

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ALEXANDRE SANTOS DA SILVA JR.

**Componentes de Interface Adaptativos em  
Tempo Real Baseados na Interação com  
Usuário e Feedback**

Monografia apresentada como requisito parcial  
para a obtenção do grau de Bacharel em Ciência  
da Computação

Orientador: Prof. Dr. Marcelo Soares Pimenta

Porto Alegre  
2022

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões

Vice-Reitora: Prof<sup>a</sup>. Patricia Helena Lucas Pranke

Pró-Reitora de Graduação: Prof<sup>a</sup>. Cíntia Inês Boll

Diretora do Instituto de Informática: Prof<sup>a</sup>. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Ciência de Computação: Prof. Marcelo Walter

Bibliotecário-chefe do Instituto de Informática: Alexsander Borges Ribeiro

*“Computers are good at following instructions,  
but not at reading your mind.”*

— DONALD KNUTH

## **AGRADECIMENTOS**

Agradeço primeiramente aos meus pais, por todo o apoio incondicional ao longo da minha vida e jornada acadêmica, que possibilitaram e incentivaram que eu chegasse até aqui.

Aos amigos, pelo companheirismo e apoio ao longo de todo o tempo em que me dediquei a este trabalho.

Ao professor Marcelo Soares Pimenta, por me orientar, aconselhar e por toda ajuda e paciência com a qual me guiou.

A todos aqueles que contribuíram, de alguma forma, para a realização deste trabalho.

## RESUMO

Interfaces de usuário atuais possuem elementos fixos e não mutáveis, como botões e textos com cores, tamanhos e limites pré-definidos, no entanto, estes componentes podem não ser adequados para todos os usuários, por questões de acessibilidade e preferência pessoal. A solução proposta visa proporcionar experiências dinâmicas e adaptativas de acordo com a necessidade de cada usuário, atualizando os componentes da interface de usuário para prover uma melhoria na interação com o software, como por exemplo ajustando tamanho de fontes, área clicável de botões, temas e até funcionalidades de acessibilidade de maneira personalizada. Para tornar possível esta adaptatividade, é necessário obter e processar dados das interações do usuário com a interface para gerar perfis de uso, que por sua vez determinarão as modificações a serem aplicadas nos elementos da interface. Esta solução é então implementada visando compreender a viabilidade desta nova experiência em componentes de interface dinâmicos, personalizados e responsivos e testada por meio de uma aplicação de prova de conceito com foco na interação e experiência dos usuários.

**Palavras-chave:** Interface de usuário. experiência de usuário. adaptatividade. customização.

# **Real Time Adaptive Interface Components Based on User Interaction and Feedback**

## **ABSTRACT**

Current user interfaces are built with fixed and immutable elements, such as buttons and text with predefined colors, sizes and limits. However, these components might not be adequate for all users, considering accessibility and personal preference. The proposed solution envisions providing dynamic and adaptive experiences according to the needs of each user, updating the user interface components and allowing for an enhancement in the software interaction, such as adjusting font sizes, buttons clickable area, themes and even accessibility functionalities in a customized manner. To make this adaptivity possible, it is necessary to collect and process data regarding the user interactions with the interface to generate use profiles, that will in turn determine the changes that have to be applied to the interface elements. This solution is then implemented with the intent to comprehend the viability of this new experience in dynamic, customized and responsive interface components and tested through the use of a proof of concept application focused on user interaction and experience.

**Keywords:** user interface, user experience, adaptivity, customization.

## **LISTA DE ABREVIATURAS E SIGLAS**

UI	User Interface
UX	User Experience
NPM	Node Package Manager
MVC	Model-View-Controller
SUS	System Usability Scale

## LISTA DE FIGURAS

Figura 2.1 Interface do aplicativo mobile do banco Itaú com problemas de responsividade após configuração para uma fonte maior no aparelho celular .....	14
Figura 2.2 Conjunto de configurações de interface do Webmail .....	14
Figura 3.1 Principal proposta de Design e exemplos de guidelines e tópicos na página principal do Google Material Design System .....	16
Figura 3.2 Páginas principais e propostas das três soluções de frameworks e bibliotecas de componentes citados: Chakra, Bootstrap e Element.....	17
Figura 4.1 Visualização das tecnologias que compõem o stack "MERN" .....	22
Figura 4.2 Aplicação de prova de conceito demonstrada em seu estado padrão e um exemplo de customização escolhido por um usuário.....	24
Figura 4.3 O componente <i>Reactive Feedback</i> visualizado em seus três estágios de interação com o usuário .....	25
Figura 4.4 O componente <i>Reactive Button</i> visualizado com suas áreas de eventos de hit (acerto) e miss (erro) de clique destacadas .....	26
Figura 4.5 Arquitetura dos componentes desenvolvidos para a prova de conceito da solução.....	27
Figura 4.6 Conjunto principal de controles utilizados na implementação da aplicação interativa para teste de usuários em foco.....	28
Figura 4.7 Telas da aplicação mobile AFK Arena contendo controles extremamente pequenos e de difícil interação.....	33
Figura 5.1 Resultado agregado das respostas à primeira pergunta do questionário: "Acredito que eu gostaria de usar esta aplicação frequentemente" .....	37
Figura 5.2 Resultado agregado das respostas à segunda pergunta do questionário: "Achei o sistema desnecessariamente complexo" .....	38
Figura 5.3 Resultado agregado das respostas à terceira pergunta do questionário: "Acredito que o sistema foi fácil de usar" .....	39
Figura 5.4 Resultado agregado das respostas à quarta pergunta do questionário: "Acredito que precisaria do suporte de uma pessoa técnica para ser capaz de usar este sistema" .....	40
Figura 5.5 Resultado agregado das respostas à quinta pergunta do questionário: "Achei que as várias funções deste sistema foram bem integradas" .....	41
Figura 5.6 Resultado agregado das respostas à sexta pergunta do questionário: "Acredito que havia muita inconsistência neste sistema" .....	42
Figura 5.7 Resultado agregado das respostas à sétima pergunta do questionário: "Eu imaginaria que a maioria das pessoas aprenderia a usar este sistema muito rapidamente" .....	44
Figura 5.8 Resultado agregado das respostas à oitava pergunta do questionário: "Achei o sistema muito complicado de usar" .....	45
Figura 5.9 Resultado agregado das respostas à nona pergunta do questionário: "Me senti muito confiante usando o sistema" .....	46
Figura 5.10 Resultado agregado das respostas à décima pergunta do questionário: "Precisei aprender várias coisas antes que pudesse seguir em frente com este sistema" .....	47

## LISTA DE TABELAS

Tabela 4.1	Dados de interações do usuário com componentes da interface .....	29
Tabela 4.2	Dados de respostas do usuário à requisições de feedback.....	29
Tabela 4.3	Dados de ações tomadas pelo sistema baseadas em interações.....	29

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>11</b>
<b>1.1 Objetivo</b> .....	<b>12</b>
<b>1.2 Estrutura do Trabalho</b> .....	<b>12</b>
<b>2 INTERFACES ATUAIS E A ESCASSEZ DE CUSTOMIZAÇÃO</b> .....	<b>13</b>
<b>3 SISTEMAS DE DESIGN, FRAMEWORKS E ESTUDOS DE USABILIDADE</b> .....	<b>16</b>
<b>4 COMPONENTES DE INTERFACE ADAPTATIVOS EM TEMPO REAL</b> .....	<b>20</b>
<b>4.1 Reactive</b> .....	<b>22</b>
<b>4.2 Descrição da Base de Dados</b> .....	<b>28</b>
<b>4.3 Ambiente de Desenvolvimento</b> .....	<b>31</b>
<b>4.4 Metodologia</b> .....	<b>32</b>
<b>4.5 Motivação</b> .....	<b>33</b>
<b>4.6 Funcionalidades</b> .....	<b>34</b>
<b>5 ANÁLISE DA SOLUÇÃO PROPOSTA</b> .....	<b>36</b>
<b>5.1 Resultados do Questionário</b> .....	<b>36</b>
<b>5.2 Síntese dos Testes Aplicados</b> .....	<b>46</b>
<b>6 CONCLUSÃO</b> .....	<b>48</b>
<b>REFERÊNCIAS</b> .....	<b>50</b>

## 1 INTRODUÇÃO

Aplicações e serviços com elementos de interface de usuário atuais implementam algum nível de dinamismo em seus componentes, como opções adicionais instanciadas após seleções, tabelas que podem ter o conteúdo filtrado ou ordenado e até mesmo configurações de temas, como *light* e *dark*. Porém todas essas customizações são feitas após interação ativa do usuário, efetivamente tendo de navegar até um painel ou página de configurações na aplicação ou selecionando opções de acordo com um fluxo de formulários, contendo apenas elementos instanciados de forma responsiva e não adaptativa.

Este comportamento predominantemente estático de interfaces atuais, por mais que seja razoavelmente agradável para a maioria dos usuários, certamente não seria identificado como completamente adequado por todos os usuários que fazem uso destas aplicações diariamente. Além de questões mais estéticas como preferências pessoais de temas, cores e posicionamento de elementos, ainda há o grande campo da acessibilidade e a necessidade de ajuste de elementos para pessoas portadoras de algum tipo de deficiência, mesmo que leve, seja esta visual, motora, ou de outra natureza, e que poderia se beneficiar, por exemplo, de componentes de interface maiores para facilitar a interação e evitar erros, fontes maiores e até mesmo um tema de alto contraste para facilitar a leitura.

A solução de Componentes de Interface Adaptativos em Tempo Real visa dar o próximo passo na capacidade de customização e dinamismo destas interfaces de usuário, permitindo que diversos componentes sejam estendidos e se adaptem de forma inteligente às necessidades ou preferências específicas de todos os usuários interagindo com o software, de forma especializada e personalizada. Com a utilização destes componentes, as interfaces passam não apenas a atender a maioria dos usuários ou apenas aqueles com gostos e necessidades mais genéricas, mas sim todos que virão a interagir com a aplicação com a mesma atenção a estas especificidades.

Dentre os componentes idealizados e desenvolvidos, a solução disponibiliza, por exemplo, botões que possuem áreas clicáveis ajustáveis - que podem aumentar e diminuir ao longo da interação com a aplicação - caso o usuário erre muitas vezes ao tentar pressionar a área próxima, requisições de feedback que podem ser instanciadas de maneira não disruptiva para que o usuário possa escolher diferentes temas, idiomas, cores personalizadas, opções de acessibilidade, como tamanho da fonte geral da aplicação e até mesmo recomendações dinâmicas, sugerindo a modificação de conteúdos que possam vir a ser mais relevantes para casos de uso específicos, entre outros.

## **1.1 Objetivo**

Este trabalho tem por objetivo então impactar as áreas de Interface e Experiência de Usuário (UI/UX) por meio de configurações dinâmicas, adaptativas e inteligentes em tempo real, baseando-se na captação e análise de dados da interação do usuário com a interface para recomendações e ajustes destas experiências e utilizando-se de componentes dinâmicos para otimizar a experiência do usuário de maneira mais informada.

Estes componentes dinâmicos da interface de usuário são atualizados ao longo da sessão para prover uma melhoria na interação com o software, como por exemplo ajustando tamanho de fontes, área clicável de botões, temas e até funcionalidades de acessibilidade de maneira personalizada. Para tornar possível esta adaptatividade, se torna necessário obter e processar dados das interações do usuário com a interface para gerar perfis de uso, que por sua vez determinarão as modificações a serem aplicadas nestes elementos da interface.

Esta solução é então implementada e aplicada em uma prova de conceito visando compreender a viabilidade desta nova experiência em componentes de interface dinâmicos, personalizados e responsivos por meio de testes de usuário, formulários sobre as interações realizadas e por fim a análise destes.

## **1.2 Estrutura do Trabalho**

Após esta introdução, os principais pontos considerados problemáticos em interfaces atuais que deram origem a este estudo são abordados na seção 2. Na sequência, a seção 3 apresenta algumas soluções em foco por outros sistemas de design, frameworks e trabalhos relacionados, enquanto a seção 4 traz uma descrição detalhada da solução proposta por este trabalho e o processo de implementação dos componentes e prova de conceito. Por fim, são abordados os resultados finais do trabalho em forma de análise de testes e dados feitos a partir de interações guiadas e questionários com usuários na seção 5.

## 2 INTERFACES ATUAIS E A ESCASSEZ DE CUSTOMIZAÇÃO

Usuários que interagem continuamente com interfaces digitais dificilmente virão a demonstrar preferências e necessidades idênticas quanto a melhor forma de visualizar e interagir com estas. Alguns usuários podem ter idade avançada e deficiência visual, necessitando de fontes maiores para maior conforto na leitura, ou problemas motores que poderiam ser mitigados com o uso de botões e controles maiores que o padrão, facilitando cliques e a interação como um todo (DAREJEH; SINGH, 2013). Até mesmo simples diferenças em preferências para cores, iluminação e contraste da tela durante a interação poderiam ser aplicadas com customização de temas. Logo, uma interface única, padrão e imutável é incapaz de atender a todos estes usuários e cenários de maneira completamente adequada.

