são preservadas na biblioteca parte pelo fato de poderem ser de uso futuro para o atual ou projetos seguintes, e parte pelo fato de incorporarem o resultado da experiência, que geralmente foi obtida pelo gasto de muito tempo e dificuldade.

2.4 Framework

Framework, traduzido do inglês como arcabouço, estrutura; na área de software, segundo Vitor Zucher (2020), é uma estrutura de códigos que tem como objetivo prover uma solução pré-fabricada, aumentando a produtividade no desenvolvimento de software.

⁵ Termo que se refere a um fragmento de código funcional que deve ser incluído em muitos lugares com nenhuma ou pouca alteração.

O uso de Frameworks pode trazer vantagens e desvantagens. Algumas das vantagens do uso de frameworks são a amplificação da produtividade e menor ocorrência de erros, uma vez que quase todas as falhas do framework são identificadas e resolvidas pelos desenvolvedores deles. Porém, uma das desvantagens do uso de alguns frameworks é a subordinação causada pela escolha, além da complexidade em configurar algumas características específicas, haja vista as características pré-fabricadas serem rígidas e dispersas no código do framework (ROVEDA, acesso em 2021).

2.5 React

React, ou ReactJS, é uma biblioteca de código aberto de JavaScript, mantida pelo Facebook, utilizada na construção de interfaces de sites e aplicações Web, baseada em componentes, partes de código funcional que podem ser reutilizadas em várias outras partes do código trabalhado.

De acordo com a documentação do React⁶:

React é uma biblioteca de JavaScript para construção de interfaces, sendo declarativa, eficiente e flexível. React lhe permite compor interfaces complexas a partir de pequenas e isoladas partes de código chamadas de “componentes”.

A biblioteca React foi utilizada no projeto por conta de sua capacidade de criar componentes, permitindo maior reutilização de código no projeto; além da fácil integração a outras bibliotecas e frameworks.

2.6 Spring Boot

Spring Boot é um framework open-source⁷ de Java, que traz agilidade no desenvolvimento diminuindo o tempo despendido com as configurações iniciais do projeto.

⁶ <https://reactjs.org/tutorial/tutorial.html#what-is-react>

⁷ De código aberto ao público

De acordo com a própria documentação “Spring Boot⁸ facilita a criação de aplicativos autônomos baseados em Spring em nível de produção que você pode “simplesmente executar”.”.

3 METODOLOGIA

De modo a utilizar os conhecimentos angariados no curso de Bacharelado em Engenharia de Software, foi definido um plano de como desenvolver o software, visando aproveitar ao máximo o conhecimento adquirido até o momento.

3.1 Primeiras definições

Primeiro, criamos personas que seriam os usuários do sistema e então criamos uma pesquisa pública, na plataforma Google Forms, que confirmou a existência de algumas personas mapeadas.

Em seguida, foram levantados os requisitos do software e, em sequência, após as validações dos requisitos, foi definido o ciclo de vida de desenvolvimento do software, sendo este baseado no Modelo Espiral.



Figura 1 - Cartões com as personas mapeadas no projeto

A validação dos requisitos é importante, porque, de acordo com Boehm (1981), aproximadamente 60% de todos os erros em projetos de desenvolvimento de softwares são gerados durante a etapa de engenharia de requisitos e são usualmente encontrados num momento futuro do desenvolvimento, causando o aumento do custo de produção (NEIVA, 2020).

⁸ <https://spring.io/projects/spring-boot>

3.2 Gerenciamento

Ao final da etapa de validação dos requisitos, definimos seguir a metodologia ágil *ScrumBan*, uma alternativa híbrida entre a metodologia ágil *Scrum* e sistema de administração *Kanban*, criado pela *Toyota* na década de 40.

Como alternativa de plataforma para gestão do projeto, escolhemos a plataforma Trello, onde podemos criar quadros que comportam listas que, por sua vez, comportam cartões.

Na plataforma, criamos oito quadros:

- [Documentação] Adoção de animais
- [Treinamento] Adoção de animais
- [Calendário] Adoção de animais
- [Sprints] Adoção de animais
- [Tasks] Marketing
- [Tasks] Recursos Humanos
- [Tasks] Tecnologia
- [Bugs] Adoção de animais

Os quadros se referem e comportam, respectivamente, a documentação do projeto; a trilha de treinamento em tecnologias e teorias, montada para integrantes do projeto; ao Calendário que utilizamos para registrar a ocorrência e anotar as pautas das reuniões do projeto; ao rastreo de sprints do projeto na metodologia Scrum; às tarefas do time de Marketing; às tarefas do time de Recursos Humanos; às tarefas do time de desenvolvimento (tecnologia); e aos bugs reportados no projeto.

