



**UEPB**

**UNIVERSIDADE ESTADUAL DA PARAÍBA - UEPB**  
**CAMPUS VII - GOVERNADOR ANTÔNIO MARIZ**  
**CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS**  
**CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**JOSÉ GILVAN DE ARAÚJO JÚNIOR**

**UM ESTUDO COMPARATIVO ENTRE ESTILOS DE INTERAÇÃO EM UMA  
APLICAÇÃO VOLTADA À ADOÇÃO DE ANIMAIS DE ESTIMAÇÃO**

**PATOS**

**2022**

JOSÉ GILVAN DE ARAÚJO JÚNIOR

**UM ESTUDO COMPARATIVO ENTRE ESTILOS DE INTERAÇÃO EM UMA  
APLICAÇÃO VOLTADA À ADOÇÃO DE ANIMAIS DE ESTIMAÇÃO**

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Computação do Centro de Ciências Sociais e Exatas Aplicadas da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de bacharel em Computação.

**Área de concentração:** Engenharia de Software.

**Orientador:** Prof. Ms. Pablo Ribeiro Suárez

**PATOS  
2022**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

A663e Araujo Junior, Jose Gilvan de.  
Um estudo comparativo entre estilos de interação em uma aplicação voltada à adoção de animais de estimação [manuscrito] / Jose Gilvan de Araujo Junior. - 2022.  
46 p. : il. colorido.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências Exatas e Sociais Aplicadas , 2022.

"Orientação : Prof. Me. Pablo Ribeiro Suárez ,  
Coordenação do Curso de Computação - CCEA."

1. Engenharia de software. 2. Aplicativo. 3. Usabilidade. 4.  
Estilo de interação. I. Título

21. ed. CDD 005.3

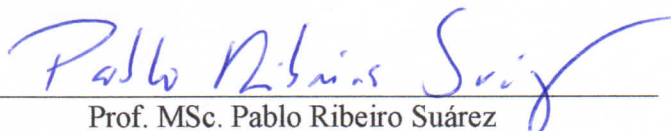
José Gilvan de Araújo Júnior

**UM ESTUDO COMPARATIVO ENTRE ESTILOS DE INTERAÇÃO EM UMA  
APLICAÇÃO VOLTADA À ADOÇÃO DE ANIMAIS DE ESTIMAÇÃO**

Trabalho de Conclusão de Curso apresentado ao  
Curso de Bacharelado em Ciência da Computação  
da Universidade Estadual da Paraíba, em  
cumprimento à exigência para obtenção do grau  
de Bacharel em Ciência da Computação.

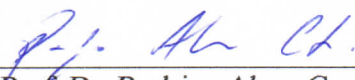
Aprovado em 01/08/2022

BANCA EXAMINADORA



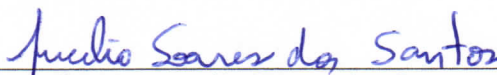
---

Prof. MSc. Pablo Ribeiro Suárez  
(Orientador)



---

Prof. Dr. Rodrigo Alves Costa  
(Examinador)



---

Prof. MSc. Jucelio Soares dos Santos  
(Examinador)

## **DEDICATÓRIA**

Agradeço ao meu orientador, o Mestre Pablo Ribeiro Suárez por ter aceitado me acompanhar neste projeto. O seu empenho foi essencial para a minha motivação à medida que as dificuldades iam surgindo ao longo do percurso.

Aos professores do curso de Ciência da Computação do Campus VII da Universidade Estadual da Paraíba que me forneceram todas as bases necessárias para a realização deste trabalho, agradeço com profunda admiração pelo vosso profissionalismo.

## **AGRADECIMENTOS**

A Deus, por permitir que eu tivesse saúde e determinação para não desanimar durante a realização deste trabalho.

Aos meus pais e irmã, que me incentivaram nos momentos difíceis e compreenderam a minha ausência enquanto eu me dedicava à realização deste trabalho.

A todos que participaram, direta ou indiretamente do desenvolvimento deste trabalho de pesquisa, enriquecendo o meu processo de aprendizado.

Aos meus colegas de curso, com quem convivi intensamente durante os últimos anos, pelo companheirismo e pela troca de experiências que me permitiram crescer não só como pessoa, mas também como formando.

A todas as pessoas que eu possa ter me esquecido nos parágrafos acima, expresso minha eterna gratidão em suas contribuições para que eu obtivesse minha educação superior.

*“Não rebrilha tudo que é ouro,  
Nem perdidos estão os que vagam;  
Não fenece o antigo tesouro,  
Nem geadas raízes apagam.”*

(J.R.R. Tolkien)

## RESUMO

A *World Wide Web* tornou-se um dos principais meios de consumo de conteúdo como um todo, principalmente na pandemia de Covid-19 de 2020. Pandemia essa que, após amenizar, trouxe um imenso abandono de animais de estimação, que eram procurados em seu início. Tendo em vista tal problema, este trabalho se propõe a prototipar soluções em dois estilos de interação diferentes, ambos no mesmo contexto de adoção de *pets*, e conseguir respostas no tangente à usabilidade. Os estilos escolhidos foram interação por menus e manipulação direta, que através do Scrum, foram prototipados utilizando o *framework* Next.JS. Como resultados, o sistema com base em menus se sobressaiu com uma margem considerável. Já nas pesquisas de usabilidade, a interface com menus também foi a vencedora, obtendo a maioria de preferência em duas das três perguntas utilizadas como métrica. Por conseguinte, foi possível comprovar a hipótese de que o sistema mais leve também foi o preferido pelos usuários, assim sendo, a escolha mais ideal para um sistema nesse contexto.

Palavras-chave: Engenharia de software; Aplicativo; Usabilidade; Estilos de interação.



## **ABSTRACT**

The World Wide Web has become one of the main means of consuming content as a whole, especially in the 2020 Covid-19 pandemic. start. In view of this problem, this work proposes to prototype solutions in two different styles of interaction, both in the same context of pet adoption, and to obtain answers regarding usability. The styles chosen were interaction through menus and direct manipulation, which, through Scrum, were prototyped using the Next.JS framework. As a result, the menu-based system excelled by a considerable margin. In the usability surveys, the interface with menus was also the winner, obtaining the most preference in two of the three questions used as a metric. Therefore, it was possible to prove the hypothesis that the lighter system was also preferred by users, thus being the most ideal choice for a system in this context.

**Keywords:** Software engineering; Application; Usability; Interaction styles.

## LISTA DE ILUSTRAÇÕES

Ilustração 1 – Metodologia do trabalho .....	16
Ilustração 2 – Comparativo de tamanho entre React e Next.js .....	23
Ilustração 3 – Tela inicial do protótipo X (manipulação direta) .....	28
Ilustração 4 – Tela de cadastro de um pet do protótipo X (manipulação direta) ...	29
Ilustração 5 – Tela de listagem de pets do protótipo X (manipulação direta) .....	29
Ilustração 6 – Tela inicial do protótipo Y (menus) .....	30
Ilustração 7 – Tela de cadastro de um pet do protótipo Y (menus) .....	30
Ilustração 8 – Tela de listagem de pets do protótipo Y (menus) .....	31
Ilustração 9 – Estrutura padrão do Next.JS .....	32
Ilustração 10 – Estrutura do animal de estimação .....	33
Ilustração 11 – Estrutura final do diretório .....	34
Ilustração 12 – Primeira e segunda seções do formulário .....	37
Ilustração 13 - Terceira seção do formulário .....	38
Ilustração 14 – Cartão de visitas distribuído nos pet shops .....	38
Ilustração 15 – Gráfico de pizza das respostas à primeira questão .....	39
Ilustração 16 – Gráfico de pizza das respostas à segunda questão .....	40
Ilustração 17 – Gráfico de pizza das respostas à terceira questão .....	40

## LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CD	Continuous Delivery
CERN	European Organization for Nuclear Research
CI	Continuous Integration
HTML	HyperText Mark-up Language
HTTP	HyperText Transfer Protocol
kB	Kilobyte
NPM	Node Package Manager
PWA	Progressive Web Application
SGML	Standard Generalized Mark-up Language
SPA	Single Page Application
WIMP	Windows, Icons, Menus, and Pointers

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>12</b>
<b>1.1</b>	<b>Cenário técnico-científico</b> .....	<b>12</b>
<b>1.2</b>	<b>Problemática</b> .....	<b>13</b>
<b>1.3</b>	<b>Relevância</b> .....	<b>14</b>
<b>1.4</b>	<b>Objetivos</b> .....	<b>14</b>
<b>1.4.1</b>	<b><i>Objetivo geral</i></b> .....	<b>14</b>
<b>1.4.2</b>	<b><i>Objetivos específicos</i></b> .....	<b>15</b>
<b>1.5</b>	<b>Metodologia</b> .....	<b>15</b>
<b>1.6</b>	<b>Estrutura do trabalho</b> .....	<b>17</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b> .....	<b>18</b>
<b>2.1</b>	<b>Design de interface Homem-Máquina</b> .....	<b>18</b>
<b>2.1.1</b>	<b><i>Linguagem de comando</i></b> .....	<b>18</b>
<b>2.1.2</b>	<b><i>Linguagem natural</i></b> .....	<b>18</b>
<b>2.1.3</b>	<b><i>WIMP</i></b> .....	<b>19</b>
<b>2.1.4</b>	<b><i>Manipulação direta</i></b> .....	<b>19</b>
<b>2.1.5</b>	<b><i>Interação através de menus</i></b> .....	<b>19</b>
<b>2.1.6</b>	<b><i>Interação através de formulários</i></b> .....	<b>20</b>
<b>2.2</b>	<b>Métodos ágeis</b> .....	<b>20</b>
<b>2.2.1</b>	<b><i>Scrum</i></b> .....	<b>20</b>
<b>2.2.2</b>	<b><i>easYProcess</i></b> .....	<b>20</b>
<b>2.3</b>	<b>Tecnologias voltadas para interação Homem-Máquina</b> .....	<b>21</b>
<b>2.3.1</b>	<b><i>HTML</i></b> .....	<b>21</b>
<b>2.3.2</b>	<b><i>React</i></b> .....	<b>22</b>
<b>2.3.3</b>	<b><i>Next.js</i></b> .....	<b>22</b>
<b>2.3.4</b>	<b><i>Material-UI</i></b> .....	<b>24</b>
<b>2.3.5</b>	<b><i>Storybook</i></b> .....	<b>24</b>
<b>2.3.6</b>	<b><i>Cypress</i></b> .....	<b>25</b>
<b>2.4</b>	<b>Considerações finais sobre o capítulo</b> .....	<b>25</b>
<b>3</b>	<b>DESENVOLVIMENTO</b> .....	<b>27</b>
<b>3.1</b>	<b>Design das interfaces</b> .....	<b>27</b>
<b>3.2</b>	<b>Codificação dos protótipos</b> .....	<b>31</b>

