

Plantão PROCON:
Defendendo seus direitos a qualquer hora, em
qualquer lugar.

Vinícius dos Santos Silveira



CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, PB

2024

Vinícius dos Santos Silveira

Plantão PROCON

Defendendo seus direitos a qualquer hora, em qualquer lugar.

Relatório Técnico apresentado ao curso Ciência de Dados e Inteligência Artificial do Centro de Informática, da Universidade Federal da Paraíba, como requisito para a obtenção do grau de Bacharel em Ciência de Dados e Inteligência Artificial

Orientador: Yuri Malheiros de Almeida Barbosa

Maio de 2024

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

S587p Silveira, Vinicius Dos Santos.

Plantão PROCON: defendendo seus direitos a qualquer hora, em qualquer lugar. / Vinicius Dos Santos Silveira. - João Pessoa, 2024.

51 f. : il.

Orientação: Thaís Rêgo.
TCC (Graduação) - UFPB/CI.

1. Desenvolvimento web. 2. ReactJS. 3. NodeJS. 4. PROCON. 5. Front-end. I. Rêgo, Thaís. II. Título.

UFPB/CI

CDU 004.774



CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

Trabalho de Conclusão de Curso de Ciência de Dados e Inteligência Artificial intitulado ***Plantão PROCON*** de autoria de Vinícius dos Santos Silveira, aprovada pela banca examinadora constituída pelos seguintes professores:

Prof. Dr. Yuri Malheiros de Almeida Barbosa
Universidade Federal da Paraíba

Prof. Dr. Thaís Gaudencio do Rêgo
Universidade Federal da Paraíba

Me. Lucas Oliveira Costa Aversari

João Pessoa, 1 de maio de 2024

“Não é a quantidade de horas que você coloca no trabalho que importa, é a quantidade de trabalho que você coloca nessas horas.” - Sam Ewing

AGRADECIMENTOS

Primeiramente, expresso minha profunda gratidão a Deus por Sua constante orientação e bênçãos ao longo desta jornada. Em momentos de indecisão e dificuldade, Sua presença foi uma luz orientadora, guiando-me pelos caminhos certos e fortalecendo-me para enfrentar os desafios mais complexos. Reconheço que sem Sua orientação e proteção, nada disso seria possível.

Sou profundamente grato à minha família, cujo apoio inabalável e encorajamento constante foram fundamentais em cada passo da minha jornada. Mesmo estando distantes em alguns momentos, o seu amor e suporte nunca vacilaram. Cada conquista alcançada é uma homenagem ao amor e à dedicação que eles sempre demonstraram por mim.

Quero também expressar minha profunda gratidão à segunda família que ganhei ao longo dessa jornada: minha família universitária. Cada amizade desempenhou um papel vital em meu crescimento pessoal e profissional. As horas passadas juntos em sala de aula, nos laboratórios e nos projetos extracurriculares foram momentos preciosos de aprendizado e camaradagem. Cada desafio enfrentado e cada conquista alcançada foram compartilhados em conjunto, fortalecendo nossos laços e enriquecendo nossa jornada acadêmica. A confiança e a inspiração mútua que encontramos uns nos outros foram fundamentais para superar obstáculos e alcançar nossos objetivos.

Foram anos de companheirismo, marcados por muitas risadas e pela formação de novas amizades, algumas das quais se distanciaram, mas tenho certeza de que aquelas que permaneceram serão para sempre. Agradeço a todos que fizeram parte desta jornada e desejo que nossa amizade perdure eternamente. Podem contar comigo para tudo o que precisarem. Amo vocês. Família NA Data Science.

Por último, mas não menos importante, expresso minha profunda gratidão à minha namorada, que desempenhou um papel fundamental em minha jornada acadêmica. Além de ser minha companheira, ela foi uma fonte constante de apoio, estímulo e motivação. Com sua dedicação e incentivo incessantes, ela esteve ao meu lado, empurrando-me para alcançar meu potencial máximo e superar todos os obstáculos que surgiram em meu caminho durante o curso. Sua presença constante, amor incondicional e compromisso com o meu sucesso foram um verdadeiro catalisador para minha conclusão do curso. Sou profundamente grato por todo o apoio e amor que ela me dedicou ao longo desses anos.

RESUMO

Com um firme compromisso em aprimorar a interação entre consumidores e o PROCON, este trabalho busca transformar a maneira como essas conexões acontecem. Utilizando tecnologias de ponta, como ReactJS para interação com o usuário, PostgreSQL para armazenamento seguro de dados e NodeJS para processamento eficiente, a aplicação se destaca por sua robustez e eficácia. Inspirada nas melhores práticas empresariais e na necessidade de proporcionar uma experiência mais ágil e eficiente aos consumidores, a solução foi submetida a um rigoroso plano de testes, garantindo usabilidade, segurança e confiabilidade em sua operação.

Palavras-chave: <Desenvolvimento web>, <ReactJS>, <NodeJS>, <PROCON>, <Front-end>.

ABSTRACT

This work focuses on the implementation of an application dedicated to optimizing communication between consumers and PROCON, seeking to revolutionize the way these interactions are carried out. Using cutting-edge technologies such as ReactJS for user interaction, PostgreSQL for secure data storage and NodeJS for efficient processing, the application stands out for its robustness and effectiveness. Inspired by best business practices and the need to provide a more agile and efficient experience for consumers, the solution was subjected to a rigorous testing plan, ensuring usability, security and reliability in its operation.

Key-words: <Web development>, <ReactJS>, <NodeJS>, <PROCON>, <Security>.

LISTA DE FIGURAS

| | | |
|----|---|----|
| 1 | Sintaxe do TailwindCSS | 21 |
| 2 | Diferença entre uma SPA e uma MPA | 25 |
| 3 | Arquitetura MVC | 26 |
| 4 | Mapa de navegação do Plantão PROCON | 33 |
| 5 | Tela de login | 34 |
| 6 | Tela de cadastro da aplicação | 35 |
| 7 | Fluxo de recuperação de senha | 36 |
| 8 | Tela principal da aplicação | 37 |
| 9 | Menu de opções do usuário | 38 |
| 10 | Modal de espera da solicitação | 39 |
| 11 | Modal de edição do perfil | 39 |

LISTA DE TABELAS

| | | |
|----|------------------|----|
| 1 | Caso de teste 01 | 41 |
| 2 | Caso de teste 02 | 42 |
| 3 | Caso de teste 03 | 42 |
| 4 | Caso de teste 04 | 43 |
| 5 | Caso de teste 05 | 44 |
| 6 | Caso de teste 06 | 45 |
| 7 | Caso de teste 07 | 45 |
| 8 | Caso de teste 08 | 46 |
| 9 | Caso de teste 09 | 46 |
| 10 | Caso de teste 10 | 47 |
| 11 | Caso de teste 11 | 47 |

LISTA DE ABREVIATURAS

AJAX Synchronous JavaScript and XML

CT Caso de teste

CSS Cascading Style Sheets

DOM Document Object Model

HTML Hypertext Markup Language

JS JavaScript

MVC Model-View-Controller

PROCON Procuradoria de Proteção a Defesa do Consumidor

SEO Search Engine Optimization

SPA Single Page Application

TS TypeScript

Sumário

| | |
|--|-----------|
| 1 INTRODUÇÃO | 15 |
| 1.1 Objetivo geral | 16 |
| 1.2 Objetivos específicos | 16 |
| 1.3 Estrutura do relatório técnico | 16 |
| 2 CONCEITOS GERAIS | 18 |
| 2.1 JavaScript | 18 |
| 2.2 HTML | 19 |
| 2.3 CSS | 19 |
| 2.4 TailwindCSS | 20 |
| 2.5 <i>Front-end</i> | 21 |
| 2.6 ReactJS | 22 |
| 2.7 Vite | 23 |
| 2.8 Single Page Application (SPA) | 24 |
| 2.9 MVC | 25 |
| 3 METODOLOGIA | 26 |
| 3.1 Visão geral | 26 |
| 3.2 Usuários | 27 |
| 3.3 Hospedagem | 27 |
| 3.3.1 <i>Front-end</i> | 28 |
| 3.3.2 <i>Back-end</i> | 28 |
| 3.4 Requisitos Funcionais | 28 |
| 3.4.1 [RF01] <i>Login</i> | 28 |
| 3.4.2 [RF02] Cadastro | 28 |
| 3.4.3 [RF03] Recuperação de senha | 29 |
| 3.4.4 [RF04] <i>Logout</i> | 29 |
| 3.4.5 [RF05] Edição de perfil | 29 |
| 3.4.6 [RF06] Alteração do tema | 29 |

| | |
|--|-----------|
| 3.4.7 [RF07] Upload de foto | 29 |
| 3.4.8 [RF08] Ver listagem de consultores | 29 |
| 3.4.9 [RF09] Ver status do consultor de forma síncrona | 30 |
| 3.4.10 [RF10] Entrada e saída da fila de espera a qualquer momento | 30 |
| 3.4.11 [RF11] Alteração de disponibilidade do consultor | 30 |
| 3.4.12 [RF12] Comunicação entre consultor e consumidor via conferência | 30 |
| 3.5 Requisitos não funcionais | 30 |
| 3.5.1 [RNF01] Responsividade | 30 |
| 3.5.2 [RNF02] Acessibilidade | 30 |
| 3.5.3 [RNF03] Segurança | 30 |
| 3.5.4 [RNF04] Idioma | 31 |
| 3.5.5 [RNF05] Ambientes e ferramentas de implementação | 31 |
| 3.5.6 [RNF06] Conexão à internet | 31 |
| 3.5.7 [RNF07] Compatibilidade | 31 |
| 3.6 Arquitetura | 31 |
| 3.7 Interface da aplicação | 32 |
| 3.7.1 Mapa de navegação | 32 |
| 3.8 Telas | 33 |
| 3.8.1 <i>Login</i> | 33 |
| 3.8.2 <i>Cadastro</i> | 34 |
| 3.8.3 <i>Recuperação de senha</i> | 35 |
| 3.8.4 <i>Home</i> | 36 |
| 3.8.5 Menu de opções do usuário | 37 |
| 3.8.6 Fila de espera | 38 |
| 3.8.7 Edição de perfil | 39 |
| 4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS | 40 |
| 4.1 Plano de testes | 40 |
| 4.2 Personas | 40 |
| 4.2.1 Persona A | 40 |

| | |
|---|----|
| 4.2.2 Persona B | 40 |
| 4.3 Casos de teste | 40 |
| 4.3.1 [CT01] Verificar se um usuário pode se autenticar na plataforma através do processo de <i>login</i> | 41 |
| 4.3.2 [CT02] Verificar se um usuário não pode fazer login com credenciais inválidas. | 42 |
| 4.3.3 [CT03] Verificar se o sistema permite a criação de novos usuários solicitantes através do processo de cadastro | 42 |
| 4.3.4 [CT04] Verificar se o sistema impede o cadastro de um usuário com um e-mail ou CPF já existente na plataforma. | 43 |
| 4.3.5 [CT05] Verificar se o usuário pode recuperar sua senha seguindo todas as etapas do processo. | 44 |
| 4.3.6 [CT06] Verificar se o sistema não permite a recuperação de senha para um e-mail não cadastrado. | 45 |
| 4.3.7 [CT07] Verificar se o usuário é desconectado da plataforma após fazer <i>logout</i> | 45 |
| 4.3.8 [CT08] Verificar se um usuário pode editar seu perfil com informações válidas. | 46 |
| 4.3.9 [CT09] Verificar se o usuário pode alterar o tema da plataforma para uma das opções disponíveis. | 46 |
| 4.3.10 [CT10] Verificar se o usuário consegue entrar na fila de espera. | 47 |
| 4.3.11 [CT11] Verificar se o usuário pode sair da fila de espera. | 47 |
| 4.4 Discussão dos resultados | 47 |
| 5 CONCLUSÕES E TRABALHOS FUTUROS | 49 |
| REFERÊNCIA | 50 |

1 INTRODUÇÃO

O atual cenário de atendimento apresenta desafios significativos que impactam diretamente a eficácia do serviço e a comunicação entre o PROCON e os consumidores. O problema central reside na ausência de um canal síncrono de teleconsultoria, dificultando a resolução imediata de questões. A falta desse canal impede que os consumidores recebam atendimento em tempo real, gerando demoras e insatisfação. Esta lacuna na comunicação também afeta a disponibilidade do serviço para a população em geral, limitando o acesso a soluções eficientes.