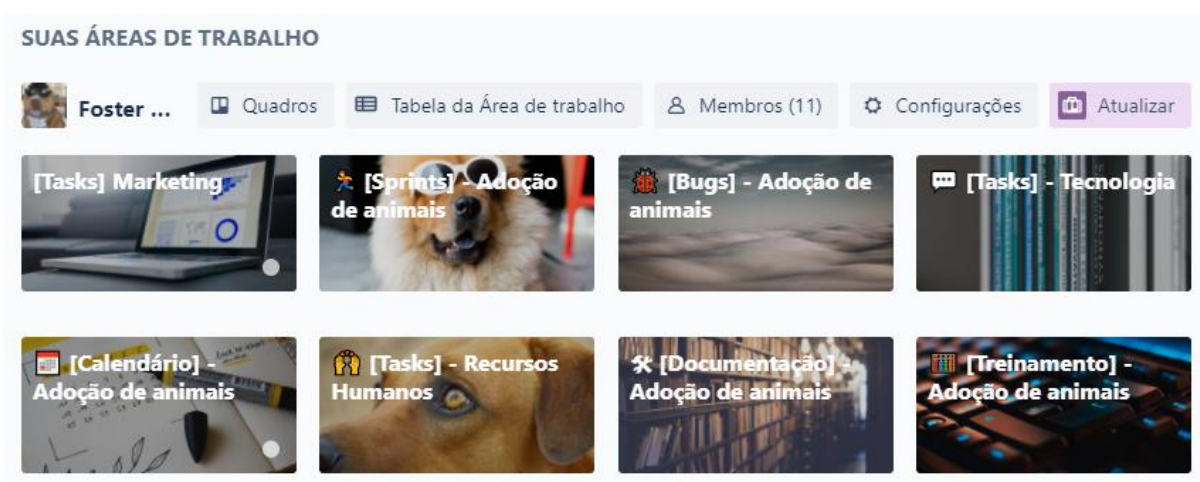


Figura 2 - Quadros criados na plataforma Trello

3.3 Modelagem

Em seguida, foi iniciada a etapa de modelagem, com a construção de alguns artefatos de documentação de software como, por exemplo, diagrama de caso de uso, documentação de caso de uso, diagrama de classe, diagrama de objetos, modelo de entidades relacionais, diagrama de entidades relacionais e diagrama de tabelas relacionais.

A construção de modelos tem sua importância pois propicia o gerenciamento da complexidade característica ao desenvolvimento de software, a comunicação entre os envolvidos no desenvolvimento, ao decréscimo dos custos de produção e à previsão da futura conduta do sistema (MIGUEL, 2019).

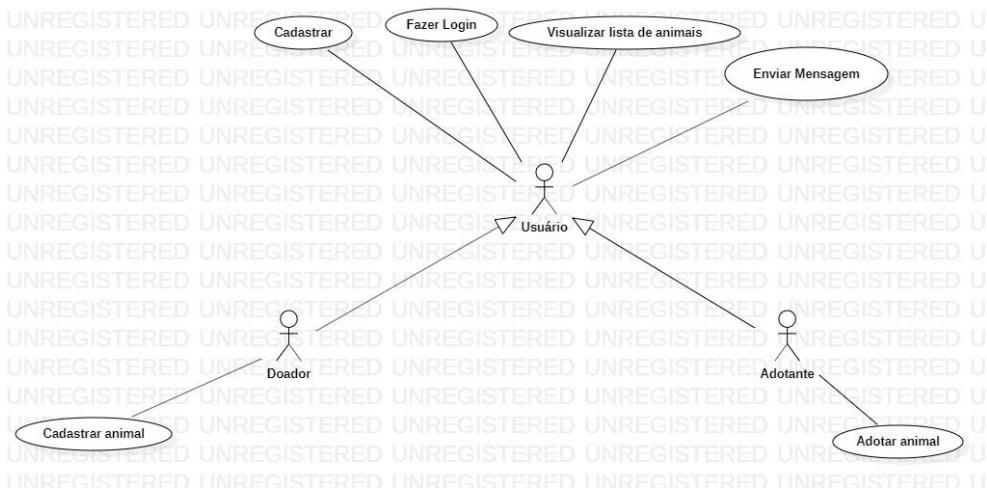


Figura 3 - Diagrama de caso de uso contemplando os usuários básicos do sistema.

