

|  |  |  |
| --- | --- | --- |
| **Sprint Documentation** | | |
| **Sprint #4 Definir padronização de configurações de acessibilidade e interface** | **Start Date: 29/10/2024** | **Final Date: 01/11/2024** | |
| **Team Members** | | | |
| Maysa Lopes de Sousa | | | |
| Marco Antônio Masson | | | |
| Franciane Ramos Franco | | | |
| Kétlyn Christini Tonholi | | | |
| Felipe villela de Souza | | | |
| Renato dos Santos Filho | | | |
| Izabelly Cristina Silva Brito | | | |
| Pedro Besteti Rotondo | | | |
| Gabriel Neto Siqueira | | | |
| Amanda Pinheiro Brito | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprint Backlog** | | | |
| Task# | Description | Start Date | Final date |
| #18 | Sprint 04 - Definir padronização de configurações de acessibilidade e interface | 29/10/2024 | 01/11/2024 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task Description** | | | | | |
| Task # | Description | Assigned To | Status | Estimated Hours | Logged Hours |
| 018 | Desenvolvemos um padrão de interface acessível e consistente para diferentes dispositivos, visando uma experiência inclusiva. As principais funcionalidades incluem ajuste dinâmico de tamanho da fonte e auto-contraste adaptável a temas claro e escuro, melhorando a legibilidade e acessibilidade. Os componentes implementados foram: Header, Footer e Layout Responsivo, garantindo adaptação em diversas telas. Utilizamos as bibliotecas React Router DOM para gerenciamento de rotas e Phosphor Icons para ícones personalizáveis e usabilidade aprimorada. | Renato dos Santos.  Maysa Lopes.  Marco Antônio.  Franciane Ramos.  Kétlyn Tonholi.  Felipe villela.  Izabelly Brito.  Pedro Besteti.  Gabriel Siqueira.  Amanda Brito. | Concluído | 20h | 20h |

|  |
| --- |
| **Sprint Results** |

Nesta sprint, demos um passo significativo na melhoria da acessibilidade do projeto, implementando ajustes essenciais que impactam diretamente a experiência do usuário. Foi feita a adição de um mecanismo que permite o ajuste do tamanho da fonte, tornando o conteúdo adaptável às necessidades visuais dos usuários. Além disso, o recurso de autocontrate foi incorporado, permitindo uma melhor visualização de elementos, especialmente para pessoas com deficiências visuais ou condições como daltonismo, melhorando a usabilidade em ambientes de pouca luz ou sob fortes reflexos.

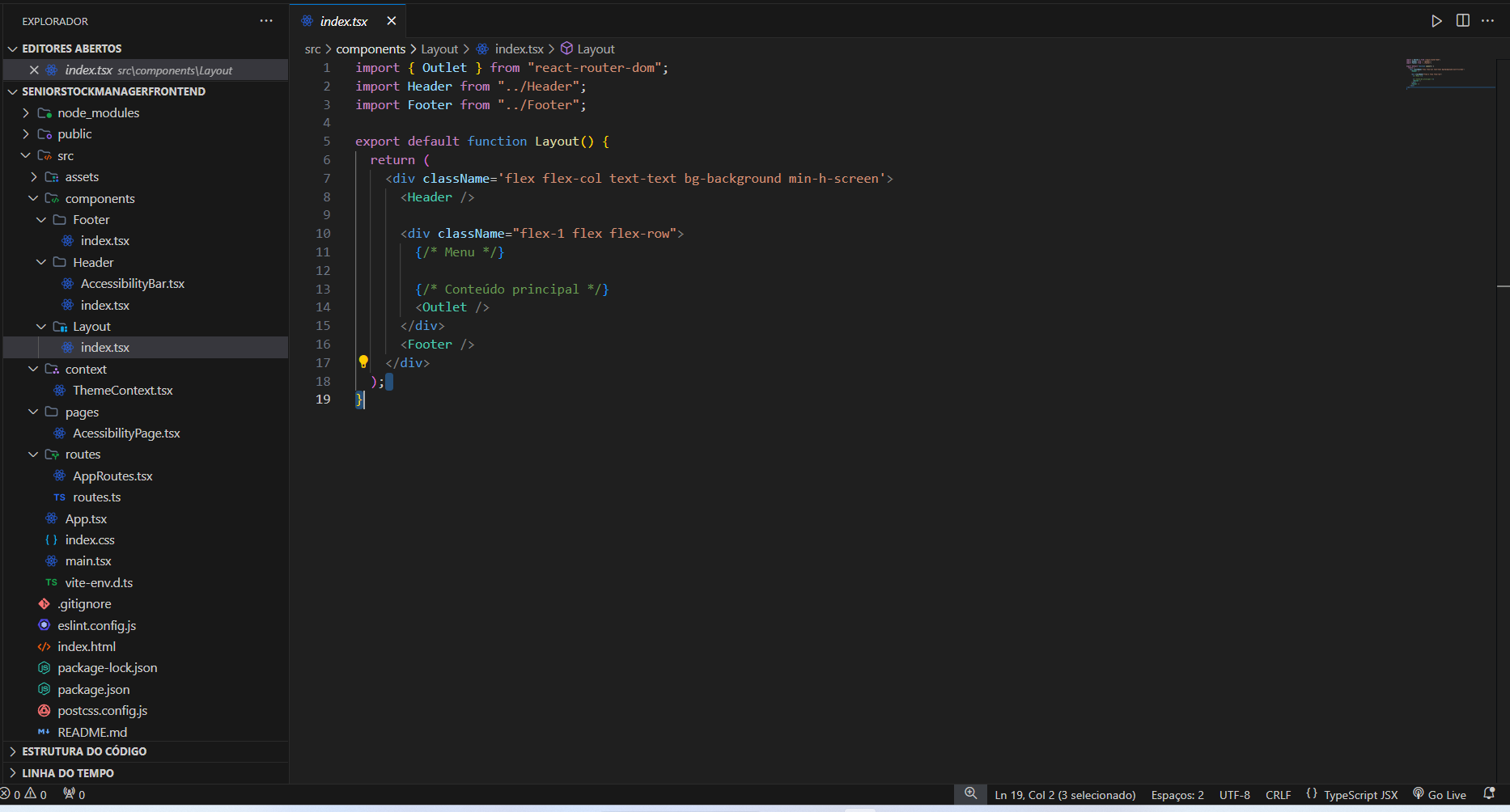
Na parte de componentização, a criação de elementos fundamentais como o Header, Footer e o Layout foi uma prioridade. O Header oferece uma navegação centralizada e consistente, facilitando o acesso rápido às funcionalidades principais da aplicação. O Footer agrega informações importantes como links institucionais, políticas de privacidade e termos de uso, mantendo um padrão de organização e clareza visual. Já o Layout foi desenhado para garantir uma estrutura intuitiva, onde o conteúdo é apresentado de maneira lógica e fluída, resultando em uma experiência do usuário muito mais agradável e eficiente. A organização modular dos componentes facilita futuras manutenções e incrementos no sistema.

