

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.js remova os imports de css

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:

```
import React from 'react';  
import './seu-arquivo.css';
```

← Os importantes

```
export default function Header(){  
  return(  
    seu código html  
  })  
}
```

6º crie seu css

7º em index.js faça o import do arquivo .js criado

```
import Nome from './pages/header/seu-arquivo';
```

8º chame seu arquivo .js adicionando antes ou após <App />:

```
<Nome />
```

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>  
}
```

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)

```
import axios from 'axios';

const api = axios.create({baseUrl:'http://localhost:3001'})

export default api;
```

3º em pages crie uma pasta chamada Users

4º em App.js o código fica assim:

```
import {useState} from 'react';
import User from './pages/Users';

function App() {
  return (
    <div className="App">
      <User/>
    </div>
  );
}
```

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

```
import React,{useState, useEffect} from "react";

import api from '../services/api';

import './style.css';

export default function User(){
  const [users, setUsers] = useState([]);
  useEffect(()=>{
    api.get('users').then(response =>{
      setUsers(response.data);
    });
  },[])

  console.log(users);

  return(
    <div id="user-container">
      <h1>Lista de usuários</h1>
      <ul className="user-list">
        {users.map(user =>(
          <li key={user.id}>
            <strong>Nome</strong>
            <p>{user.name}</p>
          )
        )}
      </ul>
    </div>
  )
}
```

```

        <strong>Email</strong>
        <p>{user.email}</p>

        <strong>Empresa</strong>
        <p>{user.empresa}</p>

        <strong>Idade</strong>
        <p>{user.idade}</p>
    </li>
  </ul>
  </div>
);
}

```

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

```

#user-container h1{
  margin-top: 10px;
  color: #da552f;
  display: flex;
  justify-content: center;
  align-items: center;
}

.user-list{
  max-width: 700px;
  margin: 20px auto 0;
  padding: 0 20px;
  list-style: none;
}

.user-list li{
  background: #fff;
  border: 1px solid #ddd;
  border-radius: 5px;
  padding: 20px;
  margin-bottom: 20px;
}

.user-list li p{
  font-size: 16px;
  color: #999;
  margin-top: 5px;
  line-height: 24px;
}

```

Rotas, Navegação e CRUD

1º crie um arquivo routes.js em src

2º adicione o seguinte código:

```
import React from 'react';
import {BrowserRouter, Route, Switch} from 'react-router-dom';
import User from './pages/Users';

export default function Routes(){
  return(
    <BrowserRouter>
      <Switch>
        <Route path = '/' component = {User}/>
      </Switch>
    </BrowserRouter>
  );
}
```

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

```
import {useState} from 'react';
import Routes from './routes';

function App() {
  return (
    <div className="App">
      <Routes/>
    </div>
  );
}
```

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

5º adicione o seguinte código:

```
import React, {useState} from 'react';
import './style.css';
import {Link, useHistory, useParams} from "react-router-dom";
import api from '../services/api';

export default function Profile(){
  const {id} = useParams();
  const history = useHistory();

  //criação de usuários
  const initUser={
    name: '',
    email: '',
    idade: 0,
    empresa: '',
  }

  const [uset, setUser] = useState(initUser);
```

```

function onChange(ev){
    const {name, value} = ev.target;
    setUser({...user, [name]:value});
    console.log(user);
}

//alteração de usuários
useEffect(() => {
    if(id) {
        api.get(`/users/${id}`).then(response => {
            setUser(...response.data);
        });
    }
}, []);

function onSubmit(ev){
    ev.preventDefault();

    const method = id ? 'put' : 'post';
    const url = id ? `/users/${id}` : 'users';

    api[method](url, user).then((response) =>{
        history.push('/');
    });
}

return(
    <div id = "profile-container">
        <h1>Cadastro</h1>
        <form onSubmit = {onSubmit}>
            <strong>Nome:</strong>
            <input name = "name" onChange = {onChange} value =
{user.name}/>

            <strong>Email:</strong>
            <input type = "email" name = "email" onChange = {onChange} value
= {user.email}/>

            <strong>Idade:</strong>
            <input name = "idade" onChange = {onChange} value =
{user.idade}/>

            <strong>Empresa:</strong>
            <input name = "empresa" onChange = {onChange} value =
{user.empresa}/>

            <div className = "actions">
                <Link className = "button" onClick = {} =>
history.push('/')>Voltar</Link>
                <button className = "button" type =
"submit">Salvar</button>
            </div>
        </form>
    </div>
)

```



```
        </div>
    </form>
</div>
);
}
```

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

```
#profile-container{
    max-width: 700px;
    margin: 20px auto 0;
    padding: 0 20px;
}

#profile-container h1{
    margin-top: 10px;
    color: #da552f;
    display: flex;
    justify-content: center;
    align-items: center;
}

#profile-container form input{
    width: 400px;
    font-size: 16px;
    display: flex;
    background: #fff;
    border: 1px solid #ddd;
    border-radius: 5px;
    padding: 10px;
    margin-bottom: 20px;
}

.button{
    height: 30px;
    width: 80px;
    border-radius: 5px;
    border: 2px solid #da552f;
    background: none;
    margin-top: 10px;
    color: #da552f;
    font-weight: bold;
    font-size: 16px;
    display: flex;
    justify-content: center;
    align-items: center;
}

.button:hover{
    background: #da552f;
    color: #fff;
}
```

```
.actions{
  display: flex;
  justify-content: space-between;
  margin-bottom: 20px;
  padding: 10px;
}
```

7º em routes.js adicione:

```
import Profile from './pages/Profile';
```

```
<Route exact path = '/' component = {User}/>
```

```
<Route path = '/create' component = {Profile}/> //nova rota
```

```
<Route path = '/update/:id' component = {Profile}/> //nova rota
```

8º na pasta users, em index.js adicione o seguinte código:

```
import {Link} from "react-router-dom";
```

```
async function handleDelete(id){
  try{
    await api.delete(`/users/${id}`);
    setUsers(users.filter(user => user.id !== id));
  }catch(err){
    alert("Erro ao deletar!");
  }
}
return([...]
<h1>Lista de usuários</h1>
<Link className = "button" id = "create-link" to = {"/create"}>Criar</Link>
[...])
  <li>
    <div className = "actions">
      <button className = "button" onClick = {() => handleDelete(user.id)} type = "button">Deletar</button>
      <Link className = "button" to = {`/update/${user.id}`}>Acessar</Link>
    </div>
  </li>
)
```

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

```
.button{
  height: 30px;
  width: 80px;
  border-radius: 5px;
  text-decoration: none;
  border: 2px solid #da552f;
  background: none;
  margin-top: 10px;
  color: #da552f;
  font-weight: bold;
  font-size: 16px;
  display: flex;
```

```
        justify-content: center;
        align-items: center;
    }

    .button:hover{
        background: #da552f;
        color: #fff;
    }

    .actions{
        display: flex;
        justify-content: space-between;
        margin-bottom: 20px;
        padding: 10px;
    }

    #create-link{
        margin: 50px;
    }
```