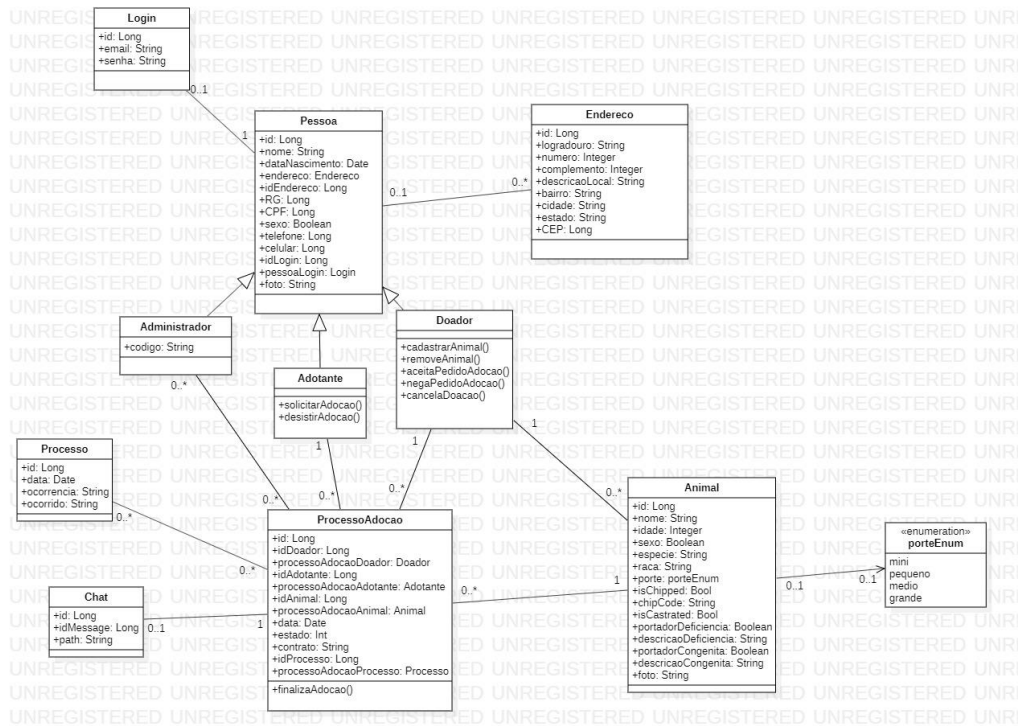


Figura 4 - Diagrama de Classes

3.4 Prototipação

Após a etapa de modelagem, iniciou-se a etapa de prototipação, onde foram feitos protótipos da interface. Foram utilizadas três aplicações para a criação dos protótipos, sendo elas Adobe XD, Adobe Photoshop e Figma. A aplicação Adobe Photoshop é uma solução paga para edição de imagens, enquanto as outras, Adobe XD e Figma, para a criação de protótipos de alta fidelidade, são gratuitas.

Conforme Kruchten (2003), os protótipos devem ser utilizados com a finalidade de reduzir riscos e incertezas quanto à viabilidade do negócio de um produto que é desenvolvido (ANDRETO, 2006). Em nosso projeto, a prototipagem de telas serviu de modo a guiar o desenvolvimento de software na camada *front-end*⁹ da aplicação.

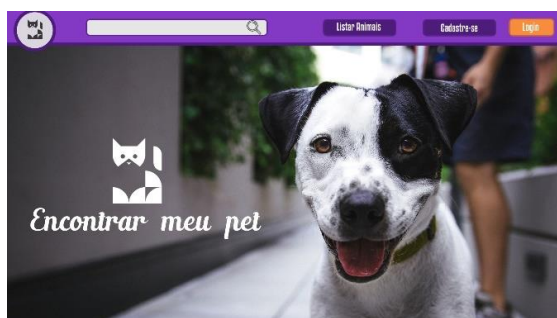


Figura 5 - Protótipo V1



Figura 6 - Protótipo V2

⁹ Área de desenvolvimento das interfaces gráficas de um software, parte da frente.