Compreendendo os desafios enfrentados no atual cenário de atendimento do PROCON, foi proposta a implementação de um sistema que não só endereça as dificuldades mencionadas, mas também visa aprimorar a eficiência do serviço. Concebido para ultrapassar as barreiras existentes na comunicação entre consumidores e o órgão, o sistema oferece uma plataforma integrada que possibilita uma interação direta e imediata. Ao garantir uma resposta ágil às demandas dos consumidores e facilitar a resolução de questões em tempo real, espera-se que o sistema não apenas otimize o atendimento prestado pelo PROCON, mas também aprimore significativamente a experiência do usuário. Destacando-se pela eficácia na teleconsultoria, o sistema permite a resolução remota de denúncias. Sua presença em totens urbanos estrategicamente localizados, juntamente com a acessibilidade via plataformas online e aplicativos móveis, oferece flexibilidade aos usuários. Além disso, a gestão inteligente de filas, baseada em algoritmos, prioriza atendimentos conforme a urgência, contribuindo para uma experiência mais rápida e eficiente. Espera-se que a implementação desse sistema resulte em uma redução significativa no tempo de espera, enquanto aumenta a acessibilidade aos serviços. Com uma interface intuitiva, busca-se proporcionar uma experiência positiva durante a teleconsultoria, visando à satisfação do usuário.

A falta da presença física do PROCON em locais específicos representa uma barreira à facilidade de acesso e registro de ocorrências por parte dos cidadãos. A necessidade de deslocamento até um ponto de atendimento dificulta a efetividade do serviço, especialmente considerando as demandas de uma sociedade cada vez mais dinâmica. Além disso, a comunicação entre o PROCON e o consumidor pode ser aprimorada. Atualmente, a falta de uma plataforma facilitadora e intuitiva dificulta o processo de registro de ocorrências. Uma solução que permita uma comunicação mais direta e eficiente entre o PROCON e o consumidor se faz necessária para superar esses desafios. Em resumo, a ausência de um canal síncrono de teleconsultoria, a limitada disponibilidade do serviço para a população, a falta da presença física do PROCON e a comunicação pouco eficiente entre PROCON e consumidor são fatores cruciais que demandam soluções inovadoras para aprimorar o sistema de atendimento e fortalecer a relação entre as partes envolvidas.

1.1 Objetivo geral

O objetivo geral deste projeto é desenvolver uma inovadora solução de teleconsultoria, projetada para ser implementada em totens distribuídos estrategicamente pela cidade. O intuito principal é facilitar o acesso da população aos serviços de consultoria e denúncia do PROCON. A iniciativa busca promover uma interação simplificada e eficiente entre os cidadãos e os serviços oferecidos, tornando-os mais acessíveis em locais estratégicos. Ao empregar a tecnologia de totens e também o acesso em dispositivos móveis, o projeto visa democratizar o acesso à informação e aos canais de denúncia, fortalecendo a participação da comunidade na promoção e defesa de seus direitos, contribuindo assim para a melhoria contínua dos serviços oferecidos pelo Procuradoria de Proteção a Defesa do Consumidor (PROCON).

1.2 Objetivos específicos

- Realizar um levantamento abrangente das regras de negócio relacionadas à teleconsultoria do PROCON, culminando na criação de um diagrama de fluxo que visualiza de forma clara e precisa os processos envolvidos.
- Desenvolver o design da solução utilizando a ferramenta Figma, assegurando um layout intuitivo, atraente e funcional. Garantir a usabilidade eficaz e a responsividade do design para proporcionar uma experiência consistente em diversas plataformas e dispositivos.
- Efetuar a implementação da solução, transformando o design concebido em um produto funcional. Utilizar as melhores práticas de desenvolvimento para garantir a eficiência, segurança e desempenho da aplicação.
- Realizar testes abrangentes em todas as fases do desenvolvimento, abordando aspectos funcionais, de usabilidade e de desempenho. Identificar e corrigir quaisquer falhas ou irregularidades para garantir a qualidade final da solução.
- Garantir a compatibilidade multiplataforma da aplicação, certificando-se de que ela seja acessível e funcional em diversas plataformas, como web, dispositivos móveis e totens distribuídos pela cidade.

1.3 Estrutura do relatório técnico

Este trabalho está estruturado em cinco capítulos. No primeiro é feita uma introdução ao tema, bem como exposição do problema, apresentação dos objetivos do trabalho. No segundo capítulo, são apresentados conceitos gerais do trabalho. No terceiro,

a aplicação Plantão PROCON é apresentada e detalhada por meio da especificação dos requisitos e modelagem da arquitetura. No quarto capítulo são apresentados o plano de teste, casos de teste e análise da avaliação do projeto. No último capítulo é apresentada a conclusão do presente trabalho e são propostas melhorias e trabalhos futuros.

2 CONCEITOS GERAIS

Nesta seção, serão abordados conceitos essenciais para compreensão do trabalho proposto. Entre eles, destacam-se JavaScript, HTML, CSS, Front-end, ReactJS, TailwindCSS, Vite, SPA e MVC, abordados nas subseções 2.1, 2.2, 2.3, 2.5, 2.6, 2.4, 2.7, 2.8, 2.9, respectivamente.

Ao explorar esses conceitos, buscamos fornecer uma base sólida para a compreensão dos elementos-chave envolvidos no desenvolvimento do presente projeto. Essa compreensão é crucial para explorar e implementar soluções de forma eficiente e alinhada com as melhores práticas do desenvolvimento web contemporâneo.

2.1 JavaScript

JavaScript (**JS**) foi criado por Brendan Eich em 1995, enquanto trabalhava na Netscape Communications Corporation. Inicialmente, era destinado a ser uma linguagem de script simples para melhorar a interatividade nas páginas web. O **JS** rapidamente ganhou popularidade e, em 1997, a Ecma International padronizou a linguagem, resultando na ECMAScript, que é a base para a implementação do **JS**.

Ao longo dos anos, o **JS** evoluiu para uma linguagem mais poderosa e versátil. O advento do Synchronous JavaScript and XML (**AJAX**) na década de 2000 permitiu que as páginas web fossem atualizadas de forma assíncrona, melhorando significativamente a experiência do usuário. Com o tempo, surgiram bibliotecas e *frameworks*, como jQuery, Angular, Vue.js e React.

Ao longo dos anos, o ecossistema **JS** cresceu enormemente, com a introdução de novas funcionalidades no ECMAScript, como o ES6 (ECMAScript 2015) e versões posteriores. Essas atualizações trouxeram melhorias significativas à linguagem, facilitando a escrita de código mais limpo e eficiente (**BANKS; PORCELLO, 2020**).

A história do **JS** é marcada por um rápido crescimento e evolução, desde seus primórdios como uma linguagem de script simples até se tornar uma parte essencial do desenvolvimento web moderno (**KEITH, 2005**).

Outro aspecto crucial a ser destacado nesta seção é que o **JS** pode ser estendido para TypeScript (**TS**)¹ o qual essencialmente incorpora uma sintaxe de tipagem ao JavaScript. Essa integração é significativa, pois oferece vantagens adicionais em termos de robustez e segurança, ao mesmo tempo que mantém a flexibilidade e poder de desenvolvimento do JavaScript.

Além disso, é importante ressaltar que o desenvolvimento web não se resume ape-

¹<https://www.typescriptlang.org>

nas ao JavaScript. De fato, ele é uma parte indispensável do processo, mas depende intrinsecamente do Hypertext Markup Language ([HTML](#)) para fornecer a estrutura básica da página. Enquanto o JavaScript é responsável pela interatividade e dinamicidade do conteúdo, o [HTML](#) é fundamental para definir a estrutura e o conteúdo estático da página. Assim, essas duas linguagens trabalham em conjunto para criar uma experiência completa e funcional para os usuários na web.

2.2 HTML

[HTML](#)² desempenha um papel fundamental no desenvolvimento de páginas web. Criado em 1991 por Tim Berners-Lee, o [HTML](#) permite a criação de documentos interpretáveis pelos navegadores, proporcionando a estrutura essencial para o conteúdo web.

Em [HTML](#), os elementos representam componentes visuais que compõem uma página, incluindo parágrafos, links e imagens, dando forma à apresentação visual da informação. O principal objetivo do [HTML](#) é facilitar a estruturação e o acesso à informação na internet, fornecendo uma linguagem padronizada para a criação de conteúdo web ([SILVA, 2008](#)).

Além do [HTML](#), outro elemento fundamental no desenvolvimento de páginas web é o Cascading Style Sheets ([CSS](#)). Criado para complementar o [HTML](#), o [CSS](#) é responsável pela apresentação visual das páginas, incluindo o layout, as cores, as fontes e outros aspectos estéticos. Ao separar o conteúdo da apresentação, o [CSS](#) permite que os desenvolvedores controlem de forma mais eficiente o estilo e o design das páginas, proporcionando uma experiência visualmente atraente e consistente para os usuários.

2.3 CSS

[CSS](#)³ é uma linguagem crucial no desenvolvimento web, criada para complementar o [HTML](#) e fornecer estilos e design às páginas. Desenvolvido pelo World Wide Web Consortium (W3C), o [CSS](#) é uma ferramenta para aprimorar a apresentação visual do conteúdo web.

O principal objetivo do CSS, conforme descrito no texto, é capacitar os desenvolvedores e designers a manter suas folhas de estilo de forma organizada e estruturada. Ao entender e aplicar os fundamentos do CSS, como herança, cascata e especificidade, os profissionais podem criar regras de estilo limpas e de fácil leitura. Além disso, o uso de padrões de arquitetura e design ajuda na organização e manutenção das folhas de estilo, enquanto os pré-processadores CSS permitem maximizar a reutilização de código. A separação de preocupações entre layout e estilo simplifica a criação de temas dinâmicos e

²<https://developer.mozilla.org/pt-BR/docs/Web/HTML>

³<https://developer.mozilla.org/pt-BR/docs/Web/CSS>

etiquetas em branco, contribuindo para uma melhor acessibilidade e suporte a tecnologias assistivas. Em última análise, o objetivo é capacitar os profissionais a implementar projetos de maneira mais rápida e fácil, aliviando-os da complexidade das folhas de estilo e promovendo um código mais limpo e elegante.(DOWDEN; DOWDEN, 2020).

