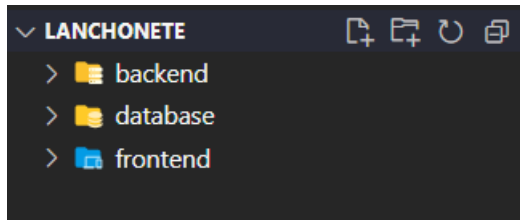


## Projeto Lanchonete – Frontend

**Objetivo:** fazer a integração do projeto da parte do backend com o frontend.

### Configuração do Projeto Lanchonete (Frontend)



Exclua a pasta frontend se ela já existir. Depois abra o terminal na pasta lanchonete e crie o projeto frontend com o seguinte comando:

- `npx create-react-app frontend`

Após o comando será criado automaticamente a pasta `node_modules` e os arquivos `package-lock.json` e `package.json`.

Com o comando **npm start** podemos testar se nossa aplicação está rodando corretamente no endereço <http://localhost:3001>.

Instale também:

- `npm install --save-dev @babel/plugin-proposal-private-property-in-object`

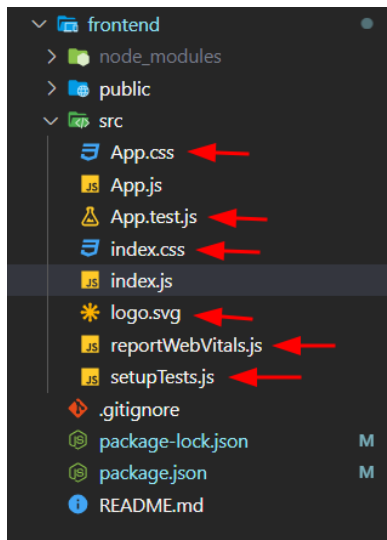
### Bibliotecas:

Assim que finalizado esse processo, no terminal, entre na pasta frontend e instale os seguintes pacotes:

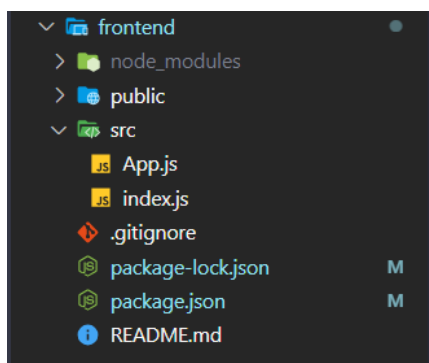
- `npm install react-router-dom axios styled-components prop-types font-awesome`

### Codificação do Projeto

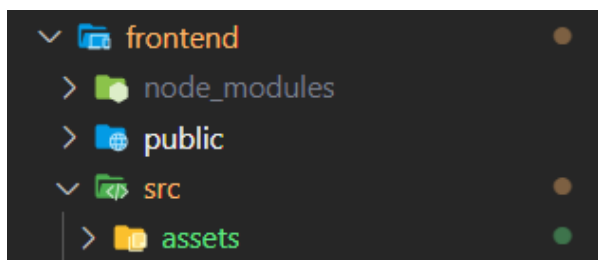
Dentro da pasta src, vamos remover os arquivos que não serão utilizados.



Deverá ficar apenas os arquivos App.js e index.js



No src vamos criar a pasta **assets** onde ficará armazenado nossas imagens do projeto.



No arquivo index.js vamos remover o que não utilizaremos.

### index.js

```
import React from 'react';  
import ReactDOM from 'react-dom/client';  
import App from './App';  
  
const root = ReactDOM.createRoot(document.getElementById('root'));
```

```
root.render(  
  <React.StrictMode>  
    <App />  
  </React.StrictMode>  
);
```

Vamos configurar o arquivo App.js

app.js

```
import React from "react";  
import Routes from "../routes";  
  
const App = () => <Routes />  
export default App;
```

Vamos criar a pasta onde ficará o css global de nossa aplicação, criando dentro do src a pasta **'styles'** e dentro dela o arquivo global.js

global.js

```
import { injectGlobal } from "styled-components";  
import "font-awesome/css/font-awesome.css";  
  
injectGlobal`  
* {  
  box-sizing: border-box;  
  padding: 0;  
  margin: 0;  
  outline: 0;  
}  
  
body, html {  
  background: #eee;  
  font-family: 'Helvetica Neue', 'Helvetica', Arial, sans-serif;  
  text-rendering: optimizeLegibility !important;  
  -webkit-font-smoothing: antialiased !important;
```

```
height: 100%;  
width: 100%;  
}  
`;  
;
```

Agora na pasta src vamos criar o arquivo de rotas 'routes.js'

routes.js

```
import React from "react";  
import {  
  BrowserRouter as Router,  
  Routes,  
  Route,  
  Navigate,  
} from "react-router-dom";  
  
const Rotas = () => (  
  <Router>  
    <Routes></Routes>  
  </Router>  
>);  
  
export default Rotas;
```

Agora no src vamos para a criação da pasta 'services', será criado os arquivos api.js e auth.js

api.js

```
import axios from "axios";  
import { getToken } from "../auth";  
  
const api = axios.create({  
  baseURL: "http://localhost:3001"
```

```

});

api.interceptors.request.use(async config => {
  const token = getToken();
  if (token) {
    config.headers.Authorization = `Bearer ${token}`;
  }
  return config;
});

export default api;

```

auth.js

```

export const TOKEN_KEY = "accessToken";
export const isAuthenticated = () => localStorage.getItem(TOKEN_KEY) !== null;
export const getToken = () => localStorage.getItem(TOKEN_KEY);

export const login = token => {
  localStorage.setItem(TOKEN_KEY, token);
}

export const logout = () => {
  localStorage.removeItem(TOKEN_KEY);
}

```

Partiremos agora para a criação das nossas páginas, para isso vamos criar a pasta **'pages'** em src.