Uma interface bem planejada e implementada pode vir a atender uma grande parte dos usuários mais comuns que acessam esta, mas sempre haverá uma parcela de usuários neste total com necessidades especiais ou preferências específicas que ou terá a sua experiência continuamente prejudicada em alguma instância, ou poderá vir até mesmo a desistir de utilizar o serviço completamente por conta destas limitações na sua interação contínua, preferindo outros serviços que forneçam capacidades de acessibilidade diferenciadas ou maior customização de seus componentes que o atual oferece.

Diversas ferramentas e parâmetros facilitadores de acessibilidade estão em crescimento no mercado, principalmente na área de serviços Web, possibilitando que softwares especializados para acessibilidade possam capturar de forma mais correta e precisa textos e controles na aplicação para comunicar ao usuário. No entanto, a maioria destas ferramentas ainda necessita de implementação especializada no Front-end da aplicação para que fique ativa e disponível, desenvolvedores precisam informar para todos os elementos relevantes campos especiais e indicadores adicionais.

Além destas, é comum também encontrar opções automáticas de tradução e aumento de fonte em navegadores e opções de smartphones. Porém, comumente, utilizar destes "facilitadores" acaba por prejudicar muitas aplicações ou sites não devidamente preparados e implementados de forma responsiva o suficiente para as diferentes proporções e caracteres que estas diferentes configurações podem trazer, por vezes escondendo textos e até mesmo controles por extrapolação de limites em modais e janelas como pode ser visto na figura 2.1. Deve ser considerada também, então, a alta complexidade de desenvolvimento para aplicações e serviços web completamente responsivos para aplicações mobile

(DEHLINGER; DIXON, 2011).

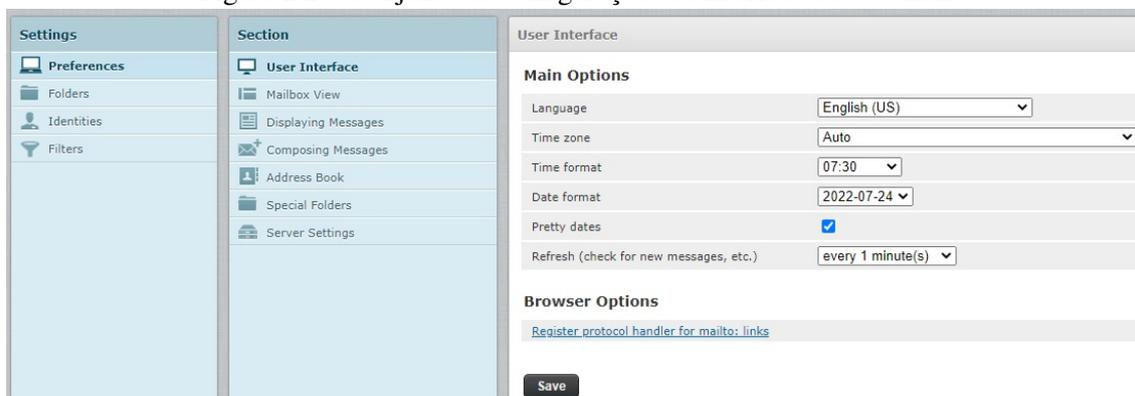
Figura 2.1 – Interface do aplicativo mobile do banco Itaú com problemas de responsividade após configuração para uma fonte maior no aparelho celular



Fonte: Aplicação Mobile do Banco Itaú

Outro fator negativo neste cenário muitas vezes é a impossibilidade de identificar estas necessidades especiais de um usuário que acessa a aplicação, como determinar o que este usuário precisa ou prefere durante sua sessão de uso do serviço. O cenário comum consiste de uma página de configurações com opções que permitem ao usuário customizar pontos específicos da interface de acordo com seus gostos e necessidades, mas que muitas vezes acaba por ter um difícil acesso ou poucas opções que efetivamente modifiquem a interface em si, como por exemplo no serviço Webmail, que conta com poucas opções no menu de interface de usuário, como podem ser visto na figura 2.2.

Figura 2.2 – Conjunto de configurações de interface do Webmail



Fonte: Página de Configurações do Serviço Web do Webmail

Supondo que todos os problemas citados anteriormente sejam resolvidos de al-

guma forma, ainda é necessário que haja uma validação da efetividade destas mudanças e ajustes na interface, e se estas realmente foram benéficas ou pelo menos bem aceitas pelo usuário. Uma interação comum do usuário com estas alterações de preferências acaba sendo ligar e desligar alternadamente modificadores ou tentar aleatoriamente aplicar diferentes valores de configurações até que encontre algo que se encaixe de certa forma em suas expectativas, por meio de tentativa e erro.

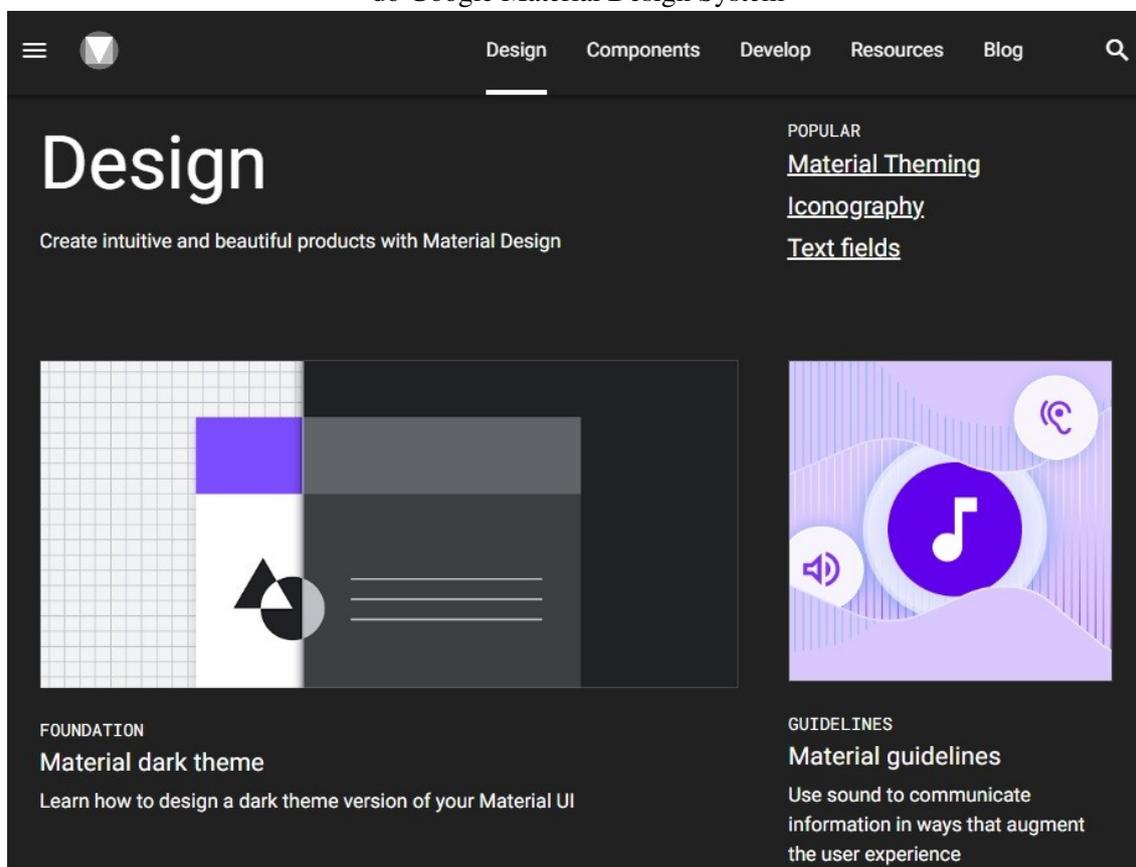
Estas situações acabam prejudicando o usuário ao longo de sua interação com a aplicação, fazendo com que este enfrente condições de interface indesejadas, perca tempo em busca de configurações para problemas específicos, descubra quais valores utilizar nos campos disponíveis ou como sinalizar a ausência de algo que considera necessário ou até mesmo um defeito na aplicação, entre outros. Além disso, este usuário provavelmente interage com múltiplas aplicações diferentes ao longo do dia, e engajar em ações para configurar e manter cada uma destas individualmente é uma quantidade de tempo e esforço considerável, se tornando uma atividade maçante.

Proporcionar uma interface apropriada para o usuário, do seu ponto de vista individual e único, não apenas traz vantagens para este, que passará a ter um tempo de interação mais agradável e adequado com o serviço, mas também para a própria aplicação e os responsáveis por esta, que terão um produto diferenciado no mercado, com o potencial de uma maior retenção de usuários e até mesmo um melhor conhecimento sobre sua base de usuários se fizerem uso destas informações de forma responsável.

### 3 SISTEMAS DE DESIGN, FRAMEWORKS E ESTUDOS DE USABILIDADE

As soluções mais comuns e presentes em aplicações populares para questões de usabilidade atualmente existem em torno de propostas para uso geral e padronização de componentes e interações, tanto em sistemas e diretrizes de design como no caso do Google Material Design System (figura 3.1) (MATERIAL. . . , 2022), quanto bibliotecas ou pacotes de componentes que seguem nativamente algumas das diretrizes mais comuns e as implementam, como Bootstrap (BOOTSTRAP. . . , 2022) (SPURLOCK, 2013), Chakra (CHAKRA. . . , 2022) e Element (ELEMENT. . . , 2022) (figura 3.2). Estas soluções buscam ater-se a padrões de usabilidade comuns que devem ser agradáveis e suficientes para a maioria dos usuários se apenas instanciados os componentes e diretrizes na implementação de uma aplicação, mas ainda assim as questões de customização e adaptabilidade ficam sob responsabilidade dos desenvolvedores, visto que estas ferramentas não costumam prever ou facilitar muitas opções de modificação de seus componentes, principalmente de forma dinâmica.

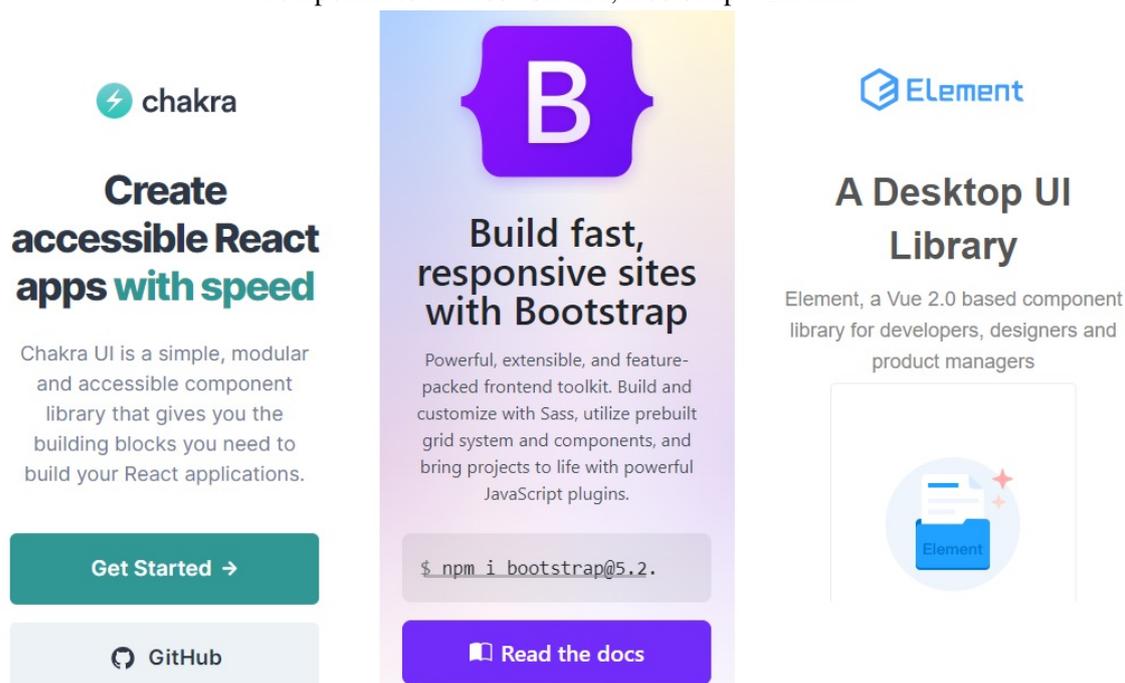
Figura 3.1 – Principal proposta de Design e exemplos de guidelines e tópicos na página principal do Google Material Design System



Fonte: Material Design: <https://material.io/design>

Um sistema de design como o Google Material tem como principal objetivo centralizar guidelines e padrões de boas práticas e experiências comprovadamente majoritariamente positivas para usuários, fornecendo tanto artigos e notícias para o aprendizado de desenvolvedores quanto efetivamente normas ou pelo menos linhas gerais para quase todos os aspectos da concepção de uma interface, desde tópicos de baixo nível como espaçamentos e margens sugeridas para botões e modais como de alto nível, como exemplos de posicionamento adequado de itens em serviços de blog, vendas, notícias, entre outros. É importante notar também que no caso do Google Material Design, além de um sistema de design, também inclui uma biblioteca de componentes chamada Material Design, atualmente na sua terceira versão, que inclui, assim como os outros frameworks citados, componentes como tabelas, painéis e *cards* já no padrão utilizado e recomendado pela própria Google nas interfaces de aplicações que desenvolve.