Ao permitir a criação de folhas de estilo externas, o CSS promove a reutilização de estilos em várias páginas, garantindo consistência e eficiência no desenvolvimento web. Sua flexibilidade também possibilita a adaptação responsiva, ajustando o layout conforme o dispositivo de visualização.

Em resumo, o CSS desempenha um papel vital na transformação de códigos HTML 2.2 em experiências visuais atrativas. Seu principal objetivo é proporcionar elegância e estilo às páginas web, garantindo uma apresentação visualmente agradável e uma navegação fluida para os usuários.

Com isso, podemos agora citar um dos muitos *frameworks* que fazem o mesmo papel do CSS, lidar com a interface. O TailwindCSS 2.4 é um *framework* que utiliza classes utilitárias para facilitar a escrita e reutilização do código de estilo escrito.

2.4 TailwindCSS

O TailwindCSS é uma ferramenta que facilita a estilização de páginas web. Enquanto no CSS 2.3 tradicional você cria regras específicas para cada estilo no seu arquivo de estilo, o Tailwind segue uma abordagem diferente, oferecendo um conjunto extenso de “classes utilitárias”.

Essas classes utilitárias são como peças prontas para uso imediato. Em vez de escrever CSS 2.3 personalizado, você aplica essas classes diretamente no seu HTML 2.2 para estilizar os elementos. Por exemplo, em vez de criar uma regra CSS 2.3 para definir a cor de um botão, você pode simplesmente adicionar a classe “bg-blue-500” para obter um botão azul. No exemplo ilustrado na Figura 1, é possível observar a sintaxe utilizada ao empregar o TailwindCSS para escrever o código (WATHAN, 2022).

A screenshot of a terminal window titled "campanha [WSL: Ubuntu] - App.tsx". The window contains 15 lines of code in a monospaced font. The code uses TailwindCSS utility classes like "flex", "flex-col", "space-y-6", "items-center", "justify-center", "h-screen", "bg-background", "font-sans", "text-4xl", "animate-spin", and "size-40".

```
1 return (
2   <div className="flex flex-col space-y-6 items-cen
3   ter justify-center h-screen bg-background">
4     <aside
5       id="logo"
6       className="flex items-center flex-col"
7     >
8       <p className="font-sans text-4xl">Plantão</p>
9       <ProconLogo />
10    </aside>
11    <Loader
12      size={40}
13      className="animate-spin"
14    >
15  </div>
);
```

Figura 1: Sintaxe do TailwindCSS

Fonte: Autoria própria

Essa abordagem classes utilitárias torna o desenvolvimento mais rápido e flexível. Você não precisa se preocupar em criar classes no seu CSS[2.3] ou escrever estilos repetitivos. O Tailwind facilita a criação de layouts e estilos de forma ágil, permitindo que você se concentre mais no HTML[2.2] e menos na criação manual de regras de estilo.

Em resumo, o TailwindCSS é uma ferramenta que simplifica o processo de estilização, oferecendo classes utilitárias pré-definidas que você pode aplicar diretamente no seu HTML[2.3]. Isso pode proporcionar uma abordagem mais eficiente e flexível para o design de páginas web.

O TailwindCSS, como mencionado anteriormente, é uma ferramenta valiosa para estilização e pode ser integrado ao HTML e JS, formando os fundamentos essenciais para o *front-end*. Esta camada é responsável não apenas pelo layout, mas também pelo design e pela interatividade da página, garantindo uma experiência envolvente para os usuários.

2.5 *Front-end*

O desenvolvimento *front-end* desempenha um papel fundamental na criação de interfaces de usuário envolventes e responsivas para aplicações web e mobile. Esta área

concentra-se nos elementos visíveis e interativos que os usuários encontram diretamente, visando proporcionar uma experiência de usuário agradável.

O *front-end* é composto por diversas tecnologias essenciais. O [HTML](#) é a linguagem de marcação que estrutura o conteúdo da página, enquanto o [CSS](#) define o estilo e layout, incluindo cores, fontes e posicionamento dos elementos. O [JavaScript](#), uma linguagem de programação, é crucial para tornar as páginas interativas, manipulando o Document Object Model ([DOM](#)) e respondendo a eventos.

Frameworks e bibliotecas são amplamente utilizados no desenvolvimento *front-end*. O React.js, mantido pelo Facebook, é uma biblioteca popular para construção de interfaces interativas, especialmente em aplicações de página única (SPA - *Single Page Application*). O Angular, desenvolvido pelo Google, é um *framework* robusto para aplicações web dinâmicas, e o Vue.js destaca-se por sua simplicidade e flexibilidade.

A responsividade é um aspecto crítico do *front-end*, garantindo que a aplicação se ajuste de maneira adequada a diferentes tamanhos de tela. Além disso, o controle de estado é essencial para manter a consistência e eficiência, sendo gerenciado localmente no *front-end* ou em conjunto com um *back-end*, dependendo da arquitetura da aplicação.

Ferramentas e práticas de testes são empregadas para garantir a qualidade do código *front-end* e identificar possíveis erros. Além disso, a integração eficiente entre o *front-end* e o *back-end* é vital para assegurar a transmissão correta de dados e a exibição adequada na interface do usuário.

Os desenvolvedores *front-end* colaboram estreitamente com designers para implementar layouts e fluxos de usuário bem projetados. Ao compreender e aplicar essas tecnologias, eles contribuem significativamente para a criação de interfaces de usuário intuitivas, atraentes e eficientes em aplicações web e mobile. Uma das muitas tecnologias usadas no *front-end* é o ReactJS⁴ que tem como principal conceito a reatividade.

2.6 ReactJS

O React JS é uma biblioteca [JS](#) amplamente usada para construir interfaces de usuário interativas e dinâmicas em aplicações web. Desenvolvido e mantido pelo Facebook, o React se destaca por sua eficiência e facilidade de desenvolvimento (RAWAT; MAHAJAN, 2020).

Ao contrário do [JS](#) convencional, o React adota uma abordagem de componentes. Isso significa que você divide a interface do usuário em pequenos pedaços autônomos chamados componentes. Cada componente é como um bloco de construção reutilizável que pode conter seu próprio código [JS](#), [HTML](#) e estilos [CSS](#).

⁴<https://react.dev>

O conceito central do React é a reatividade. Quando o estado de um componente muda, o React atualiza automaticamente a interface do usuário para refletir essas mudanças. Isso resulta em atualizações rápidas e eficientes, sem a necessidade de recarregar toda a página.

O JSX (JavaScript XML) é uma extensão do JS^{2.1} frequentemente usada no React. Ele permite que você escreva código HTML^{2.2} diretamente em seus arquivos JS^{2.1}, tornando a criação de componentes mais intuitiva e fácil de entender (GACKENHEIMER, 2015).

Outra característica importante do React é a comunidade ativa e o ecossistema expansivo. Há uma grande variedade de bibliotecas e ferramentas disponíveis que complementam o React e ajudam a abordar diferentes necessidades no desenvolvimento de aplicações web.

Em resumo, o React JS é uma biblioteca JS^{2.1} que simplifica o desenvolvimento de interfaces de usuário dinâmicas e reativas. Ao adotar a abordagem de componentes e reatividade, o React proporciona uma experiência de desenvolvimento eficiente e eficaz para a criação de aplicações web modernas.

Existem ferramentas que, quando usadas em conjunto com o ReactJS, garantem uma maior performance e desempenho. Um exemplo dessas ferramentas é o Vite. O Vite é uma ferramenta de compilação de código moderna e eficiente, projetada especificamente para oferecer um ambiente de desenvolvimento ágil.

2.7 Vite

O Vite⁵ é uma ferramenta para desenvolver sites e aplicativos web de maneira rápida e eficiente. Ao contrário de algumas ferramentas mais complexas, o Vite é conhecido por sua simplicidade e velocidade durante o desenvolvimento.

Sua principal vantagem está na forma como lida com o código. Em vez de depender de métodos mais antigos, o Vite utiliza algo chamado “ESM” (*ECMA Script Modules*), uma técnica moderna e eficiente para organizar o código. Isso significa que as mudanças feitas durante o desenvolvimento são aplicadas quase instantaneamente, tornando o processo mais rápido.

Se você está construindo um site ou aplicativo, o Vite oferece suporte para diferentes *frameworks*, como Vue.js, React e até mesmo JavaScript puro. Isso proporciona flexibilidade para escolher a tecnologia que melhor se adapta às suas necessidades.

Uma característica interessante do Vite é o “*Hot Module Replacement*” (*HMR*). Isso permite ver as mudanças feitas no código na página sem precisar recarregar tudo.

⁵<https://vitejs.dev/guide/>

Essa funcionalidade torna o desenvolvimento mais eficiente.

Para quem está começando no desenvolvimento web ou busca uma abordagem mais rápida, o Vite simplifica o processo de criação de páginas web e oferece uma experiência ágil, sendo uma ótima opção para aprender a programar ou para projetos que demandam eficiência.

Em resumo, o Vite é uma ferramenta amigável que pode facilitar a criação de sites e aplicativos, tornando o desenvolvimento web mais acessível para uma variedade de usuários.

Além disso, ao desenvolver um aplicativo Single Page Application (**SPA**) com o ReactJS, o Vite pode desempenhar um papel crucial. **SPA**s são aplicativos web que carregam uma única página **HTML** e atualizam dinamicamente a página à medida que o usuário interage com o aplicativo. O Vite, com sua capacidade de compilação rápida e recarregamento instantâneo do navegador, contribui para a criação eficiente e aprimoramento da experiência do usuário em **SPA**s desenvolvidos com o ReactJS.

2.8 Single Page Application (SPA)

Uma **SPA** é um tipo de aplicativo web que funciona em uma única página, em vez de várias páginas diferentes. Em uma SPA, todo o conteúdo é carregado uma vez no navegador, e as interações subsequentes não exigem a recarga completa da página.

Para entender melhor, imagine um site tradicional com várias páginas, onde cada link leva a uma página diferente. Em uma **SPA**, ao clicar em um link, o conteúdo da nova página é carregado dinamicamente na mesma página inicial, sem recarregar todo o site (JADHAV; SAWANT; DESHMUKH, 2015).

Para uma visualização mais clara do funcionamento de uma aplicação de página única **SPA**, uma imagem ilustrativa, como a mostrada na Figura 2, pode ser bastante esclarecedora. Nessa imagem, é possível ver claramente como o conteúdo é carregado dinamicamente na mesma página, proporcionando uma experiência de navegação fluida e sem interrupções.

Isso é possível graças ao uso de tecnologias como JavaScript e *frameworks* como React 2.6, Angular ou Vue.js. Essas tecnologias permitem a atualização dinâmica do conteúdo, sem a necessidade de carregar uma nova página do servidor.