3.5 Desenvolvimento

Após a criação de alguns protótipos baseados no fluxo de caso de uso, começamos a discutir qual tecnologia utilizar no projeto. Foi decidido a utilização do React, uma biblioteca de JavaScript, para construção da camada *front-end* da aplicação, e Java, com o framework Spring Boot, para a construção da camada *back-end*¹⁰ da aplicação.

i Trilhas de treinamento

No projeto, para que os participantes tivessem a oportunidade de aprender sobre as tecnologias e técnicas empregadas no projeto, foi criada uma trilha de treinamento no Trello, contemplando conhecimentos de modelagem, *front-end*, *back-end*, *dev-ops*¹¹ e gestão. A trilha conta com explicações sintetizadas entre os alunos bem como cursos gratuitos e/ou pagos em plataformas de ensino online.

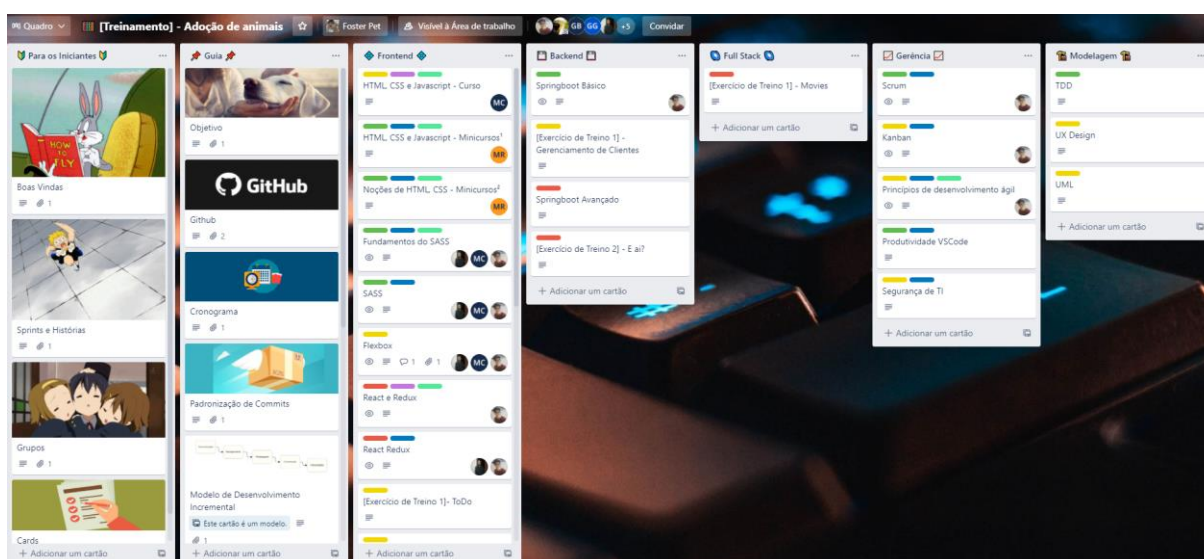


Figura 7 - Trilhas de treinamento no Trello.

ii Front-end

A camada front-end é a camada da interface gráfica de um software, no caso, de uma aplicação web ou site, aplicativos e softwares em geral (TOTVS, acesso em 2021). Essa camada também pode ser responsável por comportamentos *client-side*¹².

¹⁰ Área de desenvolvimento das estruturas de apoio e operacionais de um software, parte de trás.

¹¹ Práticas e/ou cultura que promove a colaboração entre desenvolvimento e manutenção do software.

¹² “Lado do cliente”, lado do usuário.