Figura 3.2 – Páginas principais e propostas das três soluções de frameworks e bibliotecas de componentes citados: Chakra, Bootstrap e Element



Fontes: Chakra: <https://chakra-ui.com/>, Bootstrap: <https://getbootstrap.com/>, Element: <https://element.eleme.io/>

Já os frameworks como Bootstrap e Chakra disponibilizam bibliotecas de componentes visuais e iterativos que podem ser importados e utilizados em aplicações implementadas por desenvolvedores em Front-end. Estas bibliotecas se encontram disponíveis e de fácil acesso, como no *Node Package Manager (NPM)* e implementam uma camada adicional principalmente visual mas também funcional sobre componentes básicos do HTML como botões, inputs, tabelas, formulários, entre outros. A principal contribui-

ção destas ferramentas é proporcionar a desenvolvedores um modo mais fácil e rápido de implementar interfaces com uma aparência e experiência mais agradável desde o início, além de fornecer um guia para a utilização dos componentes disponíveis. Apesar destes facilitadores, um dos grandes problemas ao utilizar um framework como este é a falta de um visual único e customizado para a aplicação, visto que os componentes em seu estado padrão destas bibliotecas já são geralmente utilizados em aplicações genéricas e seu "*look and feel*" pode ser facilmente identificado como algo comum e de baixo esforço. Aplicações de grande porte ou que necessitem de uma aparência ligada a uma marca ou uma identidade visual mais característica não tendem a possibilitar o uso destas bibliotecas, e demandam o desenvolvimento de componentes customizados pelos desenvolvedores para atender as demandas do design.

Algumas aplicações mais utilizadas no mercado como Gmail, Instagram e Slack já implementam de forma personalizada algum nível de customização da interface pelo usuário, como diferentes temas (modos *light* e *dark*), opções de linguagem e o quão compacto é mostrado o conteúdo. No entanto, estas opções, além de escassas, comumente precisam ser ativamente buscadas pelo usuário caso este esteja insatisfeito com a experiência, por meio de telas de configurações e diferentes menus e submenus que contém não apenas estas opções mas diversas outras sobre a aplicação em geral, como detalhes da conta, notificações e segurança, geralmente sendo confuso para que o usuário encontre as opções específicas desejadas.

Já no campo da pesquisa, há diferentes propostas para diversos componentes específicos, como menus (BOUZIT et al., 2016), formulários (ZIMMERMANN et al., 2015) botões, modais, entre outros. Estas propostas variam em foco entre implementações mais apropriadas de componentes para a maioria dos usuários e customizações ativas ou passivas específicas para casos de uso e tarefas, incluindo testes e validações para aplicação de cada opção. Em grande parte, as soluções são para aplicações com situações características de uso, como acesso a informações em bancos de dados e preenchimento de formulários, e não para aplicações gerais.

Há também estudos teóricos de alto nível sobre as vantagens e desvantagens que a implementação de componentes adaptativos pode trazer para aplicações e seus usuários (BENYON, 1993), salientando principalmente os custos adicionais ao desenvolver aspectos adaptativos e customizáveis em aplicações (DEHLINGER; DIXON, 2011) e o real impacto benéfico que estas alterações podem trazer nas interações e experiência dos usuários.

Por fim, identifica-se que as soluções mais populares disponíveis no mercado atualmente possuem um aspecto mais genérico em suas implementações, visto que aplicações mais básicas ou provas de conceito tendem a utilizar por praticidade, enquanto aplicações maiores e com maior investimento criam seus próprios componentes de acordo com a marca e identidade visual desejada. Já quanto a adaptatividade, as soluções encontradas se concentram mais no campo teórico e de estudo de viabilidade e impacto com usuários, comumente em pequena escala ou com foco em componentes específicos e não abordagens gerais para aplicações. A solução proposta neste trabalho busca trazer uma parte do impacto destes estudos que visam a adaptatividade e customização de componentes para uso conjunto com estas bibliotecas populares ou de fácil inserção com implementações personalizadas, criando um novo tipo de biblioteca de componentes com funcionalidades de adaptatividade integradas e de mais fácil implementação e utilização por desenvolvedores enquanto mantendo-se em guidelines de design desejáveis e adequadas como visto também em sistemas de design.

#### 4 COMPONENTES DE INTERFACE ADAPTATIVOS EM TEMPO REAL

Visando não focar no que torna usuários que interagem com aplicações similares de forma geral, mas sim distintos, a solução de Componentes de Interface Adaptativos implementa mecanismos que podem ser instanciados ao longo da jornada de utilização da aplicação de maneira natural e não disruptiva para extrair importantes informações sobre algumas das principais preferências do usuário como indivíduo, aplicando atualizações em componentes que venham a tornar a interface mais agradável de acordo com estas preferências informadas.

Para melhor conhecer os gostos ou necessidades do usuário em questão são gerados ao longo do uso da aplicação pequenos modais no canto inferior direito - permitindo que sejam facilmente ignorados caso o usuário não tenha a intenção de interagir com estes, inclusive em telas menores como de aparelhos móveis - que trazem perguntas sobre aspectos específicos da interface que podem vir a ser respondidas caso o usuário tenha interesse, atualizando assim estes aspectos de acordo com a escolha, ou mantendo-os caso sejam ignoradas. Algumas dessas interações perguntam ao usuário se este gostaria de aumentar ou diminuir o tamanho da fonte utilizada, com foco em acessibilidade, ou também solicitam que este escolha uma cor de seu gosto entre algumas opções, para customização do tema principal da aplicação, entre outras que serão abordadas em detalhes na próxima sessão.

Caso o usuário venha a interagir com um destes modais e escolha uma opção diferente da atual ou que modifique componentes, esta seleção é armazenada em um perfil específico do usuário logado e um evento é sinalizado e enviado para uma camada superior da aplicação para que os parâmetros relevantes sejam atualizados nos componentes dinâmicos. Além das mudanças aplicadas, um evento sobre a escolha também é armazenado para que o sistema, por exemplo, volte a perguntar no futuro ou deixe de perguntar novamente caso uma escolha tenha sido feita ou não.

É importante adicionar que esta solução não tem como objetivo criar um novo framework visual ou competir com as opções citadas anteriormente, como Bootstrap, Element ou até mesmo componentes feitos de forma customizada por desenvolvedores, mas sim adicionar uma nova camada de inteligência sobre estes componentes visuais. Ao encapsular um componente comum em um componente *Reactive* - nome dado a solução da camada de inteligência adaptativa que engloba os componentes comuns - este herda novas capacidades de se atualizar de acordo com as preferências do usuário, podendo ter

seu espaçamento, cores, tamanho de fonte, entre outros, alterado em tempo real.

Expandindo um pouco o exemplo que traz ao usuário a escolha de uma cor de seu gosto, esta decisão faz com que o sistema reaja a interação e aplique a cor selecionada a um parâmetro da paleta de cores do tema *Reactive* dinâmico, que atualiza a cor primária da aplicação, sendo esta disseminada para todos os componentes *Reactive* que estão conectados nativamente ao tema, atualizando, por exemplo, botões da cor padrão azul para a nova cor vermelha escolhida pelo usuário, além de também recolorir bordas sólidas de elementos de *input*, modais, e todos os componentes que sejam relevantes, ou seja, que façam uso do parâmetro de cor primária.

Outra capacidade adaptativa e um pouco mais experimental destes componentes *Reactive*, como citado anteriormente, é a de logar eventos relevantes e aplicar alterações na lógica do sistema e de componentes de acordo com estes eventos. Por exemplo, um botão *Reactive* é envolto em uma pequena margem que pode detectar se o usuário errou um clique próximo ao botão, supostamente indicando que o usuário tentou pressionar o botão mas não teve sucesso. Caso uma determinada quantidade destes eventos classificados como "*misclick*" (cliques que falharam em interagir com o botão) seja identificada, a camada superior da solução identifica que este limiar de eventos foi atingido e aplica uma alteração em todos os componentes de botão, aumentando levemente a área destes em alguns pixels, na intenção de que o usuário venha a ter mais acertos em interações futuras com estes controles.

Também é importante notar que o tema da aplicação é completamente atualizado em tempo real a cada escolha dinâmica do usuário que é recebida como um evento pelo perfil, então se um usuário responde a uma solicitação de feedback escolhendo trocar o tema de *light* para *dark*, o tema da aplicação é atualizado e todos os componentes são renderizados novamente, com background escuro, letras claras, entre outros. A aplicação e otimização desta atualização dinâmica é possível por meio da tecnologia chamada de *Virtual DOM*, disponível nativamente no framework ReactJS, que renderiza novamente apenas os componentes que efetivamente tiveram mudanças visuais do estado anterior para o novo, poupando recursos nos elementos que se mantiveram inalterados na interface.

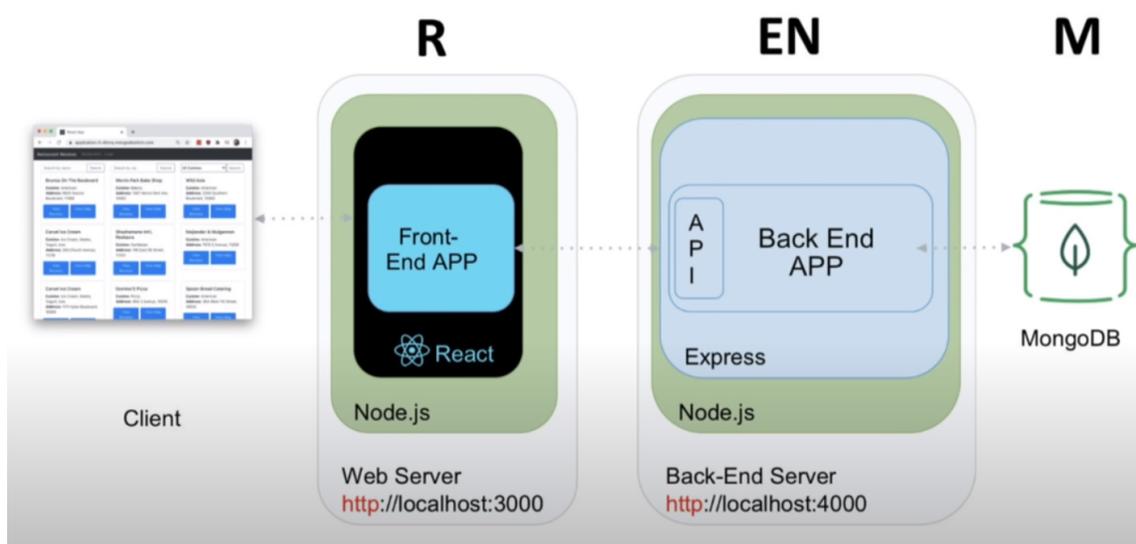
Por fim, todas estas alterações e atualizações em componentes permitem tanto que o usuário volte atrás, com um comando de *Undo* (Desfazer) nos modais de feedback, quanto apresentam opções no painel de configurações da aplicação (que também possui um atalho disponível após cada atualização de componentes) caso o usuário escolha visualizar ou alterar todos os campos possíveis e também retornar a opções anteriores. Desta

forma, a solução traz diversas opções de customização para que o usuário aplique e verifique em tempo real o resultado de suas escolhas e possa tomar as melhores decisões para customizar a sua experiência de forma instantânea e adequada para suas necessidades e preferências.

#### 4.1 Reactive

Para a implementação da solução *Reactive* - a prova de conceito dos componentes idealizados neste trabalho - foi desenvolvida uma aplicação com tecnologias e linguagens comuns e populares ao cenário de desenvolvimento atual, utilizando MongoDB, Express, ReactJS e NodeJS - conjunto também conhecido como o stack MERN - como apresentado na Figura 4.1. ReactJS foi utilizado para a criação de componentes como uma biblioteca ou *framework* (RATHNAYAKE et al., 2019), permitindo ao desenvolvedor da solução instanciar estes componentes para obter as principais vantagens propostas que serão abordadas em detalhes na sequência. Para o objetivo de coleta e observação de dados para a lógica de alteração de parâmetros de componentes e testes com usuários foi implementado um Back-end em NodeJS e Express para comunicação com um banco de dados em MongoDB e também uma versão utilizando a *local storage* do navegador.

Figura 4.1 – Visualização das tecnologias que compõem o stack "MERN"



Fonte: O Autor

Duas possibilidades de armazenamento foram testadas nesta solução como prova de conceito no modelo de serviço, utilizando *cookies* e *storages* disponíveis em navegadores para manutenção do perfil do usuário e eventos, e a criação de um servidor em nuvem

capaz de não apenas armazenar e processar essas informações por meio de uma lógica implementada em Back-end, mas também de gerar perfis traçando as preferências gerais de usuários e até mesmo agregá-las com as de outros usuários de forma anônima.

A lista de componentes *Reactive* criados especificamente para esta prova de conceito e teste da aplicação de demonstração é a seguinte:

- Gerador de pop-up para solicitação de feedback do usuário, aceitando configurações para definição de todas as modificações listadas abaixo.
- Temas gerais da aplicação nas opções *Dark*, *Light* e Alto Contraste.
- Customização da cor primária da aplicação nas opções de Vermelho, Azul e Verde.
- Linguagem da aplicação nas opções de Inglês e Português.
- Modificador de tamanho de fonte da aplicação.
- Modificador do tamanho de botões da aplicação.
  - Este último também conta com o mecanismo de reação a eventos de erro repetidos do usuário e ajuste dinâmico sem interação por feedback necessária, por meio de botões especializados.

