

Rotar com React Router Dom

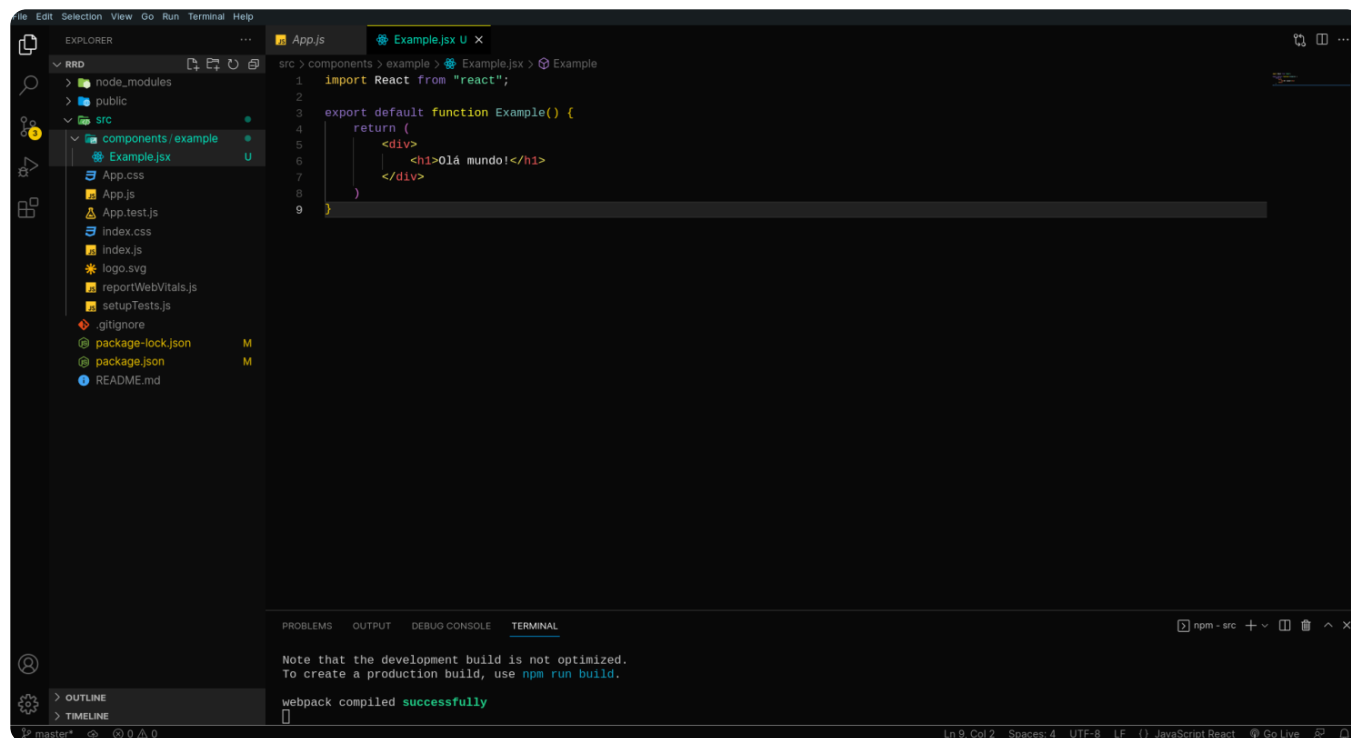
Instalação

Para fazer a instalação do React Router Dom vamos abrir o terminal na pasta do nosso projeto e utilizar o seguinte comando:

```
nmp i react-router-dom
```

Configuração inicial

Após fazer a instalação do React Router Dom vamos criar uma página para usar de exemplo. Eu criei a pasta components dentro de src e dentro de components criei uma pasta com o nome 'example' que vai ser a página para usarmos de exemplo, essa página é um componente funcional que retorna apenas uma div com uma 'Olá mundo!', além dessa página vamos criar também outra pasta com o nome Home, que será a nossa página Home, vamos configurar a navegação entre essas duas páginas.

