

Configurações e códigos simples de  
frontend com React no Linux

Ubuntu 20.04

## **Instalando o REACT:**

**Terminal: (na pasta que será seu projeto)**

\$ npm install -g create-react-app

### **Pacotes**

#### **Axios**

\$ yarn install axios (ou yarn add axios)

#### **Router Dom**

\$ npm install react-router-dom

## Criando seu projeto de front

1º em uma pasta desejada aberta no terminal digite:

```
$ create-react-app frontend
```

```
$ cd frontend
```

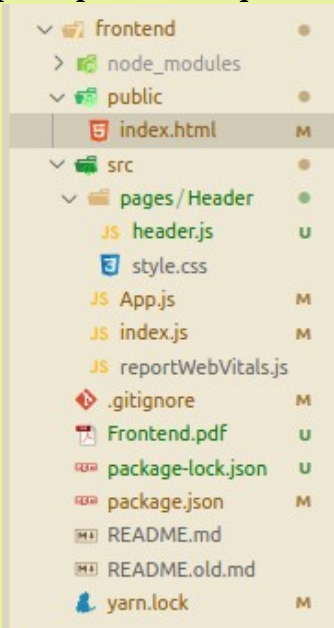
```
$ npm install yarn
```

```
$ yarn start
```

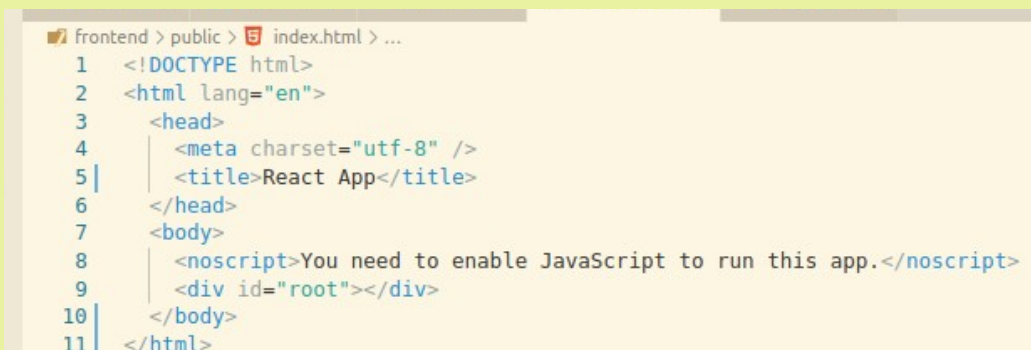
2º adicione as pastas frontend criada e o repositório backend no mesmo workspace do VS

## Limpando o projeto com o que não irá usar

1º no VS, remova arquivos desnecessários de frontend, as imagens, o txt, o ico e o manifest da pasta public, o arquivo de teste, setup, css e e o svg da pasta src



2º de index.html remova os imports de css e limpe o código



3º de App.js remover o conteúdo da div e os imports de arquivos excluídos



## Criando um novo componente

1º crie uma pasta chamada pages em src

2º crie uma pasta chamada header dentro de pages

3º crie um arquivo .js dentro da pasta header

4º crie um arquivo .css

5º adicione o código no .js criado:

```
frontend > src > pages > Header > JS header.js > ...
1  import React from "react";
2  import './style.css';
3
4  export default function Header(){
5    return(
6      <header id = "main-header">
7        | <h1>Primeiro app</h1>
8        | </header>
9      )
10   };
11 }
12
```

6º crie seu css

```
frontend > src > pages > Header > style.css > ...
1  header#main-header{
2    width: 100%;
3    height: 50px;
4    background: coral;
5    font-size: 15px;
6    font-weight: 200;
7    color: hsl(180, 100%, 27%);
8    display: flex;
9    justify-content: center;
10   align-items: center;
11 }
```

7º em index.js faça o import do arquivo .js criado e chame seu arquivo .js adicionando antes ou após <App />:

```
frontend > src > JS index.js
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import App from './App';
4  import Header from './pages/Header/header';
5  import reportWebVitals from './reportWebVitals';
6
7  ReactDOM.render(
8    <React.StrictMode>
9    | <App />
10   | <Header />
11   </React.StrictMode>,
12   document.getElementById('root')
13 );
14
15 // If you want to start measuring performance in your app, pass a function
16 // to log results (for example: reportWebVitals(console.log))
17 // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
18 reportWebVitals();
19
```

### **Ex. passando por parametro**

→ *em index.js*

```
<Nome title= “Meu app”/>
```

→ *no seu arquivo .js criado*

```
export default function Header(param){  
  return(  
    <h1>{param.title}</h1>  
  )  
}
```

### **Ex. utilizando variável que se modifica no html, ex. Contador**

*em um arquivo .js (como o seu criado ou App.js)*

#### **1. jeito que não funciona como esperado:**

```
function App(){  
  let counter = 0;  
  function increment(){  
    counter++;  
    console.log(counter);  
  }  
  
}
```