<b>3.2.1</b>	<b><i>Codificação do modelo</i></b> .....	<b>33</b>
<b>3.2.2</b>	<b><i>Modificações no primeiro protótipo</i></b> .....	<b>34</b>
<b>3.2.3</b>	<b><i>Modificações no segundo protótipo</i></b> .....	<b>35</b>
<b>3.3</b>	<b>Considerações finais sobre o capítulo</b> .....	<b>35</b>
<b>4</b>	<b>ESTUDO DE CASO</b> .....	<b>37</b>
<b>4.1</b>	<b>Pesquisas de usabilidade</b> .....	<b>37</b>
<b>4.2</b>	<b>Análise dos dados</b> .....	<b>39</b>
<b>4.3</b>	<b>Comparação com as bases de código</b> .....	<b>41</b>
<b>4.4</b>	<b>Considerações finais sobre o capítulo</b> .....	<b>41</b>
<b>5</b>	<b>CONCLUSÃO</b> .....	<b>42</b>
<b>5.1</b>	<b>Considerações finais</b> .....	<b>42</b>
<b>5.2</b>	<b>Contribuições da pesquisa</b> .....	<b>42</b>
<b>5.3</b>	<b>Limitações da pesquisa</b> .....	<b>42</b>
<b>5.4</b>	<b>Trabalhos futuros</b> .....	<b>43</b>
	<b>REFERÊNCIAS</b> .....	<b>45</b>

# 1 INTRODUÇÃO

## 1.1 Cenário técnico-científico

Durante toda a década de 1990, o cenário da tecnologia mudou drasticamente, cortesia de Tim Berners-Lee e sua invenção do protocolo HTTP como meio de transferência para sua nova linguagem HTML, que viria a revolucionar a web como um todo. O diferencial da nova linguagem, quando comparada a algumas existentes como a SGML inventada pela Hewlett-Packard, seria sua habilidade de usar *hiperlinks* para facilitar a navegação entre páginas (RAGGETT et al., 1998). Outro ponto positivo pensado pelo seu criador foi construí-la baseado em algo pré-existente, retirando assim diversas das limitações de se criar uma nova tecnologia que poderia não ser aceito.

15 anos após o lançamento do HTML 4, em julho de 2013, o Facebook realizou o *initial public release* (ou lançamento público inicial, em uma tradução livre) do React, abrindo assim o código-fonte de sua ferramenta interna para desenvolvimento web. Desde então, ele disparou em popularidade, chegando a se tornar o segundo framework mais popular de acordo com o número de estrelas no agregador de repositórios Github, chegando à marca de 181 mil na devida categoria, bem como 8,7 milhões de repositórios que dependem dele na plataforma.<sup>1</sup>

O lançamento do *React* causou uma enorme mudança no ecossistema de desenvolvimento *web*, mais especificamente na camada *View*, a responsável pela visualização das aplicações, ou como é comumente chamada no desenvolvimento web, *front-end*. A dinamicidade trazida pelo React se dava na forma de que ele era capaz de apresentar os elementos de maneira mais inteligente, muitas vezes sem precisar recarregar a página inteira.

Para Tidwell, Brewer e Valencia (2019, p. 134), “latência custa visitas”<sup>2</sup>, indicando uma verdade de que os usuários tendem a sair de páginas que demoram para carregar. Portanto, tal inteligência se via necessária no Facebook como um todo, onde os usuários possuíam feeds enormes, e tais usuários poderiam sofrer uma aversão caso a página sofresse um novo carregamento a cada comentário ou interação com a rede social (FERNANDES, 08/11/2017).

---

<sup>1</sup> Dados relacionados à 20/01/2022. Disponível em: <https://github.com/facebook/react>

<sup>2</sup> “latency costs viewers”

Três anos após o lançamento do Facebook, em outubro de 2016, a Vercel lançou seu framework baseado em React, chamado de Next.JS. Construindo com base no que funcionou na ferramenta do Facebook, o Next procurou aprimorar em novas funcionalidades, como a evidência para os motores de busca, como Google ou Bing, melhorando assim a visibilidade dos websites construídos com ele.

Websites esses pertencentes a algumas das maiores companhias da atualidade, como Nubank e Binance no setor financeiro e TikTok e Marvel na esfera do entretenimento<sup>3</sup>. Sua popularidade também se reflete no Github, onde ele possui 80 mil estrelas e 536 mil repositórios na plataforma são dependentes da ferramenta.<sup>4</sup>

## 1.2 Problemática

A pandemia causou um impacto gigantesco no mundo como um todo, seja psicológica ou financeiramente. Um aspecto relevante foi a relação entre esses impactos e os animais de estimação, principalmente cachorros. Pois no início da pandemia, as pessoas os adotaram para ter um companheiro no isolamento. Porém, quando elas começaram a precisar cortar gastos ou foram acometidas pela doença e vieram a falecer, os animais foram abandonados (VEIGA, 2020).

Todos esses fatores são refletidos nos números. De acordo com Veiga (2020), a quantidade de pessoas interessadas em encontrar novos donos para seus pets teve “um aumento de 40% da procura anterior ao período”. Ainda segundo Veiga (/2020), o estresse do isolamento também aumentou os maus tratos com os animais, resultando em “482% mais denúncias sobre o tema em comparação com o mesmo período do ano passado”.

De acordo com Genger (2019), “apesar de trabalharem nos mesmos projetos e produtos, desenvolvedores e designers costumam trabalhar separados uns dos outros em silos”<sup>5</sup>. Muitas vezes esse fenômeno ocorre pois os desenvolvedores são vistos como resolvidores de problemas e tendem a divergir sua atenção à funcionalidade da aplicação, enquanto os designers são naturalmente mais criativos e focam na experiência do usuário, repassando tal experiência para que os desenvolvedores a repliquem com o maior grau de fidelidade possível.

---

<sup>3</sup> Dados relacionados à 20/01/2022. Disponível em: <https://nextjs.org/showcase>

<sup>4</sup> Dados relacionados à 20/01/2022. Disponível em: <https://github.com/vercel/next.js>

<sup>5</sup> Even though they may be working on the same projects and products, developers and designers often work apart from one another in silos.

As razões para um desenvolvedor aprender a área tangente ao design são várias. Dentre uma das mais importantes, menciona Genger (23/05/2019), pode-se citar que em times pequenos, pode ser que não haja uma pessoa específica na área de *design*, caindo para os desenvolvedores essa responsabilidade. O cenário anterior se aplica principalmente quando trabalha-se em projetos por si só. Outra razão é pela fricção profissional que uma mudança feita pelo desenvolvedor pode acarretar em um designer, precisando então que os desenvolvedores adquiram um olhar mais aguçado para os detalhes, e que eles estão ali por um motivo.

Visando as adversidades citadas acima, esse trabalho se propõe a responder o seguinte questionamento: em um contexto relacionado à adoção de animais de estimação, e a partir de testes de usabilidade, qual dentre dois sistemas de interação prova ser a melhor solução?

### **1.3 Relevância**

Esclarecem Barbosa e Silva (2010, p. 245), que “na interação através de menus, o sistema oferece um conjunto de opções dentre as quais o usuário deve selecionar o que lhe interessa”. Shneiderman (1997) clarifica que “a manipulação direta depende da representação visual dos objetos e ações de interesse (...) cujo efeito sobre o objeto de interesse é imediatamente visível”<sup>6</sup>.

Tendo em vista ambos estilos de interações supracitados, bem como as perspectivas do desenvolvedor e do usuário final, esse trabalho possui o intuito de desenvolver dois protótipos funcionais e de alta fidelidade, cada um com as peculiaridades que decorrem de seu estilo de interação. Ao final, é esperado que o melhor design, de acordo com os testes realizados com os usuários, também seja o mais otimizado de se desenvolver, com base em dados analíticos referentes à ambas bases de código.

### **1.4 Objetivos**

#### **1.4.1 Objetivo geral**

---

<sup>6</sup> Direct manipulation depends on visual representation of the objects and actions of interest (...) whose effect on the object of interest is immediately visible



Descobrir, através de testes de usabilidade, qual o sistema de interação mais apropriado para um sistema relacionado à adoção de animais de estimação.

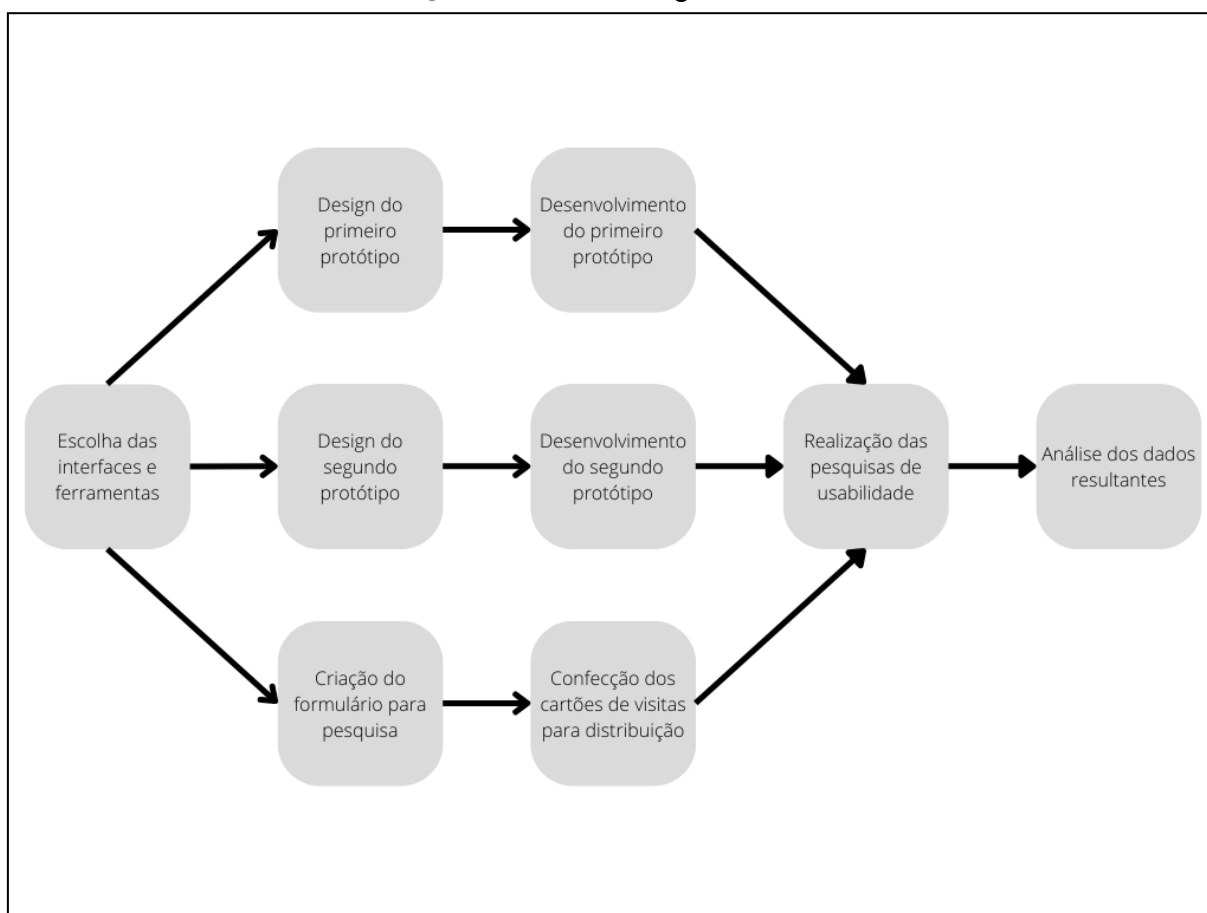
#### **1.4.2 Objetivos específicos**

- Construir a interface baseada em menus
- Construir a interface baseada em interação direta
- Desenvolver o primeiro protótipo, baseado em menus
- Desenvolver o segundo protótipo, baseado em interação direta
- Realizar testes de usabilidade com clientes de pet shops
- Cruzar os dados advindos dos testes com os dados das bases de código
- Estabelecer a interface desejada

#### **1.5 Metodologia**

A estrutura do trabalho descrita a seguir pode ser melhor visualizada na ilustração abaixo.