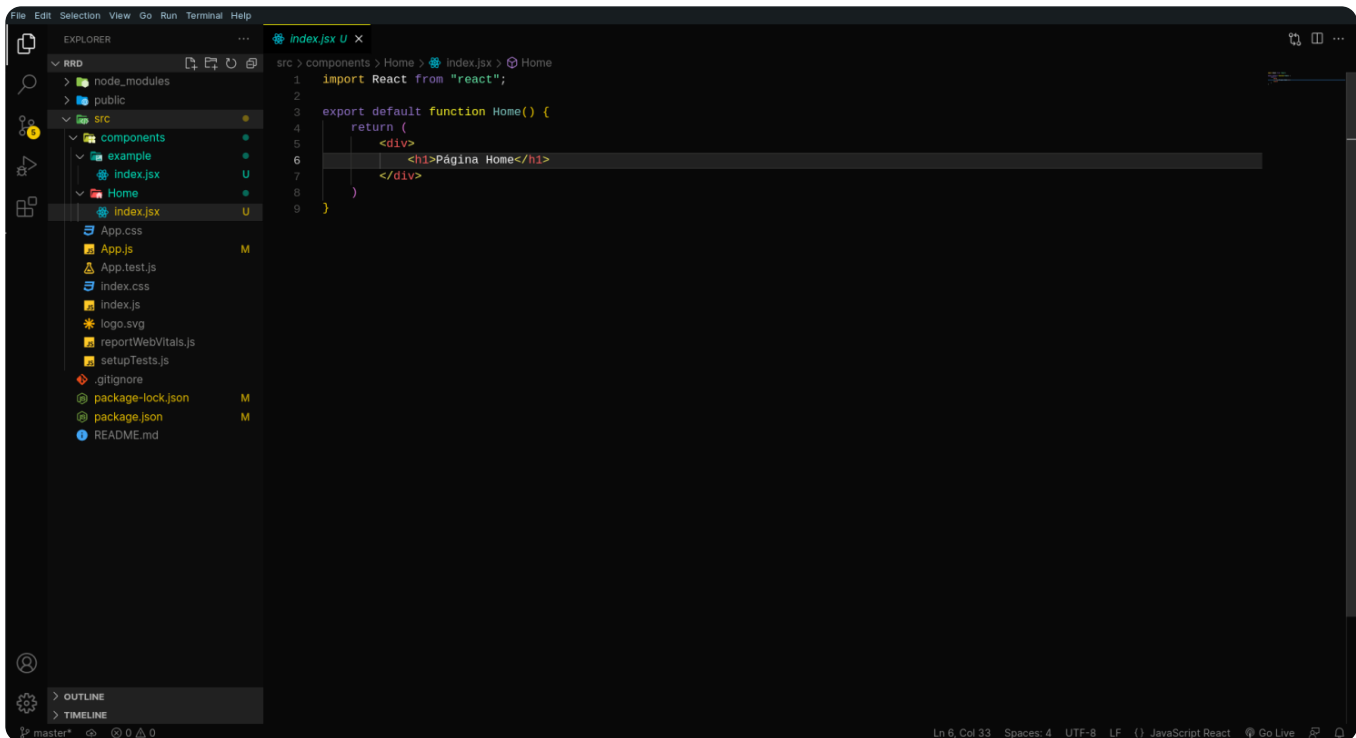


```
import React from 'react'

export default function Example() {
  return(
    <div>
      <h1>Olá mundo!</h1>
    </div>
  )
}
```

JAVASCRIPT

Home



```
import React from "react";

export default function Home() {
  return (
    <div>
      <h1>Página Home</h1>
    </div>
  )
}
```

JAVASCRIPT

Criando rotas

Após criar a nossa página de exemplo iremos criar o nosso componente de rotas. Dentro da pasta components vamos criar a pasta Routes e dentro dela vamos criar o arquivo Routes.js. Dentro do arquivo Routes.js vamos fazer as seguintes importações:

