



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

CAIO VICTOR DO AMARAL CUNHA SARMENTO

**PLACY: UMA SOLUÇÃO UI/UX E FRONT-END PARA A DIVULGAÇÃO DE
INFORMAÇÕES EM ESTABELECIMENTOS COMERCIAIS**

JOÃO PESSOA
2023

Caio Victor do Amaral Cunha Sarmento

**PLACY: UMA SOLUÇÃO UI/UX E FRONT-END PARA A DIVULGAÇÃO DE
INFORMAÇÕES EM ESTABELECIMENTOS COMERCIAIS**

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Engenharia de Computação da Universidade Federal da Paraíba como requisito complementar para obtenção do título de Bacharel em Engenharia de Computação, sob orientação do professor Sr. Derzu Omaia.

João Pessoa
2023

**Catalogação na publicação
Seção de Catalogação e Classificação**

S246p Sarmento, Caio Victor do Amaral Cunha.
Placy: uma solução UI/UX e Front-end para a divulgação de informações em estabelecimentos comerciais / Caio Victor do Amaral Cunha Sarmento. - João Pessoa, 2023.
45 f. : il.

Orientação: Derzu Omaia.
TCC (Graduação) - UFPB/CI.

1. UI. 2. UX. 3. Prototipação. 4. Front-end. 5. Estabelecimentos comerciais. I. Omaia, Derzu. II. Título.

UFPB/CI

CDU 004.4

Caio Victor do Amaral Cunha Sarmento

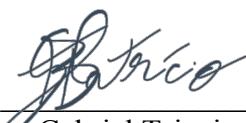
PLACY: UMA SOLUÇÃO UI/UX E FRONT-END PARA A DIVULGAÇÃO DE
INFORMAÇÕES EM ESTABELECIMENTOS COMERCIAIS

Trabalho de conclusão de curso submetido à Banca Examinadora designada pelo Curso de Graduação em Engenharia de Computação da Universidade Federal da Paraíba como requisito para obtenção do grau de Bacharel em Engenharia de Computação.

BANCA EXAMINADORA

Assinatura: 
Prof. Derzu Omaia
(Orientador)

Assinatura: 
Prof. Danielle Rousy Dias Ricarte

Assinatura: 
Gabriel Teixeira Patrício

João Pessoa, 16 de junho de 2023.

Dedico este trabalho ao meu futuro profissional, utilizando e aprimorando os conhecimentos, práticas e ferramentas aqui adquiridas.

AGRADECIMENTOS

Gostaria de expressar minha gratidão ao meu professor e orientador Derzu Omaia, por todo o suporte e compromisso, somado com sugestões e feedbacks construtivos que foram essenciais durante o desenvolvimento deste trabalho.

Agradeço à minha namorada, que sempre esteve me incentivando e apoiando emocionalmente, propondo-se a ajudar no que fosse preciso desde o início.

Sou grato à minha família, que sempre foi compreensiva e apoiadora, buscando entender sobre o tema abordado e desejando muito sucesso.

Também gostaria de agradecer a todos os meus colegas, de curso e de trabalho, que me ajudaram com técnicas, ferramentas, dicas e testes que contribuíram desde o início da construção desta aplicação.

Novamente, expresso meu sincero agradecimento a todos que contribuíram para o sucesso deste trabalho e para a minha formação acadêmica. Sou verdadeiramente grato por sua dedicação e por acreditarem no meu potencial.

“O design não é apenas o que parece e o que se sente. O design é como funciona”.

Steve Jobs

RESUMO

Com o crescente avanço tecnológico, muitos estabelecimentos precisam se adaptar para atrair e fidelizar ainda mais seus clientes. Para isso, este projeto discorre sobre a implementação de uma aplicação para dispositivos móveis, IOS e Android denominada Placy, com objetivo de possibilitar aos estabelecimentos uma solução para divulgação de informações e serviços aos clientes, como a lotação atual, lista de produtos disponíveis e programas de fidelidade. O foco central do trabalho e desenvolvimento da aplicação é abordar e utilizar conceitos de UI/UX como: usabilidade, acessibilidade e responsividade, com práticas e tecnologias eficientes e mais atualizadas do mercado front-end como: frameworks, componentização e otimização.

Palavras-chave: usabilidade, acessibilidade, responsividade, *UX design*, *UI design*, prototipação, design, front-end, framework, componentização, otimização, React, React Native, Figma.

ABSTRACT

With the growing technological advancement, many establishments need to adapt to attract and further loyal their customers. To this end, this project discusses the implementation of an application for mobile devices, IOS and Android called Placy, with the purpose of enabling establishments for a solution to disseminate information and services to customers, such as current capacity, available product list and loyalty programs. The central focus of this work and application development is to address and use UI/UX concepts such as: usability, accessibility and responsiveness, with efficient and more up-to-date practices and technologies in the front-end market such as: frameworks, componentization and optimization.

Keywords: usability, accessibility, responsiveness, *UX design*, *UI design*, prototype, design, front-end, framework, componentization, optimization, React, React Native, Figma.

LISTA DE FIGURAS

Figura 1 - Verificador de contraste seguindo as normas do WCAG.....	11
Figura 2 - Biblioteca de cores do Design System.....	12
Figura 3 - Logo do Placy.....	13
Figura 4 - Ícones do Design System.....	14
Figura 5a - Componentes do Design System.....	15
Figura 5b - Componentes do Design System.....	15
Figura 6 - Telas do design da aplicação no modo claro.....	16
Figura 7- Telas do design da aplicação no modo escuro.....	17
Figura 8 - Rede de interações entre telas da aplicação.....	17
Figura 9 - Fluxo de check-in no estabelecimento.....	19
Figura 10 - Fluxo de realização de pedidos após check-in.....	20
Figura 11a - Fluxo de reservas.....	21
Figura 11b - Fluxo de reservas.....	21
Figura 12 - Níveis de confiabilidade do Maze.co.....	25
Figura 13 - Avaliação de legibilidade dos textos no Maze.co.....	27
Figura 14 - Avaliação de navegação da aplicação no Maze.co.....	28
Figura 15 - Avaliação de satisfação geral da aplicação no Maze.co.....	28
Figura 16 - Comentários dos usuários sobre a aplicação no Maze.co.....	29

LISTA DE ABREVIATURAS

CSS	Cascading Style Sheets
DOM	Documents Object Model
HTML	HyperText Markup Language
IDE	Integrated Development Environment
JS	Javascript
UI	User Interface
UX	User Experience
W3C	World Wide Web Consortium
WAI	Web Accessibility Initiative
WCAG	Web Content Accessibility Guidelines

SUMÁRIO

1. INTRODUÇÃO.....	1
1.1 MOTIVAÇÃO.....	1
1.2 OBJETIVO GERAL.....	2
1.3 OBJETIVOS ESPECÍFICOS.....	2
1.4 TRABALHOS RELACIONADOS.....	2
2. REFERENCIAL TEÓRICO.....	4
2.1 USABILIDADE E UX.....	4
2.1.1 MÉTRICAS DE UX.....	4
2.2 ACESSIBILIDADE.....	5
2.2.1 MÉTRICAS DE ACESSIBILIDADE.....	6
2.3 RESPONSIVIDADE.....	6
2.4 DESIGN E PROTOTIPAÇÃO.....	6
2.4.1 FIGMA.....	6
2.4.2 DESIGN SYSTEM.....	7
2.4.3 PROTOTIPAÇÃO.....	7
2.5 FRAMEWORKS E TECNOLOGIAS FRONT-END.....	7
2.5.1 REACT JS.....	7
2.5.2 TYPESCRIPT.....	8
2.5.3 REACT NATIVE.....	8
2.5.4 NATIVEBASE.....	8
2.5.5 EXPO.....	9
2.6 COMPONENTIZAÇÃO.....	9
2.7 OTIMIZAÇÃO NO REACT.....	9
3. DESENVOLVIMENTO DO SISTEMA.....	11
3.1 DESIGN E PROTOTIPAÇÃO.....	11
3.2 IMPLEMENTAÇÃO FRONT-END.....	22
3.3 TESTE DE UI/UX.....	24
4. VALIDAÇÃO E RESULTADOS.....	26
5. CONCLUSÃO.....	30
5.1 TRABALHOS FUTUROS.....	31
6. REFERÊNCIAS.....	32

1. INTRODUÇÃO

No cenário de pandemia do COVID-19 e a necessidade de distanciamento social, muitos estabelecimentos se viram obrigados a se adaptar rapidamente para manter suas atividades e atender seus clientes. Nesse contexto, a tecnologia se torna uma aliada fundamental para garantir a continuidade dos negócios e a satisfação dos clientes.

Devido a isso, este trabalho foi originado de uma aplicação previamente implementada para dispositivos móveis, que tinha como objetivo apenas o controle de fluxo de pessoas em estabelecimentos, o CoronAway. Com a redução dos casos de COVID-19, a ideia pôde se expandir para um conjunto maior de soluções, o Placy, com o objetivo de possibilitar aos estabelecimentos uma forma centralizada para divulgação de informações e serviços aos clientes de maneira mais eficiente e segura. Assim, nesta aplicação, será possível fornecer informações atualizadas sobre a lotação atual do estabelecimento, a lista de produtos disponíveis, cardápios, programas de fidelidade, avaliações e reservas, de forma que os clientes possam se planejar melhor e evitar aglomerações.

Além disso, o projeto tem como foco central o desenvolvimento de uma aplicação que utilize conceitos de UI/UX como usabilidade, acessibilidade e responsividade, com práticas e tecnologias eficientes e atualizadas do mercado front-end como frameworks, componentização e otimização. Dessa forma, a aplicação será intuitiva e fácil de usar para diversos tipos de usuários e dispositivos móveis.

Portanto, este trabalho se mostra relevante e atual, principalmente frente ao cenário pandêmico ou outras situações que possam ocorrer, e que exigiu uma adaptação rápida e eficiente dos estabelecimentos para garantir a segurança e a satisfação dos clientes. Assim, a implementação da aplicação Placy contribuirá para melhorar a experiência e eficiência dos usuários e estabelecimentos.

1.1 MOTIVAÇÃO

A pandemia do COVID-19 impôs uma nova realidade aos estabelecimentos e aos consumidores, em que a preocupação com a saúde e a segurança se tornou prioridade. Nesse contexto, a implementação de uma aplicação como a Placy se torna essencial para garantir que os clientes possam tomar decisões informadas sobre sua frequência em estabelecimentos, baseados em informações atualizadas sobre a lotação. Isso não só contribui para evitar aglomerações e a propagação do vírus, mas também para aprimorar a experiência do cliente,

que pode se planejar com mais facilidade e aproveitar melhor o tempo em seus estabelecimentos favoritos.

Dessa forma, a motivação para a implementação da Placy não se restringe apenas ao contexto pandêmico, mas se estende à necessidade de oferecer soluções eficientes e tecnológicas que garantam a segurança e a satisfação dos clientes em todas as situações.

1.2 OBJETIVO GERAL

Este trabalho tem como objetivo principal a criação e implementação da aplicação Placy, para disponibilizar informações de estabelecimentos como lotação atual, lista de produtos disponíveis, programas de fidelidade, reservas e avaliações. Assim, o trabalho será focado no design, protótipo e desenvolvimento front-end, juntamente com as definições de conceitos, métodos e ferramentas utilizadas.