Vantagens de **SPA**s incluem uma experiência de usuário mais fluída, transições mais rápidas entre as seções do aplicativo e a capacidade de criar interfaces mais interativas. No entanto, é importante garantir que a aplicação seja otimizada para desempenho e Search Engine Optimization (**SEO**)⁶, já que o conteúdo não é carregado de uma nova

⁶<https://www.alura.com.br/artigos/seo-o-que-e-boas-praticas-ferramentas-estrategia>

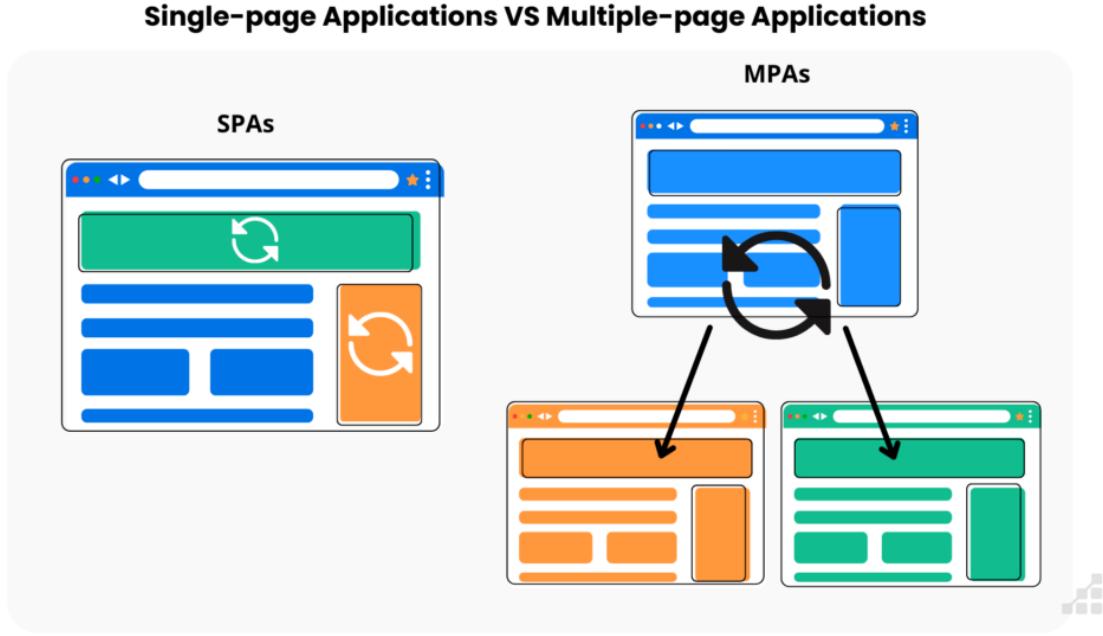


Figura 2: Diferença entre uma SPA e uma MPA

Fonte: <https://www.scalablepath.com/front-end/single-page-applications>

página..

2.9 MVC

O Model-View-Controller (**MVC**) é um modelo de organização utilizado no desenvolvimento de software. Ele divide uma aplicação em três partes principais: o Modelo (responsável pelos dados e regras de negócios), a Visão (que exibe os dados aos usuários) e o Controlador (que atua como intermediário, processando entradas dos usuários e atualizando o Modelo e a Visão conforme necessário). Essa abordagem pode proporcionar uma estrutura clara e organizada, facilitando a manutenção e compreensão do software (**DEACON, 2009**). A figura 3 ilustra o funcionamento da arquitetura **MVC**.

O Modelo (Model) gerencia dados e regras de negócios, a Visualização (View) lida com a interface do usuário, enquanto o Controlador (Controller) atua como intermediário, processando entradas dos usuários e atualizando Modelo e Visão conforme necessário. Essa abordagem estruturada torna o desenvolvimento mais organizado e compreensível, facilitando a manutenção do software.

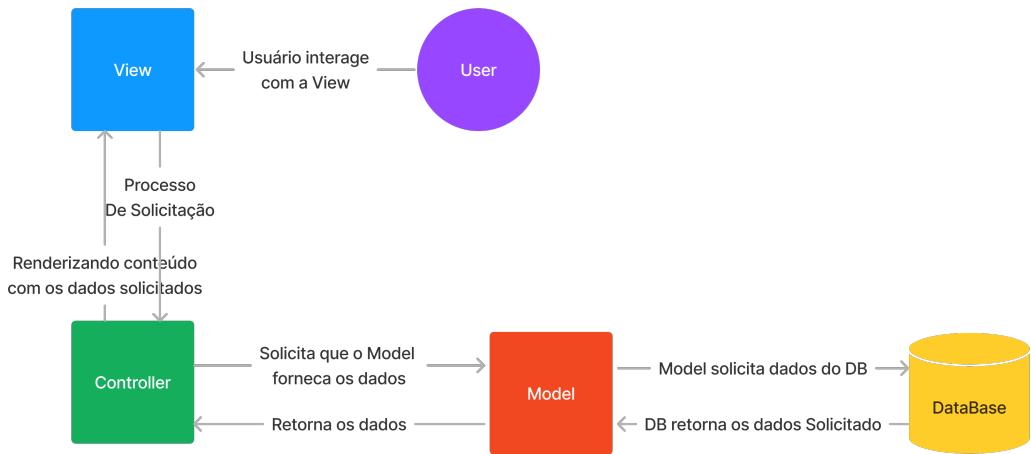


Figura 3: Arquitetura MVC

Fonte: Autoria própria.

3 METODOLOGIA

Este capítulo contém os métodos e procedimentos adotados no desenvolvimento do trabalho, desde uma visão geral do sistema, a descrição dos seus usuários e como aconteceu a sua hospedagem. Por fim, são detalhados os requisitos funcionais e não funcionais, a arquitetura do sistema, sua interface de aplicação e telas.

3.1 Visão geral

O projeto visa desenvolver uma plataforma de teleconsultoria integrada a totens espalhados pela cidade, objetivando facilitar o acesso aos serviços do **PROCON**. Esse sistema permitirá que os usuários, de forma intuitiva e acessível, consultem informações, realizem denúncias e se comuniquem com os teleconsultores para resolverem suas demandas relacionadas ao consumo. A solução busca, portanto, fortalecer os direitos dos consumidores por meio da tecnologia, proporcionando um canal direto e eficiente de comunicação entre o **PROCON** e a população.

Para atingir esses objetivos, o projeto foi estruturado em torno de objetivos específicos planejados, que incluem desde o levantamento das regras de negócio específicas para a teleconsultoria, até a implementação e testes da solução proposta. O design da plataforma, respeitando princípios de usabilidade e responsividade, foi concebido para garantir uma experiência de usuário fluida em diversos dispositivos, promovendo uma interação sem barreiras com os serviços oferecidos.

A arquitetura do projeto foi desenvolvida utilizando ReactJS em conjunto com Vite, seguindo a metodologia **MVC** para organizar o código de maneira eficiente e facilitar a manutenção. Essa escolha tecnológica não só otimiza o desenvolvimento e a performance da aplicação, mas também suporta uma escalabilidade futura. Além disso, o *back-end* da plataforma, hospedado no Oracle utilizando Kubernetes e Postgres, garante robustez, segurança e alta disponibilidade para o sistema.

As funcionalidades do sistema foram projetadas para atender às necessidades dos usuários finais, desde o *login*, até a complexidade do gerenciamento de filas de espera e a edição de perfil. O sistema teve seu *front-end* hospedado na plataforma Vercel⁷, aproveitando o plano gratuito e oferecendo uma solução econômica, sem sacrificar a qualidade ou a performance. Cada tela foi pensada para facilitar o acesso às funcionalidades, com um menu de opções claro e um modal de fila de espera que permite ao usuário a flexibilidade de sair da fila, conforme sua necessidade. Toda a aplicação foi desenhada para ser acessível e disponível em português, assegurando que a barreira do idioma não seja um impedimento para os usuários.

3.2 Usuários

O público-alvo da plataforma abrange todas as pessoas que enfrentam problemas com estabelecimentos, uma vez que o PROCON atua diretamente nas questões relacionadas aos consumidores. Esses usuários, denominados consumidores, buscam uma solução para suas preocupações, desde problemas com produtos ou serviços até questões contratuais.

Além disso, a plataforma visa atender a outra camada de usuários: os teleconsultores. Esses são profissionais de dentro do PROCON que desempenham um papel crucial no auxílio aos consumidores. Agindo de forma síncrona com as denúncias recebidas, os teleconsultores estão prontos para fornecer orientação e assistência direta aos consumidores, contribuindo para uma resolução eficiente e satisfatória dos problemas apresentados.

Dessa forma, a plataforma cria uma ponte de comunicação entre consumidores e teleconsultores, proporcionando um canal direto para a resolução de questões e fortalecendo o papel do PROCON no amparo e proteção dos direitos do consumidor.

3.3 Hospedagem

A hospedagem do *front-end* e *back-end* de um sistema trata da disponibilização e gerenciamento de ambas as partes em um site ou aplicativo.

⁷vercel.com/

3.3.1 *Front-end*

O *front-end* da plataforma foi escolhido para ser hospedado na *Vercel*⁸, uma plataforma de hospedagem e *deploy* projetada para facilitar o processo de disponibilização de sites e aplicações web. A escolha pela *Vercel* foi motivada por vários fatores, incluindo sua facilidade de uso, integração com *frameworks* modernos, como o React JS, e seu desempenho notável.

3.3.2 *Back-end*

O *back-end* da plataforma encontra-se hospedado na infraestrutura da Oracle, beneficiando-se da eficiência e segurança proporcionadas por essa provedora de serviços em nuvem. Utilizando Kubernetes como orquestrador de contêineres, a escalabilidade e a gestão dos serviços são otimizadas, permitindo um ambiente flexível e adaptável às demandas variáveis da aplicação. Além disso, a escolha do Postgres como sistema de gerenciamento de banco de dados oferece confiabilidade e desempenho, assegurando a integridade e eficiência no armazenamento e recuperação de dados. Essa arquitetura na Oracle, respaldada por Kubernetes e Postgres, representa uma estratégia sólida para suportar a infraestrutura do *back-end*, garantindo uma experiência de usuário consistente e confiável para os consumidores e teleconsultores, que utilizam a plataforma Plantão PROCON.

3.4 Requisitos Funcionais

Requisitos de um sistema são declarações claras sobre o que o sistema deve ser capaz de fazer para atender às necessidades e requisitos dos intervenientes.

3.4.1 [RF01] *Login*

O sistema deve permitir que os usuários se autentiquem na plataforma através de um processo de *login*, fornecendo suas credenciais de acesso, como e-mail e senha.

3.4.2 [RF02] *Cadastro*

O sistema deve permitir a criação de novos usuários solicitantes, através de um formulário fornecendo: Primeiro nome, Último nome, CPF, e-mail, Documento profissional, Celular e Senha.

⁸vercel.com/

3.4.3 [RF03] Recuperação de senha

Os usuários devem ter possibilidade de recuperar sua senha através do respectivo fluxo que é composto por três etapas.

- **Primeira etapa:** O usuário deve conseguir fornecer o e-mail cadastrado na plataforma.
- **Segunda etapa:** Nesta etapa, o usuário deverá fornecer o código enviado para o e-mail inserido na etapa anterior.
- **Última etapa:** O usuário deverá conseguir fornecer sua nova senha de acesso ao sistema.

3.4.4 [RF04] Logout

Deve haver uma opção para que os usuários façam *logout* da plataforma, encerrando sua sessão de uso.

3.4.5 [RF05] Edição de perfil

No sistema, deverá ser possível a alteração de alguns dados do usuário como, nome e e-mail.

3.4.6 [RF06] Alteração do tema

O usuário poderá escolher entre as três opções disponíveis: Sistema, Claro, Escuro. A opção **Sistema** deverá utilizar o tema do sistema do usuário.

3.4.7 [RF07] Upload de foto

Os usuários devem ter a capacidade de fazer o *upload* de uma foto para seu perfil na plataforma, proporcionando uma identificação visual adicional.