### Ilustração 1 – Metodologia do trabalho



Fonte: Elaborada pelo autor, 15/012022.

Como irá ocorrer o desenvolvimento tanto de telas quanto de protótipos, o Scrum se torna a melhor opção para a escolha de metodologia, pois permite uma maior agilidade. O Scrum se baseia em *sprints*, que não devem durar mais que quatro semanas, e que por sua vez possuem as fases de planejamento, execução, “reuniões diárias”, revisão e retrospectiva (Magno, 2019).

Para a construção das aplicações, será usado o Next.JS<sup>7</sup> para o *front-end*, com o auxílio da Material-UI<sup>8</sup> para uma maior agilidade e consistência entre designs. Já para a arquitetura do *back-end* e consistência de dados dos sistemas, será utilizado um banco de dados não relacional, através da ferramenta FaunaDB<sup>9</sup>, com o auxílio de armazenamento de imagens provido pelo ImgBB<sup>10</sup>.

Por último, o meio para conseguir as respostas de usabilidade ao final da construção de ambas as interfaces ficou por conta do Google Forms, um

<sup>7</sup> <https://nextjs.org/>

<sup>8</sup> <https://v4.mui.com/pt/>

<sup>9</sup> <https://fauna.com/>

<sup>10</sup> <https://imgbb.com/>

instrumento para construção e compartilhamento de formulários construído pela Google. As perguntas serão comparativas, visto que dois sistemas foram desenvolvidos, e busca-se descobrir qual é o melhor em um aspecto mais geral.

O formulário será dividido em três seções, com a primeira sendo responsável pela angariação de dados do usuário. Já a seção intermediária será onde os links dos sistemas serão colocados para que os respondentes possam testá-los, bem como uma pergunta para declarar o fizeram. Por fim, a terceira seção será onde os usuários responderam questões sobre facilidade, rapidez e design, respectivamente.

Para a distribuição desses formulários, serão confeccionados 200 cartões de visita com um QR Code que levarão aos mesmos. Esses cartões serão distribuídos de maneira aleatória e igualmente entre cinco lojas de animais de estimação na cidade de Patos, na Paraíba.

## **1.6 Estrutura do trabalho**

A monografia encontra-se composta pelo capítulo 2, alusivo ao referencial bibliográfico, onde são explorados os conceitos referentes à design, bem como a metodologia e as ferramentas utilizadas no desenvolvimento dos protótipos.

Desenvolvimento este que ocorre no terceiro capítulo, onde são desenhadas as interfaces, e os sistemas prototípicos são construídos, com base nas interfaces mencionadas.

Com as aplicações construídas, o capítulo 4 tange o estudo de caso, onde eles são utilizados pelos usuários, que respondem uma tríade de perguntas sobre a facilidade, rapidez e apresentação de cada um dos sistemas. Essas respostas são então comparadas com os resultados obtidos no capítulo 3 para obtenção do sistema mais otimizado.

No capítulo conclusivo, são elucidadas algumas considerações, bem como as contribuições e limitações do trabalho, seguida por uma lista final de trabalhos futuros possíveis.

## 2 REFERENCIAL TEÓRICO

O desenvolvimento de software em sua totalidade é fundamentado em três pilares que precisam ser bem definidos para que o produto final seja de qualidade: o design da interface, a metodologia a ser aplicada e as ferramentas a serem utilizadas. Tais pilares serão detalhados nas seções seguintes.

### 2.1 Design de interface Homem-Máquina

Toda aplicação requer um ou múltiplos designs de interface, que requerem uma elaboração cuidadosa através de um *designer*; em seguida, sua criação e execução por um programador, que se encarregará de tirar as ideias de papel e concretizá-las. A seguir, serão demonstradas alguns *designs* propostos e suas justificativas para serem ou não incluídos na comparação proposta por esse trabalho.

#### 2.1.1 Linguagem de comando

Para Barbosa e Silva (2010, p. 244), “uma linguagem de comando é organizada como um conjunto de comandos simples (...), comandos mais parâmetros (...) ou comandos seguidos de opções e argumentos”. Contudo, visto que o alvo dos sistemas a serem desenvolvidos trata-se da web, este estilo torna-se inviável, já que é mais encontrado em sistemas operacionais, e são complexos para o usuário final compreender.

#### 2.1.2 Linguagem Natural

Já a linguagem natural, “visa permitir que o usuário se expresse como em uma conversa com uma outra pessoa, utilizando seu próprio idioma” (BARBOSA e SILVA, 2010, p. 244). Entretanto, apesar da facilidade para o usuário utilizar-se dessa interação, ela é imensamente complexa de ser desenvolvida, visto que tal linguagem natural precisaria lidar com as ambiguidades originadas da linguagem expressada pelo utilizador do sistema.

### 2.1.3 WIMP

Já o estilo de interação WIMP (*Windows, Icons, Menus and Pointers*, que pode ser traduzido como janelas, ícones, menus e apontadores) é um dos casos onde um mesmo sistema utiliza-se de múltiplos atributos em distintas porções da interface (BARBOSA; SILVA, 2010). Esse estilo é normalmente reservado para aplicações complexas, o que o elimina como possibilidade de uso, visto que o sistema será bastante simples em seu uso.

### 2.1.4 Manipulação direta

O estilo de interação por manipulação direta, segundo Shneiderman (1998, p. 231) traz um “sentimento de envolvimento direto com um mundo de objetos ao invés de se comunicar com um intermediário”<sup>11</sup>. Um exemplo proeminente desse estilo de interação pode ser visto nos sistema operacional para computadores Windows, onde as pastas da área de trabalho podem ser arrastadas ao serem clicadas.

Tal estilo de interação torna-se interessante para a comparação, visto que, apesar de ser mais difundido em sistemas operacionais, é facilmente replicável em um sistema web.

### 2.1.5 Interação através de menus

A interação por menus é interessante, visto que são bastante usados em aplicativos web e outras mercadorias que emulam aplicações desktop (TIDWELL; BREWER; VALENCIA, 2019, p. 378). Segundo no contexto de sistemas operacionais, a tela de configurações no sistema operacional para celulares Android é um ótimo exemplo de design com menus, pois oferece diversas opções, desde opções relativas à tela a informações do sistema como um todo.

Novamente, a despeito do estilo de interação ser mais encontrado em sistemas operacionais, percebe-se viabilidade em uma implementação web, principalmente tendo em mente que a aplicação poderá ser acessada através de dispositivos móveis.

---

<sup>11</sup>“feeling of involvement directly with a world of objects rather than of communicating with an intermediary”

### **2.1.6 Interação através de formulários**

A interação por formulário consiste em diversos campos, comumente com alguns sendo obrigatórios, para que o sistema realize alguma ação com os dados de tais campos (BARBOSA; SILVA, 2010). O cadastro de usuários é um relevante exemplar dessa interface, bem como o registro de um novo item em determinado sistema.

Tendo em vista que o contexto é de adoção de animais de estimação, será preciso que os usuários do sistema cadastrem os animais que irão para adoção.

## **2.2 Métodos ágeis**

De acordo com Fowler et al. (2001), “Facilitar a mudança é mais eficaz do que tentar impedi-la”<sup>12</sup>. Tal frase é o início do manifesto ágil escrito em 2001, e relevante até a contemporaneidade, por lidar com vários dos problemas que atacam os desenvolvedores de software diariamente, levando a um êxodo em massa dos processos anteriores baseados em cascata, a métodos que acomodavam mudanças mais facilmente. As seções a seguir descrevem algumas dessas metodologias.

### **2.2.1 easYProcess**

Podendo ser compreendida como a junção das palavras fácil (easy) e processo (processo), a metodologia conhecida como YP “tem como objetivo auxiliar tanto a gerência do desenvolvimento de aplicações em disciplinas de engenharia de software quanto a aprendizagem dos conceitos desta disciplina na graduação” .

Apesar de poder ser adaptado para um ambiente que não seja a sala de aula, o fato do YP ter nascido nela, o deixa um tanto desinteressante para ser usado em uma aplicação que possa vir a se tornar comercialmente viável.

### **2.2.2 Scrum**

---

<sup>12</sup> Facilitating change is more effective than attempting to prevent it

Buscando revolucionar os rendimentos da profissão, desenvolvedores de software cunharam a estrutura de trabalho Scrum (MAGNO, 2019). No centro da ideologia proposta pelo Scrum, estava um esforço para tornar a indústria mais ágil, entregando pequenos incrementos do produto final, aumentando a satisfação do cliente gradativamente ao mesmo tempo mostrando o progresso feito.

Conforme afirmado por Magno (2019, p. 143): “Ao definir-se como uma estrutura de trabalho, e não como uma metodologia, o Scrum dá aos praticantes o protagonismo para definir práticas, ferramentas e mesmo processos”. Isso significa que o Scrum é flexível o suficiente para ser aplicado em projetos pessoais, apesar do foco ser em times, com o nome vindo da formação dos times de Rugby antes do jogo.

É notável que o Scrum não fornece práticas, mas uma base onde construir com as práticas que se adequem a cada situação ou time (MAGNO, 2019). Alguns dos pilares que sustentam tal estrutura são: o planejamento, onde são decididos os itens da Sprint; a Sprint, que consiste em um período de não mais que 4 semanas, quando o trabalho é feito; e a revisão, para recapitular o que foi feito na Sprint.

Tendo em vista sua popularidade crescente e a flexibilidade para múltiplos tamanhos de times, o Scrum foi a metodologia escolhida para ser utilizada no desenvolvimento.

## **2.3 Tecnologias voltadas para interação Homem-Máquina**

Por fim, as ferramentas propostas para o desenvolvimento dos sistemas são descritas nas seções a seguir.

### **2.3.1 HTML**

Antes de adentrar no tópico da tecnologia HTML em si, é relevante esclarecer um equívoco comum relativo às diferenças entre a *Internet* e a *Web*. Segundo Raggett et al. (1998, p. 2): “(...) a Internet simplesmente fornece o meio para a Web funcionar (...)”<sup>13</sup>, bem como um trilho de trem provê o meio para que o trem faça seu percurso.

---

<sup>13</sup> (...) the Internet simply provides the medium for the Web to run on (...)

Retomando o tópico da linguagem HTML, a sigla significa Linguagem de Marcação de Hipertexto, e foi lançada em conjunto com o HTTP, ou Protocolo de Transferência de Hipertexto, em 1989 por Tim Berners-Lee, na organização CERN, na Europa. (RAGGETT et al., 1998)

Assim sendo, é possível perceber que a linguagem inventada por Tim se tornou extremamente popular e aceita na comunidade. Pois como Raggett et al. (1998, p. 1) bem aponta: “HTML é a linguagem de publicação para a World Wide Web. Nos bastidores, todo documento que você vê na Web (...) é escrito usando HTML”. Tal afirmação era verdadeira em 1998, quando foi escrita, e ainda continua sendo verdade, com o HTML não dando sinais de que vai embora tão cedo.