1.3 OBJETIVOS ESPECÍFICOS

- 1) Descrever os principais conceitos e métricas de usabilidade e acessibilidade;
- 2) Definir conceitos de design e prototipação;
- 3) Definir ferramentas front-end utilizadas na implementação do trabalho;
- 4) Definir conceitos e práticas atreladas ao React como componentização e otimização;
- 5) Descrever fluxo da implementação do design e protótipo da aplicação;
- 6) Propor uma arquitetura e apresentar detalhes do desenvolvimento da aplicação;
- 7) Realizar um teste de usabilidade da aplicação com usuários.

1.4 TRABALHOS RELACIONADOS

A divulgação de informação é um tema de extrema importância, diversas plataformas e redes sociais são fundamentais para esse meio. Assim, existem algumas plataformas que são relacionadas com o tema e que influenciaram a construção deste trabalho, alguns deles são apresentados a seguir.

O Get In¹ é um aplicativo que possibilita aos usuários a realização de reservas, acompanhamento de filas, cardápio e delivery em estabelecimentos. Com essas e mais algumas funcionalidades previstas, ele já é utilizado atualmente por muitos bares e restaurantes do Brasil. Ele foi importante na inspiração do Placy pois essas funcionalidades citadas acima, além de outras como gestão de comandas e programa de fidelidade, também foram adicionadas ao trabalho;

¹ Disponível em: <<https://www.getinapp.com.br/>>. Acesso em: 10 março 2023.

O Trinks², que possui um nicho mais específico no campo das reservas, é um sistema para salão de beleza e barbearia que tem parceria com as principais redes e estabelecimentos do Brasil. Com mais de 100 mil usuários, já é um aplicativo consolidado para empresas e clientes. Foi importante para a implementação do Placy, pois o conjunto de soluções que este trabalho propõe, como reservas, não possui um nicho específico como barbearia ou restaurantes, e sim qualquer tipo de estabelecimento;

O iFood³ é uma plataforma digital de delivery de comida e produtos que conecta restaurantes e estabelecimentos com consumidores. Fundada no Brasil em 2011, a empresa oferece uma ampla variedade de opções de restaurantes e pratos, permitindo que os usuários escolham e solicitem a entrega de comida ou produtos diretamente do aplicativo. O iFood é considerado o maior aplicativo de delivery de comida da América Latina e está presente em diversos países, incluindo Brasil, México, Argentina, Colômbia, Peru, Chile e Equador. Ele foi importante na influência da implementação de pedidos e cardápios da aplicação, possuindo design e funcionalidades já consolidadas no mercado de aplicativos.

² Disponível em: <<https://www.trinks.com/>>. Acesso em: 10 março 2023.

³ Disponível em: <<https://www.ifood.com.br/>>. Acesso em: 10 março 2023.

2. REFERENCIAL TEÓRICO

Este capítulo tem por finalidade estabelecer os conceitos teóricos necessários usados no trabalho. O capítulo é iniciado com conhecimentos gerais acerca de usabilidade, acessibilidade e responsividade, para em seguida detalhar sobre ferramentas e tecnologias de design, prototipação e front-end que foram utilizados neste trabalho.

2.1 USABILIDADE E UX

Segundo Dias (2007, p. 25), o termo usabilidade começou a ser usado como substituto da expressão “user-friendly” no início da década de 1980 e teve suas raízes na ciência cognitiva, nas áreas de psicologia e ergonomia.

Usabilidade é a habilidade de um produto ser utilizado por usuários específicos para alcançar metas específicas com eficácia, eficiência e satisfação em um contexto particular de uso, de acordo com a norma ISO 9241-11 (1998).

Sendo assim, a usabilidade possui especificidades e pode ser utilizada em cenários onde existe interface com o homem. No contexto do projeto, sua modalidade é a usabilidade de software, a qual é definida como a capacidade do software de ser aprendido, compreendido, usado e apreciado pelo usuário, quando usado nas condições especificadas (ISO/IEC FCD 9126-1, 1991).

User Experience (UX), ou Experiência do Usuário em português, é um conceito que se refere a todas as interações que um usuário tem com um produto ou serviço, incluindo aspectos subjetivos como percepções, emoções e opiniões. A UX engloba áreas como a usabilidade, a acessibilidade, o design, a interação, a arquitetura da informação, entre outros, buscando proporcionar ao usuário uma experiência positiva, eficiente e satisfatória (GUBE, 2010).

2.1.1 MÉTRICAS DE UX

A experiência do usuário pode ser medida por meio de métricas, que permitem avaliar o desempenho de um sistema, site ou aplicativo. Elas são essenciais para avaliar a qualidade da usabilidade e identificar possíveis problemas que podem prejudicar a usabilidade. De acordo com os fundamentos dos autores Nielsen, J. (1994), Tullis, T., & Albert, B. (2013), Sauro, J. (2011) e Brooke, J. (1996), algumas métricas comuns de experiência do usuário incluem:

- Taxa de sucesso: a porcentagem de usuários que completam uma tarefa específica com sucesso. Ela consegue avaliar a eficácia do sistema em permitir que os usuários atinjam seus objetivos;
- Tempo de conclusão: o tempo que um usuário leva para concluir uma tarefa. Ela ajuda a avaliar a eficiência do sistema em permitir que os usuários realizem as tarefas com rapidez;
- Taxa de erro: a porcentagem de usuários que cometem erros ao tentar realizar uma tarefa. Ela é útil para identificar problemas de usabilidade e interface que podem estar impedindo que os usuários completem as tarefas com sucesso;
- Taxa de abandono: a porcentagem de usuários que abandonam o site ou aplicativo antes de concluir uma tarefa. Ela possibilita a identificação de problemas de usabilidade que podem estar impedindo que os usuários completem as tarefas com sucesso;
- Número de cliques: a quantidade de cliques que um usuário precisa para completar uma tarefa. Essa métrica ajuda a avaliar a facilidade de uso do sistema e a identificar possíveis melhorias de interface;
- Satisfação do usuário: a percepção geral do usuário em relação ao sistema, site ou aplicativo. Ela pode ser medida por meio de pesquisas de satisfação do usuário ou avaliações subjetivas.

Assim, é importante escolher as métricas mais relevantes para o contexto específico da aplicação e utilizá-las de forma consistente para medir a usabilidade ao longo do tempo e possibilitar melhorias para a experiência do usuário.

2.2 ACESSIBILIDADE

Acessibilidade é a capacidade de um sistema, serviço ou produto ser utilizado por todas as pessoas, independentemente de suas habilidades ou limitações. No campo do software, a acessibilidade na web é um conjunto de diretrizes e práticas que buscam tornar os conteúdos digitais acessíveis a pessoas com deficiência visual, auditiva, física, cognitiva ou qualquer outra limitação que possa afetar sua navegação na internet (W3C, 2022).

Essa definição é baseada nas diretrizes do W3C (World Wide Web Consortium), que é o principal órgão internacional responsável por desenvolver padrões e diretrizes para a web. O W3C define a acessibilidade como "a facilidade com que as pessoas podem usar a web, independentemente de sua capacidade ou deficiência".

2.2.1 MÉTRICAS DE ACESSIBILIDADE

O W3C define e destaca algumas diretrizes e recomendações para o desenvolvimento de conteúdos acessíveis na web e em dispositivos móveis. Algumas delas são o WCAG 2.1: Diretrizes de Acessibilidade para Conteúdo da Web (WCAG), que visam tornar o conteúdo da web mais acessível. Esta versão 2.1 do WCAG inclui algumas recomendações específicas para dispositivos móveis; e o WAI-ARIA: Iniciativa de Acessibilidade na Web (WAI) do W3C que desenvolveu o ARIA (Accessible Rich Internet Applications), sendo um conjunto de atributos que podem ser adicionados ao HTML para tornar o conteúdo mais acessível em dispositivos móveis.

Assim, como métricas de acessibilidade, podem ser utilizados alguns pontos do Mobile Web Best Practices do W3C, como: design responsivo, nível de contrastes, legibilidade do texto, navegação clara e simples, entre outros.

2.3 RESPONSIVIDADE

Segundo a W3C, uma das recomendações para acessibilidade é a responsividade, a qual é a capacidade de uma interface ou layout se adaptar a diferentes tamanhos e resoluções de tela, mantendo uma boa legibilidade e usabilidade.

O design responsivo é uma abordagem de design que permite que um site se ajuste a diferentes dispositivos e tamanhos de tela, como desktops, laptops, tablets e smartphones, e ainda assim ter um layout atraente e funcional (FRIEDMAN, 2018).

2.4 DESIGN E PROTOTIPAÇÃO

Neste tópico, serão apresentados e descritos os conceitos fundamentais de design system e prototipação, juntamente com a principal ferramenta utilizada para a criação e desenvolvimento desses elementos: o Figma.

2.4.1 FIGMA

O Figma é uma plataforma de design de interface de usuário (UI) e de experiência do usuário (UX) baseada na nuvem. Possuindo planos gratuitos e pagos, ela permite aos usuários criar, colaborar, compartilhar e testar designs de aplicativos e websites em tempo real (FIGMA, 2023).

Além disso, oferece recursos como edição simultânea, comentários em tempo real, prototipagem interativa, design responsivo, animações e integração com outras ferramentas de

design. É uma ferramenta popular entre designers e equipes de design devido à sua facilidade de uso, acessibilidade e colaboração em equipe.

2.4.2 DESIGN SYSTEM

Design system, ou sistema de design, é uma forma de padronizar o processo de design e desenvolvimento de um produto digital para garantir a consistência e a qualidade da experiência do usuário. Ele inclui elementos como paletas de cores, tipografia, ícones, padrões de layout, componentes de interface, animações, documentação, entre outros, possibilitando a equipe a fazer melhorias e verificar os padrões do projeto ao longo do desenvolvimento ("What is a Design System?" an article by Dan Mall", [s.d.]).

No Figma, é possível criar um design system que se integra ao design e ao protótipo em uma única plataforma.

2.4.3 PROTOTIPAÇÃO

A prototipação de software é uma etapa importante no processo de design de um produto de software, que permite testar e validar soluções de design e funcionalidades antes da implementação final, visando melhorar a experiência do usuário e reduzir os custos e o tempo de desenvolvimento (LANCASTER, 2004).

No Figma, é possível criar protótipos, os quais disponibilizam diversas opções e configurações, como tipo de dispositivo, tamanhos, fluxos da aplicação e animações.

2.5 FRAMEWORKS E TECNOLOGIAS FRONT-END

Os Frameworks oferecem um conjunto de funcionalidades básicas para a construção de software e permitem que os desenvolvedores se concentrem nas funcionalidades específicas de sua aplicação. Eles tendem a acelerar o desenvolvimento de software, garantir maior consistência e qualidade de código, e permitir que as equipes de desenvolvimento trabalhem de forma mais eficiente (SOMMERVILLE, 2011).

Alguns exemplos de frameworks populares incluem o React e Angular para desenvolvimento de aplicações web, e React Native e Flutter para desenvolvimento de aplicações móveis ("Stack Overflow Developer Survey 2021", [s.d.]).

2.5.1 REACT JS

Desenvolvido pelo Facebook em 2013, React JS é uma biblioteca JavaScript de código aberto utilizada para construir interfaces de usuário (UI) em aplicações web. Atualmente é um

dos frameworks mais populares para o desenvolvimento de aplicações web (META OPEN SOURCE, 2023).