3.4.8 [RF08] Ver listagem de consultores

Os solicitantes devem poder visualizar uma listagem dos consultores disponíveis na plataforma, incluindo informações como nome, especialidade e status de disponibilidade.

3.4.9 [RF09] Ver status do consultor de forma síncrona

Os solicitantes devem ter acesso ao status de disponibilidade do consultor em tempo real, possibilitando que escolham o momento mais adequado para entrar na fila de espera.

3.4.10 [RF10] Entrada e saída da fila de espera a qualquer momento

Os solicitantes devem ter a flexibilidade de entrar e sair da fila de espera a qualquer momento, permitindo-lhes gerenciar melhor seu tempo de espera.

3.4.11 [RF11] Alteração de disponibilidade do consultor

Os consultores devem ter a capacidade de indicar sua disponibilidade para atender solicitações de consultoria, indicando quando estão disponíveis para receber solicitações.

3.4.12 [RF12] Comunicação entre consultor e consumidor via conferência

realizar conferências diretas na plataforma para uma comunicação eficiente e imediata.

3.5 Requisitos não funcionais

3.5.1 [RNF01] Responsividade

A plataforma deve ser responsiva, adaptando-se a diferentes dispositivos e tamanhos de tela, garantindo uma experiência de usuário consistente e intuitiva em smartphones, tablets e computadores.

3.5.2 [RNF02] Acessibilidade

É fundamental que a plataforma seja acessível a todos os usuários, independentemente de suas habilidades ou deficiências. Deve seguir padrões de acessibilidade da web, garantindo que todos os elementos sejam percebidos, operados e compreendidos por usuários com diferentes necessidades, incluindo aqueles com deficiências visuais, auditivas ou motoras.

3.5.3 [RNF03] Segurança

A segurança da plataforma é uma prioridade e deve ser garantida em todos os níveis. Isso inclui a proteção dos dados dos usuários, como informações de *login* e dados

pessoais, por meio de técnicas de criptografia e medidas de segurança robustas. Além disso, é importante implementar controles de acesso adequados para proteger contra acessos não autorizados e garantir a integridade e confidencialidade das informações armazenadas na plataforma.

3.5.4 [RNF04] Idioma

A aplicação será desenvolvida em português, utilizando uma linguagem clara e de fácil entendimento. Todos os textos e elementos da interface serão elaborados em português, proporcionando uma experiência consistente e acessível para todos os usuários. O objetivo é garantir que a navegação na plataforma seja intuitiva e compreensível, independentemente do nível de familiaridade do usuário com o idioma.

3.5.5 [RNF05] Ambientes e ferramentas de implementação

A aplicação deverá ser desenvolvida utilizando a biblioteca ReactJS [2.6] versão 18 ou superior juntamente com a ferramenta Vite [2.7].

3.5.6 [RNF06] Conexão à internet

A aplicação requer uma conexão ativa com a internet para operar de forma eficaz e garantir todas as suas funcionalidades.

3.5.7 [RNF07] Compatibilidade

A aplicação deverá funcionar nos navegadores *Chrome* versão 108+, *Safari* versão 16+, *Opera* versão 107+ e *Edge* versão 108+.

3.6 Arquitetura

O projeto foi desenvolvido utilizando uma arquitetura **MVC** [2.9], que é um padrão de projeto amplamente adotado para o desenvolvimento de aplicações web (SINGH et al., 2020). Nessa arquitetura, a lógica de negócios da aplicação é dividida em três componentes principais: o Modelo (*Model*), a Visualização (*View*) e o Controlador (*Controller*). Além disso, o projeto utiliza a biblioteca ReactJS [2.6] para a construção da interface do usuário e o Vite, como ferramenta de desenvolvimento para a construção e o empacotamento do aplicativo.

Utilizando o Vite [2.7] como ferramenta de desenvolvimento, o projeto se beneficia de uma configuração simplificada e um tempo de compilação rápido, o que contribui para uma experiência de desenvolvimento ágil e eficiente.

Essa arquitetura proporciona uma organização clara e modular do código-fonte, facilitando a manutenção e o teste da aplicação. Além disso, a combinação do ReactJS [2.6] com o Vite [2.7] oferece uma base sólida e moderna para o desenvolvimento de aplicações web robustas e escaláveis.

3.7 Interface da aplicação

3.7.1 Mapa de navegação

O mapa de navegação (Figura 4) da aplicação inclui diversas telas com funcionalidades distintas para proporcionar uma experiência completa aos usuários. A tela de *login* permite autenticar-se na plataforma e acessar as opções de cadastro de usuários e recuperação de senha. Na tela de Cadastro, os usuários podem registrar-se na plataforma e retornar à tela de *login*. Para recuperar a senha, o processo ocorre em três etapas: na tela de Solicitar Código de Recuperação, o usuário fornece seu e-mail para receber um código de confirmação; na tela de Validar Código de Recuperação, ele insere o código recebido e tem a opção de reenviá-lo, se necessário; finalmente, na tela de Definir Nova Senha, o usuário define uma nova senha e é redirecionado para a tela de *login*. A tela de *Home* apresenta duas visualizações distintas, dependendo do papel do usuário na plataforma. Como solicitante, o usuário visualiza uma lista de consultores disponíveis, podendo entrar na fila de espera para ser atendido. Já como consultor, ele pode alterar seu status e receber solicitações. Ambos os papéis têm acesso a funcionalidades comuns, como edição de perfil, alteração de tema, *logout* e suporte. Essas telas e funcionalidades garantem uma experiência intuitiva e completa aos usuários da plataforma.

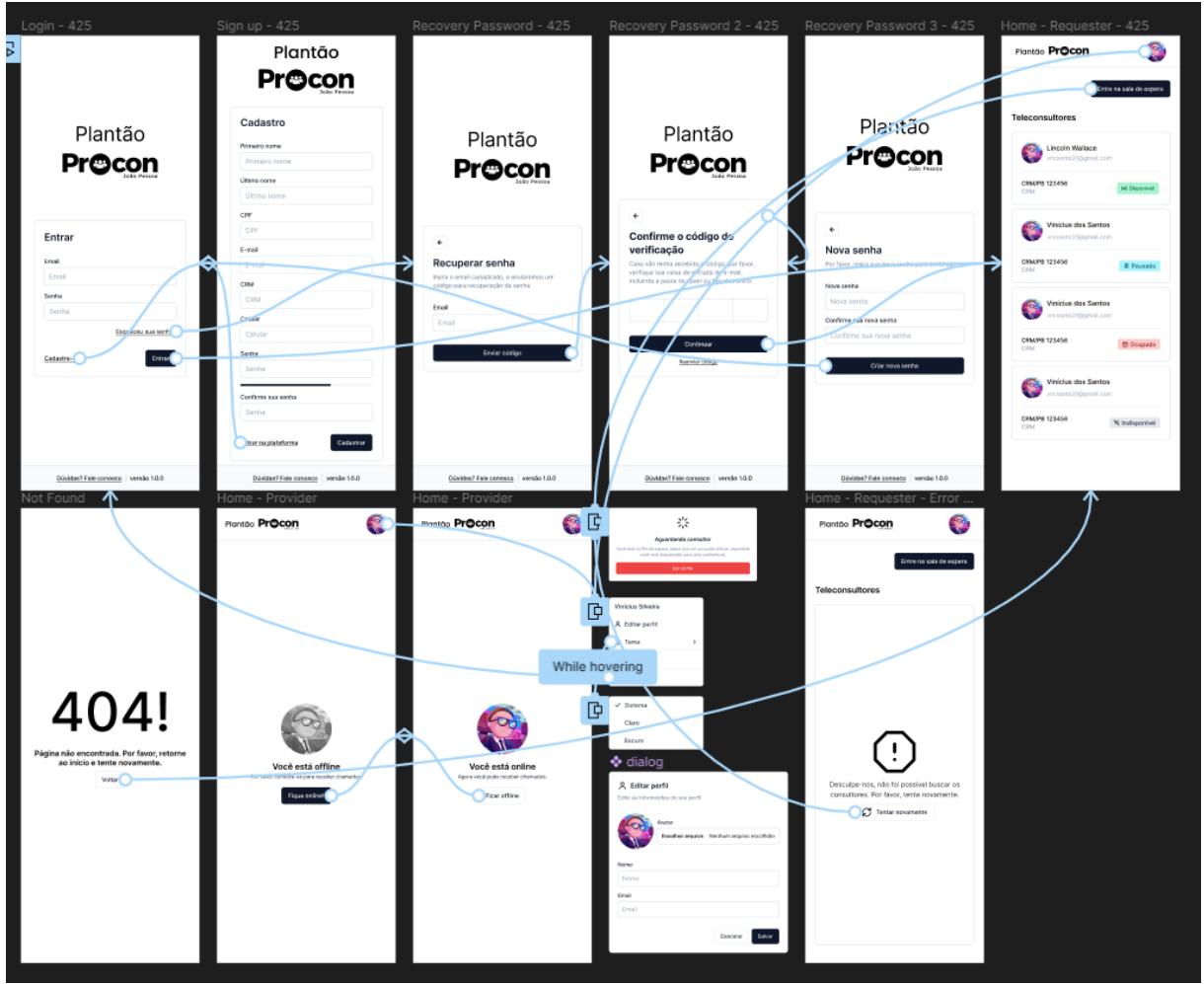


Figura 4: Mapa de navegação do Plantão PROCON

Na próxima seção, todas as telas da plataforma serão completamente explicadas, proporcionando uma compreensão detalhada de cada funcionalidade e opção disponível. Este guia abrangente permitirá que os usuários explorem todas as características da plataforma com confiança e entendimento.

3.8 Telas

3.8.1 Login

A tela de *login* (Figura 5) é a porta de entrada para a plataforma, onde os usuários podem acessar suas contas inserindo seu e-mail e senha previamente cadastrados. Além disso, nesta tela, os usuários têm acesso rápido às opções de cadastro de novos usuários e recuperação de senha, proporcionando uma experiência de entrada completa e facilitando o acesso à plataforma para usuários novos e existentes.