Para iniciarmos um projeto React, primeiro é necessário ter instalado na máquina o Node.js, um runtime¹³ de JavaScript. Em seguida, definimos uma pasta para iniciarmos o projeto. Nesta pasta, abrimos um bash¹⁴ e digitamos o comando “*npm create-react-app <nome do projeto>*” que irá iniciar o projeto e instalar as dependências básicas da biblioteca React.



```

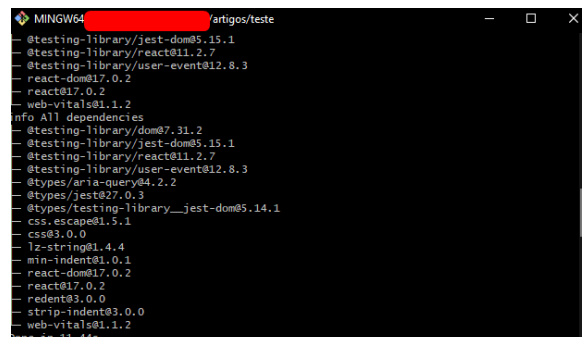
MINGW64: artigos/teste
$ cd teste
MINGW64 ~/Desktop/artigos/teste
$ npm create-react-app projeto-teste
npm: installed 67 in 13.679s

Creating a new React app in [redacted]artigos/teste/projeto-teste.

warning ..\..\..\package.json: No license field
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

yarn add v1.22.5
warning ..\..\..\package.json: No license field
[1/4] Resolving packages...
[2/4] Fetching packages...
info fsevents@1.2.13: The platform "win32" is incompatible with this module.
info "fsevents@1.2.13" is an optional dependency and failed compatibility check.
Excluding it from installation.
info fsevents@2.3.2: The platform "win32" is incompatible with this module.
info "fsevents@2.3.2" is an optional dependency and failed compatibility check.
Excluding it from installation.
[3/4] Linking dependencies...
  
```

Figura 9 - Execução do comando.



```

@testing-library/jest-dom@5.15.1
@testing-library/react@11.2.7
@testing-library/user-event@12.8.3
react-dom@17.0.2
react@17.0.2
web-vitals@1.1.2
info All dependencies
@testing-library/dom@7.31.2
@testing-library/jest-dom@5.15.1
@testing-library/react@11.2.7
@testing-library/user-event@12.8.3
@types/aria-query@4.2.2
@types/jest@27.0.3
@types/testing-library__jest-dom@5.14.1
css.escape@1.5.1
css@3.0.0
lz-string@1.4.4
min-indent@1.0.1
react-dom@17.0.2
react@17.0.2
redux@3.0.0
strip-indent@3.0.0
web-vitals@1.1.2
Done in 11.44s.
  
```

Figura 8 - Finalização do comando.

No arquivo “*package.json*” gerado, definimos quais dependências utilizaríamos no projeto, ou seja, quais outras bibliotecas ou frameworks importar.

Após a definição, executamos o comando “*npm install*” para baixarmos as dependências listadas e em seguida podemos executar o script “*npm start*” para iniciar o servidor de desenvolvimento da biblioteca React.

Uma vez aberto o servidor de desenvolvimento, pode ser realizado o desenvolvimento da camada front-end.

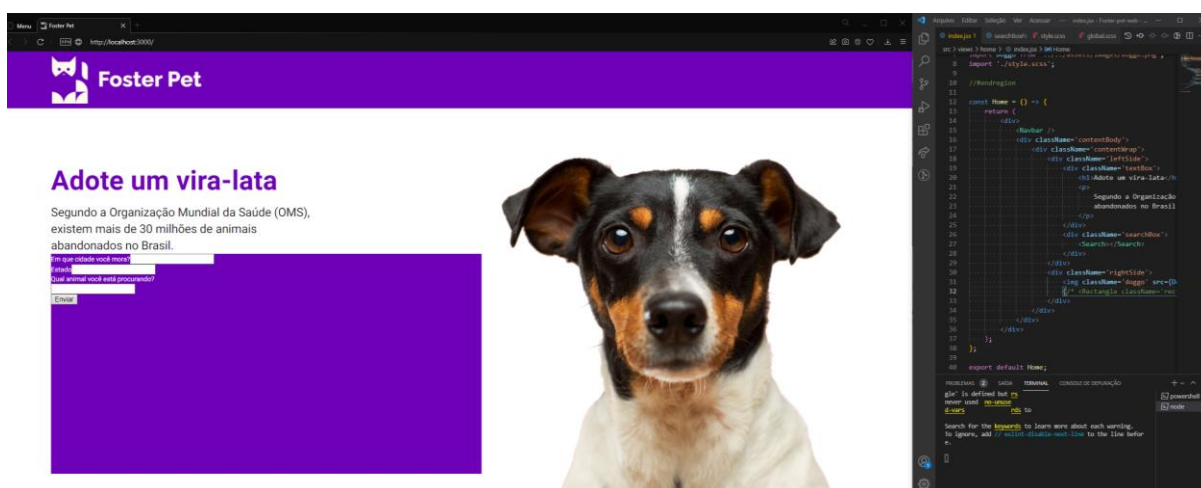


Figura 10 - Landing Page em desenvolvimento

¹³ Sistema de tempo de execução.