O React é baseado no conceito de componentes e na técnica de Virtual DOM para otimizar o desempenho da interface, atualizando apenas as partes que foram alteradas em vez de atualizar todo o DOM.

2.5.2 TYPESCRIPT

Desenvolvida pela Microsoft em 2012, o TypeScript é uma linguagem de programação de código aberto que é uma extensão da linguagem JavaScript, adicionando recursos de tipagem estática e outros recursos de programação orientada a objetos ao JavaScript. Seu objetivo é tornar o desenvolvimento de aplicativos em JavaScript mais seguro, escalável e estruturado, fornecendo um conjunto de recursos avançados de linguagem e detecção de erros de codificação em tempo de compilação em vez de tempo de execução, o que ajuda a melhorar a qualidade do código e a reduzir erros e falhas na aplicação final (MICROSOFT).

TypeScript é amplamente utilizado em aplicações web, aplicações desktop e aplicações móveis, e é suportado por muitos frameworks populares, como o Angular e o React.

2.5.3 REACT NATIVE

Desenvolvido pelo Facebook em 2015, o React Native é um framework de desenvolvimento mobile que permite a criação de aplicativos para iOS e Android usando JavaScript e React.

Como um dos frameworks mobile mais utilizados atualmente, ele permite que os desenvolvedores criem aplicativos móveis usando uma única base de código, que é compilada para código nativo para cada plataforma de destino. Isso significa que os aplicativos criados com React Native são tão rápidos e responsivos quanto os aplicativos nativos, enquanto mantêm a facilidade de desenvolvimento e a eficiência de código da plataforma web React (REACT NATIVE).

2.5.4 NATIVEBASE

NativeBase é uma biblioteca de componentes para React Native, que fornece um conjunto de componentes prontos para uso para o desenvolvimento de aplicativos móveis para iOS e Android. Esses componentes são baseados em componentes nativos de cada plataforma,

como botões, ícones, listas, formulários e muito mais, e são altamente personalizáveis e fáceis de usar (NATIVEBASE).

Com o NativeBase, os desenvolvedores podem criar interfaces de usuário consistentes e atraentes para seus aplicativos móveis, economizando tempo e esforço no desenvolvimento, além de já possuir e exigir funções de acessibilidade em seus componentes. A biblioteca é gratuita e de código aberto, e é amplamente utilizada por desenvolvedores em todo o mundo.

2.5.5 EXPO

Expo é um conjunto de ferramentas que facilita o desenvolvimento de aplicativos móveis com React Native. Ele fornece uma série de recursos, como um conjunto de componentes UI prontos para uso, um ambiente de desenvolvimento integrado (IDE) em nuvem e uma plataforma de distribuição de aplicativos móveis (EXPO).

Com o Expo, os desenvolvedores podem criar aplicativos para iOS e Android sem precisar configurar e manter as ferramentas de desenvolvimento em seus próprios computadores, podendo utilizá-los em seus próprios smartphones. Além disso, o Expo permite que os desenvolvedores acessem recursos nativos dos dispositivos móveis, como câmera, geolocalização e armazenamento, utilizando código em JavaScript ou Typescript.

2.6 COMPONENTIZAÇÃO

De acordo com a documentação do React (REACT, 2023), a componentização é um conceito fundamental que permite aos desenvolvedores dividir a interface do usuário em partes menores e independentes, chamadas de "componentes". Esses componentes são como blocos de construção reutilizáveis que podem ser combinados para criar interfaces complexas. Cada componente pode ter sua própria lógica, estados e propriedades, tornando mais fácil o gerenciamento e a manutenção do código.

Com a componentização, é possível criar interfaces de usuário flexíveis e escaláveis, além de permitir que diferentes equipes de desenvolvimento trabalhem em diferentes partes da aplicação. A componentização é uma das principais responsáveis pela popularização do React e é amplamente utilizado no desenvolvimento de interfaces de usuário para web e mobile.

2.7 OTIMIZAÇÃO NO REACT

De acordo com a documentação do React (REACT, 2023), por atualizar a interface do usuário de forma reativa, qualquer mudança no estado da aplicação pode acionar uma

renderização da interface de usuário, o que pode afetar o desempenho. Sendo assim, a otimização é um conjunto de técnicas que melhoram o desempenho da aplicação, como:

- Utilizar o `useMemo` e `useCallback` para evitar cálculos desnecessários ou funções sendo recriadas a cada renderização;
- Utilizar o lazy loading para carregar componentes somente quando necessário;
- Utilizar o virtual scrolling para renderizar somente os elementos visíveis em listas longas.

3. DESENVOLVIMENTO DO SISTEMA

Neste capítulo, será apresentada a metodologia adotada para a construção da aplicação, abrangendo o design system, a prototipação e a codificação front-end. Ao longo do capítulo, serão descritas as etapas, ferramentas e boas práticas utilizadas, visando alcançar uma aplicação final com um design atrativo, experiência de usuário intuitiva e uma implementação front-end eficiente e escalável.

3.1 DESIGN E PROTOTIPAÇÃO

O desenvolvimento da aplicação foi iniciado pelo design e protótipo do Placy na plataforma Figma. Primeiramente foi implementado o Design System, partindo da escolha da paleta de cores que seria utilizada tanto no tema claro quanto no tema escuro da aplicação. A plataforma essencial nessa escolha foi o Adobe Color⁴, que possui diversas combinações feitas por outros usuários e também permite a criação de uma paleta personalizada.

Pensando na acessibilidade, após a escolha da paleta principal, as cores que seriam utilizadas de fundo e em texto foram testadas em um verificador de contraste que segue as normas do WCAG, o Color Contrast Ratio Calculator & Checker⁵, com todas elas sendo aprovadas com WCAG AAA (nota máxima), como pode ser visualizado no exemplo a seguir, no qual um texto de cor #1F2937 é testada contra um fundo de cor #E1E1E6:

Figura 1 - Verificador de contraste seguindo as normas do WCAG.

The figure shows a screenshot of the Color Contrast Ratio Calculator & Checker tool. On the left, there are two color selection boxes. The top one is labeled 'Cor de primeiro plano' (Foreground color) with the hex code '#1F2937'. The bottom one is labeled 'Cor de fundo' (Background color) with the hex code '#E1E1E6'. Below these is a text input field containing the text 'Texto do título'. To the right of the text is a descriptive paragraph: 'Exemplo de texto: É um fato estabelecido há muito tempo que um leitor se distrai com o conteúdo legível de uma página ao olhar para seu layout.' At the bottom left, it says 'Relação de Contraste - 11.26:1'. To the right of that is a blue 'COPIAR URL' button. At the very bottom, there are two sections: 'Para texto e imagens de texto em grande escala' which includes 'WCAG AA (contraste mínimo)' and 'WCAG AAA (contraste realçado)', both with green checkmarks; and 'Para texto em grande escala e imagens de texto' which also includes 'WCAG AA (contraste mínimo)' and 'WCAG AAA (contraste realçado)', both with green checkmarks.

Fonte: Adaptado de Color Contrast Ratio Calculator & Checker (2023).

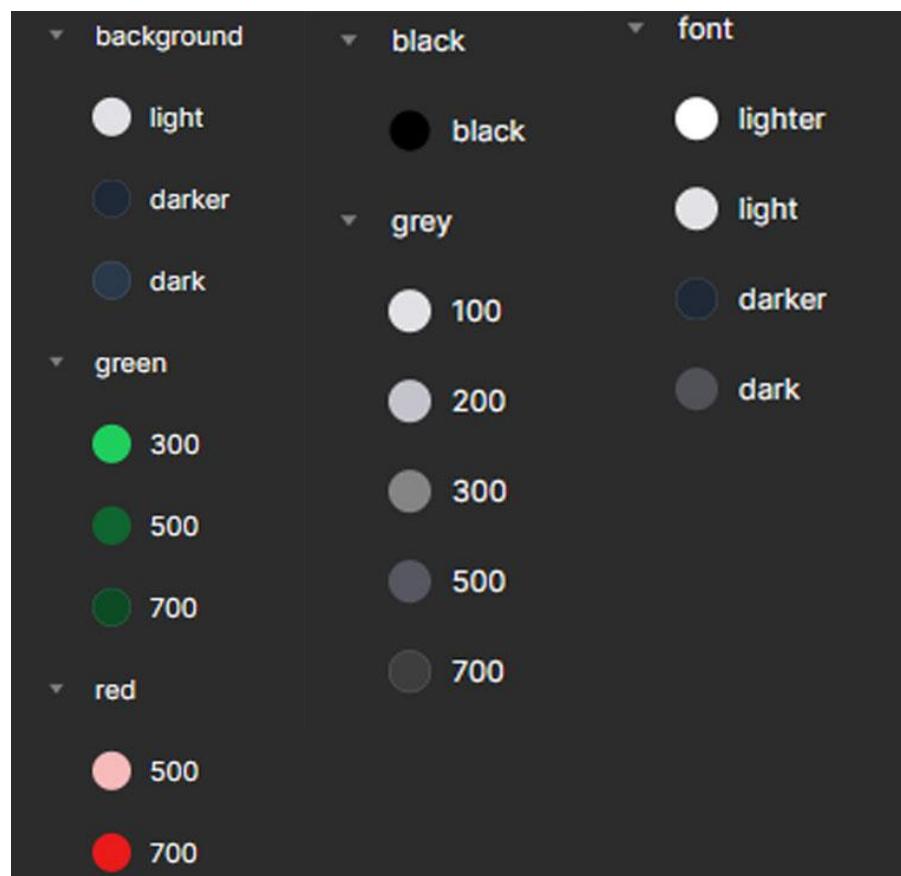
⁴ Disponível em: <<https://color.adobe.com/>>. Acesso em: 15 março 2023.

⁵ Disponível em: <<https://contrastchecker.online/>>. Acesso em: 15 março 2023.

Na imagem acima, é possível visualizar o resultado da combinação das cores do lado direito e a escala de contraste abaixo dos inputs. Os resultados da combinação se encontram na parte inferior, e em caso de aprovação na categoria, é mostrado um símbolo de confirmação em verde, do lado direito de cada categoria.

Assim, foi montada a biblioteca de cores no Figma, com a utilização do plano gratuito da plataforma, e adicionando mais algumas variantes como tonalidades de cinza, verde (cor principal de destaque), vermelho e definindo as cores a serem utilizadas em fontes e planos de fundo, como pode ser visto abaixo:

Figura 2 - Biblioteca de cores do Design System.



Fonte: Autoria própria.

Em seguida, foi feita a logo do Placy, que originou-se do marcador a ser utilizado no mapa da aplicação, transformado na letra “P” e com um olho, representando uma espécie de “mascote” da marca, como pode ser visto a seguir:

Figura 3 - Logo do Placy.



Fonte: Autoria própria.

Continuando o Design System, foram escolhidos os ícones da biblioteca Phosphor Icons⁶ que seriam utilizados na plataforma. A biblioteca também possui uma página no Figma com todos os ícones disponíveis e suas variantes, separados por categoria, o que facilita o desenvolvimento do Design System ao copiá-la para o projeto, como visto abaixo:

⁶ Disponível em: <<https://phosphoricons.com/>>. Acesso em: 16 março 2023.