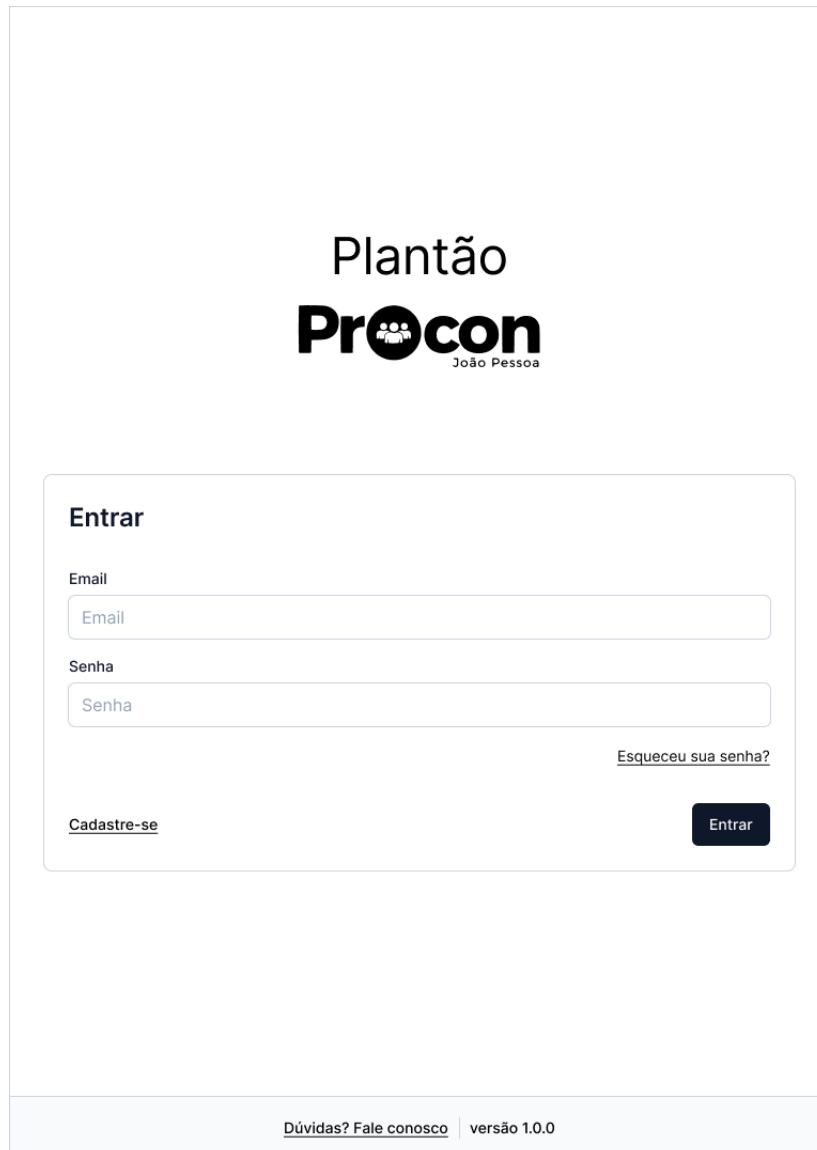


Figura 5: Tela de login
Fonte: Autoria própria.

3.8.2 Cadastro

Na tela de cadastro (Figura 6), os usuários têm a oportunidade de criar uma nova conta na plataforma preenchendo um formulário com informações básicas, como nome, e-mail e senha desejada. Essa etapa é fundamental para permitir que novos usuários se juntem à comunidade e aproveitem os recursos da plataforma. Após o cadastro bem-sucedido, os usuários podem acessar a plataforma utilizando suas novas credenciais de *login*.

The screenshot shows the registration page of the Plantão Procon João Pessoa application. At the top, the logo 'Plantão Procon João Pessoa' is displayed. Below it, the title 'Cadastro' is centered. The form contains several input fields: 'Nome completo' (Name), 'CPF', 'E-mail', 'CRM', 'Celular', 'Senha' (Password), and 'Confirme sua senha' (Confirm password). At the bottom left is a link 'Entrar na plataforma' (Enter platform) and at the bottom right is a dark blue button labeled 'Cadastrar' (Register). A horizontal bar at the bottom indicates the progress of the form completion.

Figura 6: Tela de cadastro da aplicação
Fonte: Autoria própria.

3.8.3 Recuperação de senha

O fluxo de recuperação de senha (Figura 7) guia os usuários por um processo em três etapas para recuperar o acesso à sua conta. Na primeira etapa, os usuários fornecem seu e-mail registrado na plataforma para receber um código de confirmação por e-mail. Na segunda etapa, os usuários inserem o código recebido e têm a opção de reenviá-lo, se necessário. Na terceira e última etapa, os usuários definem uma nova senha para acessar a plataforma após a confirmação bem-sucedida do código de recuperação.

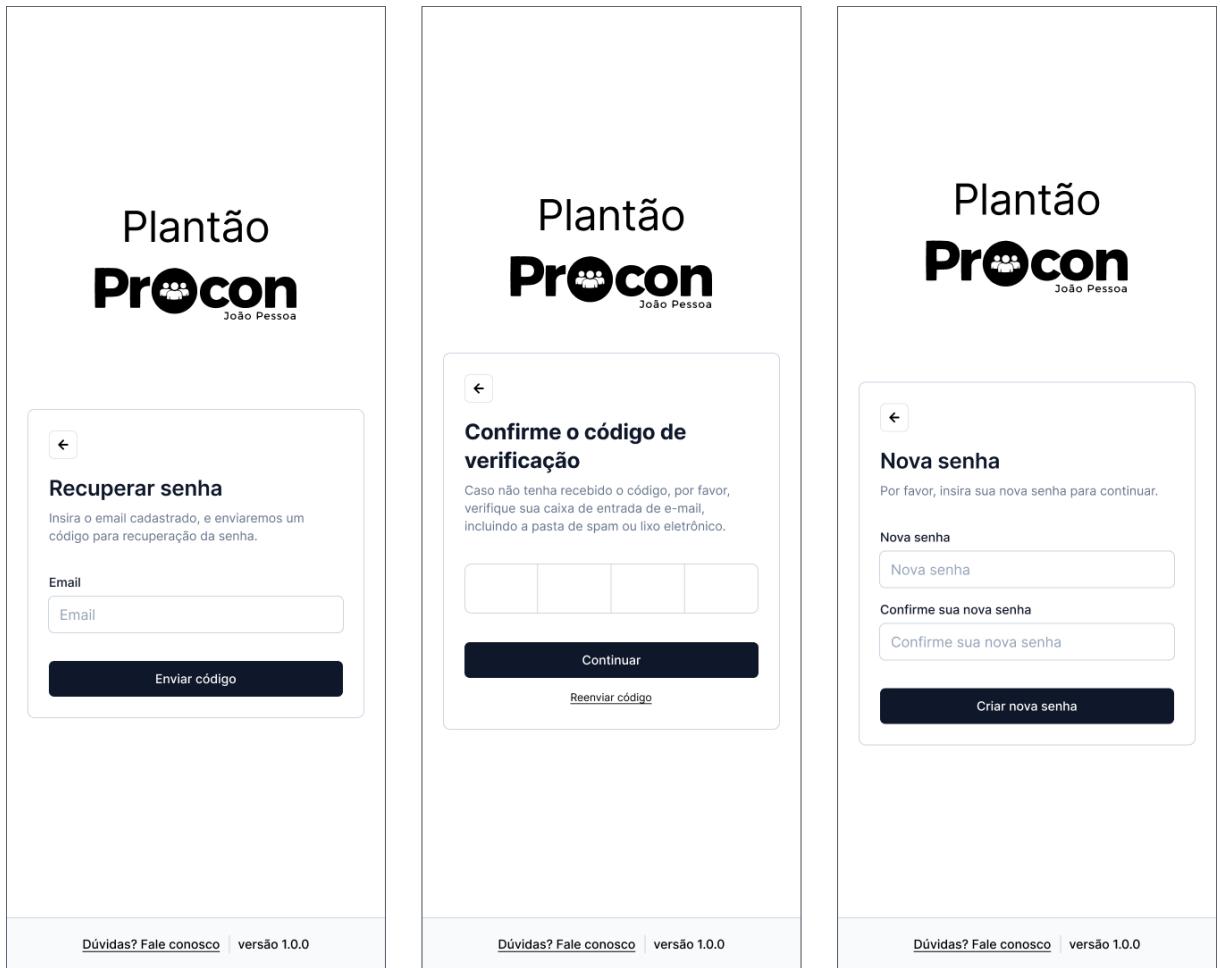


Figura 7: Fluxo de recuperação de senha
Fonte: Autoria própria.

3.8.4 *Home*

A tela principal (Figura 8) é a página inicial da plataforma, apresentando aos usuários uma visão geral do conteúdo e funcionalidades disponíveis. Dependendo do papel do usuário na plataforma, diferentes informações e opções são exibidas. Como solicitante, os usuários podem visualizar uma lista de consultores disponíveis, verificar suas informações e solicitar atendimento. Como consultor, os usuários podem acessar informações sobre suas consultas pendentes, atualizar seu status de disponibilidade e acessar recursos de suporte e configurações de perfil.

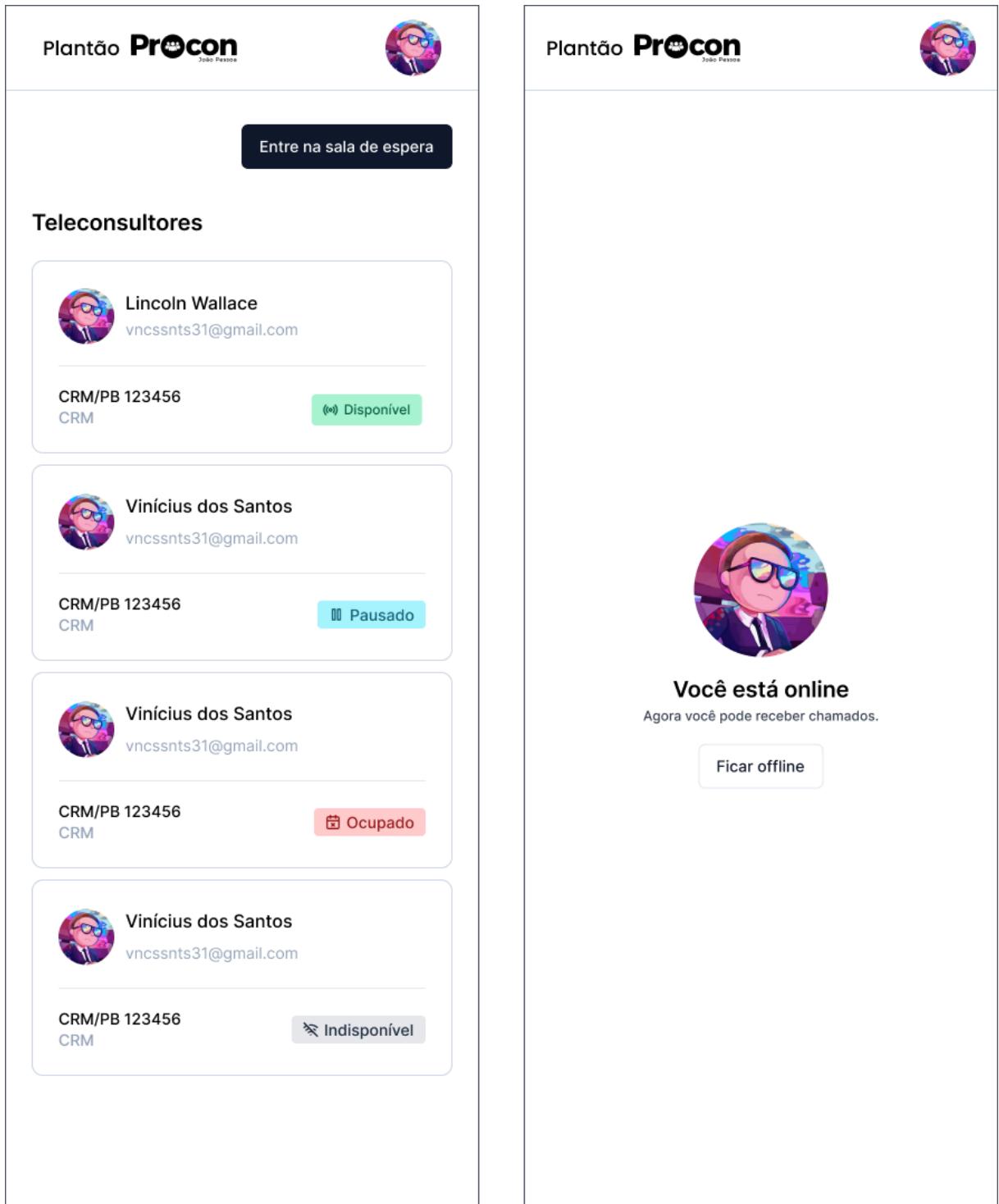


Figura 8: Tela principal da aplicação
Fonte: Autoria própria.

