Relatório – Formulário React – 03/05/2023

Nome: Bruno do Nascimento Rodrigues

1- Objetivos

Fazer o mesmo formulário da atividade passada, porém dessa vez deve ser feito em React. Essa atividade deve conter, além da tela de cadastro, a tela de login do sistema, e que seja possível alternar entre elas.

2- Atividade

O primeiro passo feito foi criar a aplicação em react e transferir o que foi feito anteriormente em html e css para a aplicação.

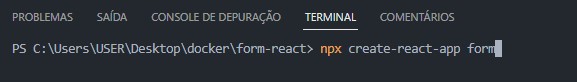


Figura 1 criando a aplicação react

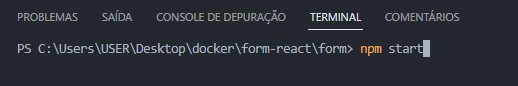


Figura 2 iniciado a aplicação

A primeira coisa a se observar é a estrutura de pasta construída após rodar o comando npx acima. O arquivo index.js está renderizando o que está no return da função App, no arquivo app.js.

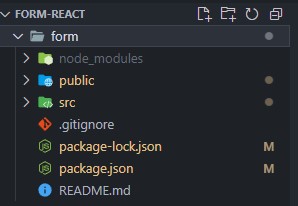


Figura 3 estrutura de pastas

Tendo isso em mente, foram criadas duas pastas dentro da pasta src, componentes e images, e seus respectivos arquivos field.js, form.js e imagens que serão utilizadas posteriormente.

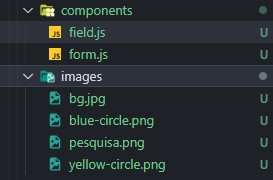


Figura 4 arquivos adicionados

Voltando ao arquivo App.js, foi excluído todo o conteúdo de dentro da <div className=”App”> e adicionado o componente <Form/> dentro da div. Observe ainda que é necessária a importação do componente para o arquivo em que estamos.

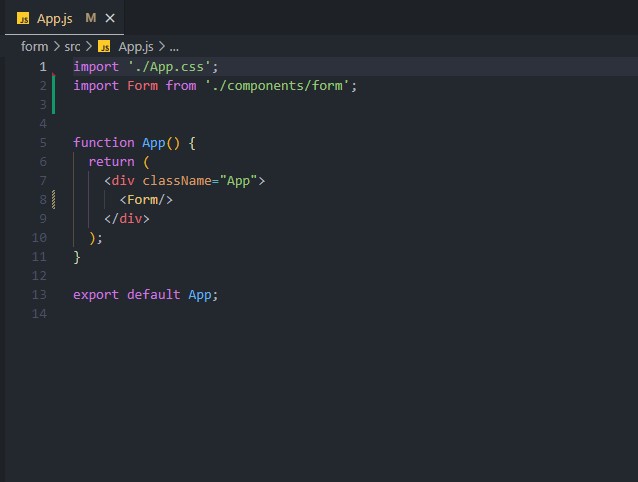


Figura 5 App.js

Para que o bootstrap funcione, vamos até a pasta public, no arquivo index.html e adicionar o link com o cdn no head.

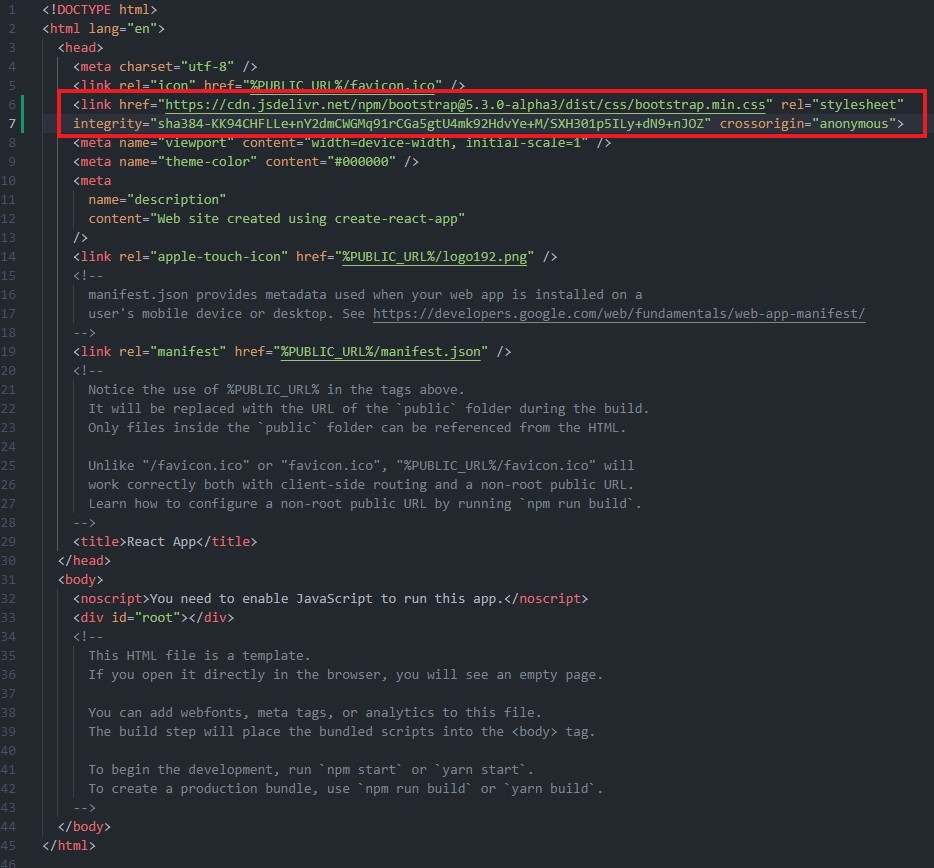


Figura 6 link bootstrap

Agora com tudo “pronto”, será feita a cópia do html e css da atividade anterior para mantermos o mesmo design.