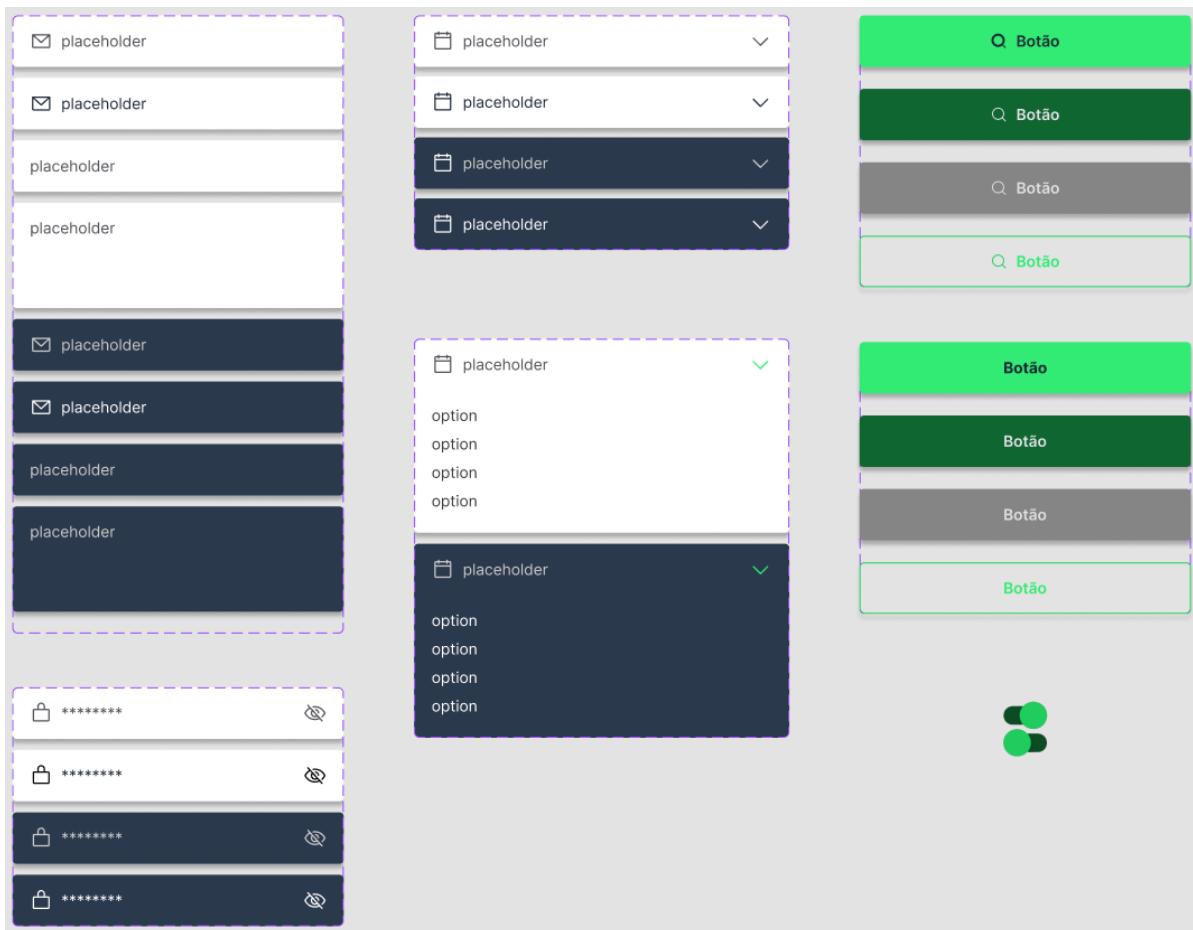
Figura 4 - Ícones do Design System.



Fonte: Autoria própria.

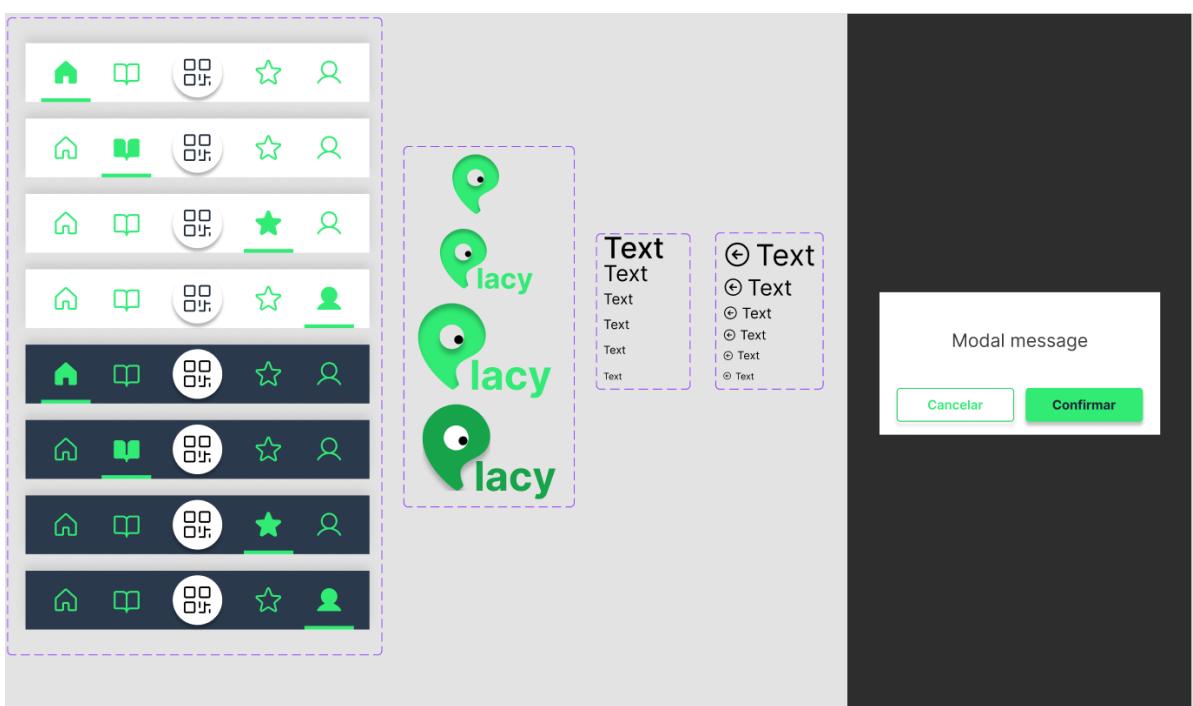
Uma das funcionalidades mais interessantes que o Figma possui é a criação de componentes, que em um contexto de React, torna o design e o protótipo mais próximos de como o aplicativo é implementado em código, já que um dos fundamentos e vantagens do React é a componentização. Sendo assim, todos os componentes, como botões, textos e campos de texto, seus comportamentos e variantes foram criados e adicionados na página “Components”, como pode ser visto nas figuras abaixo:

Figura 5a - Componentes do Design System.



Fonte: Autoria própria..

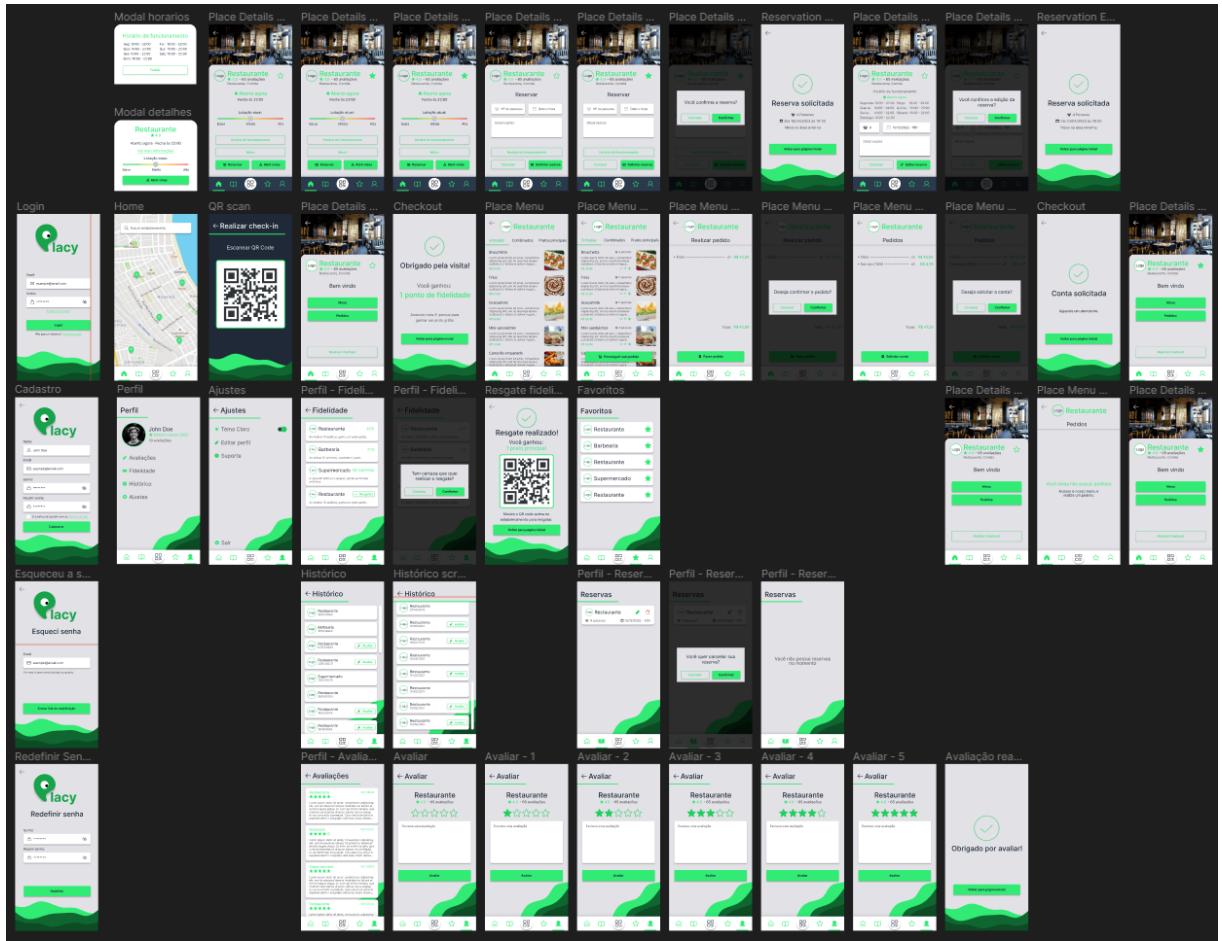
Figura 5b - Componentes do Design System.



Fonte: Autoria própria.