### 2.3.2 React

Para Griffiths e Griffiths (2021), o React é um framework de desenvolvimento excepcionalmente flexível. Pode-se constatar tal fato pela existência de outros frameworks baseados nele, como o React Native, focado em dispositivos móveis, e o Next.js ou Gatsby que incrementam o conjunto de ferramentas apresentadas pelo React.

Contudo, “projetos React são desafiadores para criar e configurar do zero”<sup>14</sup> (GRIFFITHS; GRIFFITHS, 2021, p. 8). Por conta dessa dificuldade, seus criadores elaboraram comandos para facilitar a criação de novos projetos, e que podem ser usados em conjunto com gerenciadores de pacotes, como o npm ou yarn.

O comando *create-react-app* é usado com o intuito de construir aplicações de página única, ou SPAs, que contenham opções, bem como uma estrutura padrão (GRIFFITHS; GRIFFITHS, 2021, p. 8). Entretanto, aplicações construídas em React costumam ser grandes, e existem alternativas mais otimizadas para a construção de aplicações web.

### 2.3.3 Next.js

Um dos problemas que os programadores encontraram ao manusearem o React, foi que, segundo Griffiths e Griffiths (2021, p. 23), o React constrói apenas o código do cliente. Isso quer dizer que muitos dos mecanismos de buscas, como

---

<sup>14</sup> React projects are challenging to create and configure from scratch.



Google ou DuckDuckGo, através dos seus robôs *crawlers*, não conseguem identificar o que há em aplicações React, tornando-as mais difíceis de indexar.

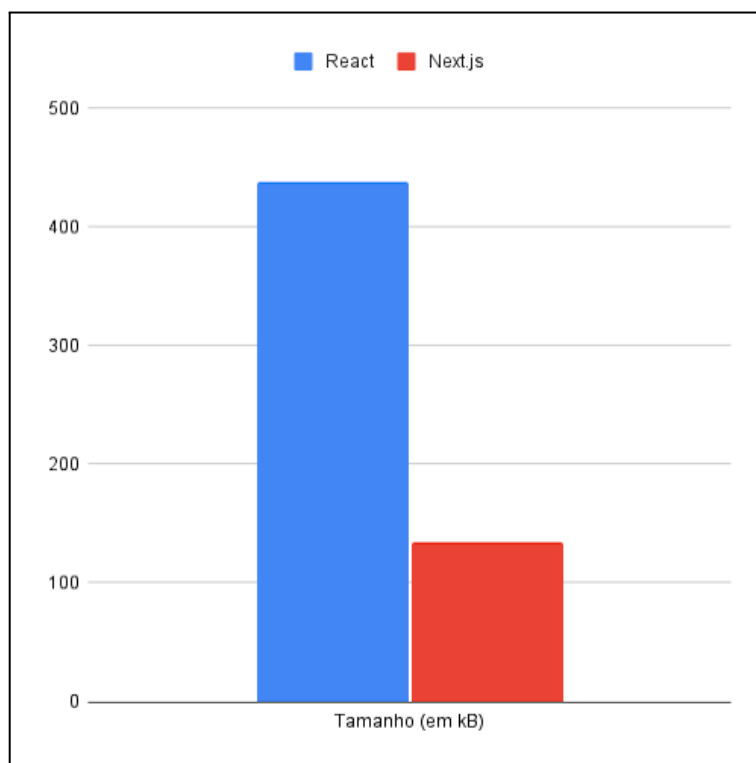
Griffiths e Griffiths (2021, p. 23) argumentam que:

O Next.js é uma ferramenta para gerar aplicações React e código de servidor. Os endpoints da API e as páginas do cliente usam convenções de roteamento padrão, tornando-os mais simples de construir e implantar do que seriam se você os gerenciasse por conta própria.

Tal simplicidade na construção pode ser vista de forma prática, ao criar os dois aplicativos padrões de cada framework, e ambos usando o gerenciador de pacotes yarn, o primeiro através do *yarn create react-app* e o segundo pelo *yarn create next-app*<sup>15</sup>. Os dados são relativos à 2022.

Desconsiderando fatores variáveis, como o tempo de instalação de cada aplicativo, que pode variar até mesmo quando executados na mesma máquina, é possível investigar um dado mais objetivo, como o tamanho da pasta em si, e perceber que o aplicativo padrão do React é aproximadamente 325% maior que o aplicativo padrão do Next.js, como demonstrado no gráfico à seguir.

**Ilustração 2** – Comparativo de tamanho entre React e Next.js



**Fonte:** Elaborado pelo autor (2022).

<sup>15</sup> Nota para a sintaxe diferente da mencionada anteriormente, pois ela se referia ao uso com o npm.

Outro bônus do Next.js é que seus “(...) aplicativos geralmente incluem algum código de servidor de API, o que é incomum para aplicativos React, que geralmente são construídos separadamente do código do servidor”<sup>16</sup> (GRIFFITHS; GRIFFITHS, 2021, p. 24). Tal diferença pronunciada permite construir aplicações mais completas sem o uso de ferramentas externas ao Next.js em si.

### **2.3.4 Material-UI**

Uma das tarefas mais demoradas no desenvolvimento de um aplicativo é escolher um sistema de design, especialmente com a falta de um profissional dedicado para tal. Portanto, se torna mais viável escolher um design já criado e estabelecido, como o Material Design, desenvolvido pelo Google, para prover uma experiência contínua entre dispositivos, tanto web, quanto mobile (GRIFFITHS; GRIFFITHS, 2021, p. 256).

A biblioteca Material-UI é uma execução da padronização Material Design criada pela Google (GRIFFITHS; GRIFFITHS, 2021, p. 78). Ela traz um amplo conjunto de componentes pré-fabricados, todos construídos em cima do que muitos consideram ser o melhor modelo de design da atualidade.

Para Griffiths e Griffiths (11/08/2021, p. 274), a “Material-UI é uma ótima biblioteca para usar e é uma das bibliotecas mais populares atualmente usadas com o React”<sup>17</sup>. O uso dela permite uma vantagem dupla, pois os usuários vão experimentar uma interface mais familiar, enquanto que o desenvolvimento será mais facilitado, pois fará uso de componentes já feitos.

### **2.3.5 Storybook**

Os componentes são os blocos de construção de todas as aplicações React, e se escritos da maneira correta, são extremamente reutilizáveis (GRIFFITHS; GRIFFITHS, 2021, p. 41). Contudo, torna-se extenuante verificar todos eles em cada um dos contextos diferentes na aplicação, sendo preferível testá-los em um ambiente isolado.

---

<sup>16</sup> (...) applications usually include some API server code, which is unusual for React applications, which are often built separately from server code.

<sup>17</sup> The Material-UI library is a React implementation of Google's Material Design standard.

Com esse intuito, foi criado o Storybook, que de acordo com Griffiths e Griffiths (2021, p. 41), pode ser definido como:

(...) uma ferramenta para exibir bibliotecas de componentes em vários estados. Você poderia descrevê-lo como uma galeria de componentes, mas isso provavelmente é pouco. Na realidade, o Storybook é uma ferramenta para desenvolvimento de componentes.<sup>18</sup>

Tendo em vista a praticidade do Storybook, é recomendável desenvolver os componentes em seu ambiente, para só então movê-los para a aplicação propriamente dita. Apesar de parecer simples, o Storybook é uma ferramenta produtiva para o desenvolvimento, pois permite que o foco seja em um componente por vez (GRIFFITHS; GRIFFITHS, 2021, p. 45).

### 2.3.6 Cypress

Do ponto de vista de Griffiths e Griffiths (2021, p. 382), no gênese de um projeto, a facilidade maior se encontra em escrever testes baseados no navegador. E no cenário de desenvolvimento, o Cypress é uma ferramenta que se destaca, por ser construída usando aspectos familiares a outras bibliotecas de teste.

Segundo Griffiths e Griffiths (2021, p. 46), a relação entre os projetos React e bibliotecas de teste é essencial, visto que:

A maioria dos projetos React inclui uma biblioteca de testes. A mais comum é provavelmente `@testing-library/react`, que vem junto com `create-react-app` (...). Mas nada supera o teste de código em um navegador real, com todas as complicações adicionais que isso implica.<sup>19</sup>

O Cypress permite realizar testes no ambiente final onde o usuário vai experimentar a aplicação, e elimina ambiguidades que poderiam ocorrer em outros modelos de testes, como os unitários. Ele também evita muitos inconvenientes dos testes de navegadores convencionais (GRIFFITHS; GRIFFITHS, 2021, p. 46).

## 2.4 Considerações finais sobre o capítulo

Tendo conhecido sobre as interfaces gráficas propostas por Barbosa e Silva, bem como as escolhidas para os protótipos, a metodologia Scrum como meio de

---

<sup>18</sup> "(...) a tool for displaying libraries of components in various states. You could describe it as a gallery for components, but that's probably selling it short. In reality, Storybook is a tool for component development."

<sup>19</sup> Most React projects include a testing library. The most common is probably `@testing-library/react`, which comes bundled with `create-react-app` (...). But nothing quite beats testing code in a real browser, with all the additional complications that entails.

desenvolvimento, e as ferramentas utilizadas para tal, torna-se necessário organizar os conceitos, e desenvolver as interfaces, que subsequentemente servirão de base para os protótipos.

## 3 DESENVOLVIMENTO

### 3.1 Design das interfaces

Alguns sistemas, tais como API's, não precisam de interfaces gráficas para serem manipulados, mas aqueles que possuem uma, necessitam de alguns blocos de construção básicos para que o sistema faça sentido. Felizmente, a *Material-UI*<sup>20</sup>, biblioteca de componente escolhida para o sistema, já provê diversos desses blocos, restando ao desenvolvedor apenas escolhê-los. Outra ferramenta que auxilia bastante na construção de interfaces é o Figma<sup>21</sup>, que foi utilizado neste trabalho.

Modelamos dois projetos de interface para o sistema Adota Fácil, a saber: i) uma das versões denominado “sistema X” é focado na interação por manipulação direta. Escolhemos um tema de fundo claro, com textos escuros, com a cor primária sendo âmbar, e azul claro para a secundária; ii) Já o “sistema Y” possui interação através de menus. Escolhemos um tema de fundo escuro, com textos claros, e uma inversão da paleta do sistema anterior, com o azul sendo a cor principal, e âmbar como a adição.

Em ambos os modelos de projetos de interface, adotamos o mesmo estilo de resolução das telas, bem como sua tipografia, o que permitiu uma avaliação mais justa na empatia dos usuários na escolha por qual ambiente se sentia mais confortável.

A resolução é algo importante de se considerar no processo de construção de uma aplicação, visto que ditará o posicionamento de itens na interface. O website *statcounter.com* provê dados sobre tais resoluções, e mostra que, no Brasil, as resoluções mais populares são 1366x768 pixels e 360x640 pixels, com 13,37% e 7,43% de presença, respectivamente<sup>22</sup>. Portanto, a primeira resolução foi escolhida para as telas de *desktop*, e a segunda para as telas de celular.