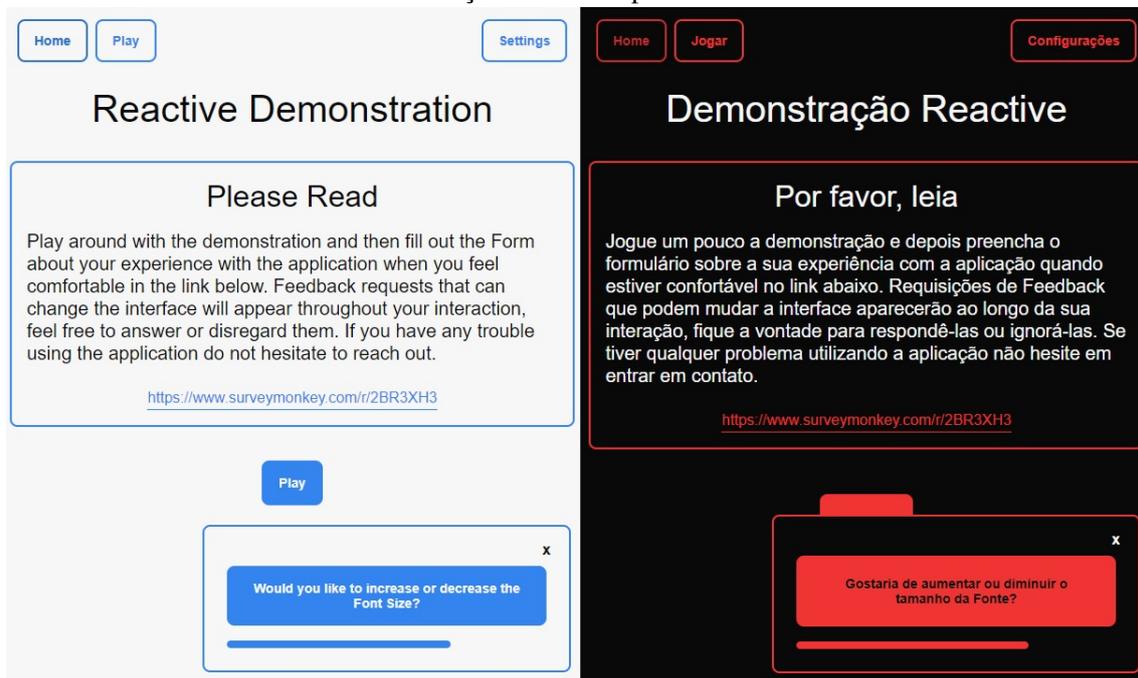
Estes componentes funcionam de forma completamente independente e atualizam suas propriedades internas de acordo com informações lidas de contextos de tema - que armazena propriedades em objetos como cor, tamanho e espaçamento - e perfil do usuário, que é armazenado como um objeto dicionário com chaves mapeando todas as propriedades que podem ser configuradas - como por exemplo a cor primária e linguagem citados acima - enquanto os valores mapeiam as escolhas do usuário - como 'azul' para cor e 'português' para a linguagem - sendo estas entradas vazias caso uma opção não tenha sido ativamente escolhida pelo usuário, caso que é devidamente tratado como verificação de nulo pela aplicação e tendo as opções *default* (padrão) ativas.

Para exemplificar um objeto do contexto de perfil do usuário, foi extraída uma amostra da aplicação em execução com praticamente todas as propriedades alteradas de alguma forma pelo usuário, tomando a seguinte forma:

- "theme": "dark",
- "primaryColor": "F03434",
- "language": "pt",
- "fontSizeModifier": 0,
- "buttonSizeModifier": 6

Neste caso, o usuário trocou do tema padrão *light* para *dark*, alterou a cor principal da aplicação de azul para vermelho (armazenado em código hexadecimal), a linguagem de inglês para português, e também aumentou o tamanho dos botões em 6 pixels (aplicado nas quatro direções) mas manteve o tamanho da fonte (neste caso o usuário ativamente escolheu outro valor e retornou ao padrão, caso contrário o campo não constaria no dicionário do perfil ou estaria nulo). A aplicação pode ser vista no seu estado padrão e no estado alterado por estas configurações na figura 4.2.

Figura 4.2 – Aplicação de prova de conceito demonstrada em seu estado padrão e um exemplo de customização escolhido por um usuário



Fonte: O Autor

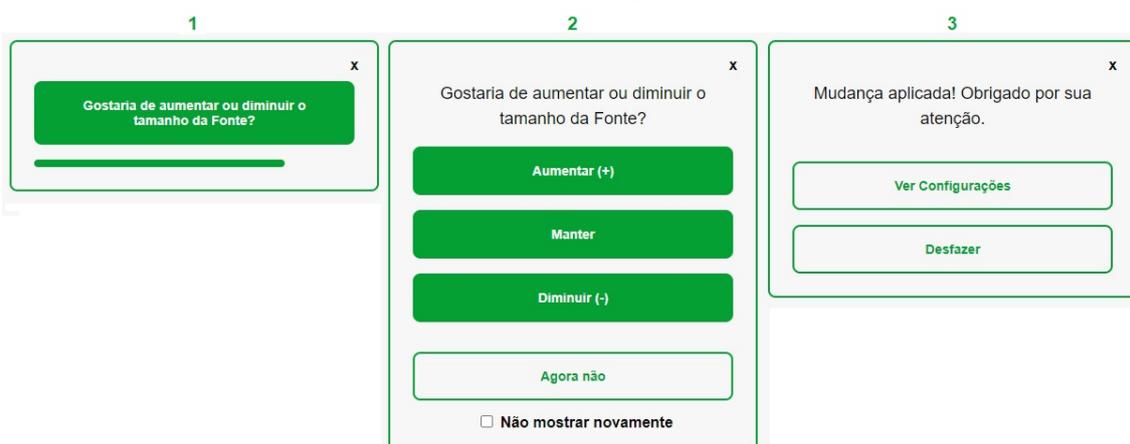
O componente de geração de pop-up para solicitação de feedback do usuário, chamado de forma simples de *Reactive Feedback*, é composto de três principais componentes - todos contidos em um modal pequeno e minimamente interruptivo à experiência do usuário instanciados no canto inferior direito da tela (como na figura 4.2) e que podem ser fechados por um ícone em formato de "x" - que correspondem às etapas do feedback (e que podem ser visualizadas na figura 4.3):

1. A pergunta inicial, ou *preview*: É composto de um simples botão contendo a pergunta referente ao feedback a ser questionado, como por exemplo "*Gostaria de selecionar um Idioma diferente?*", ou "*Gostaria de aumentar ou diminuir o tamanho da Fonte?*", e um pequeno indicador na área inferior do modal do tempo até que esta requisição seja fechada automaticamente, na forma de uma barra da cor

primária que diminui até desaparecer com o modal, após o tempo padrão escolhido de cinco segundos.

2. As opções, ou o controle de feedback em si: Além de trazer novamente a pergunta principal como descrita anteriormente, mas apenas em formato de texto - como título do modal e não um controle - traz também todas as opções referentes ao feedback ativo, como por exemplo "Português" e "Inglês" para a pergunta de idioma e "Aumentar", "Manter", e "Diminuir" para a pergunta sobre o tamanho da fonte, e por fim sempre um botão auxiliar para fechar o controle com o texto "Não agora" e um *checkbox* para "Não perguntar novamente", fazendo com que a aplicação não volte a questionar sobre o tema ativo.
3. O indicador de sucesso do feedback aplicado: Por fim, caso o usuário não feche o modal manualmente ou deixe-o fechar automaticamente e venha a escolher uma opção que atualiza a interface, é mostrada então a última etapa do modal de *Reactive Feedback*, já com a mudança aplicada e renderizada em toda a aplicação, com um texto afirmativo, "*Mudança aplicada! Obrigado por sua atenção.*", seguido de um botão de atalho que leva o usuário para a página de configurações - caso este queira reavaliar as opções escolhidas ou verificar todas as opções disponíveis - e um outro controle de "*Undo*" para desfazer a mudança da opção escolhida, voltando a configuração para a versão anterior ou a padrão do sistema, caso esta não existisse.

Figura 4.3 – O componente *Reactive Feedback* visualizado em seus três estágios de interação com o usuário



Fonte: O Autor

As opções de feedback disponíveis são apenas diferentes conjuntos de configurações contendo o título, as opções, e as ações que estas opções realizam ao serem selecionadas e desfeitas, além de qualquer customização visual dos controles (como as cores em si

para a seleção de cor primária ou uma prévia dos botões nos temas a serem selecionados) que são passadas ao controle de feedback por meio de um dicionário pré-configurado e importado por este, e que naturalmente poderiam vir a ser atualizadas e modificadas ao longo do tempo de vida da aplicação.

Já os botões especializados da aplicação, chamados de *Reactive Button*, são bem mais simples, e apenas adicionam uma camada extra de lógica e espaçamento a um componente de botão comum que pode ser passado por parâmetro e ser completamente customizado ou até mesmo vir de um framework visual que já venha a prover este componente por padrão, como um botão do Chakra ou Element. Além das capacidades comuns aos outros elementos da aplicação, como mudança de sua cor, fonte, idioma e tamanho de acordo com o contexto geral do tema, o *Reactive Button* também é capaz de identificar dois eventos por meio de *handlers*, chamados de *hit* (acerto) e *miss* (erro), que sinalizam para a central lógica da solução que o usuário acertou ou errou (na área próxima ao botão) um clique no botão em questão, adicionando informações como o id do botão, o tipo do evento (neste caso um clique) e se o evento teve ou não sucesso. As áreas relevantes para cada evento descrito podem ser visualizadas na figura 4.4.

Figura 4.4 – O componente *Reactive Button* visualizado com suas áreas de eventos de hit (acerto) e miss (erro) de clique destacadas



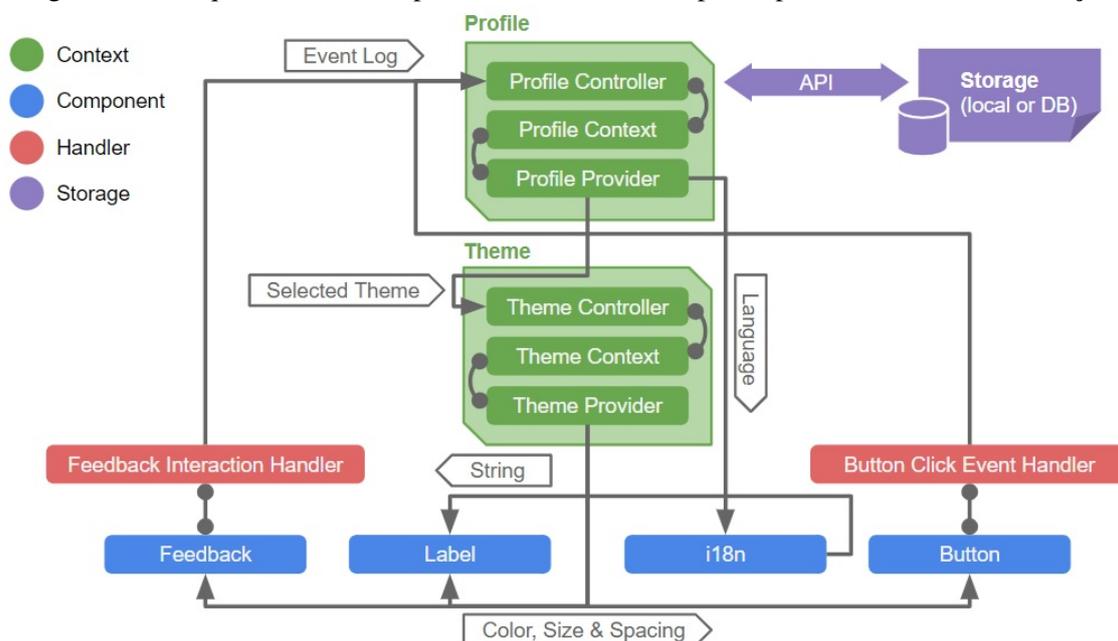
Fonte: O Autor

A lógica da solução *Reactive* que se encarregará da decisão de aumentar ou não o tamanho de todos os botões da aplicação de acordo com os eventos recebidos, caso uma quantidade de *misses* seguidos seja identificada de acordo com um limiar estipulado, e não o botão, que apenas enviará os eventos relevantes.

Os contextos de tema e perfil do usuário citados acima e detalhados visualmente na figura 4.5 são implementações de um dos facilitadores proporcionados pelo framework ReactJS, chamados de *React Hooks*, que são basicamente implementações de um padrão arquitetural de *Model-View-Controller* (MVC) (LEFF; RAYFIELD, 2001). Este *Hook* em

específico, chamado de *Context* (o *Model* do padrão arquitetural), deve ser integrado por um *Controller* (de mesmo nome no padrão) e um *Provider* (paralelo ao *View*), que basicamente controlam a entrada, manipulação e saída (ou visualização) de dados, enquanto o *Context* em si armazena e atualiza dados organizadamente, mesmo em um ambiente com processamento assíncrono, garantindo a integridade destes dados e evitando concorrência e condições de corrida, visto que vários componentes podem tentar atualizar e ler estes dados centralizados a qualquer momento. Para organização interna destes contextos e também de vários componentes, também são utilizados os *Hooks* de *State*, que provém variáveis e seus *setters* (funções que atualizam estas variáveis) de maneira conjunta, e *Effect*, que é capaz de captar mudanças nestes *States* ou eventos relacionados ao render da janela, como *mount* e *unmount* (quando o componente é instanciado ou removido do render), e disparar funções designadas pelo desenvolvedor. Esta nova maneira de implementação utilizando *Hooks* foi introduzido no framework ReactJS na versão 16.8, e substituiu o padrão anterior de criação de classes, estados e funções internas por este novo padrão de funções e *Hooks*.