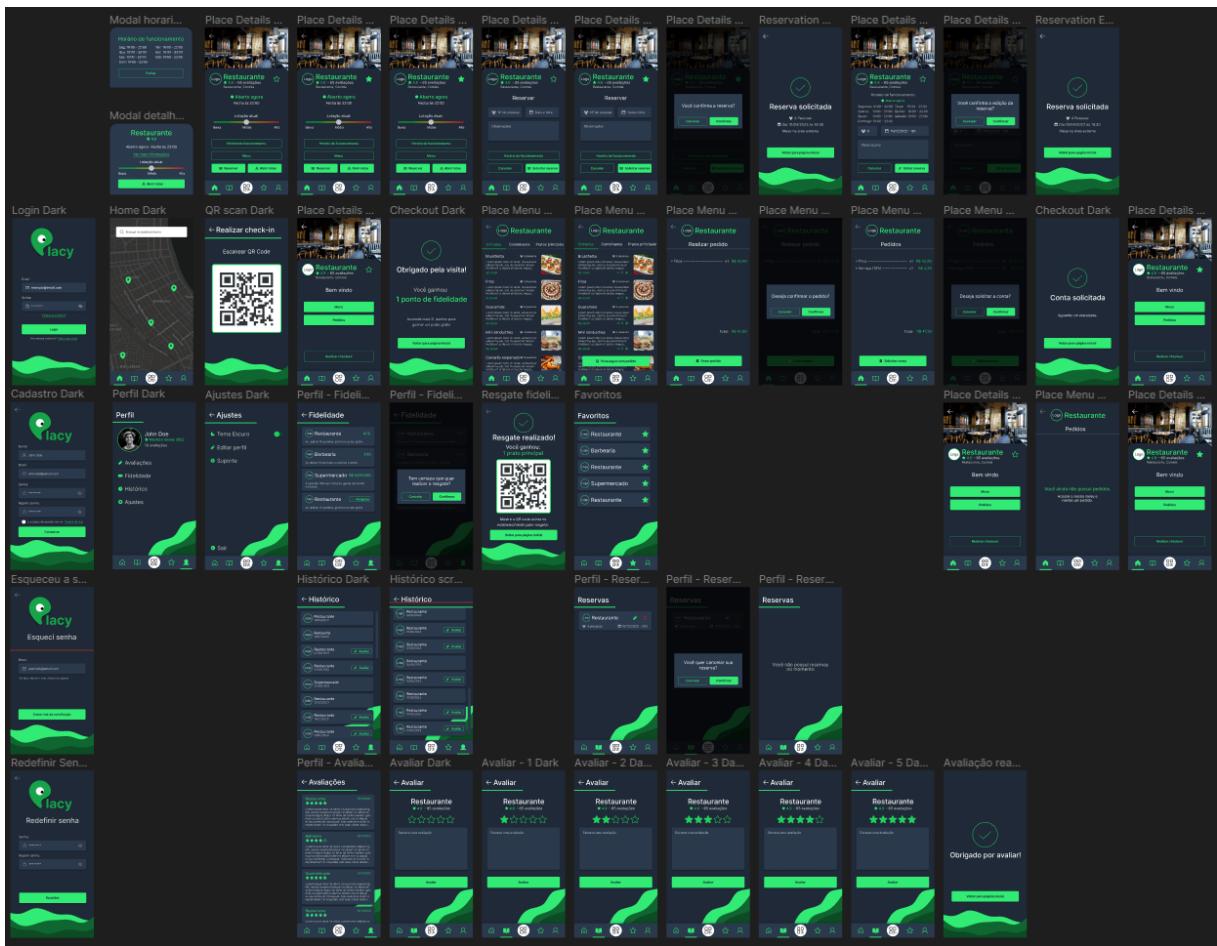
Assim, foi finalizada a parte do Design System, que configurou todos os componentes, cores e ícones que serão utilizados no desenvolvimento. Dando sequência, ainda no Figma, foram criadas na página “App Client” as telas da aplicação juntamente com suas interações e fluxo (protótipo). Em uma visão macro, é possível visualizar o conjunto de telas do modo claro e escuro, cada tela nomeada de acordo com a respectiva funcionalidade, como mostrado nas figuras a seguir:

Figura 6 - Telas do design da aplicação no modo claro.



Fonte: Autoria própria.

Figura 7- Telas do design da aplicação no modo escuro.



Fonte: Autoria própria.

Em relação ao protótipo, é possível visualizar a rede de interações entre as telas e botões e inclusive entre temas. Na imagem abaixo, cada “curva” representa uma interação do usuário com um botão, que pode ser customizada em tipo de clique, animação e direção da animação, criando, assim, as funções de navegação do protótipo.

Figura 8 - Rede de interações entre telas da aplicação.



Fonte: Autoria própria.

O design, funcionalidades, telas e fluxos foram inspirados em aplicativos já consolidados do mercado, como os mencionados na seção trabalhos relacionados: iFood, Get In e Trinks.

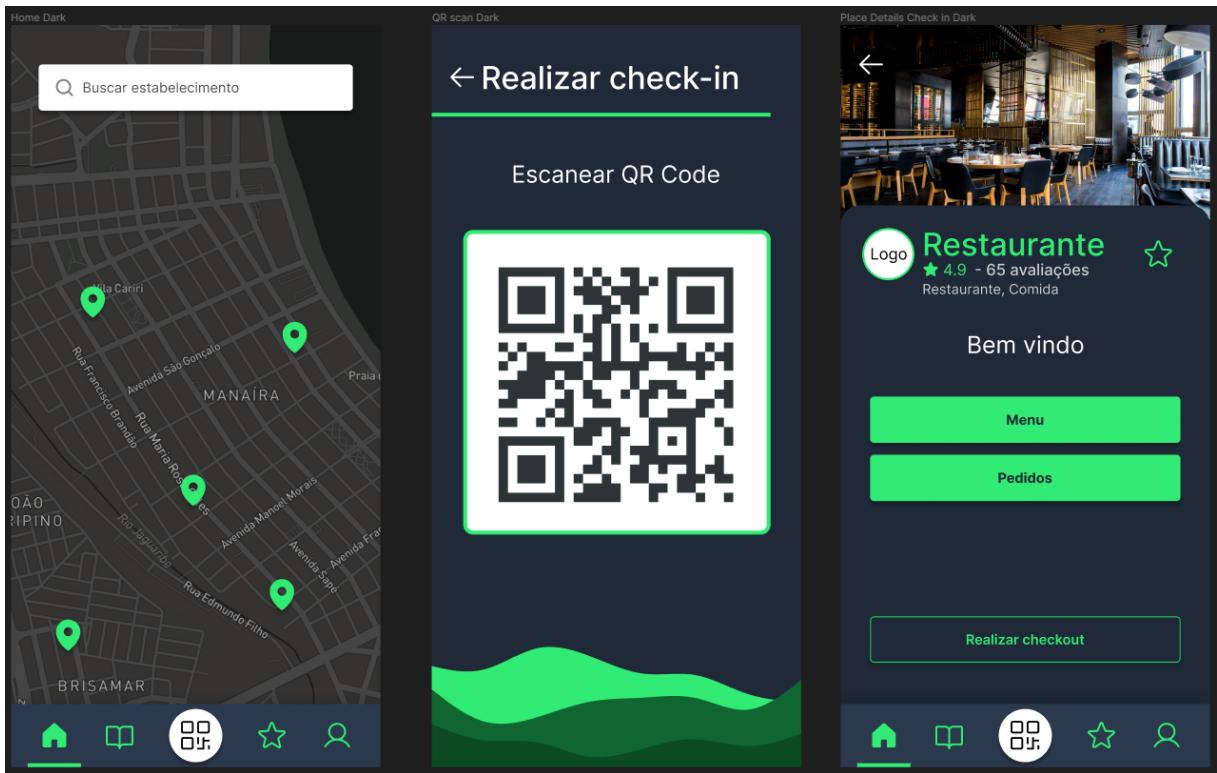
Ao analisar detalhadamente a aplicação em questão, é possível identificar uma ampla variedade de fluxos e funcionalidades presentes no protótipo, são eles: autenticação (login, cadastro e redefinição de senha), busca e detalhes de estabelecimentos, solicitação e edição de reservas, abrir rotas, visualização de menu do estabelecimento, check-in e checkout no estabelecimento, realizar pedidos e solicitar contas, visualização de progresso e resgate de programa de fidelidade, visualização de histórico de visitas, favoritar/desfavoritar e avaliar estabelecimentos, mudança de tema, ajustes no perfil e logout.

Assim, devido a essa grande quantidade de fluxos, neste momento serão apresentados apenas três exemplos, no tema escuro, com os demais podendo ser acessados por meio do design ou protótipo disponíveis no Figma⁷.

- Fluxo de check-in no estabelecimento: Na página principal da aplicação, é possível visualizar e buscar no mapa por estabelecimentos cadastrados. No centro do menu inferior, existe um botão com um QR Code, que ao ser clicado, o usuário é levado para uma tela na qual ele pode escanear o QR Code fisicamente no estabelecimento para realizar o check-in. Após escanear, existe uma tela que permite que o usuário possa ver o cardápio, pedidos ou realizar checkout. Esse fluxo é possível de ser visualizado na captura de tela abaixo, na ordem em que foi descrito:

⁷ Disponível em: <<https://www.figma.com/file/LxlfhYcvBgqrIwUtGGMXBD/>>

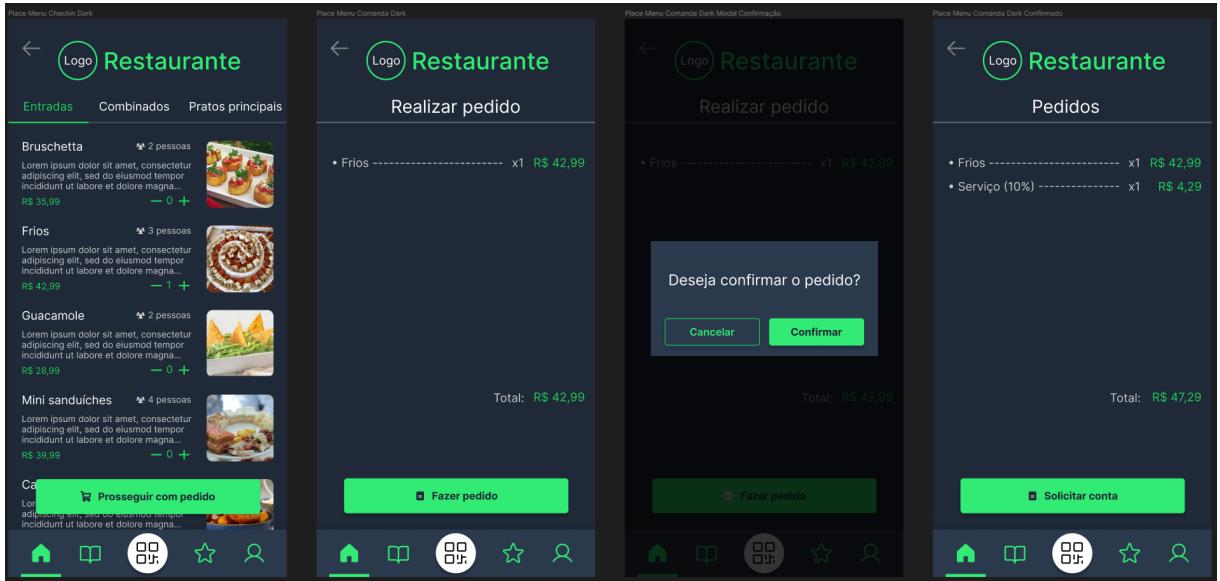
Figura 9 - Fluxo de check-in no estabelecimento..



Fonte: Autoria própria.