3.8.5 Menu de opções do usuário

Além das telas mencionadas anteriormente, a aplicação oferece um menu de usuário (Figura 9), acessível através do cabeçalho da interface. Este menu proporciona aos usuários uma série de opções para personalizar sua experiência na plataforma. Entre

as opções disponíveis estão:

- **Edição de Perfil:** Permite aos usuários atualizarem suas informações pessoais, como nome, e-mail e foto de perfil, garantindo que seus dados estejam sempre atualizados e precisos.
- **Tema:** Oferece a possibilidade de escolher entre diferentes temas de interface, permitindo que os usuários personalizem a aparência da aplicação, de acordo com suas preferências visuais e necessidades de acessibilidade.
- **Suporte:** Disponibiliza um canal direto de comunicação com a equipe de suporte técnico, onde os usuários podem relatar problemas, fazer perguntas ou solicitar assistência sempre que necessário.
- **Sair da Aplicação:** Permite aos usuários encerrarem sua sessão na plataforma, garantindo a segurança de seus dados e privacidade ao finalizarem sua utilização.



Figura 9: Menu de opções do usuário
Fonte: Autoria própria.

3.8.6 Fila de espera

Na aplicação, quando um usuário solicita atendimento e entra na fila de espera, é exibido um modal (Figura 10) para o aguardo do atendimento. Além disso, o modal oferece a funcionalidade de sair da fila de espera. essa opção dá aos usuários a liberdade de cancelar sua solicitação de atendimento e sair da fila, caso sua situação mude, ou não possam mais aguardar.

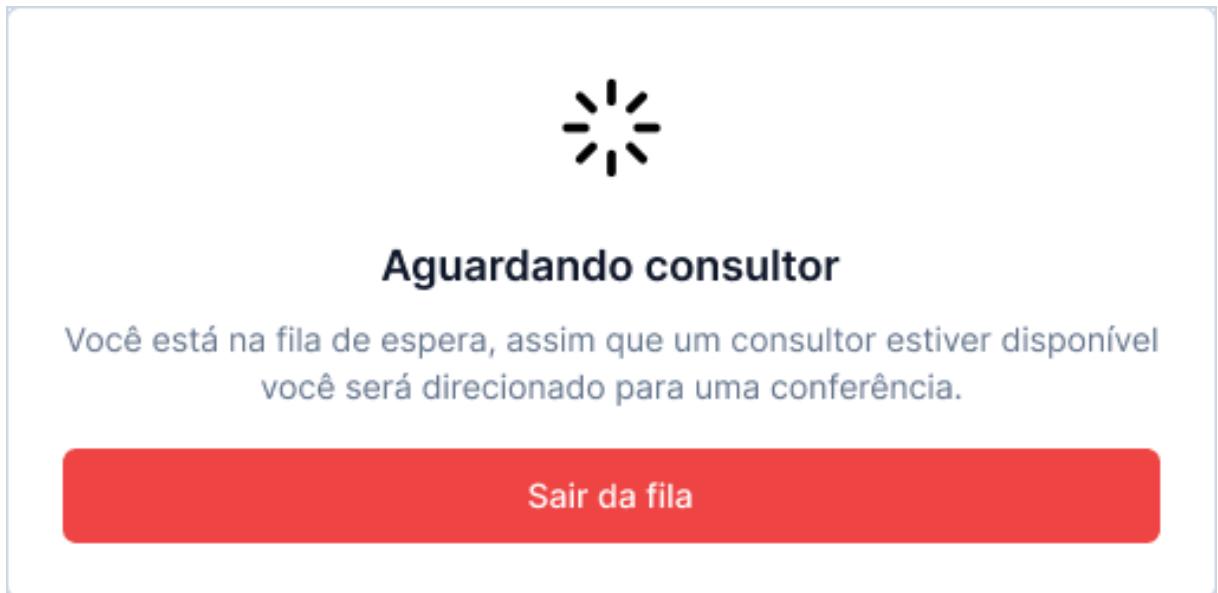


Figura 10: Modal de espera da solicitação

Fonte: Autoria própria.

3.8.7 Edição de perfil

O modal de edição de perfil (Figura 11) permite aos usuários personalizarem suas informações, incluindo foto de perfil, nome e endereço de e-mail. Essa funcionalidade oferece uma maneira simples e intuitiva para os usuários atualizarem suas informações pessoais, garantindo uma experiência personalizada na aplicação.



Figura 11: Modal de edição do perfil

Fonte: Autoria própria

4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Nesse capítulo apresentamos o plano de testes, as personas e os casos de testes.

4.1 Plano de testes

Este plano de testes tem como objetivo garantir a qualidade e funcionalidade da plataforma de teleconsultoria desenvolvida para o PROCON. Abrangendo todos os aspectos da plataforma, incluindo funcionalidades para consultores e solicitantes, processos de comunicação segurança e integrações como sistemas externos, os testes visam verificar a eficácia, usabilidade e confiabilidade da plataforma.

Será realizada uma série de cenários de teste detalhados para avaliar a funcionalidade da plataforma em diferentes situações de uso. Isso inclui testes de *login* e autenticação, funcionalidades específicas para consultores e solicitantes, comunicação entre os usuários, segurança da plataforma e integrações externas. Os resultados dos testes serão registrados e analisados para identificar quaisquer problemas ou áreas de melhoria, assegurando que a plataforma atenda aos padrões de qualidade estabelecidos.

4.2 Personas

4.2.1 Persona A

Maria é uma advogada especializada em direito do consumidor e trabalha como consultora no PROCON há cinco anos. Ela é dedicada, experiente e comprometida em ajudar os consumidores a resolverem seus problemas de forma eficaz. Maria está sempre disposta a oferecer assistência e orientação para garantir que os direitos dos consumidores sejam respeitados.

4.2.2 Persona B

João é um profissional autônomo que reside na cidade e está preocupado com seus direitos como consumidor. Ele teve algumas experiências negativas com produtos e serviços no passado e agora está determinado a buscar ajuda do PROCON para resolver seus problemas. João é proativo, diligente e está disposto a aprender mais sobre seus direitos e como protegê-los.

4.3 Casos de teste

Os casos de teste desempenham um papel crucial no desenvolvimento e na garantia da qualidade de uma aplicação. Eles consistem em uma série de cenários planejados e

executados para validar se a aplicação atende aos requisitos estabelecidos e se funciona conforme o esperado em diferentes situações. O propósito fundamental dos casos de teste é detectar defeitos e garantir que a aplicação atenda aos padrões de qualidade esperados pelos usuários finais. Ao abranger uma variedade de cenários possíveis, os casos de teste ajudam a identificar falhas, garantindo assim que a aplicação seja robusta, confiável e capaz de fornecer uma experiência satisfatória aos usuários.

4.3.1 [CT01] Verificar se um usuário pode se autenticar na plataforma através do processo de *login*

| | |
|----------------------|--|
| Objetivo | Verificar se o sistema permite que os usuários se autentiquem na plataforma, fornecendo suas credenciais de acesso, como e-mail e senha. |
| Pré-condições | Nenhuma |
| Passos de teste | <ol style="list-style-type: none"> 1. Acessar a página de <i>login</i> da plataforma. 2. Inserir o e-mail e senha associados à conta do usuário nos campos correspondentes. 3. Clicar no botão para fazer <i>login</i> na plataforma. |
| Resultados esperados | O sistema deve redirecionar o usuário, caso as credenciais estiverem válidas, para a tela principal da aplicação. |
| Resultado obtido | Bem sucedido. |
| Análise do resultado | Funcionou como o esperado. |

Tabela 1: Caso de teste 01

4.3.2 [CT02] Verificar se um usuário não pode fazer login com credenciais inválidas.

| | |
|----------------------|--|
| Objetivo | Verificar se o sistema não permite que os usuários se autentiquem na plataforma fornecendo suas credenciais de acesso inválidas, como e-mail e senha. |
| Pré-condições | Nenhuma |
| Passos de teste | <ol style="list-style-type: none"> 1. Acessar a página de <i>login</i> da plataforma. 2. Inserir o e-mail e senha nos campos correspondentes. 3. Clicar no botão para fazer <i>login</i> na plataforma. |
| Resultados esperados | O sistema deve exibir uma mensagem de erro informando que aquelas credenciais não estão cadastradas na plataforma. |
| Resultado obtido | Bem sucedido. |
| Análise do resultado | Funcionou como o esperado. |

Tabela 2: Caso de teste 02

4.3.3 [CT03] Verificar se o sistema permite a criação de novos usuários solicitantes através do processo de cadastro

| | |
|----------------------|---|
| Objetivo | Verificar se o sistema permite que os usuários criem novas contas na plataforma, fornecendo todas as informações informações obrigatórias no formulário de cadastro. |
| Pré-condições | Nenhuma |
| Passos de teste | <ol style="list-style-type: none"> 1. Acessar a página de cadastro da plataforma. 2. Preencher todos os campos obrigatórios do formulário de cadastro. 3. Clicar no botão para finalizar o cadastro na plataforma. |
| Resultados esperados | O sistema deve redirecionar o usuário para a tela de <i>login</i> permitindo que o usuário acesse sua nova conta. |
| Resultado obtido | Bem sucedido. |
| Análise do resultado | Funcionou como o esperado. |

Tabela 3: Caso de teste 03

4.3.4 [CT04] Verificar se o sistema impede o cadastro de um usuário com um e-mail ou CPF já existente na plataforma.

| | |
|----------------------|---|
| Objetivo | Verificar se o sistema permite que os usuários criem novas contas na plataforma, fornecendo todas as informações informações obrigatórias no formulário de cadastro. |
| Pré-condições | Nenhuma |
| Passos de teste | <ol style="list-style-type: none"> 1. Acessar a página de cadastro da plataforma. 2. Preencher todos os campos obrigatórios do formulário de cadastro. 3. Clicar no botão para finalizar o cadastro na plataforma. |
| Resultados esperados | O sistema deve apresentar uma mensagem de erro informando que o e-mail ou CPF já está cadastrado na plataforma. |
| Resultado obtido | Bem sucedido. |
| Análise do resultado | Funcionou como o esperado. |

Tabela 4: Caso de teste 04

4.3.5 [CT05] Verificar se o usuário pode recuperar sua senha seguindo todas as etapas do processo.