Figura 4.5 – Arquitetura dos componentes desenvolvidos para a prova de conceito da solução



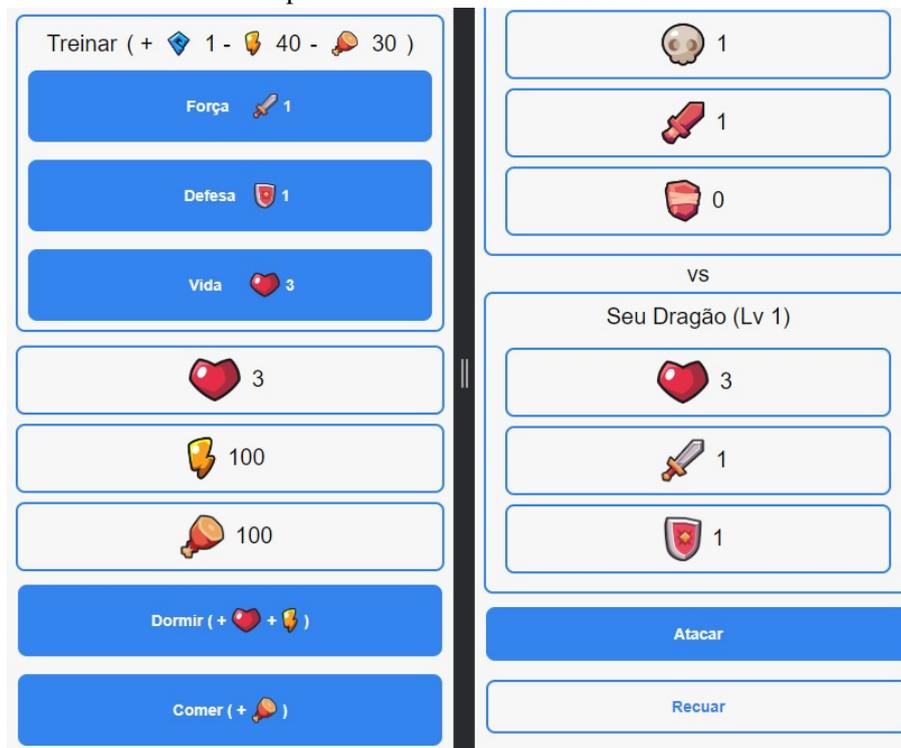
Fonte: O Autor

É importante notar que esta solução, como citado anteriormente, não visa competir com os frameworks e bibliotecas já existentes e estabelecidos no mercado, mas sim adicionar uma nova camada de lógica e customização dinâmica a estes ou a componentes próprios dos desenvolvedores de maneira adicional. O formato final visualizado para esta prova de conceito é de uma biblioteca ou pacote, como no caso do *Node Package Mana-*

ger (NPM), formato em que os frameworks citados anteriormente são disponibilizados, para que desenvolvedores possam mais facilmente adicionar e integrar esta adaptabilidade por meio da interação com usuários, tornando a experiência de suas aplicações mais apropriadas, personalizadas e interessantes para estes.

Por fim, apenas para o propósito de teste de usuários e implementação de uma aplicação baseada na prova de conceito, foi desenvolvido um breve jogo interativo (podendo ser completado entre 2 a 5 minutos) que demanda do usuário interações com os componentes desenvolvidos para progredir. O teste e o formulário em si serão descritos em mais detalhes na seção 5. Alguns controles em foco da aplicação podem ser visualizados na figura 4.6.

Figura 4.6 – Conjunto principal de controles utilizados na implementação da aplicação interativa para teste de usuários em foco



Fonte: O Autor

## 4.2 Descrição da Base de Dados

A Base de Dados que viabiliza esta solução armazena informações tanto das interações relevantes do usuário com a interface em si quanto das respostas a perguntas de feedback que são instanciadas e respondidas. Componentes como o *Reactive Feedback* e *Button* são responsáveis por disparar estes eventos com as informações relevan-

tes, enquanto a central de processamento destes eventos presente no *Context* principal da aplicação prepara o *payload* para ser processado e armazenado. Estas requisições, suas respostas e as ações tomadas pelo sistema para modificá-las também são armazenadas.

Estas interações com o usuário são armazenadas da seguinte forma, com exemplos de alguns dados para melhor compreensão:

Tabela 4.1 – Dados de interações do usuário com componentes da interface

<b>user_id</b>	<b>interaction_id</b>	<b>element_id</b>	<b>event_info</b>	<b>timestamp</b>
"alexandre"	"button_click"	"submit_account_creation_form_button"	"{ click_success: false, click_error_offset: [2px, -5px] }"	"2021-09-20 11:52:23"
"alexandre"	"button_click"	"submit_account_creation_form_button"	"{ click_success: true }"	"2021-09-20 11:52:25"
"admin"	"table_click"	"user_info_table"	"{ table_row_click: \"user_name\" }"	"2021-09-20 11:55:34"

Tabela 4.2 – Dados de respostas do usuário à requisições de feedback

<b>user_id</b>	<b>feedback_id</b>	<b>answer</b>	<b>timestamp</b>
"alexandre"	"theme_selection"	"{ answered: true, response: \"dark\" }"	"2021-09-17 10:31:56"
"admin"	"font_size_feedback"	"{ answered: false }"	"2021-09-20 10:35:22"
"alexandre"	"font_size_feedback"	"{ answered: true, response: \"too_small\" }"	"2021-09-20 10:45:02"

Também é necessário manter dados sobre o histórico de ações que foram tomadas pelo sistema, provenientes de interações ou feedbacks identificados.

Tabela 4.3 – Dados de ações tomadas pelo sistema baseadas em interações

<b>user_id</b>	<b>action_id</b>	<b>event</b>	<b>timestamp</b>
"alexandre"	"theme_change"	"{ previous_theme: \"light\", new_theme: \"dark\" }"	"2021-09-17 10:32:00"
"alexandre"	"font_change"	"{ previous_size: \"18\", new_size: \"20\" }"	"2021-09-20 10:45:02"

Com estas três principais fontes de dados, inicialmente, temos um volume pequeno de informações no sistema. Para possibilitar testes mais céleres com os usuários a quantidade de prompts para feedback e também o roteiro de interações terá de ser mais

intenso do que um caso de uso comum, que é esperado que venha a ter no máximo um pedido de feedback e muito poucas ações de alteração no sistema por dia.

Os receptores principais dos dados de entrada - vide qualquer interação do usuário com componentes dinâmicos relevantes do sistema e com pedidos de feedback - são responsáveis por, quando houver uma nova entrada relevante, gerar um processo de decisão para avaliar se uma ação deverá ser tomada para identificar e modificar algum elemento na interface, seja aumentar a área clicável de um botão, diminuir a fonte, mudar o tema, entre outras. Independentemente da iteração ou ação, todos estes dados ficam armazenados no sistema de forma histórica enquanto o usuário estiver cadastrado neste sistema ou caso alguma configuração de tempo ou tamanho máximo para armazenamento destes dados limite este processo.

Naturalmente, é possível limitar a quantidade de entradas por usuário ou totais no banco de dados dependendo do caso de uso do sistema principal, em que a solução aqui descrita poderia oferecer limites para configuração por usuários administradores do sistema e vir a se comportar de forma mais independente em frente a interações e não aguardar um histórico maior destas para intervir com alguma ação do sistema. Estas configurações e definições que possam vir a ser interessantes para uma solução em produção foram estudadas também ao longo da implementação do software de testes.

É importante também ressaltar que estes dados tiveram seu armazenamento e comunicação desenvolvidos de duas maneiras para testes:

- Inicialmente tendo os dados captados de um Front-end e enviados por uma *RESTful API* que se comunica com o Back-end por payloads *JSON*, o Back-end, por sua vez, executa qualquer tratamento dos dados e parâmetros para atualizações e armazenamento do Banco de Dados, sendo capaz de agregação de dados, processamento de decisão descentralizado e maior persistência dos dados de usuários.
- E adicionalmente uma implementação mais simples utilizando apenas mecanismos no Front-end, utilizando-se de uma *local storage* para armazenamento dos dados e tendo as decisões de configurações e atualizações feitas por componentes internos especializados. Esta implementação, por ser mais simples, permite velocidades maiores na interação com o usuário e respostas praticamente instantâneas aos eventos e ajustes, sendo por isso utilizada nos testes de usuário, mas falha com a persistência e agregação de dados caso o usuário limpe os dados do seu navegador ou até mesmo troque para outro dispositivo ou navegador no processo.

### 4.3 Ambiente de Desenvolvimento

O desenvolvimento desta solução foi implementado em um computador com as seguintes especificações:

- Processador: Intel Core i7 6700HQ 2.60GHz
- Memória: 16Gb RAM
- Placa Gráfica: NVIDIA Quadro M1000M
- Armazenamento: 512Gb SSD
- Sistema Operacional: Windows 10

A linguagem de programação utilizada para o desenvolvimento foi principalmente JavaScript, com o framework ReactJS para os componentes de interface de usuário, incluindo também HTML e CSS, NodeJS e Express para processamento dos dados recebidos da interface e comunicação com o banco de dados, RESTful API para comunicação entre estes, e MongoDB Atlas (versão Cloud) para o banco de dados em si.

Como uma das possibilidades de produção desta solução é se tornar uma biblioteca independente ou até mesmo uma aplicação como serviço para implementação de interfaces de usuário, apenas bibliotecas básicas - como os frameworks citados anteriormente - foram utilizados.

A escolha destas tecnologias se dá em maior parte pela popularidade destes frameworks e a capacidade de integração com soluções já existentes e potencialmente futuras, possibilitando a criação de bibliotecas reutilizáveis ou uma aplicação *standalone* a ser integrada em sistemas mais complexos. As tecnologias de Front-end e banco de dados são as que mais possuem capacidade de impactar a integração com outras soluções, desde que estas utilizem também RESTful APIs para comunicação. O banco de dados escolhido tem base na naturalidade em que novas entradas recebidas como JSON ou documentos são inseridos como tabelas e documentos para armazenamento, facilitando a inserção de verificação de dados relevantes para o sistema, principalmente visto que as entradas serão como fichas criadas para cada usuário com informações e chaves simples por item, e o quanto essas informações puderam ser facilmente transformadas em objetos para armazenamento local no navegador.

#### 4.4 Metodologia

A implementação desta solução e da prova de conceito ocorreu por meio da metodologia Agile Scrum, com interações e testes constantes na implementação. As sprints tiveram duração de duas semanas com objetivos de alto nível para cada, sendo planejadas em mais detalhes e divididas em tarefas para o backlog no início de cada período e reavaliado o progresso ao final. A maior adaptação da técnica neste caso foi o trabalho individual, e não em grupo, consolidando os papéis de planejamento, revisão e execução em um único participante do processo e excluindo a necessidade de eventos como *daily stand-ups* e *Sprint Reviews*. O objetivo principal do uso da técnica foi a organização do processo e melhor rastreamento do desenvolvimento nas etapas traçadas para o projeto, divididas da seguinte forma:

1. Design inicial em alto nível da solução e implementação base dos principais componentes dinâmicos de interface, incluindo gerenciadores de estilos, temas, fontes e texto, botões, pop-ups para feedback, entre outros.
2. Desenvolvimento da opção de armazenamento mais simples em Front-end, utilizando a tecnologia de *local storage* para armazenamento dos dados de uso no próprio navegador do usuário, incluindo uma opção de análise e coleta de dados mais leve para a solução e possibilitando testes de viabilidade iniciais.
3. Implementação do Back-end para processamento dos dados enviados pela solução integrada na interface de usuário e tomada de decisões para ações e armazenamento de dados no banco, além da configuração do banco de dados em si.
4. Implementação do sistema de comunicação e API com os payloads idealizados e melhorias nos componentes básicos existentes, além do polimento da solução.
5. Implementação da aplicação de prova de conceito e testes na qual os componentes de interface dinâmicos são integrados, possibilitando a coleta de dados e posteriormente testes com usuários.
6. Estudo e criação de roteiro de testes de usuários a serem conduzidos para avaliação do valor e eficácia da solução, além da coleta de dados iniciais durante interações com a aplicação de teste para geração de dados, testes, validações e iterações na implementação.
7. Condução de testes de usuários com a solução da prova de conceito e aplicação de formulário para resposta sobre a experiência da interação.

8. Avaliação dos resultados obtidos nos testes de usuários e escrita dos resultados finais do experimento para demonstração e encerramento.