- Fluxo de realização de pedidos após check-in: Partindo da última tela do fluxo anterior, é possível realizar pedidos a partir do menu de produtos, que no caso, como o exemplo do protótipo é um restaurante, é um cardápio. No menu, é possível selecionar itens e a quantidade de cada um deles para montar um pedido, que é mostrado juntamente com os valores por item e valor total do pedido, como visto na segunda tela abaixo. Ao clicar no botão de “Fazer pedido”, existe um modal de confirmação, que ao confirmar, é levado para uma nova tela semelhante a anterior, mas com o resumo de todos os pedidos realizados e taxas (caso existam). O fluxo completo descrito acima pode ser visto a seguir:

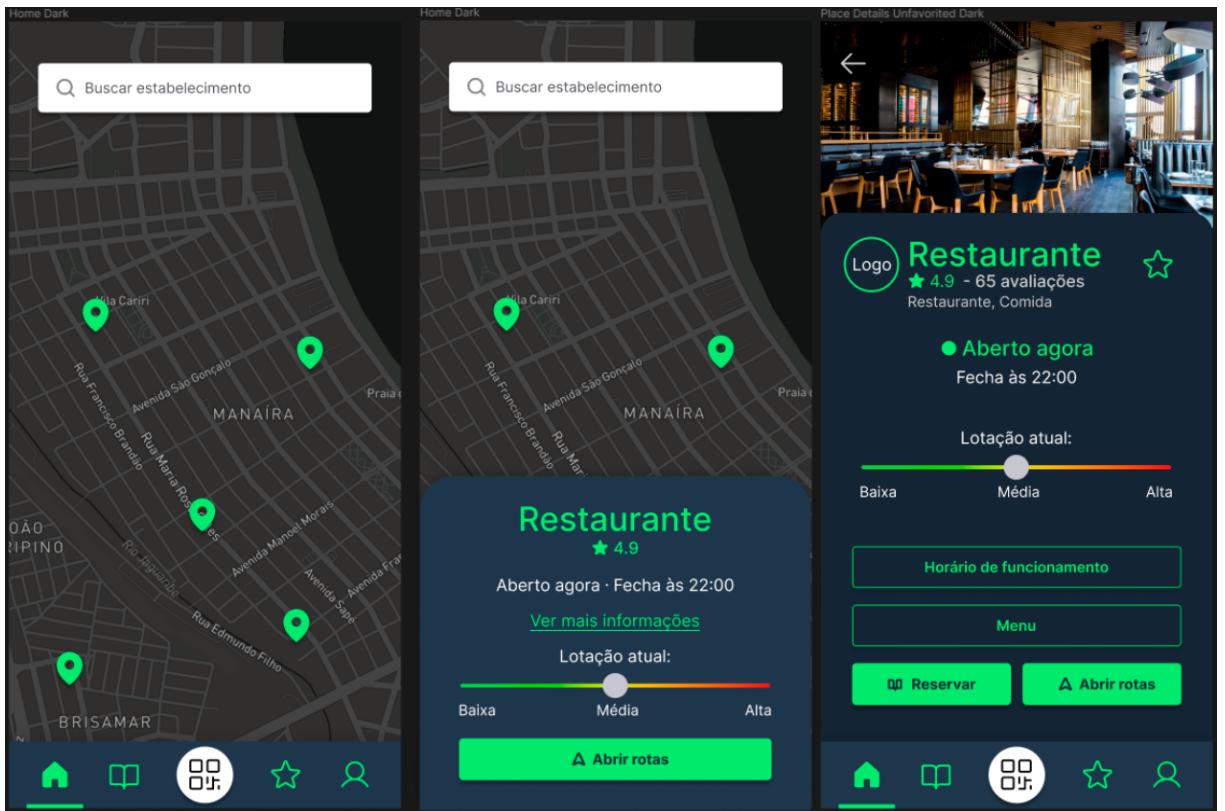
Figura 10 - Fluxo de realização de pedidos após check-in.



Fonte: Autoria própria.

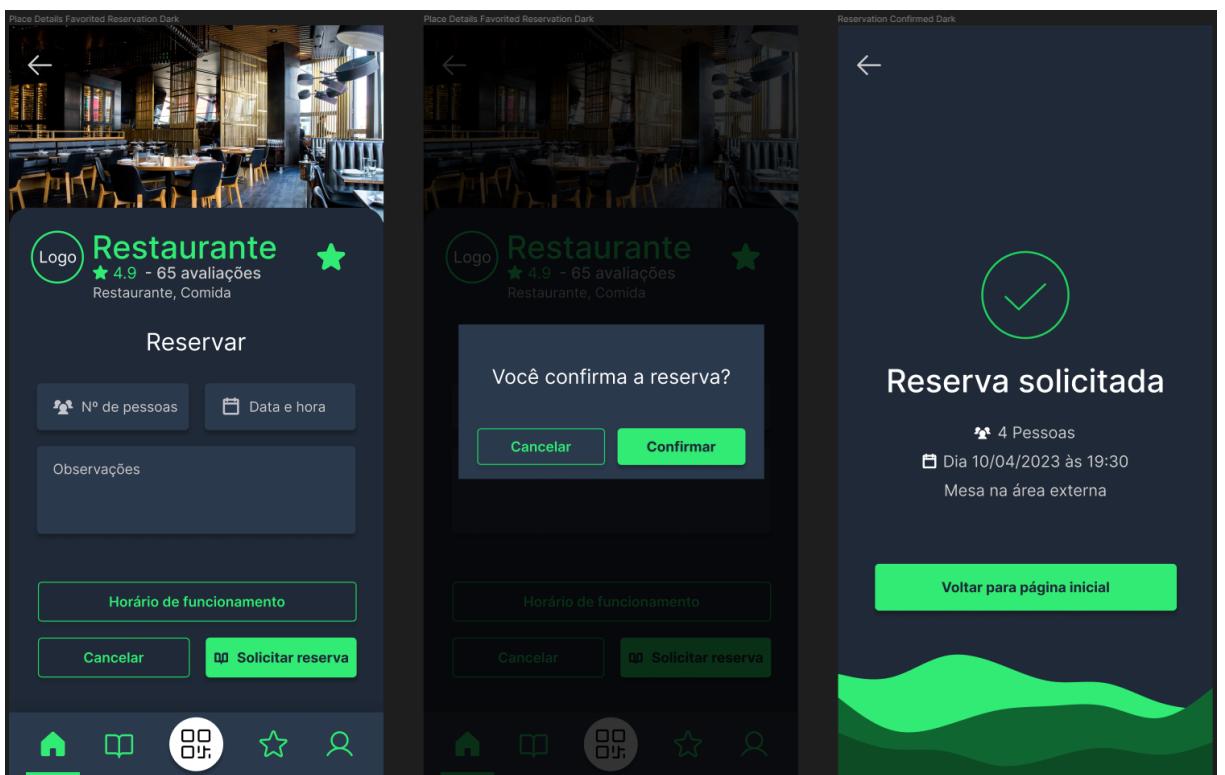
- Fluxo de reservas: A partir da página principal (mapa), ao clicar em um estabelecimento, um modal com algumas informações sobre o local aparece. Clicando em “Ver mais informações”, o usuário é levado para a tela com mais detalhes sobre o estabelecimento, juntamente com algumas opções como “Horário de funcionamento”, “Menu”, “Reservar” e “Abrir rotas”. Essas telas e comportamentos podem ser vistos na primeira imagem abaixo. Dando sequência, ao clicar em “Reservar”, uma nova tela com informações sobre a reserva aparece para ser preenchida, que após preencher e clicar em “Solicitar reserva”, um modal de confirmação aparece, que caso seja confirmado, aparece uma tela de confirmação da solicitação de reserva com as informações preenchidas, como pode ser visto na segunda imagem abaixo:

Figura 11a - Fluxo de reservas.



Fonte: Autoria própria.

Figura 11b - Fluxo de reservas.



Fonte: Autoria própria.

Assim, utilizando princípios de usabilidade, acessibilidade e responsividade, a prototipação foi realizada de forma iterativa, acelerando e facilitando consequentemente o processo de desenvolvimento da aplicação e permitindo refinamentos e melhorias contínuas. Com o design e prototipação bem estruturados, o próximo passo é implementar o projeto em React Native, visando à construção de uma aplicação robusta, funcional e eficiente.

3.2 IMPLEMENTAÇÃO FRONT-END

A estrutura base da aplicação em React Native foi implementada seguindo um fluxo similar ao descrito anteriormente para a montagem do design system. Inicialmente, o projeto foi criado utilizando a plataforma Expo, onde foi estabelecido o uso do Typescript como linguagem de programação. Além disso, a biblioteca NativeBase foi adicionada ao projeto, com o propósito de agilizar e facilitar o desenvolvimento dos componentes a serem utilizados.

A escolha do Typescript como linguagem de programação traz diversos benefícios para o desenvolvimento da aplicação em React Native. Ele ajuda a evitar erros comuns durante o desenvolvimento, melhorando a qualidade e robustez do código. Além disso, ele oferece recursos avançados de autocompletar e validação de código, o que agiliza o desenvolvimento e facilita a detecção de erros antes mesmo da execução do código.

Por sua vez, a inclusão da biblioteca NativeBase no projeto traz vantagens significativas. Além de acelerar o desenvolvimento dos componentes, a biblioteca oferece recursos nativos de acessibilidade. Isso garante que a aplicação seja mais inclusiva, permitindo que usuários com necessidades especiais possam utilizar a aplicação de forma mais acessível e amigável. Com o NativeBase, é possível implementar elementos de interface acessíveis, como leitores de tela e recursos de navegação por gestos, proporcionando uma experiência de usuário aprimorada para todos os usuários.

Assim, após a instalação inicial do projeto, foram desenvolvidos os temas da aplicação, tanto para o modo claro quanto para o modo escuro. Esses temas foram criados com base nos esquemas de cores definidos previamente na biblioteca de cores, que foram estabelecidos durante o processo de design no Figma. Além disso, os tamanhos e as fontes a serem utilizadas nos textos foram especificados nos temas.

No arquivo de temas, o NativeBase proporciona a capacidade de criar e personalizar componentes padrão, substituindo o comportamento nativo da biblioteca. Isso inclui componentes como botões, campos de entrada e textos, juntamente com suas variantes, conforme definidos no design system. Dessa forma, ao importar um componente do NativeBase, ele já será customizado de acordo com os padrões estabelecidos para a aplicação,

evitando a necessidade de retrabalho em cada instância do componente. Essa funcionalidade contribui significativamente para a eficiência e consistência no desenvolvimento da interface da aplicação.

Após a definição e configuração dos temas, procedeu-se à criação das rotas da aplicação, que desempenham um papel fundamental na determinação das telas e da forma de navegação adotada. As rotas são categorizadas em rotas de autenticação e rotas da aplicação. As rotas de autenticação abrangem funcionalidades como cadastro, login e recuperação de senha, e são implementadas utilizando o tipo de navegação *Stack Navigation*, que segue uma estrutura de pilha, permitindo que as telas sejam sobrepostas de forma sequencial. Por sua vez, as rotas da aplicação são acessíveis somente para usuários autenticados e utilizam o tipo de navegação *Bottom Tabs*, que apresenta abas inferiores para facilitar a navegação entre diferentes seções da aplicação.