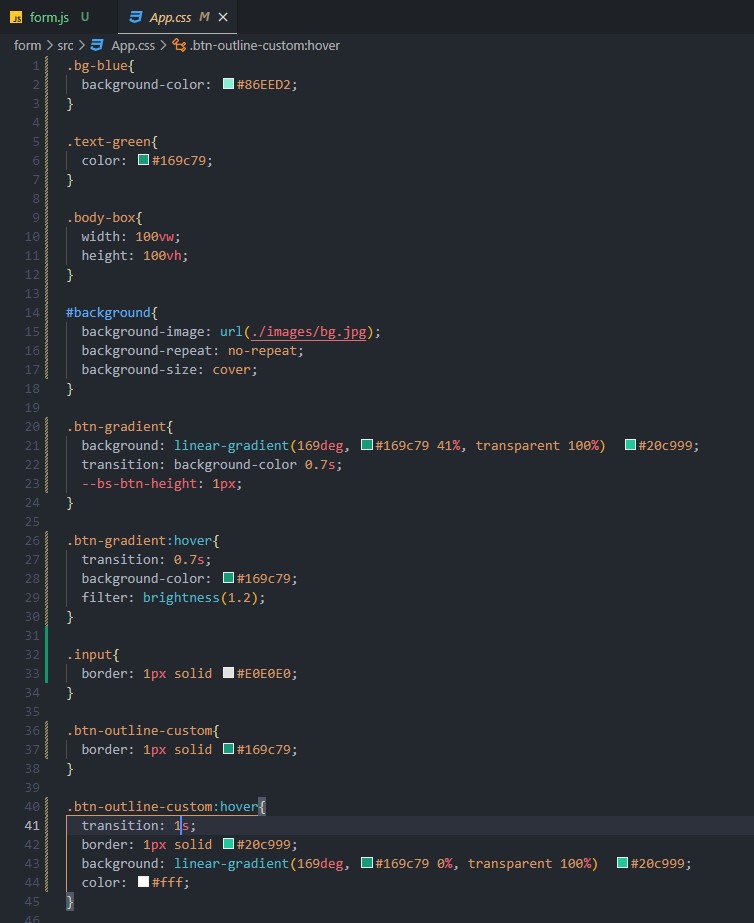


Figura 7 arquivo css

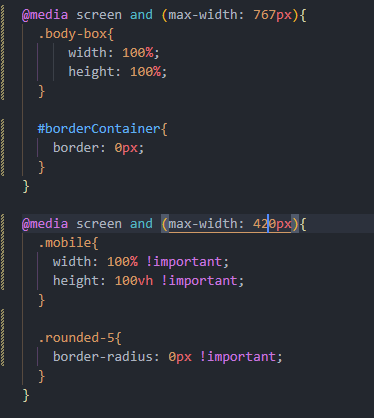


Figura 8 continuação css

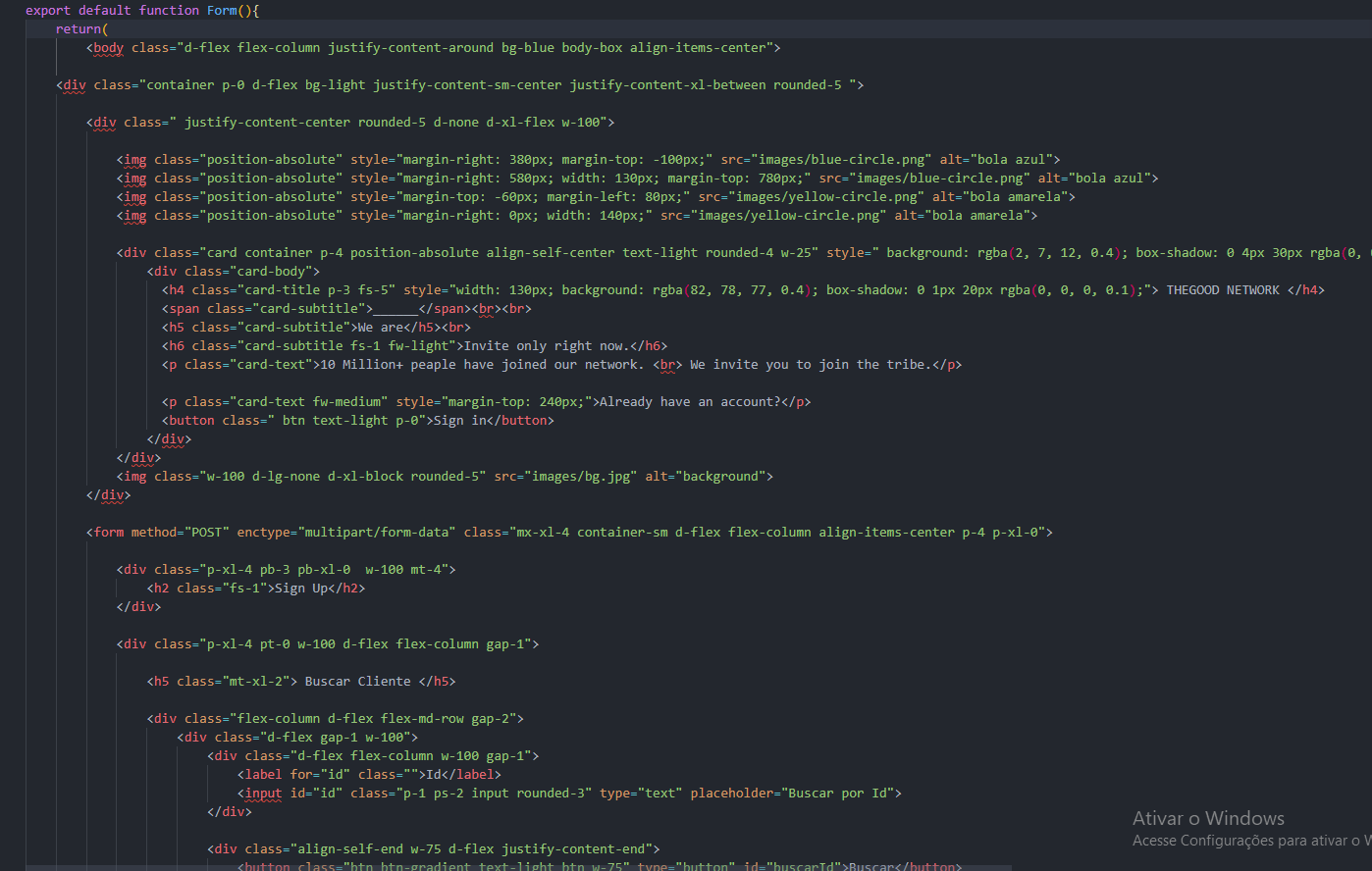


Figura 9 html copiado

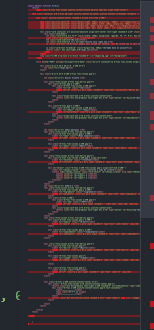


Figura 10 erros

Ao colar o html, vamos ter diversos erros, isso é normal pois o JSX exige o fechamento de todas as tags, diferentemente do html e também devemos alterar a class para className, pois class é reservada pelo react. Além disso, se faz necessário importar as imagens para o arquivo, e usar o style da forma correta no JSX.