¹⁴ Um interpretador de comandos.

iii Back-end

A camada back-end é a camada de desenvolvimento que está no lado do servidor (*server-side*), conhecida como “aquilo que está por trás de uma aplicação” (SOUTO, 2019). Essa camada se concentra em tornar possível as ações de valor do produto como fazer login, finalizar um pedido, fazer uma busca, por exemplo; sendo assim, a parte do software que faz a conexão com bases de dados e executa e realiza a autenticação de ações (REVELO, acesso em 2020).

A camada back-end do projeto foi construída em Java, baseada na arquitetura REST, visando ser uma aplicação RESTful ao aplicar as exigências da arquitetura (GEEKHUNTER, acesso em 2021).

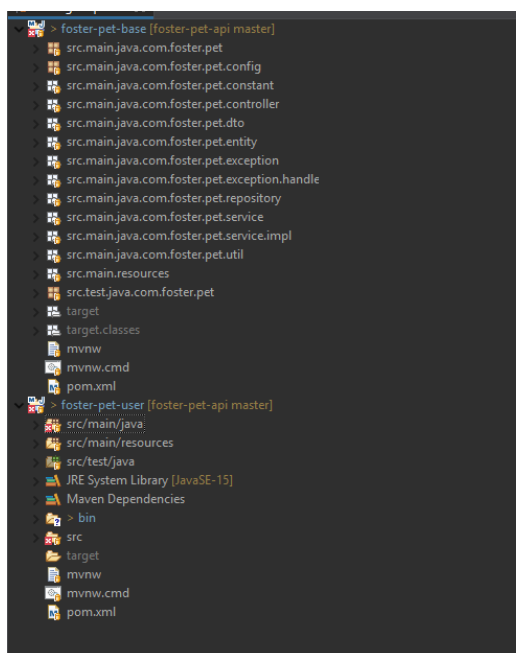


Figura 11 - Diretório dos projetos back-end

4 RESULTADOS E DISCUSSÃO

O desenvolvimento de softwares é algo comum e relevante nos dias atuais, pois existem com o propósito de sanar problemas e saciar necessidades humanas.

Dessa forma, visando resolver o problema de abandono e facilitando o processo de doação e adoção de animais, surgiu a ideia de desenvolver um software que fosse uma plataforma que viabilizasse esse tipo de processo.

As tecnologias empregadas no processo de desenvolvimento cumpriram com o planejado, restando a continuação do desenvolvimento da plataforma.

Com as pesquisas realizadas com o público, numa amostra de 61 pessoas, notou-se que a plataforma teve 67,2% de aceitação no quesito “usaria algum serviço de adoção de animais pela internet, tanto para DOAR quanto ADOTAR animais?”, seguido por 21,3% indecisos e 11,5% decididos em não utilizar.

Usaria algum serviço de adoção de animais pela internet, tanto para DOAR quanto ADOTAR animais ?
61 respostas

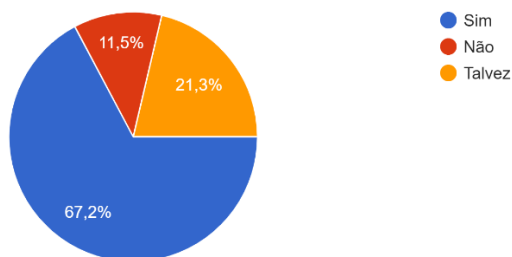


Figura 12 - Gráfico de setores (pizza) sobre a intenção de utilização de uma plataforma online para adoções e doações de animais

5 CONSIDERAÇÕES FINAIS

No momento em que Ada Lovelace percebeu o potencial da Máquina Analítica, de Charles Babbage, podemos dizer que a humanidade demos o primeiro passo de uma gigantesca caminhada da informatização da vida humana. Pelos esforços de todos os envolvidos nessa trilha, hoje vivemos numa sociedade onde quase tudo é facilitado ou sustentado pela computação; seja a administração de uma empresa de dimensão global, sejam os cálculos e processamentos de dados que nos permitem ver o longínquo universo ainda observável, até o pedido de um produto pela internet ou uma conversa mundana em aplicativos de mensagens que substituíram as analógicas cartas.