A implementação das rotas em uma aplicação viabiliza a criação das respectivas telas e componentes. Cada rota corresponde a uma tela específica, como no caso desta aplicação, o perfil do usuário, histórico, favoritos, configurações, entre outras. Os componentes, por sua vez, são elementos mais específicos do código, baseados no importante conceito de componentização do React. Eles podem representar itens recorrentes em uma ou mais telas, ou serem separados em arquivos independentes para aprimorar a organização e legibilidade do código-fonte. Exemplos de componentes criados na aplicação incluem a barra de lotação dos estabelecimentos, indicadores de carregamento, cabeçalhos das telas, modais, entre outros. Essa abordagem permite reutilizar e modularizar elementos visuais e funcionais, promovendo uma estrutura coerente e escalável para a aplicação.

As telas e componentes foram desenvolvidos utilizando *hooks*, que são ferramentas altamente recomendadas no React para otimização de desempenho. Na aplicação em questão, foram utilizados *hooks* como `useMemo` e `useCallback`, os quais desempenham um papel fundamental na otimização de cálculos computacionais intensivos e na redução da re-renderização desnecessária dos componentes. Além disso, outra ferramenta importante empregada foi a `FlatList`, do React Native, que permite renderizar listas longas e otimizar o carregamento e renderização de itens de forma eficiente. Essas ferramentas e técnicas contribuíram para melhorar o desempenho e a eficiência da aplicação, proporcionando uma experiência fluida ao usuário.

No contexto da responsividade, a aplicação foi cuidadosamente desenvolvida para garantir compatibilidade com uma ampla gama de dispositivos móveis, incluindo aparelhos com sistemas Android e iOS, que apresentam diversos tamanhos e proporções de tela.

Durante o processo de desenvolvimento, foram realizados testes em dispositivos específicos para avaliar o desempenho da aplicação em diferentes cenários. Os testes abrangeram dois aparelhos Android da Samsung, o Galaxy S10+ e o Galaxy S23 Ultra, bem como um dispositivo iOS, o iPhone 13. Em todas essas plataformas, a aplicação se adaptou de forma adequada, respeitando as margens e proporções de cada dispositivo, proporcionando uma experiência consistente e satisfatória aos usuários.

Durante os testes com os dispositivos da Samsung, foi possível utilizar ferramentas nativas do sistema operacional de leitura de tela e de inversão de cores. A primeira, denominada *TalkBack*, funcionou conforme esperado para descrever, via áudio, os textos, imagens, ícones, botões e demais componentes da aplicação e informar qual a ação que deverá ser feita para interagir com determinado item. A segunda, serve para inverter os esquemas de cores do dispositivo, ou seja, os tons escuros se tornam claros e os tons claros se tornam escuros, tornando o conteúdo mais legível e acessível para pessoas com deficiências visuais ou sensitivas.

3.3 TESTE DE UI/UX

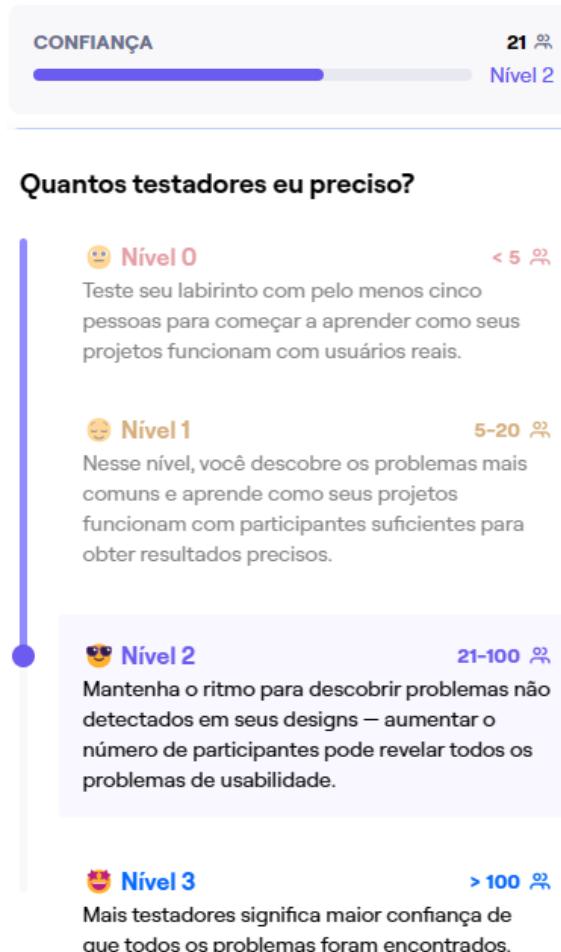
Para medir a taxa de satisfação dos usuários, com base em algumas das métricas mencionadas anteriormente, foi utilizada a plataforma Maze.co⁸. O Maze.co é uma plataforma online de design e pesquisa de experiência do usuário que permite criar e executar testes de usabilidade interativos. Com planos gratuitos e pagos, os designers podem conectar seus protótipos interativos do Figma e coletar feedback e métricas sobre a usabilidade e a experiência dos usuários. Ele oferece recursos como testes remotos, questionários, análise de métricas e compartilhamento de resultados. É uma ferramenta popular entre profissionais de UX e equipes de design para melhorar a usabilidade e a experiência do usuário em seus produtos digitais.

De acordo com o Nielsen Group, liderado pelo renomado especialista em usabilidade Jakob Nielsen, quando se trata de testes de usabilidade e pesquisa de experiência do usuário, a confiabilidade dos resultados geralmente aumenta à medida que o número de testadores aumenta. O número ideal de testadores pode variar dependendo do contexto e dos objetivos específicos do teste, mas estudos mostraram que mesmo com um pequeno número de testadores, é possível identificar a maioria dos problemas de usabilidade (NIELSEN, 1994).

⁸ Disponível em: <<https://maze.co/>>. Acesso em: 20 maio 2023.

Além disso, o Nielsen Group recomenda que para obter insights confiáveis, geralmente são necessários cerca de 5 a 8 testadores. Eles afirmam que a cada novo testador, é provável que novos problemas sejam descobertos, mas a taxa de descoberta diminui à medida que mais testadores são adicionados. Com cerca de 5 testadores, provavelmente serão identificados a maioria dos problemas graves de usabilidade, enquanto adicionar mais testadores ajudará a descobrir problemas mais sutis (NIELSEN, 1994). No Maze.co, essa taxa de confiabilidade é medida em quatro níveis, no qual quanto maior a quantidade de testadores, maior o nível e maior a confiabilidade. Por fim, os níveis podem ser visualizados na imagem abaixo, com o nível dois destacado, como sendo o atingido durante o teste da aplicação com 21 pessoas.

Figura 12 - Níveis de confiabilidade do Maze.co.



Fonte: Autoria própria.

Dessa forma, é possível analisar os resultados do teste, concluir quais os pontos que foram considerados positivos na aplicação e identificar melhorias que poderão ser ajustadas futuramente.

4. VALIDAÇÃO E RESULTADOS

O Maze.co foi utilizado em sua versão gratuita para realizar o teste de usabilidade com vinte e um usuários voluntários, na qual consistia em um limite de dez telas disponíveis para implementação do teste.

A primeira tela foi utilizada como “boas vindas”, apresentando a ideia da aplicação e informando instruções necessárias para completar o teste. Da segunda à sexta tela, foram feitas as tarefas a serem realizadas pelos usuários. Da sétima à nona tela, foram feitas avaliações com nota para coletar satisfação dos testadores. Por fim, na décima tela, um campo opcional de avaliação aberto para sugestões de melhorias ou elogios.

Detalhando cada etapa do teste e iniciando pelas tarefas, foram solicitadas atividades aos usuários que envolviam algumas das funcionalidades mais importantes do sistema, e que possuíssem fluxos completamente distintos, para fornecer uma visão abrangente de sua usabilidade. As tarefas foram:

- Fazer uma reserva em um estabelecimento;
- Fazer check-in em um estabelecimento e realize um pedido;
- Resgatar um prêmio de fidelidade;
- Mudar o tema da aplicação para o tema claro;
- Escrever uma avaliação de estabelecimento que já foi visitado.

As tarefas coletam informações como taxa de sucesso, taxa de missão inacabada, taxa de cliques incorretos, duração média, taxa de usabilidade, mapa de calor de toques na tela e geram relatórios como “telas para refazer”, “telas para ajustar” e “telas legais”. Como a etapa das tarefas era o primeiro contato dos testadores com a aplicação, alguns obtiveram um tempo mais elevado do que outros para completar algumas das atividades, a taxa de cliques errados e duração média foi elevada, mas foi registrado que a maioria deles, pelo menos 18 dos 21 testadores, conseguiram terminar todas elas.

Ao analisar isoladamente as métricas de cada tarefa, foi percebido uma pequena dificuldade nas tarefas 1 e 5. A primeira era relativa ao botão de “Abrir rotas” que atraía mais a atenção do que o link de “Ver mais informações”, o qual deveria ser utilizado na atividade. Já a quinta, era a tarefa mais complicada, dado que na aplicação, é possível avaliar um estabelecimento de duas formas, via checkout de um estabelecimento e pelo histórico após visitar um estabelecimento, mas a primeira alternativa tinha um fluxo parecido com o de outra tarefa, sendo assim, foi escolhida a segunda alternativa. Dessa forma, alguns usuários não

associaram o requisitado na tarefa ao histórico de visitas, com alguns deles ainda não tendo descoberto a tela do histórico, e finalizaram pelo método do checkout do estabelecimento.

Dando sequência, é possível coletar os resultados das avaliações dos usuários de acordo com algumas métricas de UI/UX para medir a usabilidade e acessibilidade da aplicação. Novamente, por utilizar o plano gratuito do Maze.co, foi possível realizar apenas três avaliações, mas dando a liberdade de adicionar observações na última tela para complementar a avaliação dos usuários. As métricas avaliadas foram legibilidade dos textos, qualidade navegação e satisfação geral com a aplicação, medidas numa escala de 0 a 7, de acordo com a Escala Likert.

Os testes de usabilidade frequentemente utilizam escalas de 0 a 7 para medir a satisfação dos usuários e obter feedback quantitativo sobre a experiência do usuário. Essas escalas são conhecidas como escalas Likert de sete pontos e são amplamente utilizadas devido à sua facilidade de uso, capacidade de capturar nuances nas respostas dos usuários e permitir análises quantitativas e comparativas dos resultados obtidos (HARTSON; PYLA, 2012).

Sendo assim, os resultados finais das avaliações dos testadores de acordo com as métricas previamente mencionadas podem ser visualizadas abaixo, com a quantidade de respostas e média no canto superior direito:

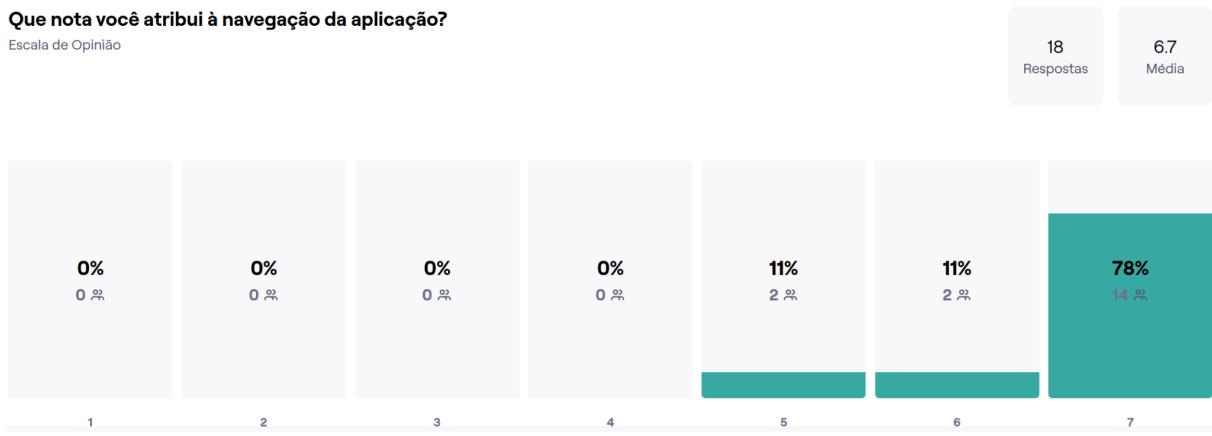
Figura 13 - Avaliação de legibilidade dos textos no Maze.co.