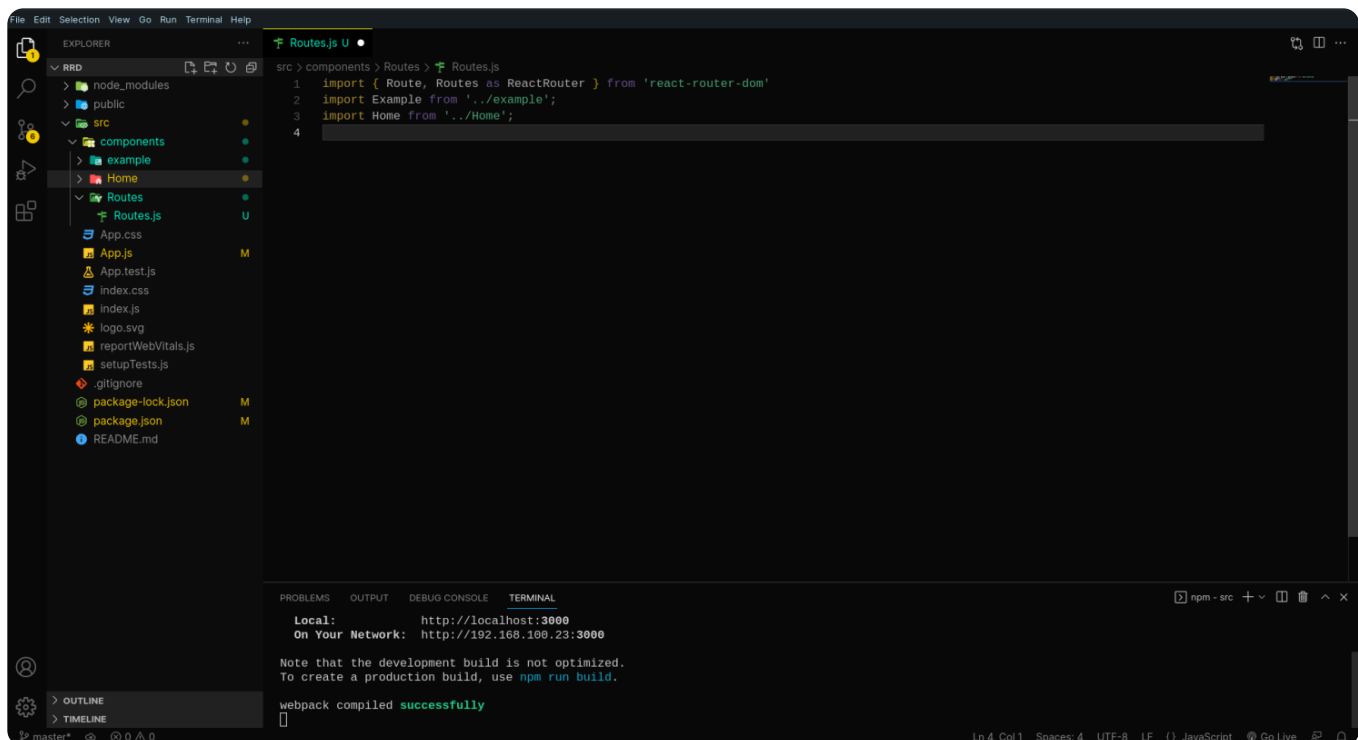
```
import { Route, Routes as ReactRouter } from 'react-router-dom'
```

Essa importação se trata dos métodos que usaremos do React Router Dom para criar as nossas Rotas, nessa importação eu usei um alias para diferenciar o nome do método Routes da nossa função Routes. Nesse arquivo vamos importar também as nossas páginas de exemplo que criamos anteriormente:

```
import Example from '../example'

import Home from '../Home'
```