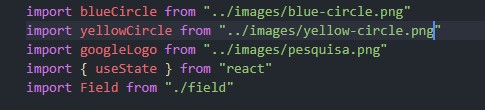


Figura 11 importação das imagens

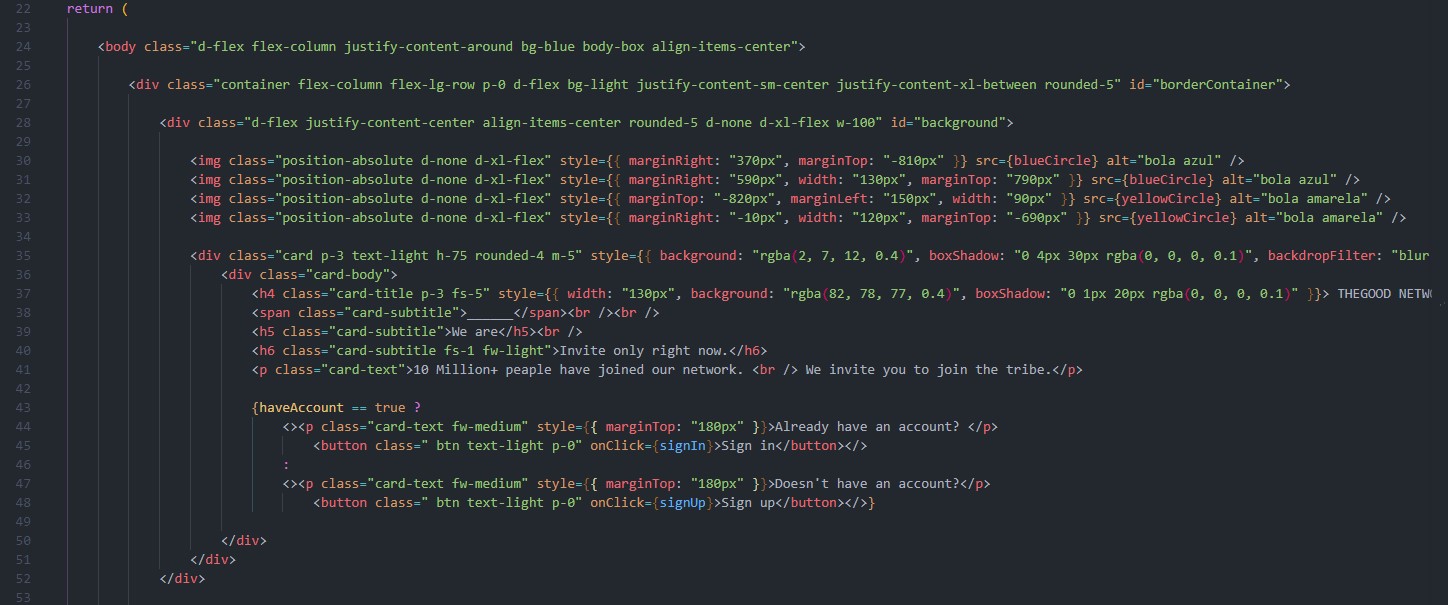


Figura 12 componente form 1/3

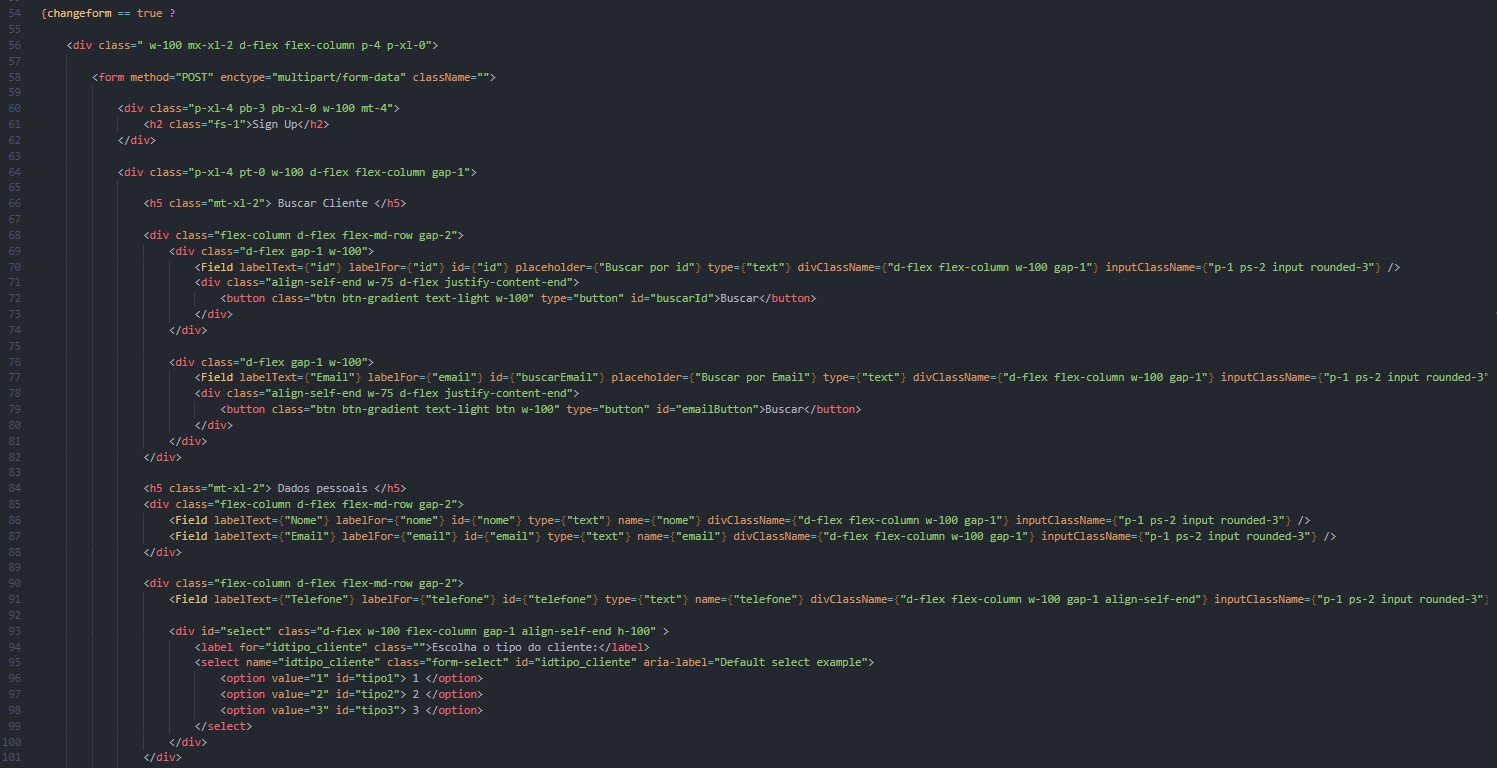


Figura 13 componente form 2/3

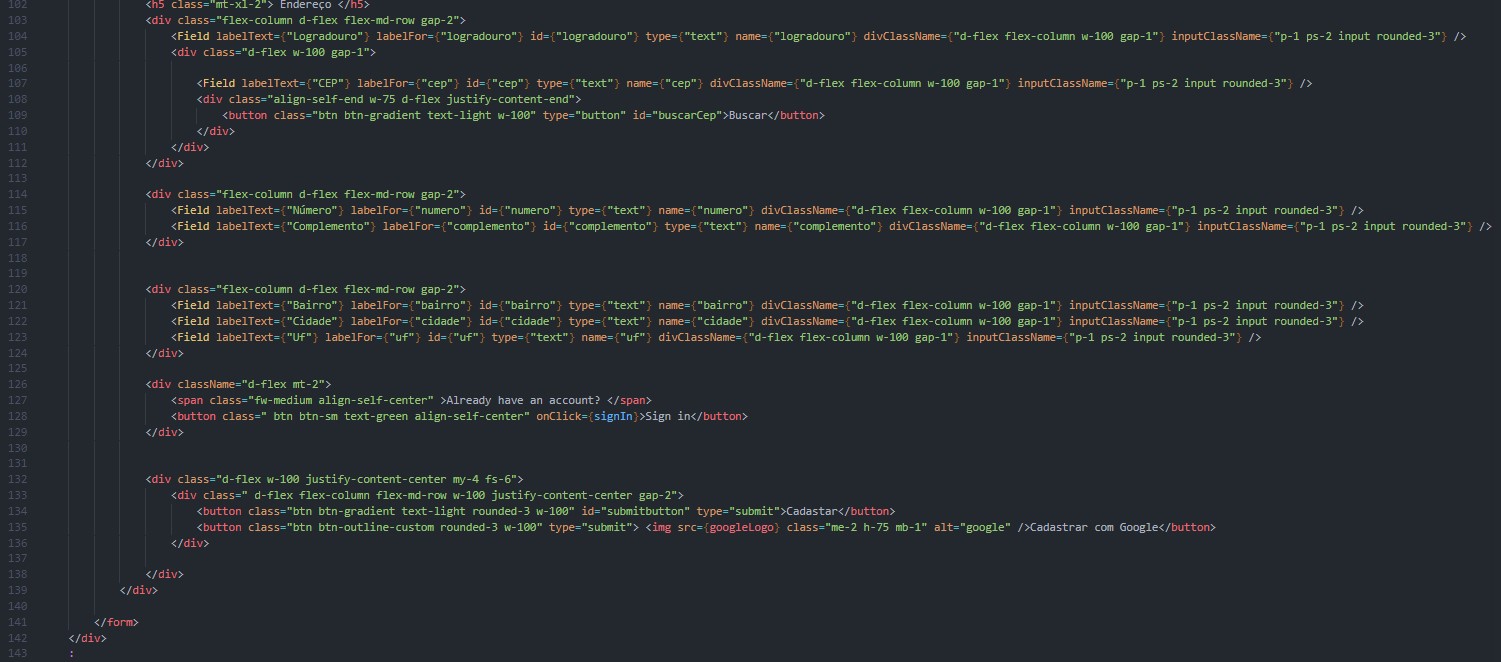


Figura 14 componente form 3/3

Como as capturas de tela foram feitas posteriormente ao trabalho finalizado, existem coisas diferentes que serão explicadas. Na figura 11 e na figura 12 foram feitas renderizações condicionais. Caso condição atendida? renderizar 1 : renderizar 2, sendo os “ : ” caso a condição seja falsa.