#### 4.5 Motivação

A motivação que deu origem às funcionalidades idealizadas inicialmente no sistema originam de diversas experiências negativas com interfaces de usuário acumuladas ao longo de anos de uso de aplicações e serviços. Um exemplo de uma dificuldade recorrente é durante a interação com uma aplicação de mais de 5 anos no mercado que enfrenta o mesmo problema desde seu lançamento, controles demasiadamente pequenos e que podem ser fechados ou ter outros efeitos indesejados se clicados na área próxima e não na área exata do controle, como pode ser visualizado na figura 4.7, prejudicando a experiência dos usuários da aplicação diariamente. Outra questão comum é a completa falta de diferentes temas na aplicação, como o tema *dark* ou a ausência de controles fáceis para a configuração e aplicação deste, sendo a interação com a interface em ambientes escuros ou à noite extremamente desconfortável.

Figura 4.7 – Telas da aplicação mobile AFK Arena contendo controles extremamente pequenos e de difícil interação



Fonte: Aplicação Mobile - AFK Arena

Estas principais dificuldades, encontradas repetidamente em interações diárias com diversas aplicações foram as principais fontes dos componentes *Reactive* para au-

mento adaptativo de botões em caso de erros repetidos e da requisição de feedback para escolha de diferentes temas e cores, além da organização da aplicação de testes com um padrão de design adequado e organização de uma página de configurações simples e compreensiva.

#### **4.6 Funcionalidades**

As funcionalidades do sistema envolvem os casos de uso citados anteriormente, e se baseiam principalmente na capacidade de observação, captura e armazenamento das interações de um usuário com a interface de uma aplicação e a tomada de ações para modificação de componentes da interface de acordo com os registros de interações recebidos.

Quanto a coleta de dados do usuário para posterior aplicação de mudanças, esta é dividida em duas principais categorias:

- Observação passiva de interações com componentes suportados na interface de usuário, capturando eventos relevantes como cliques e seu estado de sucesso, dentro de áreas úteis pré-definidas.
- Atuação dentro de períodos definidos de tempo com geração de requisições de feedback para o usuário, com captura e análise das respostas, podendo gerar ações de modificação da interface em tempo real de acordo com a opção escolhida, ou apenas sendo fechadas manualmente pelo usuário ou automaticamente caso ignoradas por um período pré-definido de tempo.

É importante salientar que no caso do banco de dados externo e gerenciado por um Back-end, a coleção de dados que será utilizada também terá o campo “user”. Logo, nos exemplos anteriores, caso não haja ainda uma entrada para o usuário “alexandre”, será criada uma nova entrada de perfil para este usuário em específico, e qualquer futura query para criação de ações específicas para um usuário naturalmente se limitará a queries pelo perfil de dados deste. O caso do armazenamento local do Front-end não conta com esta informação de usuário, por ser mais simples, e esperar a interação com um usuário único, sem visar uma futura agregação de dados de vários usuários, por exemplo.

Já o sistema de atuação desta solução é mais complexo, e não conta apenas com escritas de dados mas também com análises e ações sobre estas informações, podendo então gerar saídas na aplicação como aumento ou diminuição do tamanho de fontes e áreas clicáveis de botões e atualizações de temas ou opções de acessibilidade disponíveis,

e atua da seguinte maneira:

- Criação de ações na aplicação com objetivo de modificação de componentes suportados na interface de usuário, sendo ativado a cada resposta para solicitação de feedback com requisição de alteração de configuração, como mudança de tema ou aumento de fontes e área de botões. Estas ações também podem ser criadas de acordo com observações das interações do usuário, como cliques na área próxima do botão repetidamente antes de um clique correto, entre outros, podendo então aumentar ou diminuir a área de contato destes controles suportados. Para isso, são necessárias também as seguintes funcionalidades:
  - Criação de uma nova ação na base de dados
  - Acesso a todas as respostas relevantes a feedbacks fornecidas pelo usuário
  - Acesso a todos as interações de um usuário com o sistema que sejam relacionadas ao evento de clique de um botão específico para criação de uma ação de aumento de área daquele botão

## 5 ANÁLISE DA SOLUÇÃO PROPOSTA

Para análise e avaliação do trabalho, foram criados uma aplicação de teste interativa e um formulário online, fornecendo aos usuários uma breve interação com os componentes desenvolvidos para a prova de conceito e uma avaliação no formato padronizado do *System Usability Scale* (SUS) (BROOKE, 1995) para que os usuários pudessem dar seu feedback em relação a experiência com as funcionalidades apresentadas.

A aplicação de teste e o formulário, com as instruções necessárias contidas em ambos, foram criados, disponibilizados de forma online e compartilhados em redes sociais pessoais e grupos internos de trabalho ao longo de uma semana para interação e preenchimento das respostas. Ao final do período de testes, foram adquiridas no total sete respostas ao questionário, que serão agregadas, avaliadas e apresentadas nas próximas seções.

O perfil geral dos usuários convocados a responder o questionário foi dos 25 aos 34 anos, com variados níveis de experiência em interação com aplicações digitais e pelo menos ensino médio completo, a maioria com ensino superior completo ou em andamento, adicionalmente. A maioria dos usuários também é familiarizado com a interação com computadores.

Para o teste, os usuários eram inicialmente levados por meio de um link para a aplicação de teste principal, que em sua página inicial (que pode ser vista na figura 4.2) continha instruções gerais para que este interagisse com o jogo interativo (em foco na figura 4.6) por alguns minutos (tempo no qual poderia vir a completá-lo) e então poderia seguir para responder o questionário cujo link também havia sido disponibilizado.

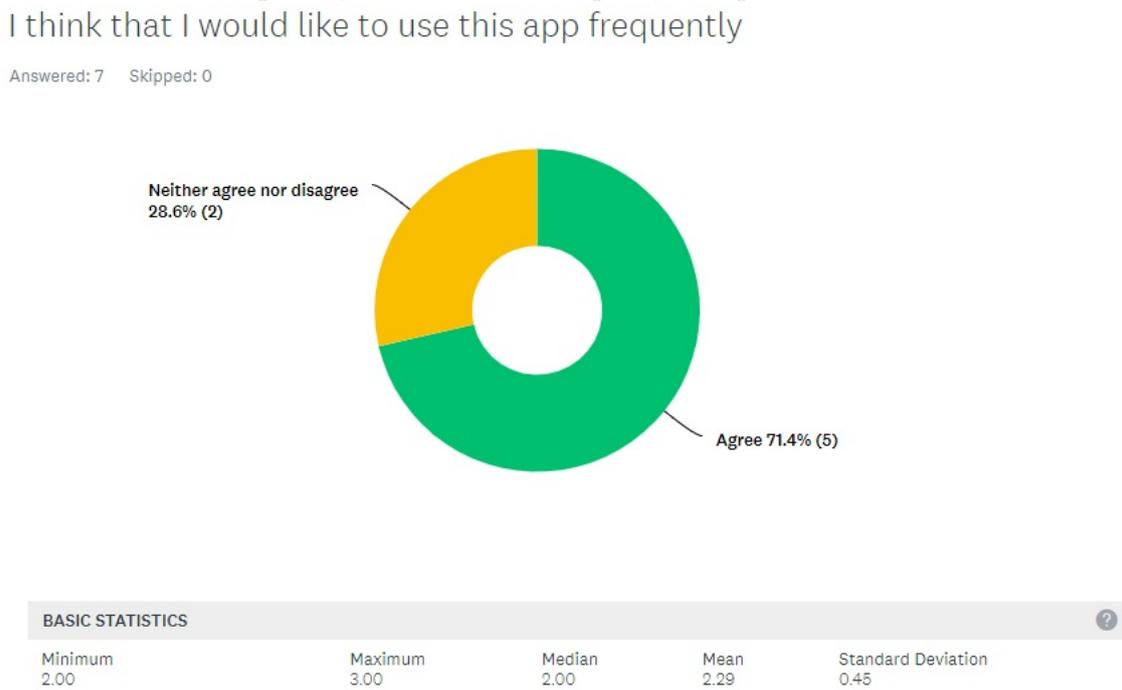
Além do questionário, foram conduzidos em menor número alguns testes de usuário mais iterativos com a aplicação de teste, em que o condutor do teste interagiu com o usuário a fim de coletar informações mais intrínsecas do sistema e obter insights mais subjetivos quanto a interação. Os principais aprendizados adquiridos durante estes testes são descritos como comentários ao longo da análise de resultados do questionário na próxima seção.

### 5.1 Resultados do Questionário

Analisaremos individualmente os resultados agregados de cada uma das dez perguntas do questionário no formato do SUS nas próximas seções e posteriormente con-

siderações finais sobre estes dados. As perguntas seguem linhas gerais de facilidade de uso, integração e consistência do sistema, visando avaliar se um usuário com poucas instruções, dicas ou experiência guiada de qualquer forma, conseguiria interagir de forma satisfatória e natural tanto com os componentes principais desenvolvidos, customizando a interface de acordo com suas preferências, quanto com a aplicação interativa de teste.

Figura 5.1 – Resultado agregado das respostas à primeira pergunta do questionário: "Acredito que eu gostaria de usar esta aplicação frequentemente"



Fonte: O Autor

A primeira pergunta do questionário, "Acredito que eu gostaria de usar esta aplicação frequentemente", cujos resultados agregados podem ser vistos na figura 5.1, teve um retorno majoritariamente positivo, mas não expressivo. Visto que a aplicação de teste era apenas um jogo simples e não utilitária de qualquer forma, ou seja, naturalmente uma aplicação que não seria utilizada frequentemente ou diariamente, é esperado que os usuários acabem por não concordar completamente.

Quando questionados sobre os pop-ups de feedback em específico, vários comentaram a frequência grande de perguntas, cuja configuração para este teste estava com um intervalo de 40 segundos. Devido a natureza rápida do teste de poucos minutos e as cinco diferentes possibilidades de customização, um intervalo menor do que o indicado teve de ser implantado. Para aplicações de maior porte, configurações com intervalos maiores - de até mesmo dias - seriam recomendados, para que os usuários não acabem por se sentir sobrecarregados com interações.

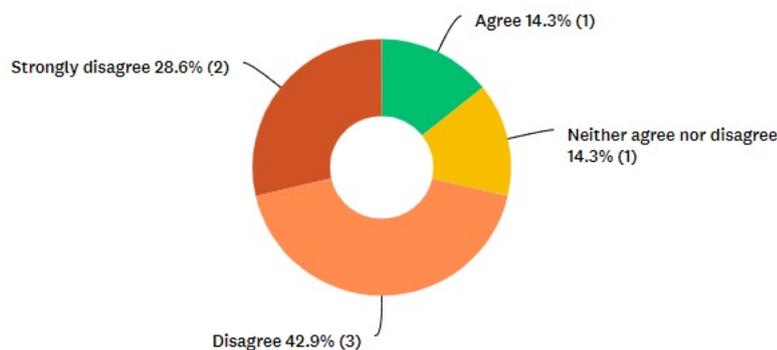
A maioria dos usuários achou muito interessante as configurações disponíveis, e também interagiu com várias das opções, explicando que esta porção de customização da aplicação seria bem-vinda em outras soluções que utiliza mais frequentemente, e que apenas o jogo em si realmente não seria algo que jogaria todo dia, como esperado.

Aplicar os componentes adaptativos a uma aplicação mais comum, como um protótipo de rede social simples ou ferramenta de chat, mais comuns no dia-a-dia dos usuários de serviços digitais, ou trazer uma pergunta um pouco diferente focada na funcionalidade em si da customização ativa poderia trazer resultados mais positivos em um novo teste futuro.

Figura 5.2 – Resultado agregado das respostas à segunda pergunta do questionário: "Achei o sistema desnecessariamente complexo"

I found the system unnecessarily complex

Answered: 7 Skipped: 0



BASIC STATISTICS				
Minimum	Maximum	Median	Mean	Standard Deviation
2.00	5.00	4.00	3.86	0.99

Fonte: O Autor

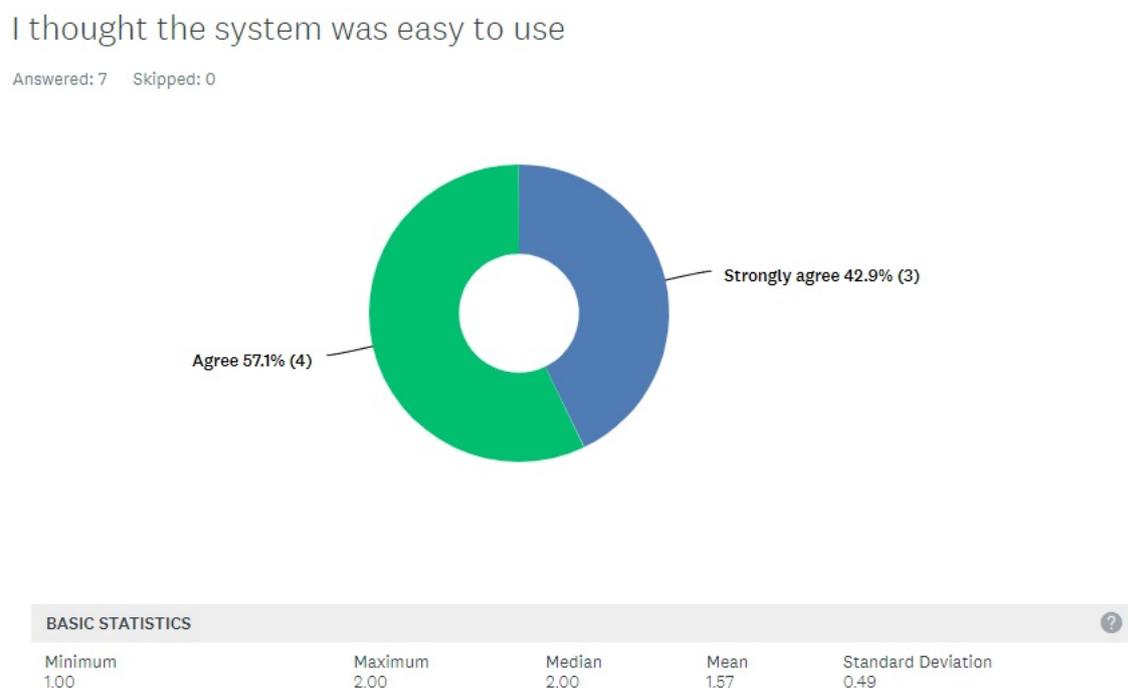
Já a segunda pergunta do questionário, "Achei o sistema desnecessariamente complexo", com os resultados agregados visíveis na figura 5.2, trazem uma das maiores variâncias entre os resultados de todo o questionário, apesar da maioria e média ainda serem favoráveis a aplicação. A origem desta variância, após algumas consultas aos respondentes, se devia em grande parte a quantidade de opções de customização em comparação com a simplicidade da aplicação de testes, sendo comum o retorno de que uma aplicação tão simples não necessitaria de tantas configurações.