JAVASCRIPT



```
import { Route, Routes as ReactRouter } from 'react-router-dom'

import Example from '../example';

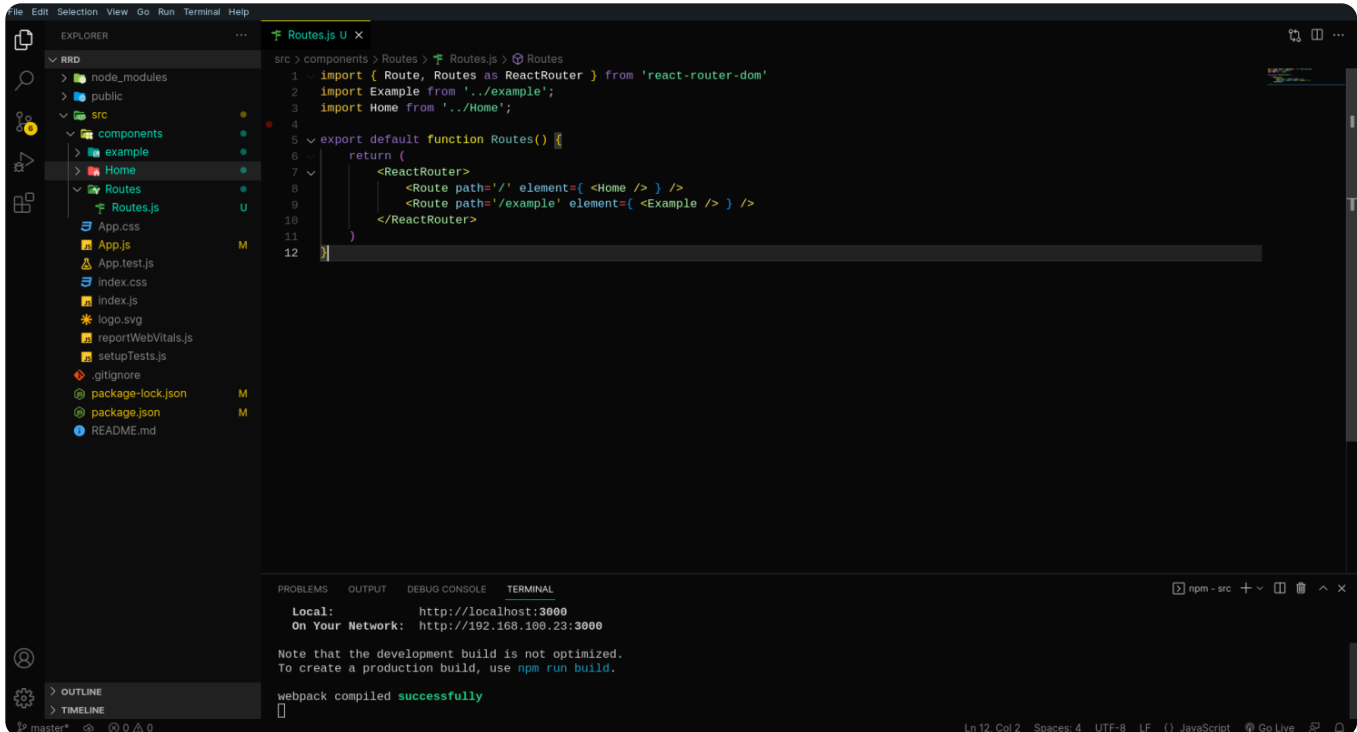
import Home from '../Home';
```

JAVASCRIPT

É nesse componente que iremos importar todas as rotas de nossa aplicação, para criar uma rota fazemos da seguinte forma:

Abrimos o `ReactRouter` em forma de tag e dentro dele nós adicionamos um `Route` com duas propriedades, elas são:

- `path` - Usamos para indicar o caminho que redireciona para a rota;
- `element` - O componente que será renderizado quando o `path` for acessado;



```
import { Route, Routes as ReactRouter } from 'react-router-dom'
import Example from '../example';
import Home from '../Home';

export default function Routes() {
  return (
    <ReactRouter>
      <Route path="/" element={ <Home /> } />
      <Route path="/example" element={ <Example /> } />
    </ReactRouter>
  )
}
```