Para que funcionem, é necessário criar as consts changeform e haveAccount. Poderiam ter qualquer outro nome, e duas funções signIn e signUp.

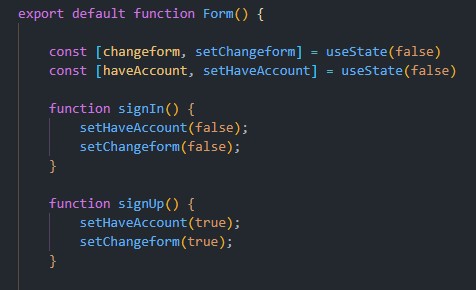


Figura 15 States

O componente <Field/> é o campo do formulário, que, por ser feito como um componente permite que o mesmo trecho JSX, crie vários campos de forma flexível, pois recebe as propriedades do componente.

Foi utilizado o formato de escrever as propriedades que serão recebidas dentro da função, ao invés do “props” somente. Dessa forma acredito que fique mais legível o código para a leitura.

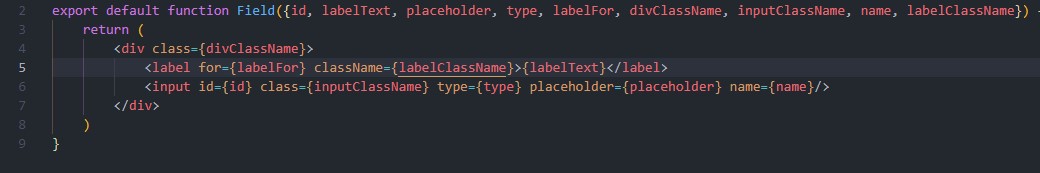


Figura 16 Field.js

Com isso finalizado e adicionando o componente substituindo os campos no arquivo form.js ficou finalizado a atividade do formulário em react. Clicando nos botões sign in e sign up dentro do form ou o que está na parte de baixo da imagem, é possível alternar entre os dois formulários, de cadastro e de login.