Várias necessidades e atividades humanas, essenciais ou triviais, foram atendidas pela computação. Da mesma forma que utilizamos redes sociais e aplicativos para nos relacionarmos, podemos estender o conceito para uma aplicação que nos permita adotar animais de estimação, para, não somente, preencher uma possível solidão com o carinho de um pet, mas, também, de forma a dar um lar àqueles pets que necessitam.

Tendo isso em mente, como trabalho futuro, é esperado que o sistema continue com seu desenvolvimento e que seja finalizado e que consiga cumprir seus objetivos iniciais.

REFERÊNCIAS

FARACO, C. B.; **Interação humano-animal**. Ciência veterinária nos trópicos, v. 11, n. 1, p. 31-35, 2008.

JAGHER, D.; GONÇALVES, M.; BUENO, J.; **"Lambeijos: sistema de busca e adoção de animais."** DE SOUZA, A. S. "Direitos dos animais domésticos: análise comparativa dos estatutos de proteção." Revista de Direito Econômico e Socioambiental 5.1 (2014): 110-132.

ANDRADE, L. P.; SILVA, L. L. F.; DAMASO, J. C. G.; **Administração para proteção de animais**. Rev. Ciênc. Ext. v.16, p.320-331, 2020.

CAVALCANTI, A. V. A.; **Animais abandonados: uma perspectiva de ONGs quanto ao problema público da proteção animal no município de João Pessoa/PB**. BS thesis. 2021.

ANDRETO, D. E.; et al.; **Desenvolvimento de software: uso da prototipação na fase de concepção do modelo de processo de software RUP**. Iniciação Científica Cesumar, v. 8, p. 27-33, 2006.

FREEMAN, P.; **REUSABLE SOFTWARE ENGINEERING: A Statement of Long-Range**. Irvine, California, nov./1980. Disponível em: <https://bit.ly/3104zUr>. Acesso em: nov. 2021.

INSTITUTO PET BRASIL. **País tem 3,9 milhões de animais em condição de vulnerabilidade**. Disponível em: <https://bit.ly/3AoQGLr>. Acesso em: mar. 2020.

DEVMEDIA. **Ciclos de Vida do Software**. Disponível em: <https://bit.ly/3CRhpl5>. Acesso em: nov. 2021.

LE WAGON. **O que é um framework? Pra que serve e por que você deveria saber?**. Disponível em: <https://bit.ly/3l8kvLd>. Acesso em: nov. 2021.

KENZIE. **O que é framework, para que serve, vantagens e desvantagens**. Disponível em: <https://bit.ly/3l23l6C>. Acesso em: nov. 2021.

TOTVS. **Front end: O que é, como funciona e qual a importância**. Disponível em: <https://bit.ly/3p0Fdht>. Acesso em: nov. 2021.

SOUTO, M. **O que é front-end e back-end?**. ALURA, set./2019. Disponível em: <https://bit.ly/30VZhZU>. Acesso em: nov. 2021.

GEEKHUNTER. **Sua API não é RESTful: Entenda por quê**. Disponível em: <https://bit.ly/30Zef1p>. Acesso em: nov. 2021.

BECODE. **O que é API? REST e RESTful? Conheça as definições e diferenças!**. Disponível em: <https://bit.ly/3l3m0Dv>. Acesso em: nov. 2021.

REVELO. **Back-End, Front-End, Full-Stack: entenda as carreiras de desenvolvedor**. Disponível em: <https://bit.ly/3l8Rf7f>. Acesso em: nov. 2021.

ROCKCONTENT. **Referências Bibliográficas da ABNT: qual é o padrão e como fazer a referência bibliográfica em um artigo?**. Disponível em: <https://bit.ly/310exVL>. Acesso em: nov. 2021.

WILKES, M.; WHEELER, D.; GILL, S.; **The Preparation of Programs for an Electronic Digital Computer**. Addison-Wesley, 1951.

MIGUEL, M. A.; **Análise de Sistemas: Material Completo**. 3. ed. 2019.

NEIVA, F. W.; **Requisitos de Software**, 2020.

TAVARES, J. F.; **Reutilização de Software**, 2020.