#### **2. jeito que funciona**

```
import {useState} from 'react';
```

```
function App(){  
  const [counter, setCounter] = useState(0); //inicializando em 0  
  function increment(){  
    setCounter(counter + 1);  
  }  
  
  return {  
    <div>contador: {counter}</div>  
  
    <button onClick={increment}>Incremento</button>  
  }  
}
```

**Ao final do teste, exclua a pasta header e remova o import e <Header /> de index.js**

## Integrando backend com o frontend utilizando o Axios

1º crie uma pasta em src chamada services

2º crie um arquivo api.js e adicione o seguinte código: (usado para fazer a comunicação)

```
frontend > src > services > JS api.js > ...
1  import axios from 'axios';
2
3  const api = axios.create({baseUrl: 'http://localhost:3001'})
4
5  export default api;
6
```

3º em pages crie uma pasta chamada Users

4º em App.js o código fica assim:

```
frontend > src > JS App.js > ...
1  import React from "react";
2  import User from './pages/Users/users';
3
4  function App() {
5    return (
6      <div className="App">
7        <User/>
8      </div>
9    );
10 }
11
12 export default App;
13
```

5º na pasta Users crie um arquivo users.js e adicione o seguinte código:

```
frontend > src > pages > Users > JS users.js > ...
1  import React,{useState,useEffect} from "react";
2  import api from '../services/api';
3  import './style.css';
4
5  export default function User(){
6    const [users, setUsers] = useState([]);
7    useEffect(()=>{
8      api.get('users').then(response =>{
9        setUsers(response.data);
10      });
11    },[])
12
13    console.log(users);
14    return(
15      <div id="user-container">
16        <h1>Lista de usuários</h1>
17        <ul className="user-list">
18          {users.map(user =>(
19            <li key={user.id}>
20              <strong>Nome</strong>
21              <p>{user.name}</p>
22              <strong>Email</strong>
23              <p>{user.email}</p>
24              <strong>Idade</strong>
25              <p>{user.idade}</p>
26              <strong>Empresa</strong>
27              <p>{user.empresa}</p>
28            </li>
29          ))}
30        </ul>
31      </div>
32    );
33  };
34 }
```

6º na pasta Users crie um arquivo style.css e adicione o seguinte código:

```
frontend > src > pages > Users > style.css > ...
1  #user-container h1{
2      margin-top: 10px;
3      color: #da552f;
4      display: flex;
5      justify-content: center;
6      align-items: center;
7  }
8
9  .user-list{
10     max-width: 700px;
11     margin: 20px auto 0;
12     padding: 0 20px;
13     list-style: none;
14 }
15
16 .user-list li{
17     background: #fff;
18     border: 1px solid #ddd;
19     border-radius: 5px;
20     padding: 20px;
21     margin-bottom: 20px;
22 }
23
24 .user-list li p{
25     font-size: 16px;
26     color: #999;
27     margin-top: 5px;
28     line-height: 24px;
29 }
30
```



## Rotas, Navegação e CRUD

1º crie um arquivo routes.js em src

2º adicione o seguinte código:

```
Frontend > src > JS routes.js > ...
1 import React from 'react';
2 import {BrowserRouter, Route, Switch} from 'react-router-dom';
3 import User from './pages/Users';
4
5 export default function Routes(){
6   return(
7     <BrowserRouter>
8       <Switch>
9         <Route path = '/' component = {User}/>
10      </Switch>
11    </BrowserRouter>
12  );
13 }
```

3º em App.js, ao invés de chamar Users, chame o routes:

4º crie em pages uma pasta chamada Profile e um arquivo chamado index.js e adicione o seguinte código:

```
Frontend > src > pages > Profile > JS index.js > ...
1 import React, {useState, useEffect} from 'react';
2 import './style.css';
3 import {Link, useHistory, useParams} from 'react-router-dom';
4 import api from '../../services/api';
5
6 export default function Profile(){
7   const {id} = useParams();
8   const history = useHistory();
9
10  //criação de usuário vazio
11  const initUser={
12    name: '',
13    email: '',
14    idade: 0,
15    empresa: '',
16  }
17
18  const [user, setUser] = useState(initUser);
19
20  //get para alteração de usuários
21  useEffect(() => {
22    if(id) {
23      api.get(`/users/${id}`).then(response => {
24        setUser(...response.data);
25      });
26    }
27  }, [id]);
28 }
```

← 1

2 →

```
Frontend > src > pages > Profile > JS index.js > Profile
29
30 function onSubmit(ev){
31   ev.preventDefault(); //pagina não regarrega novamente
32
33   //definindo se é create ou update para o método e url
34   const method = id ? 'put' : 'post';
35   const url = id ? `users/${id}` : '/users';
36
37   api[method](url, user).then((response) =>{
38     history.push('/');
39   });
40
41   //mudando valores [create/update]
42   function onChange(ev){
43     const {name, value} = ev.target;
44     setUser({...user, [name]:value});
45     console.log(user);
46   }
47 }
```