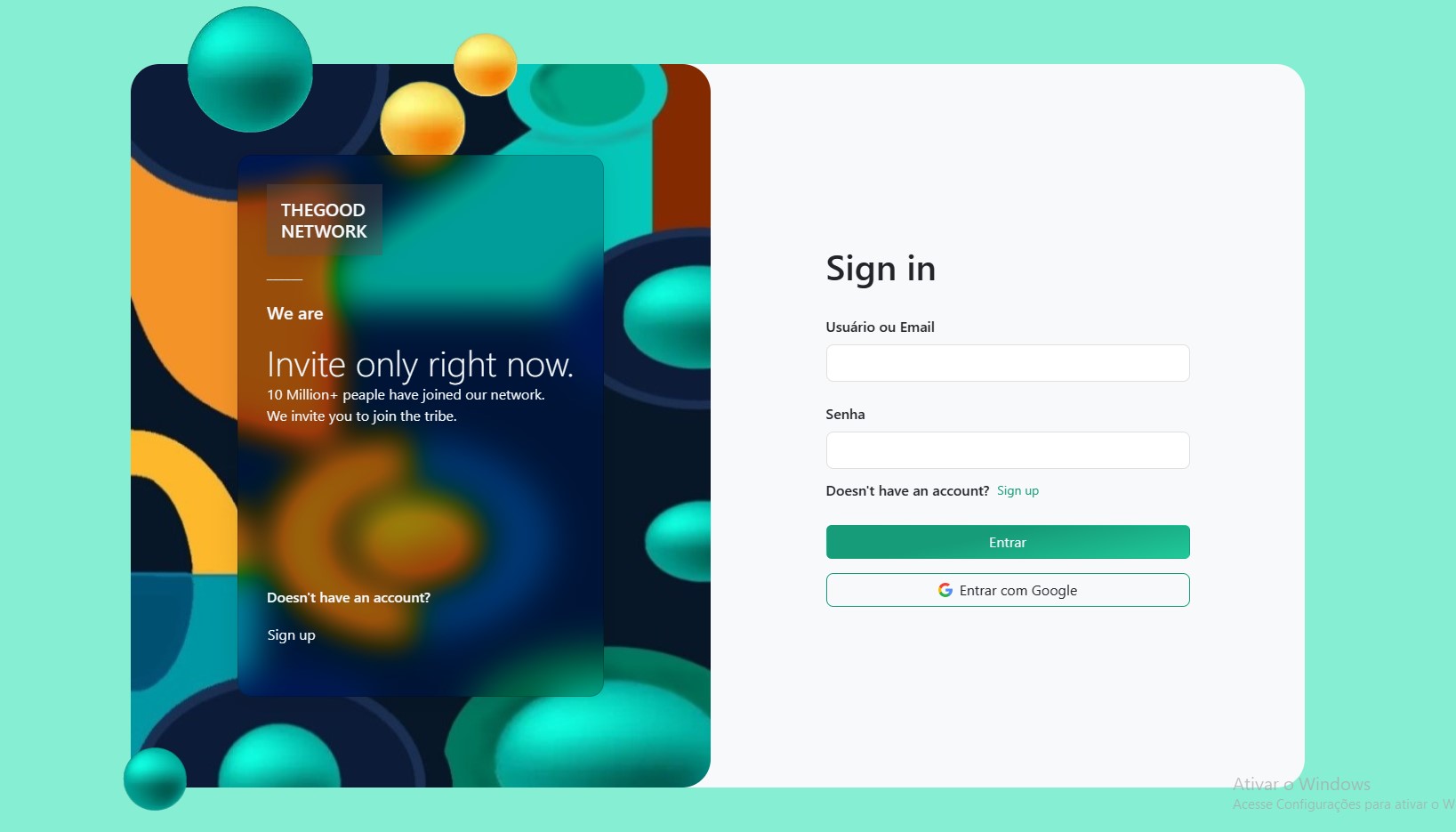


Figura 17 form Sign in

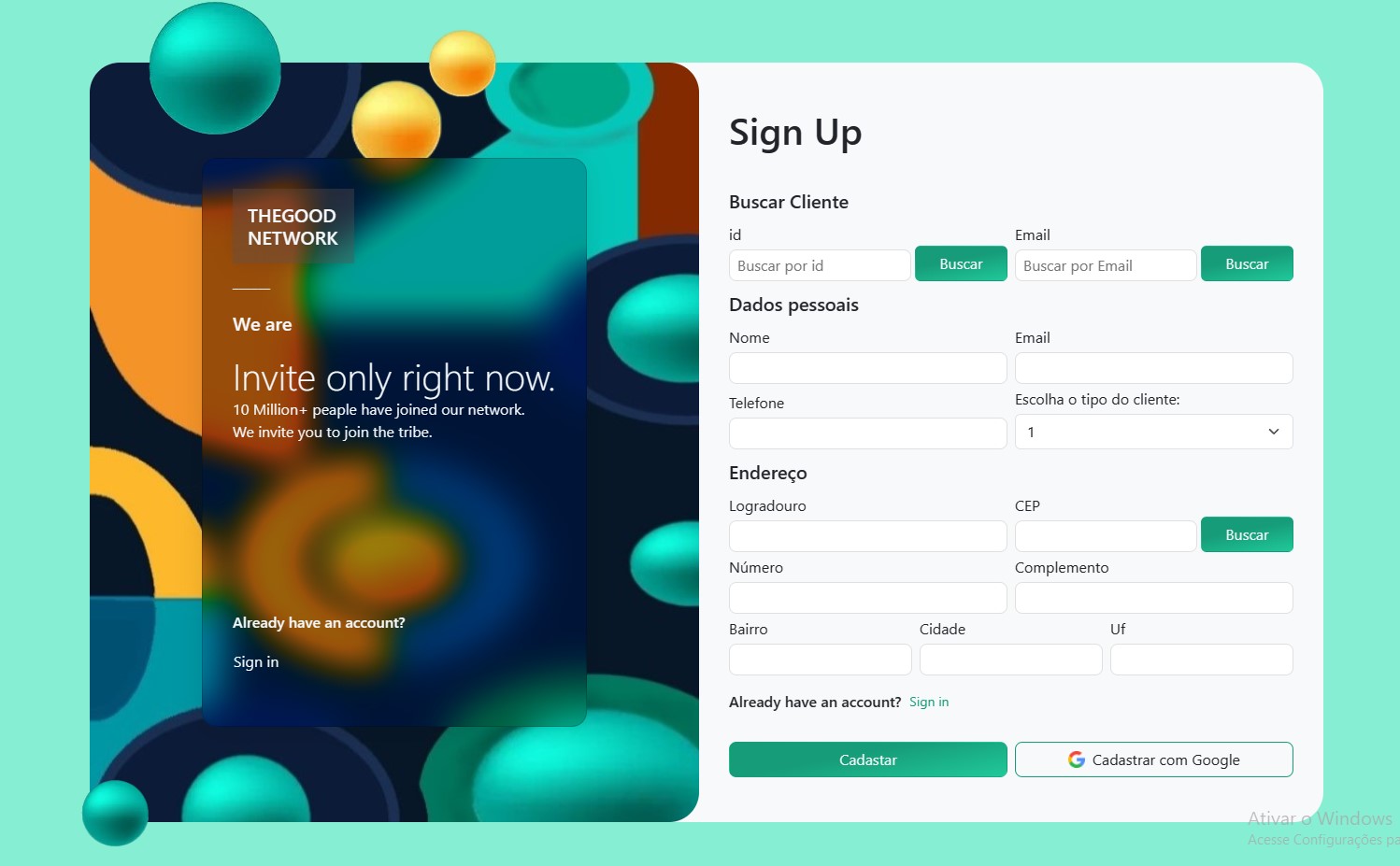


Figura 18 form Sign up

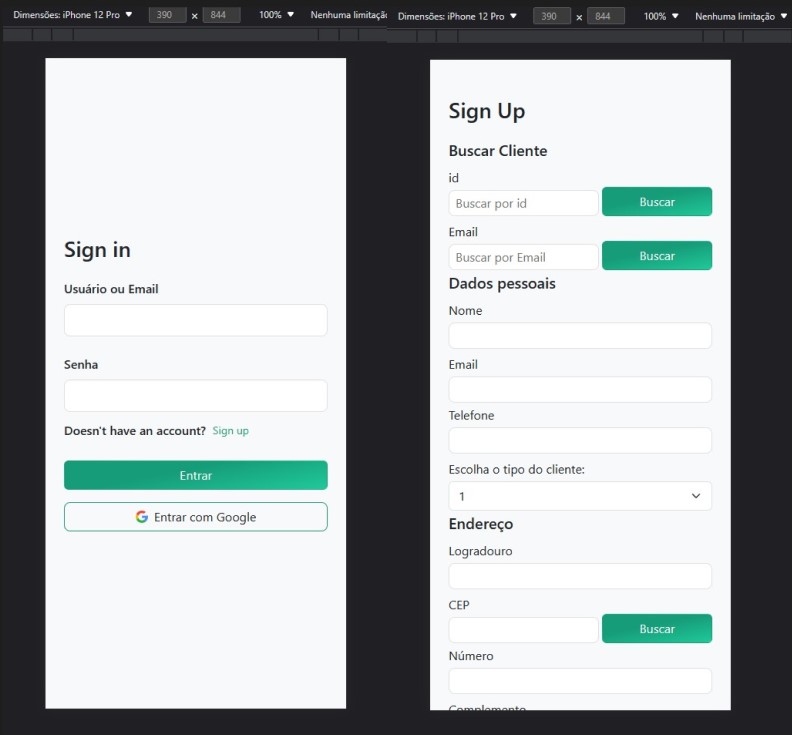


Figura 19 form em telas mobiles

3-Atualizações

Com o propósito de melhorar o código e dividir melhor entre os componentes, foram criados 3 novos arquivos para dividir o formulário de cadastro, login e também o card lateral ao formulário, tornando assim um código mais limpo de ser lido.