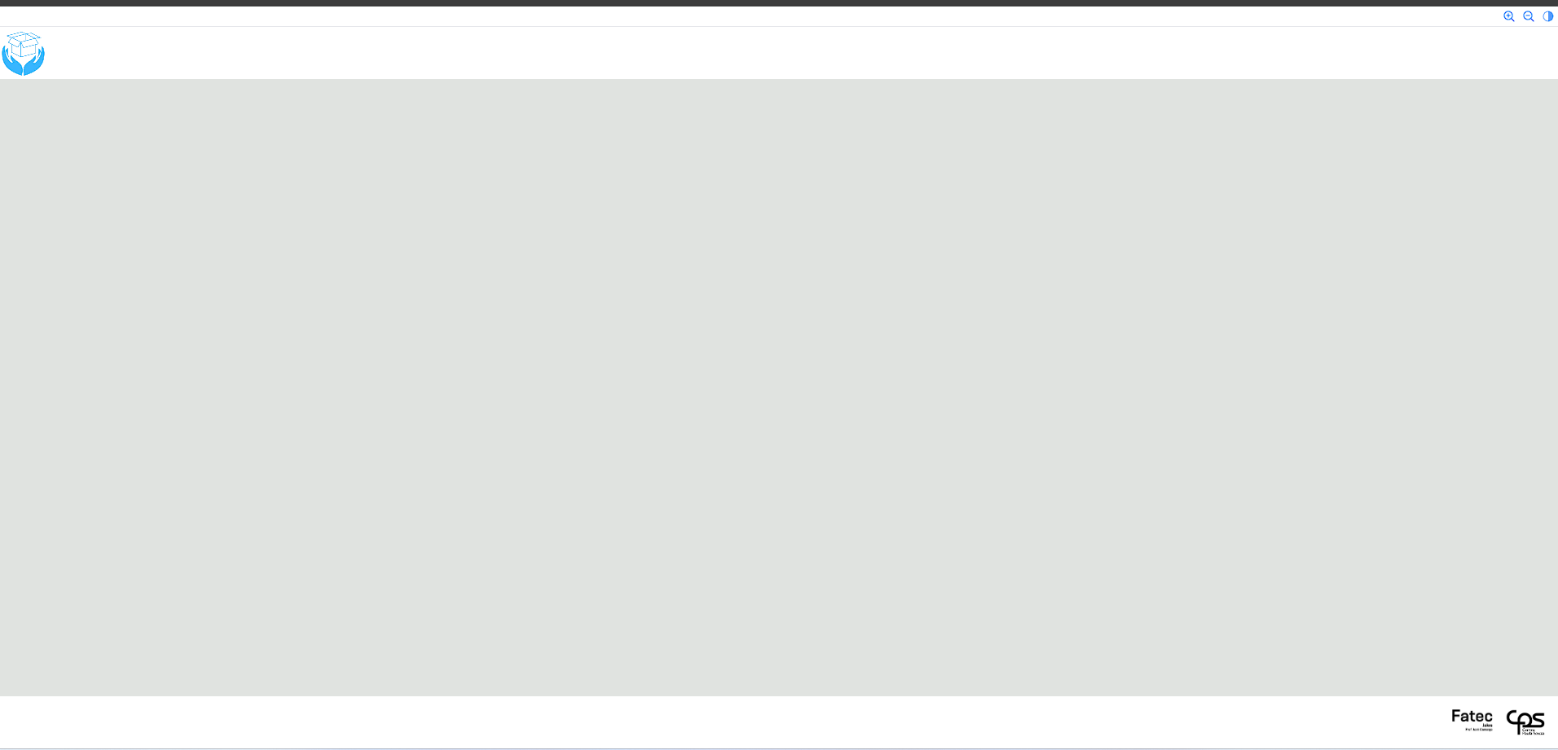
Em relação à navegação, a implementação das rotas foi um ponto crucial. Utilizando a biblioteca React Router DOM, conseguimos estruturar a navegação interna de maneira dinâmica e fluida, permitindo a troca de páginas sem recarregar o site inteiro. Este recurso melhora a performance do sistema e proporciona uma experiência de navegação mais rápida e contínua. A divisão clara das rotas também facilita a manutenção e a escalabilidade do sistema, permitindo que novas páginas ou funcionalidades sejam adicionadas de maneira organizada e sem impacto significativo no restante da aplicação.