A tipografia é algo que requer muito estudo para que elas tenham sintonia, e a biblioteca do Material<sup>23</sup> já possui uma fonte padrão, bem como fontes alternativas

---

<sup>20</sup> Disponível em: <https://mui.com/>

<sup>21</sup> Disponível em: <https://www.figma.com/>

<sup>22</sup> Dados relacionados à 15/04/2022. Disponível em: <https://gs.statcounter.com/screen-resolution-stats/all/brazil>

<sup>23</sup> Disponível em: <https://mui.com/>

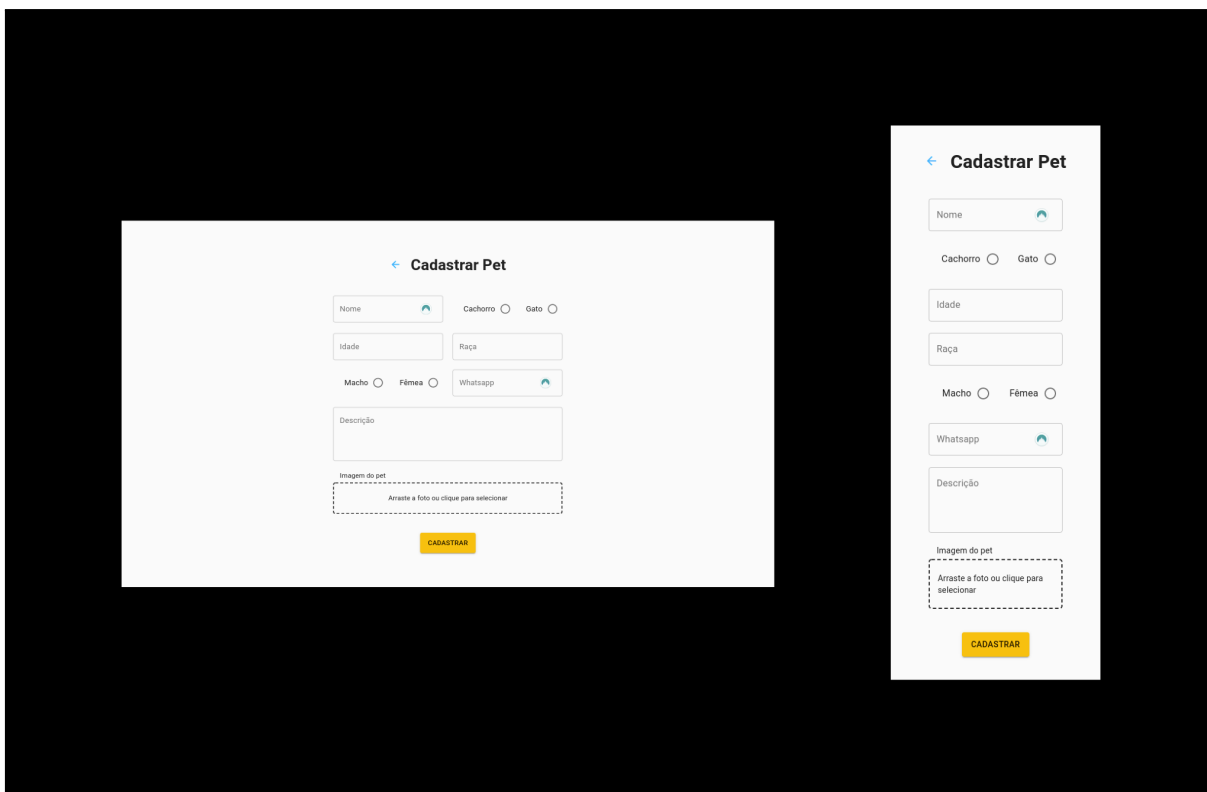
para diferentes sistemas. Tendo em vista o uso do padrão do Material nas cores, consideramos apropriado manter o padrão da biblioteca, utilizando a fonte Roboto.

Considerando todos os pontos demonstrados acima, a seguir apresentamos todas as capturas de tela que demonstram os protótipos gerados na ferramenta Figma para os protótipos utilizando manipulação direta podem ser vistos abaixo.

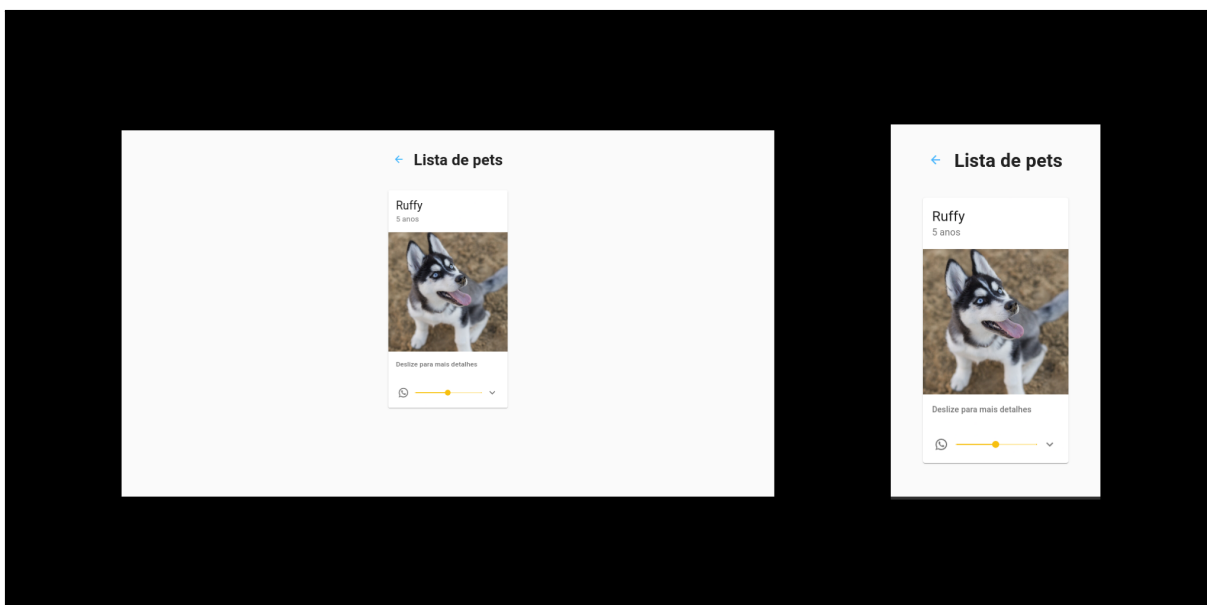
### Ilustração 3 – Tela inicial do protótipo X (manipulação direta)



**Fonte:** Elaborado pelo autor, 13/03/2022.

**Ilustração 4** – Tela de cadastro de um pet do protótipo X (manipulação direta)

Fonte: Elaborado pelo autor, 13/03/2022.

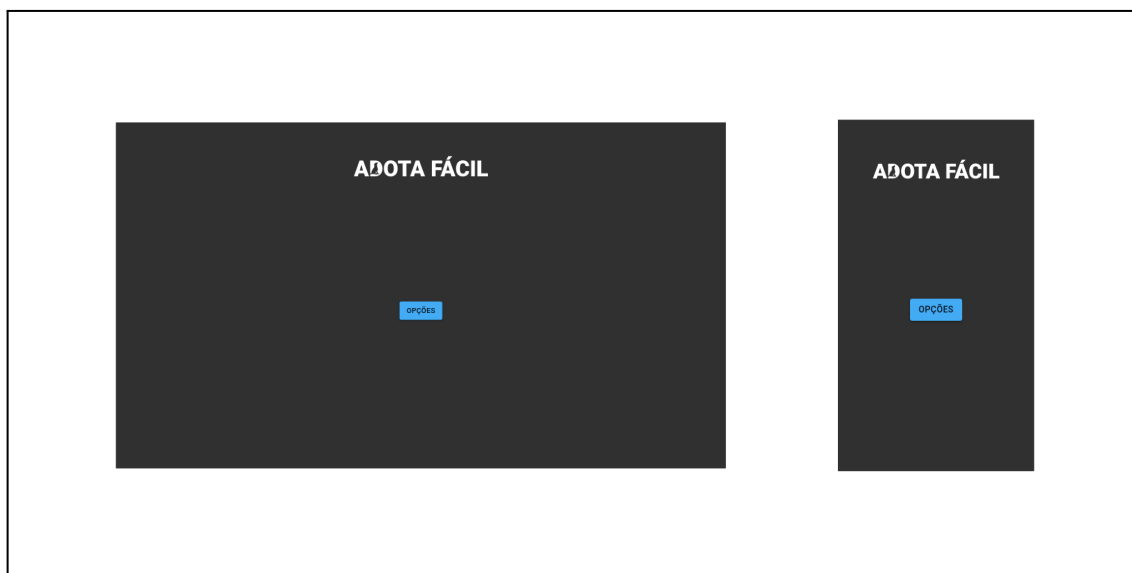
**Ilustração 5** – Tela de listagem de pets do protótipo X (manipulação direta)

Fonte: Elaborado pelo autor, 13/03/2022.

Em seguida, podem ser vistos os artefatos produzidos para o sistema com interação baseada em menus, que possuiu algumas telas similares, porém com

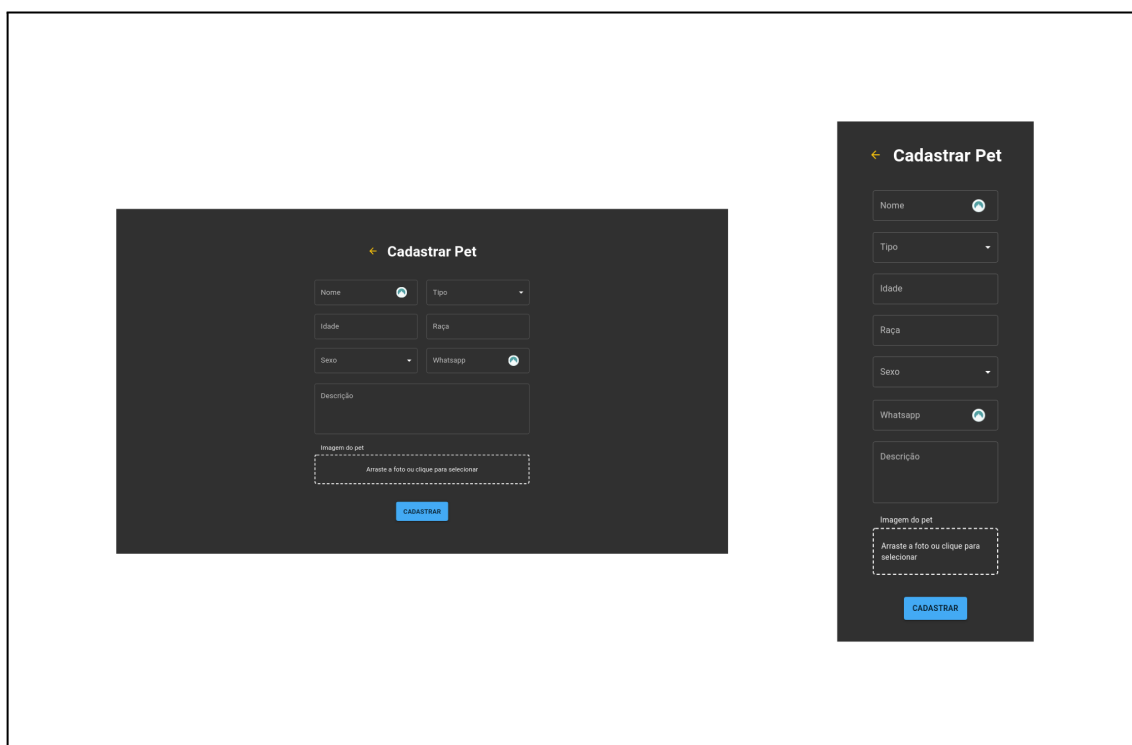
diferenças em características chave no que diz respeito ao acesso às funcionalidades, como o acesso às opções, que são através de *sliders* no protótipo X, e ficam dentro de um menu no protótipo Y.