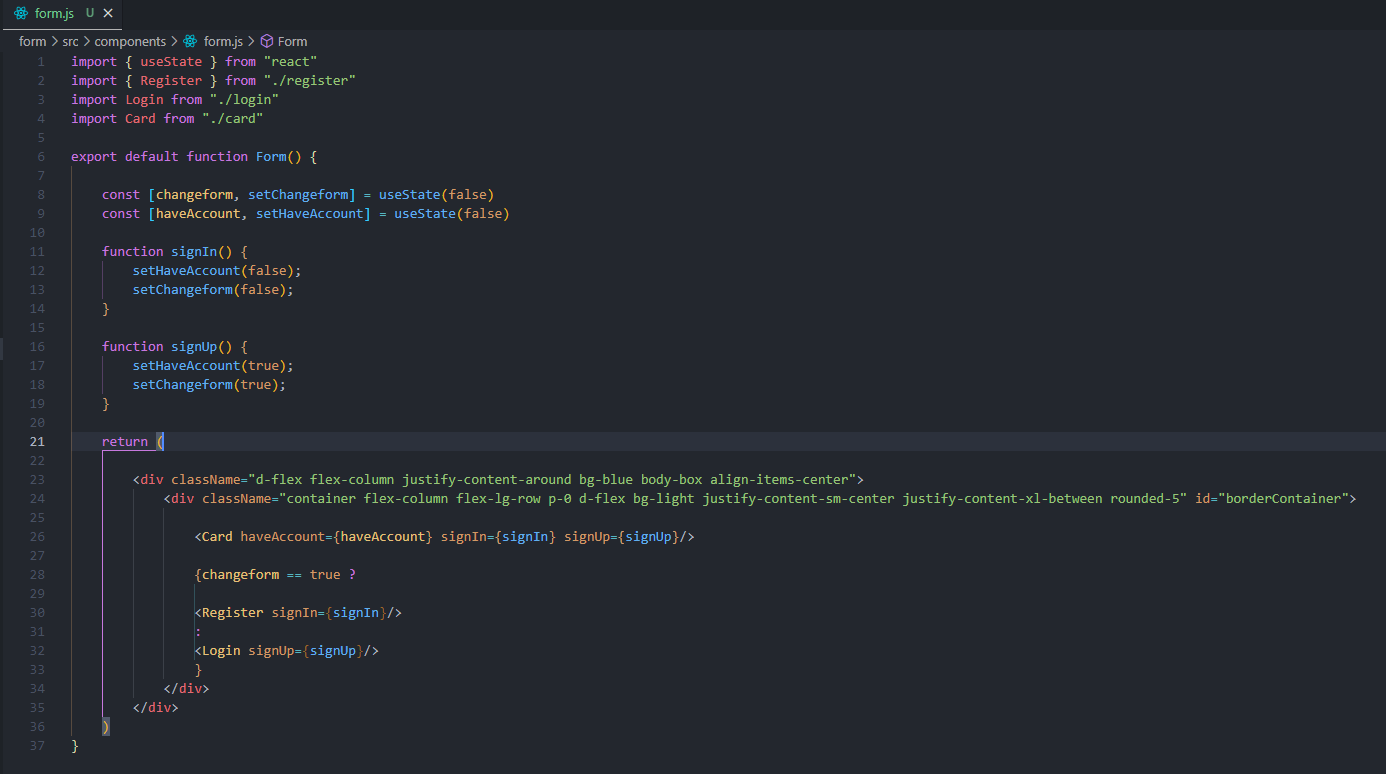


Figura form.js atualizado

Nessa nova atualização, estou passando a função e o state requerido como propriedades dos componentes criados, podendo assim, ter acesso a eles em outro arquivo.