Fonte: Autoria própria.

Assim, a legibilidade de textos possui média 6.9/7 baseada em 18 respostas, representados por 89% dos usuários que atribuíram nota 7 e 11% com nota 6.

Figura 14 - Avaliação de navegação da aplicação no Maze.co.



Fonte: Autoria própria.

Assim, a navegação da aplicação possui média 6.7/7 baseada em 18 respostas, representados por 78% dos usuários que atribuíram nota 7, 11% com nota 6 e 11% com nota 5.

Figura 15 - Avaliação de satisfação geral da aplicação no Maze.co.



Fonte: Autoria própria.

Assim, a satisfação geral da aplicação possui média 6.9/7 baseada em 18 respostas, representados por 89% dos usuários que atribuíram nota 7 e 11% com nota 6.

Por fim, na última tela do teste, é possível visualizar os comentários realizados pelos usuários sobre sugestões ou melhorias a respeito da aplicação. Envolvendo elogios e melhorias, os comentários podem ser visualizados na imagem abaixo:

Figura 16 - Comentários dos usuários sobre a aplicação no Maze.co.

(Opcional) Você tem alguma sugestão de melhoria ou comentário sobre o app?

Open Question

18 Responses

QUOTES

"Talvez melhorar o input do botão de avaliação, pois não sei se ícone de lápis é melhor pra representar a ação"

Tester #164686220 · May 20th 2023, 3:54:21 pm

"Ta top"

Tester #164025717 · May 17th 2023, 2:09:02 pm

"amei"

Tester #163466384 · May 15th 2023, 10:06:10 pm

"Fiquei um tempão sem descobrir a parte da fidelidade, seria interessante adicionar um textinho informando onde fica cada campo. Tipo, mudar a cor do app, obviamente fica nas configurações do perfil. Mas, o lance da fidelidade e a avaliação tbm não achei tão intuitivo, tava procurando em histórico e nas outras abas. Mas tá muito massa msm!! "

Tester #163445349 · May 15th 2023, 8:11:16 pm

"Um pouco confuso para chegar na avaliação do restaurante "

Tester #163411086 · May 15th 2023, 5:56:06 pm

"Top"

Tester #163409335 · May 15th 2023, 5:38:53 pm

Fonte: Autoria própria.

Dessa forma, algumas das dificuldades demonstradas pelas métricas das tarefas foram confirmadas, principalmente no caso da tarefa 5, que pedia para avaliar um estabelecimento já visitado, representadas pelo quarto e quinto comentários. Como já mencionado, a tarefa 5 era realmente a mais difícil e pedia um fluxo menos intuitivo dos disponíveis para avaliar um estabelecimento no aplicativo, provocando a dificuldade demonstrada pelo teste.

5. CONCLUSÃO

Este trabalho teve como foco o desenvolvimento e a descrição detalhada da aplicação Placy. Desde a fase inicial de design e prototipação até a implementação do front-end, todos os aspectos foram abordados com precisão e dedicação.

Ao longo deste estudo, foram abordados conceitos fundamentais de usabilidade e acessibilidade, juntamente com suas métricas relevantes. A compreensão desses conceitos é essencial para o desenvolvimento de aplicações eficientes e acessíveis aos usuários.

No contexto do design e da prototipação, foram explorados conceitos-chave que contribuem para a criação de interfaces intuitivas e agradáveis. A utilização desses conceitos ao longo do desenvolvimento da aplicação permitiu a elaboração de protótipos eficientes, que foram refinados por meio de revisões e iterações, resultando em uma interface finalizada alinhada com os objetivos propostos.

Durante a implementação front-end, foram empregadas ferramentas adequadas, que desempenharam um papel crucial no processo de desenvolvimento. Essas ferramentas facilitaram a criação de uma arquitetura flexível e a otimização do desempenho da aplicação. Além disso, foram explorados conceitos e práticas relacionados ao React, como componentização e otimização, que permitiram o desenvolvimento de uma aplicação de alta qualidade.

A proposição de uma arquitetura sólida foi um dos pilares deste trabalho. Detalhes relevantes sobre o desenvolvimento da aplicação foram apresentados, incluindo tecnologias utilizadas, estrutura do código e integrações realizadas. Além disso, foram mencionadas ferramentas adicionais que facilitaram e que contribuíram para a obtenção de uma aplicação bem estruturada, funcional e acessível.

Um teste de usabilidade foi conduzido com usuários, fornecendo informações valiosas para a avaliação da aplicação. Os procedimentos mais apropriados adotados por testes de UI/UX foram utilizados, como tarefas e avaliações. Os feedbacks dos usuários desempenharam um papel significativo nas decisões de design e implementação, possibilitando aprimoramentos e refinamentos adicionais.

Em resumo, este trabalho de conclusão de curso cumpriu seus objetivos, fornecendo uma compreensão aprofundada dos conceitos de usabilidade, acessibilidade, design e prototipação. Além disso, apresentou uma arquitetura sólida, detalhes do desenvolvimento da aplicação e resultados de um teste de usabilidade com usuários. As contribuições deste trabalho são relevantes para a área de usabilidade, acessibilidade e desenvolvimento de

aplicações para dispositivos móveis, fornecendo um fluxo de desenvolvimento de um aplicativo para futuras pesquisas.

5.1 TRABALHOS FUTUROS

Após analisar o teste de usabilidade, é possível identificar pontos positivos e negativos existentes na aplicação. No que se diz respeito a ajustes de funcionalidades e designs existentes, é importante rever os pontos que devam atrair mais atenção na aplicação, para minimizar a taxa de cliques errados ou caminhos incorretos para realizar alguma ação.

Além disso, é possível a criação do design e protótipo dos estabelecimentos, visto que este trabalho aborda o lado da versão cliente do aplicativo.

Em relação a funcionalidades que possam ser adicionadas na aplicação, algumas das opções são: pagamento pelo próprio aplicativo, sistema de delivery de produtos, sistema de recomendações baseado nos interesses e histórico e uma área social, permitindo que usuários visualizem avaliações e recomendações de outros usuários pelo perfil deles.

6. REFERÊNCIAS

Get In - Descubra novos restaurantes, entre na fila e faça reservas online. Disponível em: <<http://www.getinapp.com.br>>. Acesso em: 01 março 2023.

TRINKS.COM, P. PARA SALÃO DE B. -. **Programa para salão de Beleza** - Trinks.com. Disponível em: <<https://www.trinks.com/>>. Acesso em: 1 março 2023.

iFood. Disponível em: <<https://www.ifood.com.br>>. Acesso em: 1 março 2023.

CYBIS, W.; BETIOL, A.; FAUST, R. Ergonomia e Usabilidade: Conhecimentos, Métodos e Aplicações. 2. ed. São Paulo: Novatec Editora, 2010.

DIAS, Cláudia. Usabilidade na Web: criando portais mais acessíveis. 2. ed. Rio de Janeiro: Alta Books, 2007. 296 p.

ISO/IEC 9126. Software product evaluation: Quality characteristics and guidelines for their use. 1991.

ISO 9241 Parte 11. Requisitos Ergonômicos para Trabalhos de Escritórios com Computador. Parte 11: Orientações sobre Usabilidade, 1998.

Color Contrast Ratio Calculator & Checker. Disponível em: <<https://contrastchecker.online/>>. Acesso em: 24 maio. 2023.

NIELSEN, J. Usability Engineering. Amsterdam: Morgan Kaufmann, 1994.

TULLIS, T., & ALBERT, B. Measuring the user experience: Collecting, analyzing, and presenting usability metrics. Morgan Kaufmann, 2013.

SAURO, J. A Practical Guide to the System Usability Scale: Background, Benchmarks & Best Practices. Measuring Usability LLC, 2011.

BROOKE, J. SUS: A quick and dirty usability scale. In Jordan, P.W., Thomas, B., Weerdmeester, B.A., & McClelland, I.L. (Eds.), Usability Evaluation in Industry. Taylor & Francis, 1996.

W3C. Introduction to Web Accessibility. Disponível em: <<https://www.w3.org/WAI/fundamentals/accessibility-intro/>>. Acesso em: 10 março 2023.

GUBE, J. What Is User Experience Design? Overview, Tools And Resources — Smashing Magazine. Disponível em: <<https://www.smashingmagazine.com/2010/10/what-is-user-experience-design-overview-tools-and-resources/>>. Acesso em: 10 março 2023.

FRIEDMAN, V. Responsive Web Design – What It Is And How To Use It — Smashing Magazine. Disponível em: <<https://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/>>. Acesso em: 11 março 2023.

FIGMA. About Figma, the collaborative interface design tool. Disponível em: <<https://www.figma.com/about/>>. Acesso em: 12 março 2023.

“What is a Design System?” an article by Dan Mall. Disponível em: <<https://danmall.com/posts/what-is-a-design-system/>>. Acesso em: 12 março 2023.

LANCASTER, A. Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces. IEEE Transactions on Professional Communication, v. 47, n. 4, p. 335–336, dez. 2004. Acesso em: 12 março 2023.

SOMMERVILLE, I. Engenharia de Software. 10. ed. São Paulo: Pearson, 2011.

Stack Overflow Developer Survey 2021. Disponível em: <<https://insights.stackoverflow.com/survey/2021#most-popular-web-frameworks/>>. Acesso em: 14 março 2023.

META OPEN SOURCE. React. Disponível em: <<https://react.dev/>>. Acesso em: 12 março 2023.

MICROSOFT. TypeScript Documentation. Disponível em: <<https://www.typescriptlang.org/docs/>>. Acesso em: 12 março 2023.

REACT NATIVE. Getting Started. Disponível em: <<https://reactnative.dev/docs/getting-started/>>. Acesso em: 12 março 2023.

NativeBase: Universal Components for React & React Native. Disponível em: <<https://nativebase.io/>>. Acesso em: 13 março 2023.

EXPO. Expo Documentation. Disponível em: <<https://docs.expo.dev/>>. Acesso em: 13 março 2023.

REACT. Your First Component. Disponível em: <https://react.dev/learn/your-first-component>. Acesso em: 14 março 2023.

REACT. Optimizing Performance. Disponível em: <<https://legacy.reactjs.org/docs/optimizing-performance.html>>. Acesso em: 14 março 2023.

HARTSON, Rex; PYLA, Pardha. The UX Book: Process and Guidelines for Ensuring a Quality User Experience. San Francisco: Morgan Kaufmann, 2012.