**FRONT-END**

No desenvolvimento do nosso projeto do TCC, um dos problemas que tivemos que pensar foi a interface do sistema, com o nosso propósito de construir um site de empregos a nível mundial focado para vagas do Brasil, precisaríamos de uma interface simples de se usar, fácil de se aprender e sem muita burocracia para que os candidatos não desistam facilmente de criar sua conta e aplicar-se para uma vaga criada no nosso site. Então, para planejarmos sobre, criamos alguns esboços das telas utilizando o software Figma, com ele pudemos decidir as cores principais do site e alguns layouts de tela para as funcionalidades que o usuário irá interagir conforme figura **<insert image number here>**

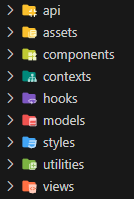
A screenshot of a computer

Description automatically generated

Para o desenvolvimento da interface decidimos usar o React, pela sua facilidade de criação de telas e quantidade de bibliotecas que podem ser integradas com o intuito de facilitar ainda mais a criação de pequenos componentes já com padronizados. O React é uma biblioteca open source criada pela empresa Meta com a ajuda da comunidade, lançado em 29 de maio de 2013.

Sobre as bibliotecas que utilizamos com o React, a primeira delas é a “styled-components”, uma biblioteca que permite utilizarmos estilização para componentes em específico sem que seja utilizada uma estilização global importando arquivos CSS, como é feito por padrão no React sem qualquer biblioteca. Também utilizamos react-router-dom, essa biblioteca nos permite mapear rotas do navegador para componentes do React, por exemplo, quando o usuário acessar o domínio com o caminho /home, um componente gerado será renderizado. A biblioteca mais importante utilizada no nosso projeto foi a Material UI, ela nos fornece diversos componentes prontos, a possibilidade de reutilização de componentes auxiliou muito no tempo de desenvolvimento.

A estrutura do projeto front-end foi pensada visando separar cada responsabilidade do código fonte, para que seja fácil escalar e implementar novas funcionalidades.



Na pasta de api temos todos as funções que foram implementadas para realizar requisições para o back-end, cada sub-diretório contém chamadas de acordo com sua entidade. A pasta de assets do projeto contém todos os arquivos de mídia utilizadas no projeto, como imagens, áudios, vídeos, gif etc.

Já a pasta de components contém todos os componentes que podemos criar utilizando React, esses componentes são trechos de código HTML, CSS e Javascript que podem ser reutilizados em diversas partes do nosso sistema, sendo extremamente poderoso e simples de ser feito, os componentes também podem receber propriedades, tornando-o personalizável para cada parte do código.

A pasta de contexts possui contextos que podem ser utilizados em qualquer parte da aplicação, permitindo salvar dados e funções que serão reutilizados, evitando passar dados via componentes. Os contextos permitiram a redução da complexidade do projeto e a melhor manutenibilidade do mesmo, gerando mais segurança e estabilidade ao front-end.

A pasta de hooks contém funcionalidades que podem ser reutilizadas sobre os componentes, hooks são funções facilitadas que podem ser utilizadas para diversos princípios, como buscar dados, configurar variáveis, executar funcionalidade em qualquer ciclo de um componente, etc. No nosso projeto criamos um para que certas páginas do site só fossem acessadas após o usuário ou empresa terem sido autenticadas.

A pasta de models contém todas as estruturas de dados que foram mapeados durante o desenvolvimento da aplicação, possuindo também algumas funcionalidades sobre a própria estrutura. Essa parte do código fonte está relacionadas com diversas outras partes do sistema, como por exemplo páginas que simbolizam essas estruturas, chamadas de API que retornam estruturas mapeadas, criação de modelos no back-end etc.

A pasta de styles contém tudo o que é relacionado com a estilização global da aplicação, dentro do arquivo global.css podem conter códigos CSS que irão ser utilizados no projeto inteiro. Já o arquivo theme.ts contém a estilização global relacionada ao framework Material UI, o mesmo além de nos fornecer diversos componentes prontos, ele também nos dá a possibilidade de estilizarmos o mesmo, como alterar cores, tamanho de fontes, etc.

A pasta de utilities possui todos os arquivos que contém funcionalidades úteis que possam ser reutilizadas ao longo do projeto como por exemplo, formatação de datas, formatação de campos, utilidades para textos, etc.

Por fim, a pasta de views contém todas as páginas do site, nela são armazenadas diversos componentes, tais como as próprias páginas. Cada página deve ter uma rota que deve ser mapeada no arquivo “router.ts” que se encontra na raiz do projeto, dentro desse arquivo contém uma lista com vários itens nos quais devem mapear um componente com sua rota.