JAVASCRIPT

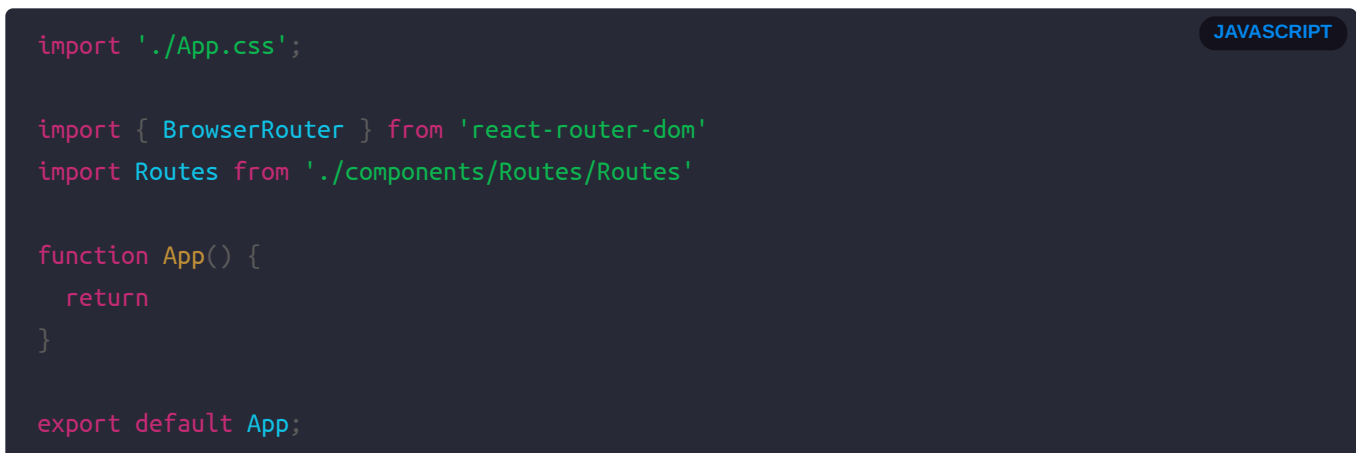
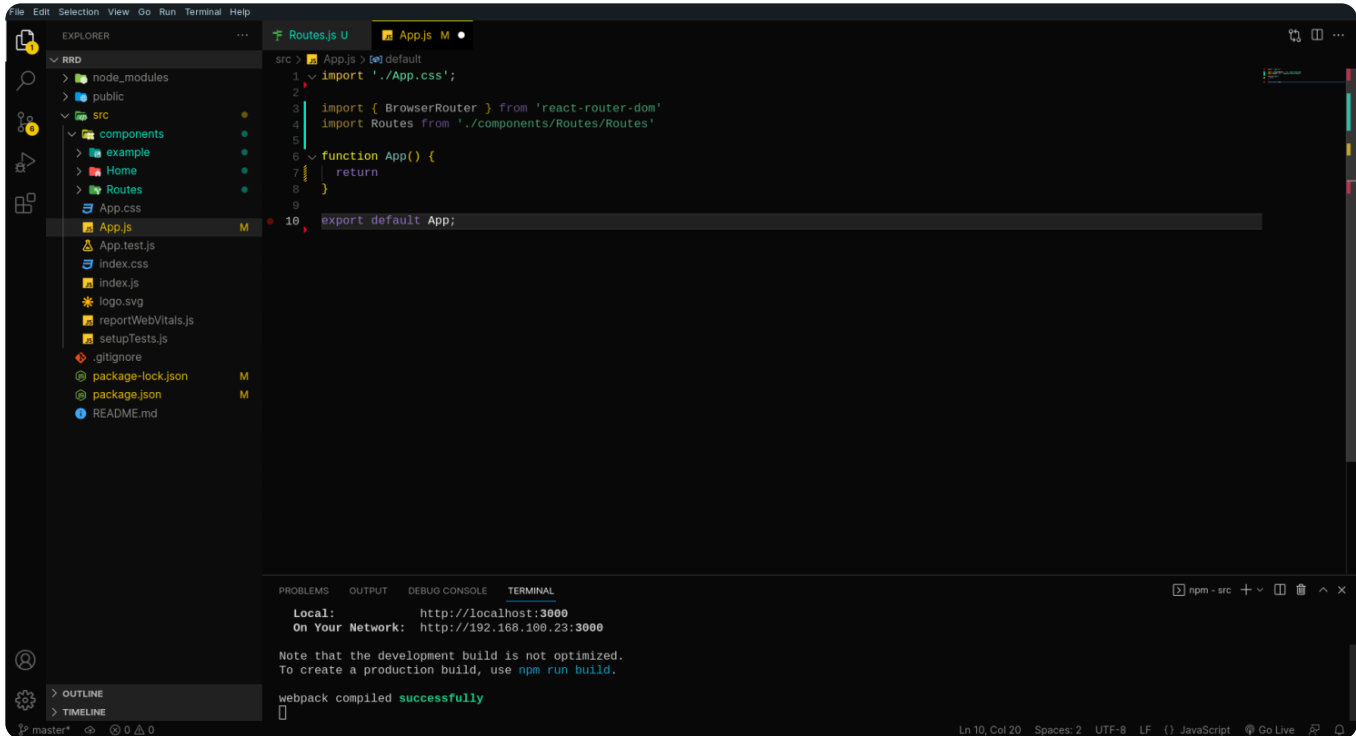
Agora no nosso componente `App.js` iremos importar do `React Router Dom` o `BrowserRouter` e também o nosso componente `Routes`.

```
import { BrowserRouter } from 'react-router-dom'

import Routes from '../components/Routes/Routes'
```