Depois de criada, também vamos criar a pasta da página de **'Login'**. Nela vamos criar os arquivos **index.js** e **style.js**

index.js

```

import React, { useState } from "react";

```

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

import api from "../../services/api";

import { Container, Form } from "./style";

import Logo from "../../assets/senac.png";

const SignIn = () => {

  const [email, setEmail] = useState("");
  const [senha, setSenha] = useState("");
  const [error, setError] = useState("");


  const navigate = useNavigate();


  const handleSignIn = async (e) => {

    e.preventDefault();

    if (!email || !senha) {

      setError("Preencha email e senha para continuar!");

      return;

    }

    try {

      const response = await api.post("/signin", { email, senha });

      localStorage.setItem("accessToken", response.data.accessToken);

      navigate("/app");

    } catch (err) {

      setError("Houve um problema com o login, verifique suas credenciais!!");

    }

  };


  return (

    <Container>

      <Form onSubmit={handleSignIn}>

        <img src={Logo} alt="logo_senac" />
```

```
<input
  type="email"
  placeholder="Endereço de Email"
  onChange={(e) => setEmail(e.target.value)}
/>
<input
  type="password"
  placeholder="Senha"
  onChange={(e) => setSenha(e.target.value)}
/>
<button type="submit">Entrar</button>
{error && <p>{error}</p>}
</Form>
</Container>
);
};
export default SignIn;
```

### style.js

```
import styled from "styled-components";

export const Container = styled.div`
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100vh;
`;

export const Form = styled.form`
  width: 400px;
  background: #fff;
```

```
padding: 20px;

display: flex;

flex-direction: column;

align-items: center;

img {

  width: 300px;

  margin: 10px 0 40px;

}

p {

  color: #ff3333;

  margin-bottom: 15px;

  border: 1px solid #ff3333;

  padding: 10px;

  width: 100%;

  text-align: center;

}

input {

  flex: 1;

  height: 46px;

  margin-bottom: 15px;

  padding: 0 20px;

  color: #777;

  font-size: 15px;

  width: 100%;

  border: 1px solid #ddd;

  &::placeholder {

    color: #999;

  }

}

button {

  color: #fff;
```



```
font-size: 16px;

background: #fc6963;

height: 56px;

border: 0;

border-radius: 5px;

width: 100%;
}

hr {

margin: 20px 0;

border: none;

border-bottom: 1px solid #cdcdcd;

width: 100%;
}

a {

font-size: 16;

font-weight: bold;

color: #999;

text-decoration: none;
}

`;
```

Agora vamos retornar ao arquivo 'routes.js' para configurar a nossa página de login

routes.js

```
import React from "react";

import {

  BrowserRouter as Router,

  Routes,

  Route,

  Navigate,

} from "react-router-dom";

import Login from "../pages/Login";
```

```
const LoginPage = () => <Login />;

const Rotas = () => (

  <Router>

    <Routes>

      <Route path="/" element={<LoginPage />} />

    </Routes>

  </Router>

);

export default Rotas;
```

Agora vamos testar se nossa aplicação está funcionando rodando o comando **npm start**. Deverá aparecer a tela de login da nossa aplicação.



Em seguida vamos criar a tela da nossa página principal. Na pasta pages vamos criar outra pasta com o nome **'MainPage'** de dentro dela o arquivo **'index.js'**.

index.js

```
import React from "react";

const MainPage = () => {

  return (

    <div>
```

```
    <h2>Bem vindo à pagina principal</h2>

    {/* conteudo da pagina principal */}

  </div>

);

};

export default MainPage;
```

Em seguida vamos retornar ao arquivo 'routes.js' para adicionarmos a nossa página principal.

routes.js

```
import React from "react";

import {
  BrowserRouter as Router,
  Routes,
  Route,
  Navigate,
} from "react-router-dom";

import { isAuthenticated } from "../services/auth";
import Login from "../pages/Login";
import MainPage from "../pages/MainPage";

const LoginPage = () => <Login />;

const AppPage = () => {
  if (!isAuthenticated()) {
    return <Navigate to="/" replace />;
  }
  return <MainPage />;
};

const Rotas = () => (
```

```
<Router>

  <Routes>

    <Route path="/" element={<LoginPage />} />

    <Route path="/app" element={<AppPage />} />

  </Routes>

</Router>

);

export default Rotas;
```

Agora vamos testar! Primeiro vamos abrir o terminal na pasta backend para iniciar o servidor com o comando **npm start**, em seguida vamos rodar o mesmo comando em outro terminal na pasta frontend. Dessa forma, vamos abrir a página de login no endereço <http://localhost:3001>.

Vamos fazer um login com um usuário já cadastrado no banco de dados. Se tudo estiver certo você deve ser logado e redirecionado a página principal.

---

## Bem vindo à pagina principal

Essa mensagem deve aparecer para você no seu navegador. Parabéns!