**Ilustração 6** – Tela inicial do protótipo Y (menus)



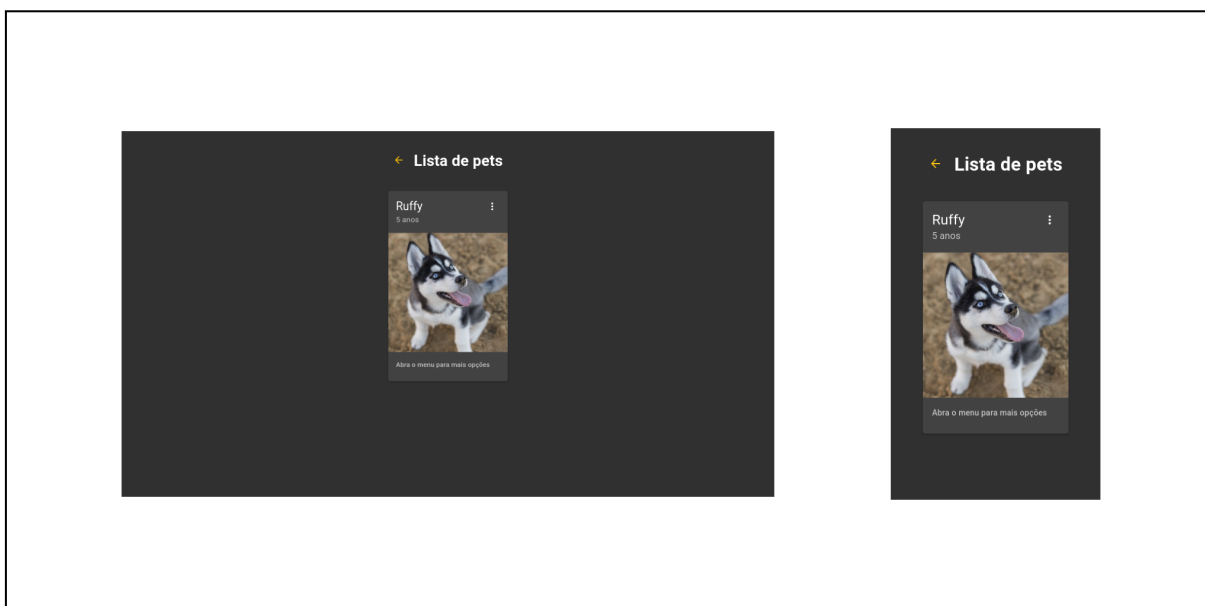
**Fonte:** Elaborado pelo autor, 15/03/2022.

**Ilustração 7** – Tela de cadastro de um pet do protótipo Y (menus)



**Fonte:** Elaborado pelo autor, 15/03/2022.



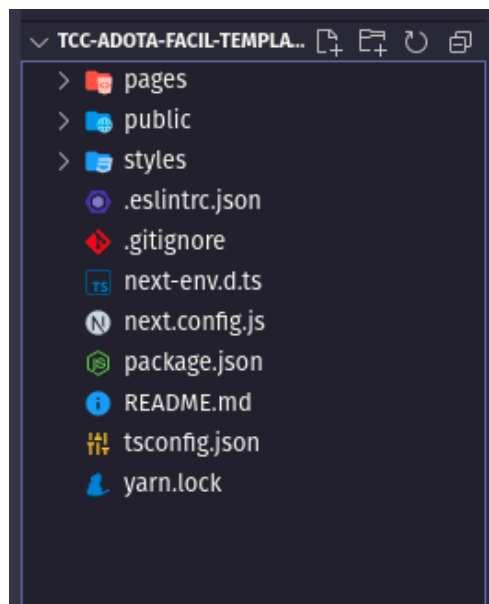
**Ilustração 8** – Tela de listagem de pets do protótipo Y (menus)

**Fonte:** Elaborado pelo autor, 15/03/2022.

**3.2 Codificação dos protótipos**

Após o design das telas de ambas as aplicações, torna-se necessário traduzi-las para algo utilizável, e com uma certa lógica por trás. Como mencionado no capítulo 2, os protótipos serão desenvolvidos para a web, com o auxílio da ferramenta Next.JS. Uma vez que tais interfaces serão comparadas, é preciso partir de um mesmo ponto de partida, e por isso um modelo base foi desenvolvido, e a partir dele divergiu ambos protótipos, cada um com seu respectivo modelo de interação.

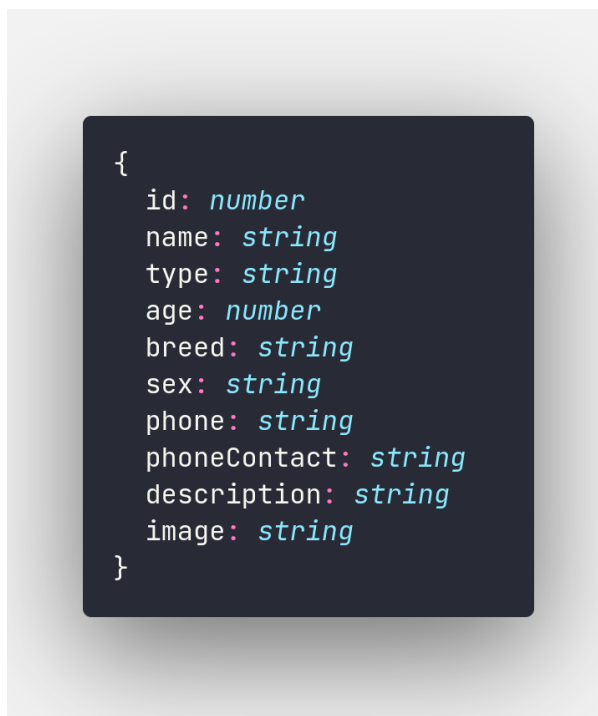
Abaixo, na ilustração 9, é possível ver como o Next.JS estrutura um projeto base, criado a partir do comando recomendado pela criadora da ferramenta.

**Ilustração 9** – Estrutura padrão do Next.JS

**Fonte:** Elaborado pelo autor, 15/04/2022.

Em função do compartilhamento de dados entre ambos os sistemas, foi escolhida também uma estrutura base que todos os animais de estimação deveriam possuir. Tais dados são identificador único, nome, tipo, idade, raça, sexo, telefone do dono, *url* de contato através do Whatsapp, descrição, bem como uma imagem do pet. A estrutura mais detalhada e com os respectivos tipos pode ser visualizada a seguir.

### Ilustração 10 – Estrutura do animal de estimação



Fonte: Elaborado pelo autor, 17/04/2022.

#### 3.2.1 Codificação do modelo

O primeiro passo na codificação do protótipo modelo foi configurar a ferramenta sobre a qual ele foi construído, ou seja, o Next.JS. As duas configurações mais importantes foram feitas em cima do modo *strict* do React, bem como em relação aos domínios das imagens utilizadas na aplicação, que serão da plataforma ImgBB. Por último, em função das telas desenhadas para dispositivos móveis, foi feita a configuração do PWA, para que o website possa ser instalado, como um aplicativo convencional, em smartphones.

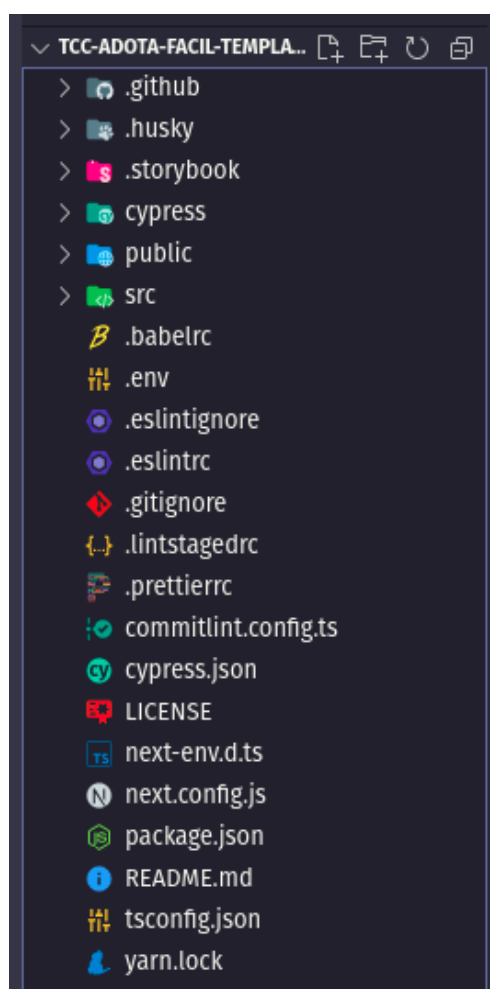
Em seguida, foi necessário configurar duas ferramentas que não impactam no aplicativo diretamente, mas facilitam no desenvolvimento e manutenção da base de código. A primeira ferramenta será o Cypress, um *framework* para testes do fluxo das aplicações desenvolvidas. A segunda, e última, será o Storybook, que auxilia na visualização de componentes da aplicação de maneira isolada.

Por fim, a organização do back-end, ou o local onde toda a lógica será armazenada para propósitos de consistência de dados, foi feita com um par de recursos que trabalham em conjunto. O primeiro é o ImgBB, que serve para

armazenar as imagens, e gerar um link para a mesma. Tal link foi armazenado em um banco de dados não relacional, chamado FaunaDB, junto do restante das informações que o usuário fornece sobre o animal de estimação.

Após todas essas configurações feitas, o protótipo sofreu um aumento de 481% em quantidade de arquivos, passando de 16 para 93, bem como um acréscimo de 706% no seu tamanho, mudando de 113.9 kB para 919.1 kB. A estrutura final do diretório pode ser vista abaixo.

**Ilustração 11** – Estrutura final do diretório



Fonte: Elaborado pelo autor, 20/04/2022.

### 3.2.2 Modificações no primeiro protótipo

O primeiro protótipo que divergiu do modelo, foi o focado na manipulação direta, e que foi nomeado de protótipo X. Para ele foi escolhido o uso de *sliders*, ou controles deslizantes, principalmente para opções binárias sobre o sistema. Na tela

inicial, foi colocado um *slider* com as opções para visualizar a listagem de animais de estimação ou cadastrar um novo.