3 →

```
Frontend > src > pages > Profile > JS index.js > Profile
47
48 return(
49   <div id = "profile-container">
50     <h1>Cadastro</h1>
51     <form onSubmit = {onSubmit}>
52       <strong>Nome:</strong>
53       <input name = "name" onChange = {onChange} value = {user.name}/>
54       <strong>Email:</strong>
55       <input type = "email" name = "email" onChange = {onChange} value = {user.email}/>
56       <strong>Idade:</strong>
57       <input name = "idade" onChange = {onChange} value = {user.idade}/>
58       <strong>Empresa:</strong>
59       <input name = "empresa" onChange = {onChange} value = {user.empresa}/>
60
61       <div className = "actions">
62         <Link className = "button" onClick = {} => history.push('/')>Voltar</Link>
63         <button className = "button" type = "submit">Salvar</button>
64       </div>
65     </form>
66   </div>
67 );
68
69 }
```



6º crie um arquivo style.css na pasta Profile e adicione o seguinte código:

```
frontend > src > pages > Profile > style.css > ...
1  #profile-container {
2    max-width: 700px;
3    margin: 20px auto 0;
4    padding: 0 20px;
5  }
6
7  #profile-container h1 {
8    margin-top: 10px;
9    color: #da552f;
10   display: flex;
11   justify-content: center;
12   align-items: center;
13 }
14
15 #profile-container form input {
16   width: 400px;
17   font-size: 16px;
18   display: flex;
19   background: #fff;
20   border: 1px solid #ddd;
21   border-radius: 5px;
22   padding: 10px;
23   margin-bottom: 20px;
24 }
```

```
frontend > src > pages > Profile > style.css > ...
25
26 .button {
27   height: 30px;
28   width: 80px;
29   border-radius: 5px;
30   border: 2px solid #da552f;
31   background: none;
32   margin-top: 10px;
33   color: #da552f;
34   font-weight: bold;
35   font-size: 16px;
36   display: flex;
37   justify-content: center;
38   align-items: center;
39 }
40
41 .button:hover {
42   background: #da552f;
43   color: #fff;
44 }
45
46 .actions {
47   display: flex;
48   justify-content: space-between;
49   margin-bottom: 20px;
50   padding: 10px;
51 }
52
```

7º em routes.js adicione:

```
frontend > src > JS routes.js > ...
1  import React from 'react';
2  import {BrowserRouter, Route, Switch} from 'react-router-dom';
3  import User from './pages/Users/users';
4  import Profile from './pages/Profile/index';
5
6  export default function Routes(){
7    return(
8      <BrowserRouter>
9        <Switch>
10         <Route exact path = '/' component = {User}/>
11         <Route path = '/create' component = {Profile}/>
12         <Route path = '/update/:id' component = {Profile}/>
13       </Switch>
14     </BrowserRouter>
15   );
16 }
17
18
```

8º na pasta users, em users.js modifique o código para:

```
frontend > src > pages > Users > JS users.js > User > users.map() callback
1 import React,{useState,useEffect} from "react";
2 import api from '../../services/api';
3 import './style.css';
4 import {Link} from 'react-router-dom';
5
6 export default function User(){
7   const [users, setUsers] = useState([]);
8   useEffect(()=>{
9     api.get('users').then(response =>{
10       setUsers(response.data);
11     });
12   },[])
13
14   async function handleDelete(id){
15     try{
16       await api.delete(`/users/${id}`);
17       setUsers(users.filter(user => user.id !== id));
18     }catch(err){
19       alert("Erro ao deletar!");
20     }
21   }
22
23   return(
24     <div id="user-container">
25       <h1>Lista de usuários</h1>
26       <Link className = "button" id = "create-link" to = {"/create"}>Criar</Link>
27
28       <ul className="user-list">
29         {users.map(user => (
30           <li key={user.id}>
31             <strong>Nome</strong>
32             <p>{user.name}</p>
33             <strong>Email</strong>
34             <p>{user.email}</p>
35             <strong>Idade</strong>
36             <p>{user.idade}</p>
37             <strong>Empresa</strong>
38             <p>{user.empresa}</p>
39
40             <div className = "actions">
41               <button className = "button" onClick = {()=> handleDelete(user.id)} type = "button" >Deletar</button>
42               <Link className = "button" to = {`/update/${user.id}`} >Acessar</Link>
43             </div>
44           </li>
45         ))}
46       </ul>
47     </div>
48   );
49
50 }
```

9º na pasta users, no arquivo .css adicione o seguinte código:

```
frontend > src > pages > Users > style.css > .button:hover
31 .button {
32   height: 30px;
33   width: 80px;
34   border-radius: 5px;
35   text-decoration: none;
36   border: 2px solid #da552f;
37   background: none;
38   margin-top: 10px;
39   color: #da552f;
40   font-weight: bold;
41   font-size: 16px;
42   display: flex;
43   justify-content: center;
44   align-items: center;
45 }
46
47 .button:hover {
48   background: #da552f;
49   color: #fff;
50 }
51
52 .actions {
53   display: flex;
54   justify-content: space-between;
55   margin-bottom: 20px;
56   padding: 10px;
57 }
58
59 #create-link {
60   margin: 50px;
61 }
```