Para a parte visual, utilizamos a biblioteca Phosphor Icons, que nos proporcionou uma vasta gama de ícones modernos e personalizáveis. A escolha dessa biblioteca se deve à sua flexibilidade, permitindo ajustar o estilo dos ícones para se alinhar com o tema da interface. Os ícones foram distribuídos estrategicamente ao longo da aplicação, guiando o usuário em sua navegação e tornando a interface mais amigável e intuitiva. Essa escolha foi fundamental para melhorar a comunicação visual e a navegabilidade.

Por fim, adotamos um padrão de organização de pastas que facilita a manutenção do código e o trabalho colaborativo entre os membros da equipe. Nossa estrutura no VS Code seguiu um modelo claro e objetivo, onde os arquivos foram divididos por funcionalidades e responsabilidades, evitando redundâncias e garantindo que o código fosse facilmente acessível e entendível por todos os desenvolvedores. Essa prática permite maior escalabilidade, já que novos desenvolvimentos podem ser integrados sem comprometer a organização existente. O modelo utilizado pode ser consultado no link de referência que foi seguido.

**Telas produzidas**





**Descrever o que foi bem durante a sprint – aprendizados**

Durante esta sprint, avançamos significativamente na acessibilidade e organização do projeto. Implementamos com sucesso o ajuste dinâmico de tamanho de fonte e o recurso de auto contraste, que irá melhorar a experiência para usuários com necessidades visuais, como daltonismo ou em ambientes com pouca luz.

Na parte de componentização, criamos o Header, Footer e o Layout, que trazem consistência à navegação e estrutura da interface. O Header facilitara o acesso às funções principais, enquanto o Footer organizara informações como políticas e links importantes. O Layout responsivo garantira que o conteúdo se ajuste adequadamente a diferentes dispositivos, proporcionando uma experiência mais fluida.

A navegação interna foi aprimorada com o uso do React Router DOM, permitindo trocas de páginas sem recarregar a aplicação, o que aumentou a performance. Também usamos a biblioteca Phosphor Icons para adicionar ícones modernos e personalizáveis, melhorando a interface visual e tornando-a mais intuitiva.

Além disso, adotamos uma organização clara de pastas no VS Code, o que facilitou a colaboração entre os membros da equipe e a manutenção do código. Esse padrão modular permitirá maior facilidade na integração de novos recursos, garantindo uma escalabilidade mais organizada.

**Descrever o que não pode ser implementado com justificativa**

**1. Funcionalidade de Personalização de Temas**

Embora tenhamos implementado ajustes de acessibilidade, a personalização avançada de temas, permite aos usuários escolher entre várias paletas de cores e estilos de interface, não foi implementada. Essa funcionalidade requer mais tempo para desenvolver e testar, assegurando que todas as combinações de cores mantenham-se a conformidade com as diretrizes de acessibilidade. Para garantir a qualidade do que já foi implementado, decidimos priorizar essa funcionalidade para uma sprint futura.

**2. Sistema de Notificações**

O desenvolvimento de um sistema de notificações para alertar os usuários sobre atualizações ou mensagens importantes não foi concluído. Essa funcionalidade é importante para melhorar a comunicação com os usuários, mas exigiria uma integração mais complexa com o backend e testes adicionais. Devido a limitações de tempo e recursos, optamos por adiar essa implementação para focar nas melhorias de acessibilidade e na estrutura básica da aplicação.