A maioria dos usuários focou sua atenção nas mudanças de tema e cores da aplicação, visto que o tema *light* era o padrão e atualmente o tema *dark* é preferido pela maioria

dos usuários, enquanto que a mudança da cor da aplicação acarretava também na mudança de algumas artes do jogo, trazendo personagens diferentes e levando a mais variações.

As alterações de tamanho de botão e fonte foram indicadas como as mais excessivas, e potencialmente as maiores razões que colaboraram para as respostas tendendo a complexidade adicional do sistema. A aplicação foi desenvolvida já com padrões de design adequados em consideração, então é esperado que a maior parte dos usuários estivesse confortável com os padrões estabelecidos. Uma amostra de usuários com algum grau de deficiência visual ou idade avançada poderia demonstrar que estas configurações mais específicas poderiam vir a ser mais benéficas.

Figura 5.3 – Resultado agregado das respostas à terceira pergunta do questionário: "Acredito que o sistema foi fácil de usar"



Fonte: O Autor

Quanto a terceira pergunta do questionário, "Acredito que o sistema foi fácil de usar", cujos resultados agregados podem ser vistos na figura 5.3, as respostas foram majoritariamente positivas, dada a natureza do estudo e aplicação de técnicas predominantemente visando a facilidade de uso, tanto nos componentes da prova de conceito quanto na aplicação interativa de teste, são um dos principais indicadores do benefício que padrões de design aliados a adaptatividade e customização podem em conjunto melhorar a experiência de usuário.

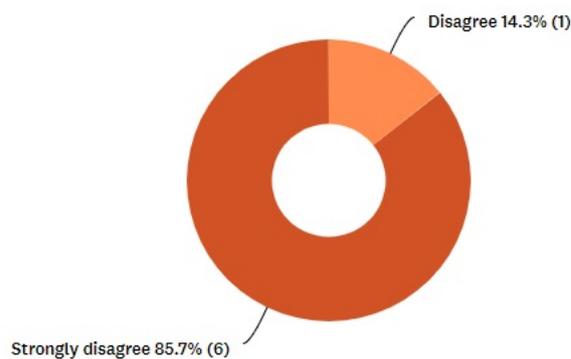
Quando abordados sobre esta questão, a maioria dos usuários não apontou nenhuma dificuldade específica na utilização do sistema, e os que trouxeram algum aspecto,

citaram apenas algum bug visual localizado ou que os controles iniciais do jogo em si poderiam ser melhor explicados por meio de um tutorial para que fosse mais evidente a progressão de tarefas, mas que após interagir com os controles disponíveis o objetivo ficou claro.

De acordo com este retorno, acredita-se então que os principais motivos para a pequena hesitação ao não escolher a opção de concordância mais alta são estes problemas mais atrelados a jogabilidade da aplicação de teste em si do que do uso dos componentes *Reactive Feedback* e *Button* implementados, que não foram trazidos como impedimentos ou dificultadores de qualquer forma.

Figura 5.4 – Resultado agregado das respostas à quarta pergunta do questionário: "Acredito que precisaria do suporte de uma pessoa técnica para ser capaz de usar este sistema"  
I think that I would need the support of a technical person to be able to use this system

Answered: 7 Skipped: 0



BASIC STATISTICS				
Minimum	Maximum	Median	Mean	Standard Deviation
4.00	5.00	5.00	4.86	0.35

Fonte: O Autor

A quarta pergunta do questionário, "Acredito que precisaria do suporte de uma pessoa técnica para ser capaz de usar este sistema", para a qual os resultados agregados são visíveis na figura 5.4, tem o resultado mais fortemente decisivo do questionário, refletindo também a facilidade de uso do sistema como um todo e a simplicidade da aplicação de teste.

Praticamente nenhum comentário adicional foi trazido quanto a esta questão, os usuários reforçaram que o sistema era fácil de usar, como trazido na questão 3, e apenas um usuário trouxe a questão de que imaginava que talvez pessoas idosas ou não familiares

com jogos digitais poderiam vir a ter dificuldade utilizando o sistema, o que é relevante.

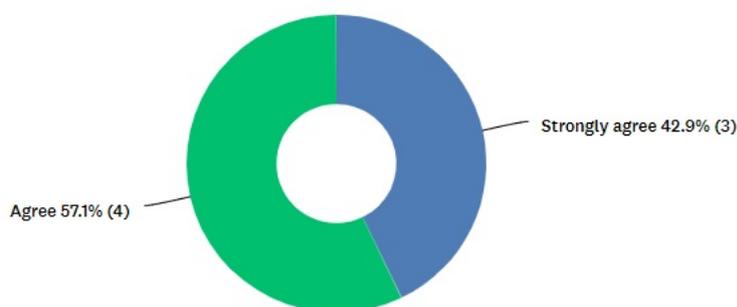
A aplicação de teste foi desenvolvida com base na jogabilidade de um "mini-game" muito famoso dos anos 90, conhecido popularmente como Tamagotchi, em que o jogador treinava um pet, comumente de origem mítica, do seu estágio de ovo até a idade avançada, alimentando-o, fazendo-o descansar, entre outros. Devido a faixa etária comum dos respondentes deste questionário, vários reconheceram o estilo da aplicação de teste, o que pode ter influenciado também a identificação das tarefas, jogabilidade e objetivo do jogo digital. Usuários com idade mais avançada e, portanto, menos familiaridade com este estilo específico de aplicação mais comum a uma geração poderiam certamente vir a ter mais dificuldade de associar as tarefas disponibilizadas a um cenário reconhecido.

Como trazido anteriormente, um teste futuro com faixas etárias mais avançadas e usuários mais leigos em conhecimentos computacionais e do campo da aplicação de teste podem trazer resultados mais distribuídos a este estudo.

Figura 5.5 – Resultado agregado das respostas à quinta pergunta do questionário: "Achei que as várias funções deste sistema foram bem integradas"

I found the various functions in this system were well integrated

Answered: 7 Skipped: 0



BASIC STATISTICS				
Minimum	Maximum	Median	Mean	Standard Deviation
1.00	2.00	2.00	1.57	0.49

Fonte: O Autor

Seguindo para a quinta pergunta do questionário, "Achei que as várias funções deste sistema foram bem integradas", com os resultados agregados visíveis na figura 5.5, podemos observar novamente respostas predominantemente positivas e refletindo praticamente a mesma proporção da pergunta 3, sobre a facilidade de uso do sistema.

Nesta questão, foram trazidos novamente alguns pontos sobre a frequência dos

pop-ups de requisição de feedback e a pequena dificuldade inicial ao interagir com a aplicação de teste, com a ausência de tutoriais ou explicações mais significativas sobre o sistema.

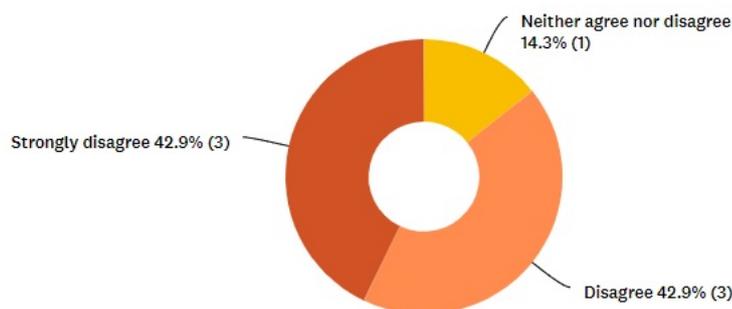
Alguns usuários também apontaram, por vezes, a falta de conexão entre uma requisição de feedback e a tarefa que estavam executando, utilizando o exemplo de estarem jogando e terem sido questionados sobre o tamanho da fonte mais de uma vez. Avaliando as opções escolhidas, o usuário havia interagido e modificado o sistema para as outras opções, como tema, linguagem e cor, e já definido-as, então os únicos feedbacks restantes eram sobre os tamanhos de fonte e botões da aplicação.

Uma melhoria futura para o sistema de geração de requisições de feedback, além da trazida anteriormente sobre o aumento do período entre as requisições, seria utilizar os eventos de feedbacks já instanciados e não repetir os mesmos questionamentos já trazidos por um período mais longo de tempo, como por exemplo dias ou semanas, e instituir também um máximo de requisições, como 3, até que o sistema pare completamente de questionar sobre esta configuração, visto que alguns usuários podem vir a não interagir com o pop-up e sinalizar a opção de "Não mostrar novamente".

Figura 5.6 – Resultado agregado das respostas à sexta pergunta do questionário: "Acredito que havia muita inconsistência neste sistema"

I thought there was too much inconsistency in this system

Answered: 7 Skipped: 0



BASIC STATISTICS				
Minimum	Maximum	Median	Mean	Standard Deviation
3.00	5.00	4.00	4.29	0.70

Fonte: O Autor

Quanto a sexta pergunta do questionário, "Acredito que havia muita inconsistência neste sistema", cujos resultados agregados são visíveis na figura 5.6, apesar de uma par-

cela significativa dos usuários discordar fortemente do ponto trazido, ainda há indecisão ou falta de confiança completa quanto ao tópico.

Um dos principais motivos trazidos pelos respondentes que ficaram um pouco confusos foram as diferentes formas que controles e informações tomaram na aplicação de teste ao longo do uso. Enquanto algumas informações, ou textos, eram trazidos de forma solta, algumas possuíam bordas, o que acabava por confundir o usuário com botões secundários, também compostos de texto e borda, apesar de o texto no segundo caso seguir a cor primária da aplicação enquanto no primeiro este seguia apenas as cores do tema, utilizando cores claras ou escuras dependendo do plano de fundo da aplicação.

O respondente mais experiente no campo de design visual, que trabalha especificamente com esta área, sugeriu uma distinção mais clara entre elementos que podem ter interação pelo usuário e os que são apenas informativos, retirando as bordas no caso do último. Além disso, a presença de ícones, além de texto, em alguns botões principais acabou também por tirar o foco de outros botões sem ícones, causando uma diferença de importância subjetiva não intencional entre as tarefas disponíveis.

Portanto, os comentários foram majoritariamente direcionados ao padrão de design aplicado de forma geral, e praticamente inexistente em relação aos componentes *Reactive*, que, acredita-se, por terem estado mais em foco durante o desenvolvimento, acabaram por ter seus designs mais bem representados quando comparados com a aplicação como um todo, fazendo com que elementos mais secundários como os citados como potenciais problemas viessem a provocar mais o foco do usuário.

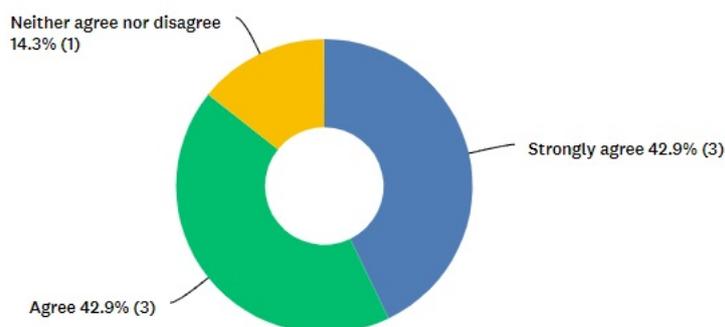
A sétima pergunta do questionário, "Eu imaginaria que a maioria das pessoas aprenderia a usar este sistema muito rapidamente", para a qual os resultados agregados podem ser vistos na figura 5.7, teve respostas com uma divisão muito similar a questão 6 anterior, expondo também uma falta de confiança completa dos usuários ao demonstrar a concordância com o tópico.

Além de reforçar os pontos trazidos anteriormente, como a potencial confusão no design entre componentes específicos de visualização e interação, os usuários também ressaltaram novamente a questão da curva inicial de aprendizado que poderia vir a ser facilitada com um direcionamento mais explicativo ou guiado nos primeiros momentos da aplicação iterativa de teste.

