DESAFIO 1

De maneira inicial, criei uma função com parâmetro str. Depois criei uma string vazia que recebe allyHub. Na sequência, utilizei FOR, onde i recebe o tamanho da string (str.length) e a cada iteração o i vai decrementar, de modo que allyHub a cada iteração vai recebendo 1 espaço, porém, de modo inverso, ou seja, de trás para frente.

DESAFIO 2

Inicialmente, comecei organizando a pasta src do arquivo criado pelo create-react-app, sendo importante para o projeto apenas os que deixei na pasta mesmo. O arquivo principal para o projeto utilizei o App.jsx, onde iniciei a execução do desafio organizando os componentes que minha UI teria, ou seja, os formulários tanto de dados pessoais quanto de destino (fiz essa primeira parte na própria quinta-feira, após finalizar o desafio 1 e investi 1h para pensar na estrutura que queria e iniciar com a criação do projeto, incluindo o respositório no GitHub.) Na sequência, sexta-feira, investi um total de 5h, entre a execução dos componentes e estilização pelo css. Assim, criei os inputs com placeholders na div formData sobre os dados pessoais e, com relação à div formLocale, pesquisei a biblioteca sobre o multiselect, pois nunca havia usado em algum projeto. Pesquisei, li a documentação, entendi a proposta e iniciei a execução no código fazendo o import dela. Para complementar especificamente essa div formLocale, criei duas const (países e cidades) com as propriedades que teriam que ser usadas na bilioteca do Select (value e label) e, assim, consegui aplicar a opção de multiselect. Nesse momento, antes de integrar a API, já que a estrutura do desafio estava bem caminhada, preferi dar atenção para o css e iniciei a estilização: utilizei cores de background, já que é imprescindível o frontend se colocar no lugar do usuário e produzir uma UI estilizada e que garanta conforto para o usuário. Após finalizar o código e deixá-lo pronto para entregar, revisei e pude notar que poderia ter deixado os formulários dentro de apenas uma div, pra aplicar as propriedades uma vez só. Após estilizar a UI, passei para o ponto que ainda tenho dificuldade e demorou mais tempo, que é utilizar a API, porém, com dedicação as coisas saem. Para finalizar a sexta integrei apenas a API country e deixei a city para sábado. Criei um arquivo constants e atribuí uma const API\_URL que recebe o url para utilizar no projeto e em app.jsx importei o axios e API\_URL. Esse momento do projeto se destinou à utilização de React Hooks, sendo que no caso, importei e utilizei o useState e useEffect. Criei duas const pra setar o country e a city pra receber useState e na sequência passei pro useEffect, e utilizei a requisição get para as rotas de city e country, sendo este o momento que o frontend precisa entender a documentação da API pra poder pegar e atribuir os valores corretamente e fazer a integração da API corretamente no código. Dentre os valores na API, verifiquei que é "data" e escolhi o "code" e o "name\_ptbr" para retornar o value e label. Utilizei o método map pra invocar a const data e retornar o array de resultado tanto pra city quanto pra country e pra finalizar peguei o setCity e setCountry pra retornar a const que mapeei pra retornar os resultados de city e country da API. Como falei antes, essa foi a parte mais tive dificuldade, principalmente para adequar a lógica com a API proposta. Investi no sábado mais 1h pra fechar a lógica e finalizar o código e mais 1h pra fechar a estilização. Acredito que consegui colocar em prática o que venho estudando e codando, bem como utilizando as práticas de cosume, como pesquisar e ir executando no código.

const paises = [

{ value: 'brasil', label: 'Brasil' },

{ value: 'eua', label: 'Estados Unidos' },

{ value: 'inglaterra', label: 'Inglaterra' },

{ value: 'africa do sul', label: 'África do Sul' },

];

const cidades = [

{ value: 'São Paulo', label: 'São Paulo' },

{ value: 'ny', label: 'New York' },

{ value: 'london', label: 'Londres' },

{ value: 'cidade do cabo', label: 'Cidade do Cabo' },

];