| | |
|----------------------|---|
| Objetivo | Verificar se o processo de recuperação de senha permite que o usuário restaure sua senha com sucesso, seguindo todas as etapas necessárias. |
| Pré-condições | O usuário possui uma conta registrada na plataforma e esqueceu sua senha. |
| Passos de teste | <ol style="list-style-type: none"> 1. Acessar a página de recuperação de senha da plataforma. 2. Inserir o endereço de e-mail associado à conta do usuário. 3. Clicar no botão para enviar o código de recuperação. 4. Verificar a caixa de entrada do e-mail fornecido e anotar o código de recuperação recebido. 5. Voltar à página de recuperação de senha e inserir o código de recuperação recebido. 6. Prosseguir para a próxima etapa após inserir o código corretamente. 7. Inserir a nova senha desejada nos campos fornecidos. 8. Confirmar a nova senha nos campos fornecidos. 9. Clicar no botão para concluir a recuperação de senha. |
| Resultados esperados | O sistema deve permitir que o usuário siga todas as etapas do processo de recuperação de senha com sucesso e, após concluir o processo, o usuário poderá realizar <i>login</i> na plataforma com sua nova senha. |
| Resultado obtido | Bem sucedido. |
| Análise do resultado | Funcionou como o esperado. |

Tabela 5: Caso de teste 05

4.3.6 [CT06] Verificar se o sistema não permite a recuperação de senha para um e-mail não cadastrado.

| | |
|----------------------|---|
| Objetivo | Verificar se o sistema impede a recuperação de senha, quando um usuário tenta usar um endereço de e-mail, que não está cadastrado na plataforma. |
| Pré-condições | O usuário tenta recuperar sua senha, mas usa um endereço de e-mail, que não está associado a nenhuma conta na plataforma. |
| Passos de teste | <ol style="list-style-type: none"> 1. Acessar a página de recuperação de senha da plataforma. 2. Inserir um endereço de e-mail, que não esteja cadastrado na plataforma. 3. Clicar no botão para enviar o código de recuperação. |
| Resultados esperados | O sistema exibe uma mensagem de erro informando que o endereço de e-mail inserido não está cadastrado na plataforma e que a recuperação de senha não pode ser concluída. |
| Resultado obtido | Bem sucedido. |
| Análise do resultado | Funcionou como o esperado. |

Tabela 6: Caso de teste 06

4.3.7 [CT07] Verificar se o usuário é desconectado da plataforma após fazer *logout*.

| | |
|----------------------|--|
| Objetivo | Verificar se o sistema desconecta adequadamente o usuário da plataforma quando ele realiza o <i>logout</i> . |
| Pré-condições | O usuário está logado na plataforma. |
| Passos de teste | <ol style="list-style-type: none"> 1. Localizar e clicar no botão de <i>logout</i> na interface da plataforma. 2. Aguardar a confirmação de que o <i>logout</i> foi realizado com sucesso. |
| Resultados esperados | O sistema encerra a sessão do usuário e redireciona para a página de <i>login</i> da plataforma. |
| Resultado obtido | Bem sucedido. |
| Análise do resultado | Funcionou como o esperado. |

Tabela 7: Caso de teste 07

4.3.8 [CT08] Verificar se um usuário pode editar seu perfil com informações válidas.

| | |
|----------------------|--|
| Objetivo | Verificar se o sistema permite que um usuário edite seu perfil, fornecendo informações válidas. |
| Pré-condições | O usuário está logado na plataforma. |
| Passos de teste | <ol style="list-style-type: none"> 1. Acessar o modal de edição de perfil na plataforma. 2. Inserir informações válidas nos campos editáveis. 3. Salvar as alterações realizadas no perfil. 4. Verificar se as alterações foram salvas corretamente. |
| Resultados esperados | O sistema deve mostrar uma mensagem de sucesso e apresentar os novos dados na tela. |
| Resultado obtido | Bem sucedido. |
| Análise do resultado | Funcionou como o esperado. |

Tabela 8: Caso de teste 08

4.3.9 [CT09] Verificar se o usuário pode alterar o tema da plataforma para uma das opções disponíveis.

| | |
|----------------------|---|
| Objetivo | Verificar se o sistema permite que um usuário altere o tema da plataforma para uma das opções disponíveis. |
| Pré-condições | O usuário está logado na plataforma. |
| Passos de teste | <ol style="list-style-type: none"> 1. Acesse a página de configurações ou preferências da plataforma. 2. Selecione uma das opções de tema disponíveis, como “Claro”, “Escuro” ou “Sistema”. 3. Salve as alterações realizadas nas configurações. 4. Verifique se o tema da plataforma foi alterado corretamente, conforme a seleção feita pelo usuário. |
| Resultados esperados | O sistema deve alterar o tema em tempo real, após o usuário escolher a opção desejada. |
| Resultado obtido | Bem sucedido. |
| Análise do resultado | Funcionou como o esperado. |

Tabela 9: Caso de teste 09

4.3.10 [CT10] Verificar se o usuário consegue entrar na fila de espera.

| | |
|----------------------|---|
| Objetivo | Verificar se o sistema permite que os usuários entrem na fila de espera para serem atendidos por um consultor. |
| Pré-condições | O usuário está logado na plataforma |
| Passos de teste | <ol style="list-style-type: none"> 1. Acesse a página, ou funcionalidade, que permite entrar na fila de espera na plataforma. 2. Verifique se há uma opção clara para entrar na fila de espera visível na interface. 3. Clique na opção para entrar na fila de espera. |
| Resultados esperados | O sistema deve apresentar um modal informando que o usuário está na fila de espera. |
| Resultado obtido | Bem sucedido. |
| Análise do resultado | Funcionou como o esperado. |

Tabela 10: Caso de teste 10

4.3.11 [CT11] Verificar se o usuário pode sair da fila de espera.

| | |
|----------------------|--|
| Objetivo | Verificar se o sistema permite que os usuários saiam da fila de espera, caso desejem não aguardar mais o atendimento por um consultor. |
| Pré-condições | O usuário está logado na plataforma e já está na fila de espera para ser atendido por um consultor. |
| Passos de teste | <ol style="list-style-type: none"> 1. Clicar no botão "Sair da fila de espera". |
| Resultados esperados | O sistema deve fechar o modal de fila de espera e o usuário não deve ser chamado para nenhuma conferência. |
| Resultado obtido | Bem sucedido. |
| Análise do resultado | Funcionou como o esperado. |

Tabela 11: Caso de teste 11

4.4 Discussão dos resultados

Os resultados obtidos na realização dos testes indicam que a aplicação apresenta um bom funcionamento em todas as funcionalidades implementadas. Durante os testes, todas as funcionalidades foram avaliadas e aprovadas com sucesso, demonstrando que a plataforma é funcional e robusta. No entanto, embora os resultados sejam geralmente positivos, é importante ressaltar que sempre há oportunidades de melhoria. Embora nenhum problema significativo tenha sido identificado durante os testes, algumas áreas podem ser

aprimoradas para aumentar ainda mais a qualidade e a usabilidade da aplicação. Em suma, os testes realizados desempenharam um papel crucial na garantia da qualidade da plataforma, assegurando que ela atenda aos requisitos funcionais e não funcionais estabelecidos.

5 CONCLUSÕES E TRABALHOS FUTUROS

Nesta pesquisa, os objetivos propostos de desenvolver uma plataforma de teleconsultoria para o PROCON, visando facilitar o acesso dos consumidores a assistência jurídica e orientações sobre seus direitos, foram alcançados. Ao longo do projeto, exploramos conceitos fundamentais como JavaScript, HTML, CSS, TailwindCSS, Vite, *Single Page Application* (SPA), MVC, TypeScript e ReactJS, que foram essenciais para o desenvolvimento da plataforma.

Durante o processo, foram identificados e superados diversos desafios, desde a elaboração das personas, até a definição dos requisitos funcionais e não funcionais do sistema. Através de uma abordagem colaborativa e iterativa, a aplicação foi desenvolvida e como resultado conseguiu desenvolver uma plataforma que atende às necessidades, tanto dos consultores do PROCON, quanto dos consumidores.

As principais descobertas deste trabalho incluem a importância da interação entre consultores e solicitantes por meio de videoconferência, a necessidade de garantir a segurança e proteção dos dados dos usuários, e a relevância das SPA para oferecer uma experiência de usuário mais fluida e responsiva.

Ao refletir sobre os desafios enfrentados durante o desenvolvimento do projeto, reconhece-se a importância de uma abordagem ágil e adaptativa, bem como a necessidade de uma comunicação eficaz e colaborativa. As lições aprendidas ao longo do processo serão valiosas para futuros projetos e contribuem para o crescimento profissional e pessoal.

Em última análise, a plataforma de teleconsultoria desenvolvida representa uma contribuição significativa para a área, ao fornecer uma solução inovadora e acessível para auxiliar os consumidores na defesa de seus direitos. Espera-se que este trabalho inspire futuras pesquisas e iniciativas na área de proteção ao consumidor e promova uma sociedade mais justa e equitativa para todos.

Além dos resultados obtidos até o momento, há um horizonte de oportunidades que se vislumbra para ampliar ainda mais o alcance e a eficácia da plataforma desenvolvida. Considerando a constante evolução das demandas e tecnologias, é crucial antecipar e planejar os próximos passos para garantir a contínua relevância e utilidade do sistema. Portanto, apresentamos a seguir uma lista de trabalhos futuros que representam não apenas um avanço no projeto atual, mas também um compromisso com a inovação e a excelência na prestação de serviços ao público-alvo:

- Implementação de uma “Tela de Organização” para facilitar a gestão interna dos consultores e recursos.
- Desenvolvimento de um “Dashboard da Organização” para fornecer uma visão geral

das atividades e métricas relevantes.

- Integração de um “Bate-papo via texto” para permitir uma comunicação imediata entre consultores e consumidores.
- Adição de funcionalidade de “Abertura de Chamado” para facilitar o registro e acompanhamento de questões dos consumidores.
- Desenvolvimento de uma aplicação móvel (Android / iOS) para aumentar a acessibilidade e conveniência dos serviços oferecidos.
- Implementação de um “Sistema de Notificação” para alertar os usuários sobre alterações nos chamados e outras atualizações importantes.

REFERÊNCIAS

- BANKS, A.; PORCELLO, E. **Learning React: modern patterns for developing React apps.** [S.l.]: O'Reilly Media, 2020.
- DEACON, J. Model-view-controller (mvc) architecture. **Online**[Citado em: 10 de março de 2006.] <http://www.jdl.co.uk/briefings/MVC.pdf>, v. 28, p. 61, 2009.
- DOWDEN, M.; DOWDEN, M. **Architecting CSS: The programmer's guide to effective style sheets.** [S.l.]: Apress, 2020.
- GACKENHEIMER, C. **Introduction to React.** [S.l.]: Apress, 2015.
- JADHAV, M. A.; SAWANT, B. R.; DESHMUKH, A. Single page application using angularjs. **International Journal of Computer Science and Information Technologies**, Citeseer, v. 6, n. 3, p. 2876–2879, 2015.
- KEITH, J. A brief history of javascript. **DOM Scripting: Web Design with JavaScript and the Document Object Model**, Springer, p. 3–10, 2005.
- RAWAT, P.; MAHAJAN, A. N. Reactjs: a modern web development framework. **International Journal of Innovative Science and Research Technology**, v. 5, n. 11, p. 698–702, 2020.
- SILVA, M. S. **Criando sites com HTML: sites de alta qualidade com HTML e CSS.** [S.l.]: Novatec Editora, 2008.
- SINGH, S. et al. Mvc framework: a modern web application development approach and working. **International Research Journal of Engineering and Technology**, v. 7, n. 01, p. 51–55, 2020.
- WATHAN, A. **Tailwind CSS-Rapidly build modern websites without ever leaving your HTML.** [S.l.]: Rapidly Build Modern Websites without Ever Leaving Your HTML. [https](https://tailwindcss.com/), 2022.