Já na tela de visualização de pets, eles foram usados abaixo da imagem do animal para dar as opções de contatar através do Whatsapp, ou mostrar mais detalhes. E no desenho do registro, os *sliders* não estão presentes, sendo substituídos por botões de rádio para os campos de gênero e tipo, para que fossem significativamente diferentes do segundo protótipo.

As modificações acima mencionadas causaram uma diminuição de 1,07% na quantidade de arquivos, indo para 92, e uma redução de 1,04% no tamanho, passando para 909,5 kB. Visto que as mudanças foram em arquivos internos, a ilustração da pasta final pode ser vista na Ilustração 11.

### **3.2.3 Modificações no segundo protótipo**

Já na segunda iteração do modelo base, foi escolhida uma abordagem mais mínima na interface, visto que todas as opções seriam acessíveis através de menus. Na página inicial, encontra-se um menu com o texto “Opções” com as possibilidades de ir para as duas telas restantes.

Em virtude dos campos do cadastro de um pet serem em sua maioria abertos, a possibilidade de colocar um menu tornou-se possível apenas nas escolhas de tipo e gênero. Mantendo-se na temática minimalista, o controle deslizante que descansava abaixo da imagem no protótipo anterior, deu lugar a um menu de três pontinhos que dá acesso a ambas as opções presentes no anterior, ou seja, contatar pelo Whatsapp ou ver mais informações.

As mudanças no segundo protótipo foram mais significativas que no primeiro, com uma mingueta para 89 arquivos, ou 4,30%, bem como para 904,3 kB, uma perda de 1,61%. Bem como o primeiro protótipo, o segundo também retém a exata estrutura externa do modelo, que pode ser vista na Ilustração 11.

## **3.3 Considerações finais sobre o capítulo**

Apesar do processo de design de interfaces ser uma tarefa multidisciplinar e que requer equipes grandes (BARBOSA; SILVA, 2010), o Material-UI consegue auxiliar o design, por já conter técnicas difundidas.

Com os protótipos prontos, realizamos as pesquisas de usabilidade com os usuários, bem como avaliar os dados, e analisá-los em conjunto com os resultados deste capítulo.

## 4 ESTUDO DE CASO

### 4.1 Pesquisas de usabilidade

Como ferramenta de criação das pesquisas de usabilidade, foi utilizada a ferramenta Google Forms, que permite dividir o formulário em seções. A primeira seção serviu como recolhimento de e-mails, bem como uma pergunta sobre o interesse do usuário em receber atualizações sobre o trabalho.

Já a segunda seção teve o intuito de mostrar ambos acessos aos sistemas para os usuários, e para verificar que eles realmente tinham acessado e utilizado os sistemas. Uma ilustração com as duas primeiras seções pode ser vista abaixo.

**Ilustração 12 - Primeira e segunda seções do formulário**

The image displays two screenshots of a Google Forms survey titled "Pesquisa de Usabilidade".

**Left Screenshot (Página 1 de 3):**

- Title: Pesquisa de Usabilidade
- Subtitle: Pesquisa referente ao software web de adoção de animais de estimação Adota Fácil
- Link: [Faça login no Google](#) para salvar o que você já preencheu. [Saiba mais](#)
- Requirement: \*Obrigatório
- Field: E-mail \* (Seu e-mail)
- Question: Você gostaria de ser notificado sobre os resultados dessa pesquisa? \*
  - Sim
  - Não
- Buttons: Próxima, Limpar formulário
- Footer: Este formulário foi criado em Universidade Estadual da Paraíba. [Denunciar abuso](#)

**Right Screenshot (Página 2 de 3):**

- Title: Pesquisa de Usabilidade
- Link: [Faça login no Google](#) para salvar o que você já preencheu. [Saiba mais](#)
- Requirement: \*Obrigatório
- Section Header: Visita e uso dos sistemas
- Text: Nessa seção, você visitará os sistemas e os utilizará como bem entender
- Item 1: Adota Fácil Y (<https://adota-facil-y.vercel.app/>)
- Item 2: Adota Fácil X (<https://adota-facil-x.vercel.app/>)
- Question: Declaro que acessei e utilizei os sistemas acima mencionados, e estou pronto para responder as perguntas que dizem respeito a tais sistemas: \*
  - Sim
- Buttons: Voltar, Próxima, Limpar formulário
- Footer: Este formulário foi criado em Universidade Estadual da Paraíba. [Denunciar abuso](#)

Fonte: Elaborada pelo autor, 20/01/2022.

Após as primeiras seções, a terceira seção do formulário mostra perguntas aos usuários, onde eles precisam votar em qual dos sistemas foi o melhor nos aspectos de facilidade, rapidez e apresentação como um todo. Uma ilustração com tal seção pode ser melhor visualizada abaixo.

### Ilustração 13 - Terceira seção do formulário

**Pesquisa de Usabilidade**

[Faça login no Google](#) para salvar o que você já preencheu. [Saiba mais](#)

\*Obrigatório

**Perguntas sobre os sistemas**

Nessa seção, você responderá algumas perguntas sobre qual sistema se provou ser o mais interessante em diversos quesitos

Qual, dentre os dois sistemas, foi o mais fácil de se utilizar? \*

Sistema Y

Sistema X

Qual, dentre os dois sistemas, foi o mais rápido na utilização? \*

Sistema Y

Sistema X

Qual, dentre os dois sistemas, possuiu a melhor apresentação? \*

Sistema Y

Sistema X

**Fonte:** Elaborada pelo autor, 24/01/2022.

Para as pesquisas de usabilidade, foram impressos 100 cartões de visita com um código QR (Fig. 14) que direcionava para o formulário da pesquisa. Os cartões foram distribuídos em conjuntos de 20 cartões para 5 pet shops, que concordaram em deixá-los para que seus clientes pudessem responder à pesquisa. Obtivemos um retorno de 16 respostas.

### Ilustração 14 – Cartão de visitas distribuído nos pet shops



**Fonte:** Elaborado pelo autor (2022).



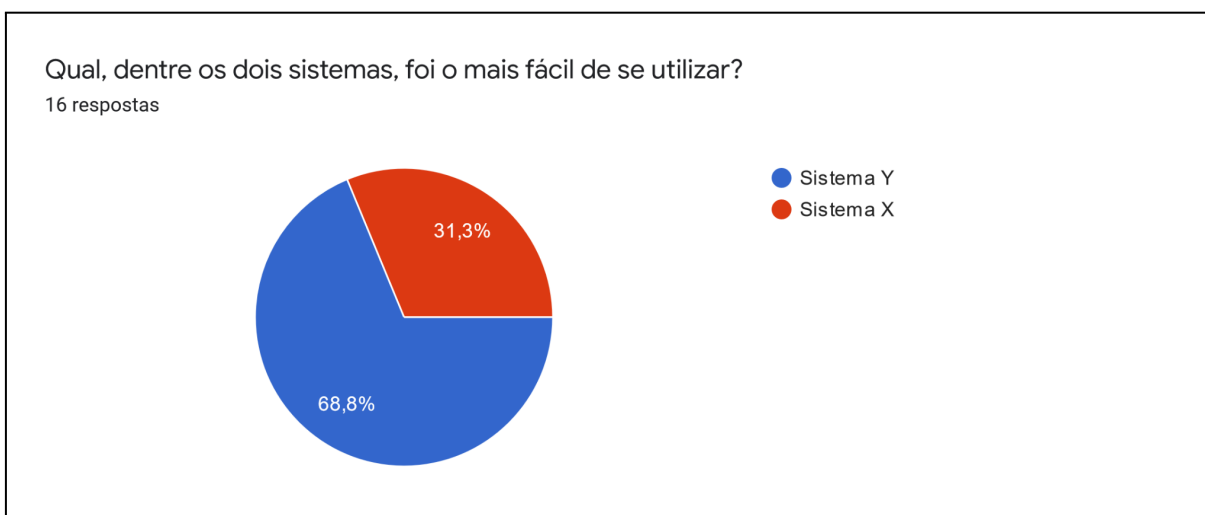
Após duas semanas de obtenção de respostas, os cartões foram recolhidos, em suma totalizando 38 cartões, o que equivale a um total de 62% de retenção de pessoas que pegaram um cartão. Desses 62 cartões que foram pegos, 16 pessoas responderam à pesquisa, o que equivale a 16% do total de cartões, mas a 25,80% dos cartões que foram pegos por algum cliente dos pet shops.

## 4.2 Análise dos dados

Visto que o formulário foi dividido em três seções, mas que a primeira foi uma validação de e-mails, e a segunda foi a apresentação dos protótipos, bem como a confirmação de que os usuários fizeram uso deles. Portanto, as perguntas foram concentradas na terceira seção, que será a analisada nos parágrafos a seguir.

Na primeira pergunta, que dizia respeito à velocidade do sistema, a divisão entre os sistemas X e Y chegou perto de um terço para o protótipo Y, com os remanescentes dois terços dos usuários, tendo o protótipo X como o mais rápido. A distribuição pode ser melhor visualizada no gráfico abaixo.

**Ilustração 15** – Gráfico de pizza das respostas à primeira questão

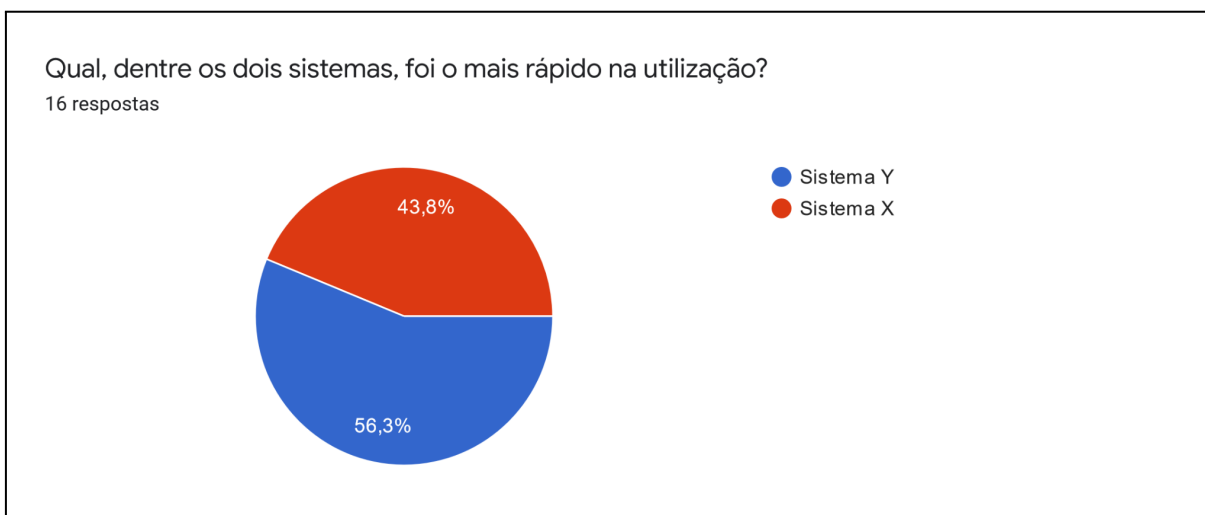


**Fonte:** Elaborado pelo autor (2022).

Já a segunda pergunta, que tange a rapidez do sistema, provou ser uma questão mais difícil para os usuários, que se provaram estar divididos. Na questão da rapidez, 9 pessoas afirmaram que o sistema Y era mais rápido, enquanto que 7

responderam que o X era o mais rápido dos protótipos. O arranjo de respostas pode ser melhor visto na ilustração abaixo.

