Front-end

No desenvolvimento do nosso projeto do TCC, um dos problemas que tivemos que pensar é a interface do sistema pois é uma das partes mais importantes porque 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, alguns layouts de tela para as funcionalidades que o usuário irá interagir conforme figura (x.x)

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 muitas quantidades de biblioteca que podem ser integradas, com o intuito de facilitar ainda mais a criação de pequenos componentes que são usados nas telas e já com padronizações criadas por outros desenvolvedores. O React é uma biblioteca criada pela empresa Meta com a ajuda da comunidade, com data de lançamento inicial em 29 de maio de 2013, sendo open source onde qualquer pessoa pode contribuir para o projeto, o React está sendo muito utilizado em diversas áreas e empresas, portando temos muita facilidade de encontrar conteúdo sobre a maioria das coisas que procuramos desenvolver, e problemas que identificamos no meio do caminho.

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 com componentes do React, por exemplo, quando o usuário acessar o nosso domínio com o caminho /home, então podemos mapear para que algum componente que criamos possa ser renderizado nesse momento.

A biblioteca mais importante utilizada no nosso projeto foi a Material UI, ela nos fornece diversos componentes prontos, no qual precisamos apenas passar algumas propriedades conforme a necessidade de cada desenvolvimento, nosso projeto teve muito mais agilidade devido a isso, pois não precisamos ficar criando componentes do zero, ela também facilita a criação de sites responsivos, não foi o foco do nosso projeto, mas caso quiséssemos seria muito fácil fazer uma adaptação.

**Estrutura do projeto**

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

**Diretorio de api**

Na pasta de api temos todos as funções que foram implementadas para realizar requisições para o back-end, cada diretório contém chamadas de acordo com sua entidade, então a pasta “job” por exemplo contém todas as chamadas de API que tem relação com os trabalhos.

A screenshot of a computer

Description automatically generated

**Pasta de assets**

A black rectangular object with white text

Description automatically generated

A pasta de assets do projeto contém todos os arquivos de midia que irão ser utilizadas no projeto, como imagens, aúdios, vídeos, gif, figurinhas etc.

**Pasta de components**

A black background with white text

Description automatically generated

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

**Pasta de contexts**

A black rectangular object with a white border

Description automatically generated with medium confidence

A pasta de contexts possuí contextos que podem ser utilizados em qualquer parte da aplicação, com ela podemos salvar dados e funções que serão reutilizados, ela possuí uma vantagem muito grande comparado com passar dados via componentes, pois se fossemos fazer pelo componentes teríamos que ter uma hierarquia de propriedades enorme, o que tornaria a manutenção da mesma muito complicada, também tornaria o código frágil fazendo com que caso um componente no meio da arvore quebre, façam com que todos os outros que estão abaixo dele também pare de funcionar, no nosso caso utilizamos muito para tratar a parte de autenticação na interface, fornecendo todos os dados do usuário ou empresa logados, métodos para deslogar, tipo de usuário logado etc.

**Pasta de hooks**

A black rectangular object with a white stripe

Description automatically generated

Sobre a pasta de hooks, ela contém funcionalidades que podem ser reutilizadas sobre os componentes, hooks nada mais são que uma função facilitada que podem ser utilizados 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 nosso projeto só fossem acessadas após o usuário ou empresa terem feitos login.

**Pasta de models**

A screenshot of a computer

Description automatically generated

A pasta de models contém todas as estruturas de dados que foram mapeados durante o desenvolvimento da aplicação, ela contém tipos criados com Typescript, na qual contém todos os dados de uma estrutura, também possuí 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 aqui, criação de módelos no back-end etc.

**Pasta de styles**

A black rectangular object with white text

Description automatically generated

A pasta de styles contém tudo que seja relacionado a estilização global da aplicação já que estilizações locais de componentes podem ficar dentro da pasta deles ou no seu próprio arquivo, dentro do arquivo global.css podem conter códigos CSS que irão ser utilizados no projeto inteiro, então se quisermos alterar a estilização de uma tag span por exemplo, colocaríamos a forma da estilização dela nesse arquivo, já o arquivo theme.ts contém a estilização global relacionada ao framework Material UI, o mesmo além de nós fornece diversos componentes prontos, ele também nos dá a possibilidade de estilizarmos o mesmo, como alterarmos cores principais, tamanhos de fontes, as fontes etc.

**Pasta de utilities**

A screenshot of a computer

Description automatically generated

Essa pasta possuí todos os arquivos que contém funcionalidades uteis que podem ser usadas ao longo do projeto como por exemplo, formatação de datas, formatação de campos, utilidades para textos, etc. Qualquer coisa que é útil e pode ser reutilizada deve ficar dentro desse diretório.

**Pasta de views**

A screenshot of a computer

Description automatically generated

A pasta de views contém todas as páginas do site, nela contém diversos componentes e elas mesmo também são um componente, então na prática ela é o início de uma hierarquia de componentes e códigos HTML, CSS e Typescript. 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 items na qual deve mapear um componente com um texto que nada mais é a rota do site.

**Resultado**

O resultado do Brazil Carrer Hub foi um projeto que facilitou muito para a aplicação de vagas de todos que pretendem trabalhar no Brasil, sua interface trouxe uma experiência fácil para o usuário, sem fazer com que o usuário passe muito tempo procurando por um emprego, a interação visual do site com o servidor ficou muito boa, com vários pontos dinâmicos, sem problemas de performance etc. O projeto se tornou uma boa oportunidade de investimento para trazer mão de obra qualificada de qualquer lugar do mundo para o Brasil, assim como promover o trabalho para Brasileiros residentes no qual resulta em geração de empregos.

O projeto resolve o problema de falta de emprego em cidades pequenas pois dentro do site já possuí a possibilidade de criação de vagas remotas, fazendo com que uma pessoa com ou sem uma especialidade possa trabalhar de qualquer lugar.

Um outro problema que o site facilita é a comunicação, pois no cadastro do usuário eles podem registrar algum dado que a empresa possa vir a se comunicar, como e-mail, telefone, website etc. A empresa também pode fornecer alguns dados na qual o usuário pode pesquisar mais sobre a mesma.