Em geral, no entanto, praticamente todos os respondentes reforçaram que acreditam que os usuários do sistema provavelmente aprenderiam de forma célere a lidar com as tarefas apresentadas, mas que estes pontos em específico poderiam apresentar-se como

Figura 5.7 – Resultado agregado das respostas à sétima pergunta do questionário: "Eu imaginaria que a maioria das pessoas aprenderia a usar este sistema muito rapidamente"  
I would imagine that most people would learn to use this system very quickly

Answered: 7 Skipped: 0



BASIC STATISTICS				
Minimum	Maximum	Median	Mean	Standard Deviation
1.00	3.00	2.00	1.71	0.70

Fonte: O Autor

empecilho para alguns usuários mais leigos, e que a resolução destes pontos poderia trazer uma melhora para o entendimento geral da aplicação.

Já a oitava pergunta do questionário, "Achei o sistema muito complicado de usar", cujos resultados agregados podem ser verificados na figura 5.8, traz um paralelo interessante com os resultados da pergunta 3, quanto a facilidade de uso do sistema, em que os respondentes aparentam um pouco mais de relutância a discordar ou discordar fortemente quanto ao quão complicado o sistema foi de usar do que a concordar ou concordar fortemente que o sistema foi fácil de usar.

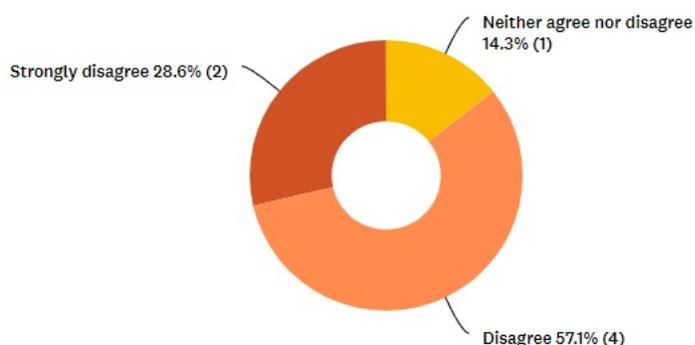
Nesta pergunta, um novo tópico foi trazido por alguns usuários, focando na repetição de tarefas para conclusão do objetivo da aplicação, e o quanto esta repetição poderia se tornar um pouco complicada ou pelo menos monótona após um certo limiar, e que a progressão, no quesito de identificação de um final para o jogo, poderia ter ficado mais clara. A progressão de níveis e estágios do personagem em si estava satisfatória, mas em que ponto esta progressão chegaria ao fim que causava, de acordo com alguns usuários, uma incerteza ou ansiedade quanto ao final da interação.

Além deste novo ponto abordado, a falta de clareza de alguns controles e visualizações foi reforçado pelos usuários.

Figura 5.8 – Resultado agregado das respostas à oitava pergunta do questionário: "Achei o sistema muito complicado de usar"

I found the system very cumbersome to use

Answered: 7 Skipped: 0



BASIC STATISTICS				
Minimum	Maximum	Median	Mean	Standard Deviation
3.00	5.00	4.00	4.14	0.64

Fonte: O Autor

A nona pergunta do questionário, "Me senti muito confiante usando o sistema", com os resultados agregados visíveis na figura 5.9, por sua vez, traz um resultado comparável a questão 2, que aborda a complexidade do sistema, e assim como a comparação da questão 8 com a 3, um pouco mais de concordância dos usuários com a forma assertiva do que a discordância com o contrário.

Quando abordados sobre esta questão em específico, os usuários apenas trouxeram novamente alguns dos pontos principais que causaram confusão na interação com a aplicação, como as diferenças entre controles e visualizações e a interação inicial com o jogo. Reforçaram que se sentiriam mais confiantes com a presença de um indicador ou guia para as tarefas iniciais da aplicação e uma melhor demonstração do progresso em relação ao total da experiência.

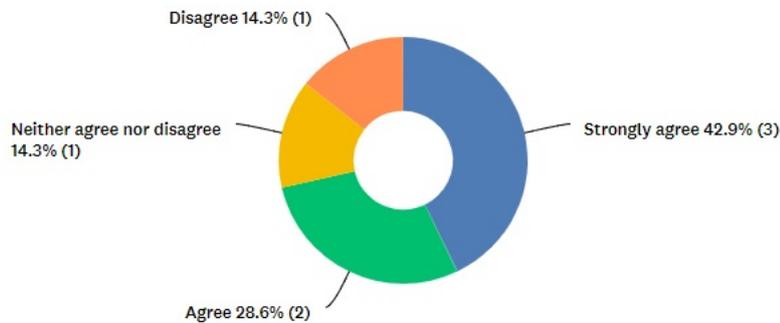
E por fim, a décima e última pergunta do questionário, "Precisei aprender várias coisas antes que pudesse seguir em frente com este sistema", para a qual os resultados agregados são visíveis na figura 5.10, traz um prospecto mais positivo em relação a complexidade da aplicação do que a questão 2 e 9, em que os respondentes mais fortemente discordam com a necessidade do aprendizado de novas técnicas do que nos questionamentos anteriores.

Neste caso foi trazida a atenção um pouco mais para os componentes da prova

Figura 5.9 – Resultado agregado das respostas à nona pergunta do questionário: "Me senti muito confiante usando o sistema"

I felt very confident using the system

Answered: 7 Skipped: 0



BASIC STATISTICS				
Minimum	Maximum	Median	Mean	Standard Deviation
1.00	4.00	2.00	2.00	1.07

Fonte: O Autor

de conceito, como esperado. Um usuário questionou qual era a lógica para que os botões tivessem seus tamanhos ajustados automaticamente, e que talvez uma regra explícita em alguma parte da aplicação poderia vir a ser interessante. Outro usuário comentou a aplicação instantânea de mudanças do controle de feedback, o quanto isso era diferente e interessante de interagir com, mas que, naturalmente, era um novo tipo de interação a ser considerado em uma aplicação.

A existência de diversas opções para configurações dos componentes visuais da aplicação também foi abordado, de forma positiva.

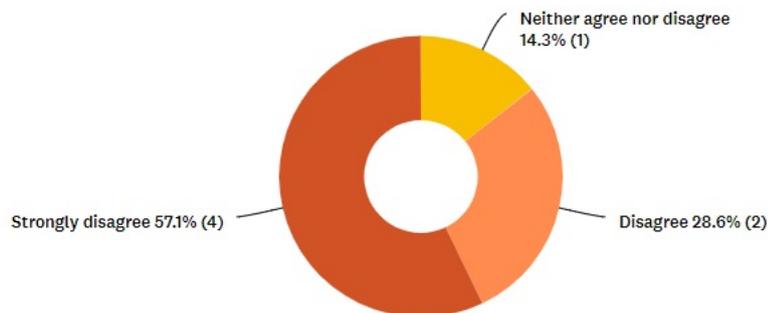
## 5.2 Síntese dos Testes Aplicados

Analisando os resultados do questionário de forma geral, a recepção da técnica aplicada no estudo foi predominantemente positiva. Identificou-se que o maior atrito reportado pelos usuários vinha de componentes e interações fixos e de menor relevância para o teste, e não dos componentes *Reactive* em foco, que quando trazidos em questionamentos mais específicos, eram considerados de fácil uso e utilidade interessante.

O resultado pelo sistema SUS e os critérios de pontuação aplicados culminou no total de 93,6 pontos percentuais agregando todas as questões e respostas obtidas e

Figura 5.10 – Resultado agregado das respostas à décima pergunta do questionário: "Precisei aprender várias coisas antes que pudesse seguir em frente com este sistema"  
I needed to learn a lot of things before I could get going with this system

Answered: 7 Skipped: 0



BASIC STATISTICS				
Minimum	Maximum	Median	Mean	Standard Deviation
3.00	5.00	5.00	4.43	0.73

Fonte: O Autor

extraindo a média de concordância destas, buscando-se então um valor acima da média de 68. A avaliação do teste então pelo questionário no modelo SUS pode ser considerado um sólido sucesso, com uma recepção extremamente positiva dos respondentes.

Os usuários também apontaram os potenciais da solução para um enfoque em questões de acessibilidade, e como usuários com dificuldades visuais ou motoras poderiam se beneficiar das alterações propostas pelos controles disponibilizados.

Foram também sugeridas algumas melhorias ao sistema pelos usuários - além das resoluções de bugs ou alterações de design citadas anteriormente - como a customização (e acréscimo) do intervalo entre as requisições de feedback e a inclusão de informações sobre o sistema de decisões e mais opções tanto na página de configuração quanto aos nas requisições instanciadas, incluindo por exemplo a transparência quanto a regras para comportamentos automáticos ou "inteligentes" do sistema e novas regras como a capacidade de habilitar e desabilitar por partes ou totalmente estas ações.

## 6 CONCLUSÃO

Este trabalho teve como objetivo avaliar a viabilidade e a aceitação de um sistema de interface inovador com elementos dinâmicos e adaptativos, colocando a técnica e sua implementação à prova, evidenciando um contraste com sistemas com os quais interagimos diariamente. Foi possível, com a implementação de vários destes componentes idealizados e a lógica e interação com dados necessários para executar cada funcionalidade destes, identificar então que a solução além de factível pode causar um impacto positivo, causando uma melhoria na experiência para usuários, visto as capacidades que foram atingidas e os resultados nos testes de usuário e questionário SUS.

A solução final implementada incluiu os componentes *Reactive* de interação com botões e a comunicação de eventos que o usuário conseguiu ou não clicar com sucesso na área do botão, as requisições de feedback e as interações com este, capaz de identificar tanto se uma resposta é dada - e qual a escolha - quanto se o componente foi ignorado ou fechado, além da aplicação em tempo real das configurações, com testes feitos para escolha de temas, cores, idioma, e tamanho de fontes e botões. O sistema também é capaz então de não apenas armazenar estas interações e feedbacks mas também tomar ações para aplicar as sugestões ou identificar as mudanças que precisam ser feitas em tempo real, como aplicar uma área clicável maior nos botões da interface, trocar o tema escolhido pelo usuário ou aumentar e diminuir a fonte de acordo com o feedback provido.

O principal desafio, que também acabou se provando um dos maiores aprendizados, foi a implementação de todo o escopo de trabalho no stack de tecnologias escolhido e nas diferentes formas idealizadas, por todas as diferentes áreas e tecnologias que foram utilizadas. Considerando todos os campos e diferentes técnicas para interação com elementos visuais no Front-end, leitura e armazenamento de dados no Back-end, a padronização dos dados e payloads para a API e a administração dos dados em si no banco de dados e no próprio navegador, lidando com todos os diferentes aspectos e áreas de uma aplicação, o que hoje é muito comum denominarmos de uma posição "Full-stack".

Os próximos passos visualizados são, além da possibilidade de implementação de mais componentes dinâmicos e adaptativos, como tabelas, modais e menus, baseados neste trabalho inicial, são o estudo do seu lançamento como uma biblioteca ou framework colaborativo e escalável, a extensão do impacto positivo na experiência de usuários com necessidades especiais e idades distintas e a aplicação e integração destes com diferentes padrões de aplicações com tarefas e cenários de teste diversos para a avaliação da difusão

que esta técnica pode vir a ter se aplicada de forma adequada e otimizada em soluções de grande porte.

## REFERÊNCIAS

BENYON, D. Accommodating individual differences through an adaptive user interface. **Human Factors in Information Technology**, Citeseer, v. 10, p. 149–149, 1993.

BOOTSTRAP · The most popular HTML, CSS, and JS library in the world. 2022. Available from Internet: <<https://getbootstrap.com/>>.

BOUZIT, S. et al. A design space for engineering graphical adaptive menus. In: . [S.l.: s.n.], 2016. p. 239–244.

BROOKE, J. Sus: A quick and dirty usability scale. **Usability Eval. Ind.**, v. 189, 11 1995.

CHAKRA UI: Simple, Modular and Accessible UI Components for your React Applications. 2022. Available from Internet: <<https://chakra-ui.com/>>.

DAREJEH, A.; SINGH, D. A review on user interface design principles to increase software usability for users with less computer literacy. **Journal of computer science**, Science Publications, v. 9, n. 11, p. 1443, 2013.

DEHLINGER, J.; DIXON, J. Mobile application software engineering: Challenges and research directions. In: **Workshop on mobile software engineering**. [S.l.: s.n.], 2011. v. 2, p. 29–32.

ELEMENT - A Desktop UI Toolkit for Web. 2022. Available from Internet: <<https://element.eleme.io/#/en-US>>.

LEFF, A.; RAYFIELD, J. T. Web-application development using the model/view/controller design pattern. In: IEEE. **Proceedings fifth ieee international enterprise distributed object computing conference**. [S.l.], 2001. p. 118–127.

MATERIAL Design. 2022. Available from Internet: <<https://material.io/design>>.

RATHNAYAKE, N. et al. A framework for adaptive user interface generation based on user behavioural patterns. In: **2019 Moratuwa Engineering Research Conference (MERCOn)**. [S.l.: s.n.], 2019. p. 698–703.

SPURLOCK, J. **Bootstrap: responsive web development**. [S.l.]: "O'Reilly Media, Inc.", 2013.

ZIMMERMANN, G. et al. Patterns for user interface adaptations. In: ZHOU, J.; SALVENDY, G. (Ed.). **Human Aspects of IT for the Aged Population. Design for Aging**. Cham: Springer International Publishing, 2015. p. 426–436. ISBN 978-3-319-20892-3.