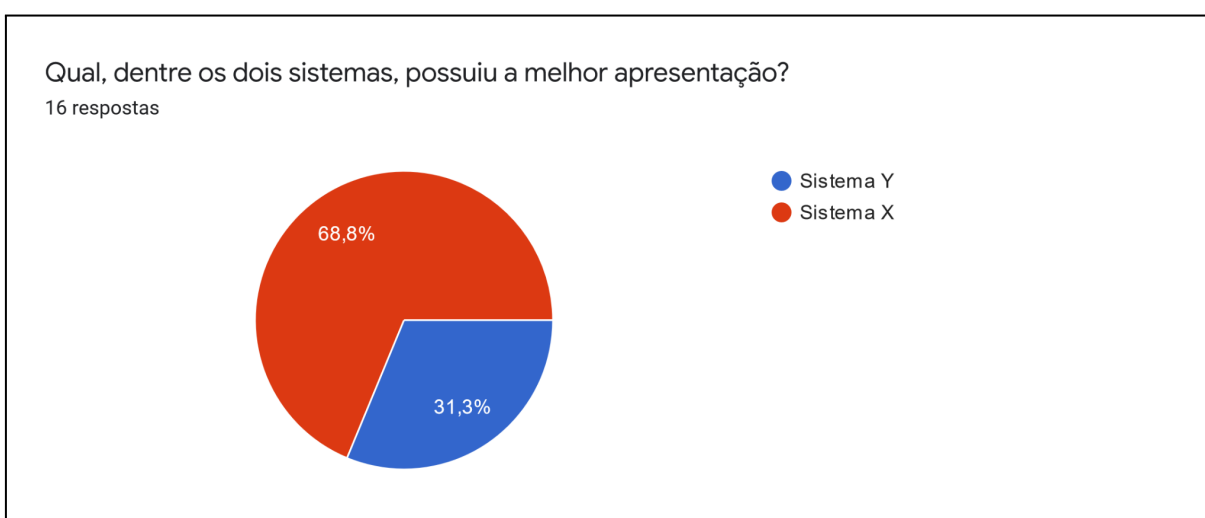
**Ilustração 16** – Gráfico de pizza das respostas à segunda questão



**Fonte:** Elaborado pelo autor, 2022.

Por último, a terceira pergunta concerne à apresentação dos sistemas, ou seja, o design propriamente dito, e as respostas foram inversas às respostas da primeira pergunta. Enquanto a aplicação Y ficou com 11 respostas favoráveis, sua contraparte obteve cinco respostas positivas.

**Ilustração 17** – Gráfico de pizza das respostas à terceira questão



**Fonte:** Elaborado pelo autor, 22/05/2022.

### **4.3 Comparação com as bases de código**

Após uma destilação dos dados oriundos das pesquisas de usabilidade, é possível perceber que o aplicativo Y foi o preferido dos usuários, ganhando nos quesitos facilidade de uso e rapidez. Em contrapartida, o aplicativo X ganhou do Y apenas na apresentação de sua interface.

Relembrando as nomenclaturas utilizadas no design das interfaces, é possível perceber que o sistema Y foi desenvolvido com interação através de menus em mente. Enquanto isso, o sistema denominado X foi desenhado e feito com manipulação direta como alvo.

Retornando às descobertas feitas no capítulo anterior, referente ao desenvolvimento, o protótipo que se provou ser o mais otimizado foi o segundo, com porcentagens maiores de diminuição com relação ao primeiro.

### **4.4 Considerações finais sobre o capítulo**

Apesar da baixa taxa de respostas, tendo em vista o tempo, pode ser considerado uma quantia considerável, chegando a uma média de 1 resposta por dia. A referência cruzada entre os dados do código e respostas também se provou ser algo positivo, visto que as mudanças foram pequenas, mas que fazem a diferença em sistemas maiores.

## **5 CONCLUSÃO**

### **5.1 Considerações finais**

Porquanto a hipótese de que o programa mais otimizado, que foi feito com base em menus, com relação ao código seria o preferido pelos usuários, ela se provou ser real.

Os objetivos foram alcançados dentro do prazo, tanto no fronte do objetivo geral, quanto dos objetivos específicos. A metodologia utilizada também se provou ideal, mostrando que o Scrum se consolidou no mercado como o método definitivo para desenvolvimento de software, devido à sua flexibilidade.

Dado que o campo da engenharia de software como um todo está sempre evoluindo, a escolha da bibliografia refletiu isso com bastante proeza. Com livros recentes no tocante tanto à biblioteca React quanto à design, eles provaram ser uma ótima base para o que foi construído em cima deles.

### **5.2 Contribuições da pesquisa**

O principal benefício da investigação realizada neste trabalho foi o discernimento entre as interfaces no nicho específico de adoção de animais de estimação. Ainda que a interação por manipulação direta possa funcionar para outras finalidades, os donos de animais de estimação apresentaram preferência pela interação através de menus.

Houve também a prova de interesse da população por tal aplicativo e, apesar da baixa taxa de conversão de cartões pegos em pesquisas respondidas, esse número poderia ter sido alavancado, caso mais tempo tivesse sido dedicado a tal porção da produção, ou com uma abordagem mais ativa.

### **5.3 Limitações da pesquisa**

O escopo foi uma das grandes barreiras do projeto, tanto temporal, quanto espacial. Uma vez que a pesquisa foi realizada apenas na cidade de Patos, na Paraíba, seus resultados não podem ser facilmente extrapolados para outras regiões.

Já o fator tempo desempenhou um papel crucial, pois com um prazo maior, uma pesquisa mais minuciosa poderia ter sido realizada. Dado mais tempo, teria sido possível também que mais pessoas pegassem os cartões de visita, aumentando o espaço amostral como um todo.

No tocante ao estilo de pesquisa, os formulários focaram em respostas quantitativas, e após a resposta dos questionários, foi enviado um e-mail pedindo um *feedback* aprofundado sobre os sistemas, porém sem sucesso. Idealmente, essa opinião teria sido incorporada ao formulário original, algo que pode ser corrigido em um trabalho posterior.

Ademais, devido às condições pandêmicas e suas consequências, o pesquisador não teve a oportunidade de conduzir o experimento nas condições habituais em um ambiente controlado.

Barbosa e Silva (2010, p. 12) explanam que “idealmente, a responsabilidade de cuidar de IHC deve ser atribuída a uma equipe multidisciplinar”. Considerando que a autoria deste trabalho foi singular, as interfaces criadas através da ferramenta Figma ficaram aquém de seu potencial máximo, e poderiam ser incrementadas futuramente.

#### **5.4 Trabalhos futuros**

A despeito do fato dos softwares serem embarcados com a tecnologia PWA, um aplicativo propriamente dito teria um impacto exponencialmente maior, e com uma versão *desktop* com bases solidificadas, uma versão *mobile* se torna mais fácil, pois seria apenas uma adaptação.

Levando em consideração o escopo dos testes realizados na aplicação, a arquitetura de *back-end* da mesma foi bem limitada, porém um trabalho futuro poderia implementar autenticação, bem como a diferenciação entre donos de *pets* e os *pet shops* em si.

A autenticação traz consigo uma leva de outras funcionalidades e telas que seriam de crucial importância serem adicionadas ao software. Duas páginas que são intrinsecamente conectadas a tal fluxo são a de *login* e a de perfil do usuário, e ambas poderiam ser abordadas em um trabalho vindouro.

No segundo capítulo, tocante ao referencial bibliográfico, foram abordados estilos de interação além da manipulação direta e interação por menus. Tais estilos

poderiam servir como releituras dos protótipos feitos neste trabalho, e serem introduzidos em comparações futuras.

## REFERÊNCIAS

BARBOSA, Simone; SILVA, Bruno. **Interação humano-computador**. Elsevier Brasil, 2010.

FACEBOOK. **GitHub - facebook/react: A declarative, efficient, and flexible JavaScript library for building user interfaces**. Disponível em: <https://github.com/facebook/react>. Acesso em: 02/01/2022.

FAUNA. **Fauna | The data API for modern applications**. Disponível em: <https://fauna.com/>. Acesso em: 07/01/2022.

FERNANDES, D. **React do zero: componentização, propriedades e estado**. 08/11/2017. Disponível em: <https://blog.rocketseat.com.br/react-do-zero-componentizacao-propriedades-e-estado/>. Acesso em: 02/01/2022.

FOWLER, Martin et al. The agile manifesto. **Software development**, v. 9, n. 8, p. 28-35, 2001.

GARCIA, Francilene Procópio et al. easYProcess: Um Processo de Desenvolvimento para Uso no Ambiente Acadêmico. In: **XII WEI-Workshop de Educação em Computação, XXIV Congresso da Sociedade Brasileira de Computação**. 2004.

GENGER, D. **How to Approach Design for Developers | Toptal**. Disponível em: <https://www.toptal.com/designers/ui/design-for-developers>. Acesso em: 05/01/2022.

GRIFFITHS, D.; GRIFFITHS, D. **React Cookbook: Recipes for Mastering the React Framework**. 1. ed. [S.l.]: O'Reilly Media, 11/08/2021. 697 p

IMGGB. **ImgBB — Alojamento de imagens grátis / envio de imagens**. Disponível em: <https://imgbb.com/>. Acesso em 07/01/2022.

MAGNO, A. **Tire seu projeto do papel com Scrum: Atitudes e práticas para realizar seus projetos no trabalho e na vida**. 1. ed. [S.l.]: Leya, 06/12/2019. 144 p.

MATERIAL-UI. **Material-UI: Um framework popular de React UI**. Disponível em: <https://v4.mui.com/pt/>. Acesso em: 20/01/2022.

RAGGETT, Dave et al. **Raggett on HTML 4**. Addison-Wesley Longman Publishing Co., Inc., 1998.

RIBEIRO, M. **Como elaborar um formulário de pesquisa perfeito**. 31/10/2017. Disponível em: <https://pluga.co/blog/como-elaborar-um-formulario-de-pesquisa/>. Acesso em: 07/01/2022.

SHNEIDERMAN, Ben. Direct manipulation for comprehensible, predictable and controllable user interfaces. In: **Proceedings of the 2nd international conference on Intelligent user interfaces**. 1997. p. 33-39.

SHNEIDERMAN, Ben. Designing the user interface: strategies for effective human-computer. **Interaction**, v. 3, 1998.

TIDWELL, Jenifer; BREWER, Charles; VALENCIA, Aynne. **Designing interfaces: patterns for effective interaction design**. Third edition. Beijing [China] ; North Sebastopol, CA: O'Reilly, 2020.

VEIGA, E. **A 'epidemia de abandono' dos animais de estimação na crise do coronavírus - BBC News Brasil**. 30/07/2020. Disponível em: <https://www.bbc.com/portuguese/brasil-53594179>. Acesso em: 05/01/2022.

VERCEL. **GitHub - vercel/next.js: The React Framework**. Disponível em: <https://github.com/vercel/next.js>. Acesso em: 03/01/2022.

VERCEL. **Next.js by Vercel - The React Framework**. Disponível em: <https://nextjs.org/>. Acesso em: 03/01/2022..