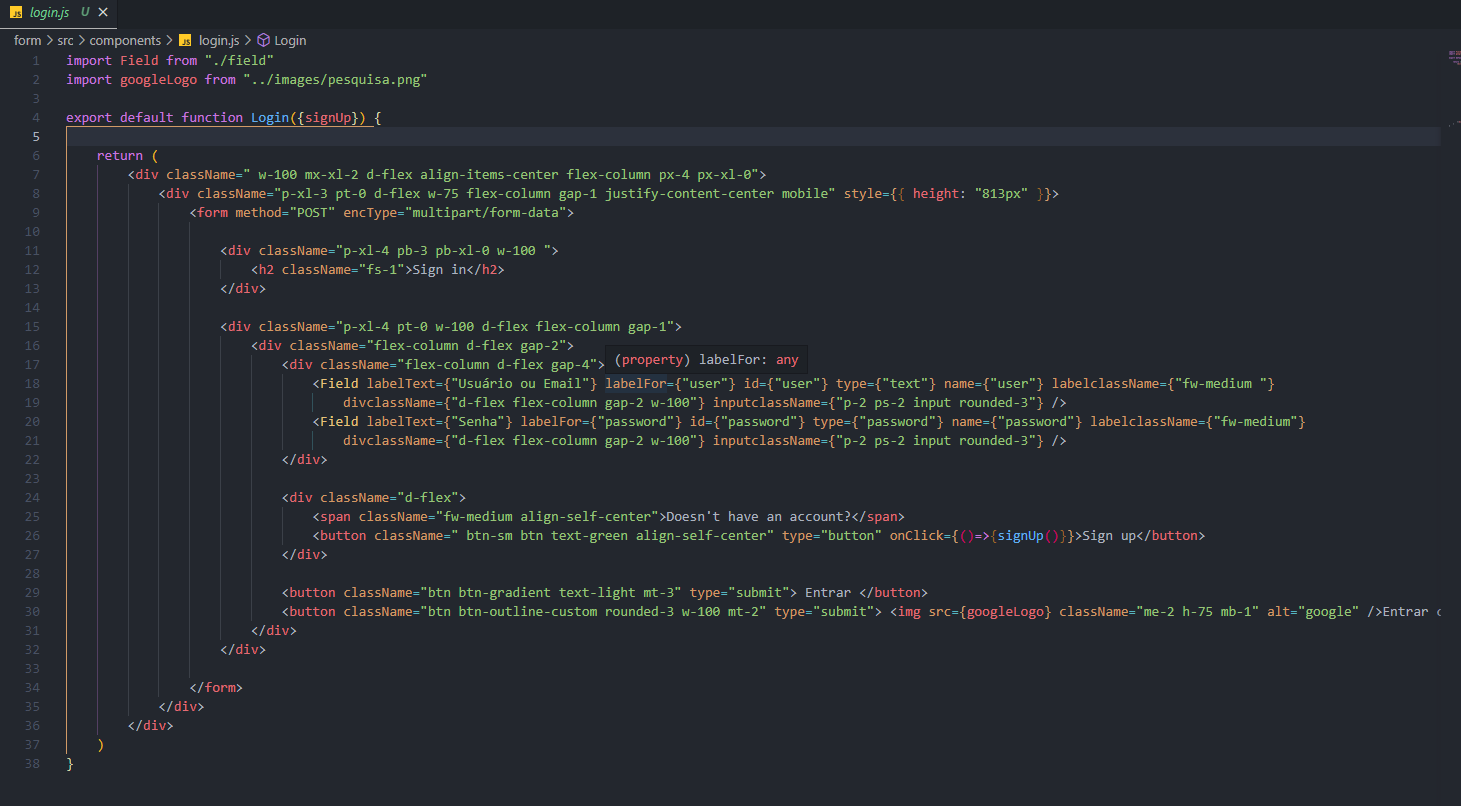


Figura login.js

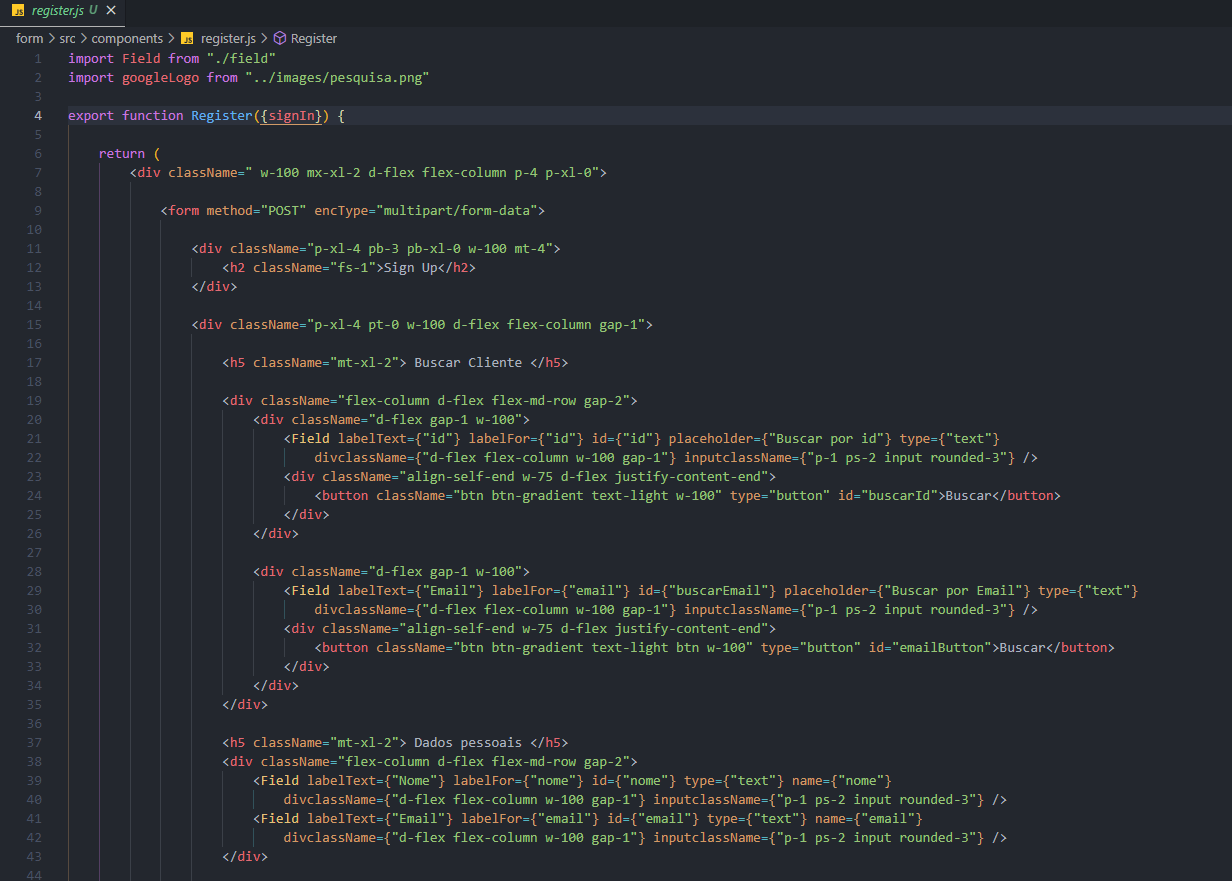


Figura register.js



Figura card.js

4- Conclusão

Com o trabalho concluído, podemos destacar que foi uma atividade interessante, dar um passo à frente e utilizar novas tecnologias é sempre bom e engrandecedor. Perceber a facilidade para mudar a composição da tela com o uso do hook useState e de componentes facilita bastante o trabalho.