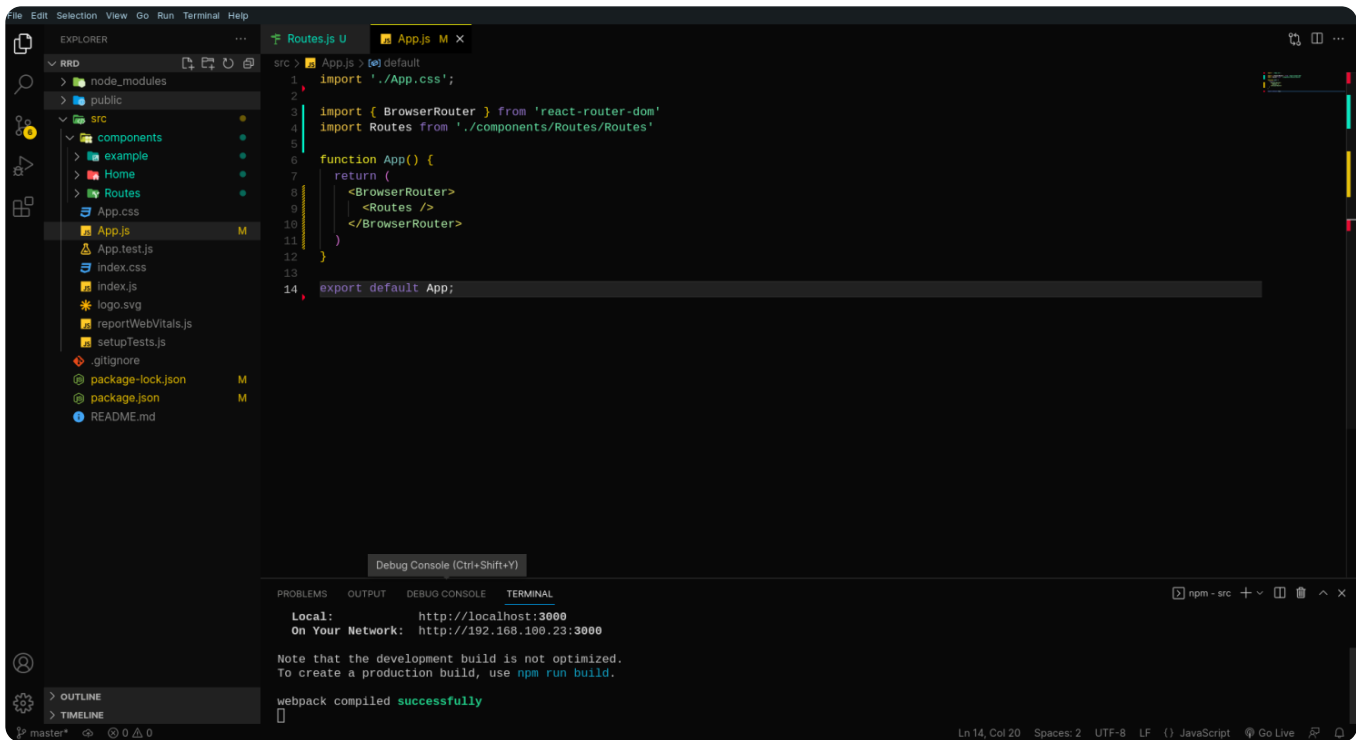
JAVASCRIPT

Podemos limpar tudo que tem no nosso componente App e deixar apenas o Return por enquanto.



Após fazer as devidas importações vamos fazer o seguinte:

Abriremos o BrowserRouter como uma tag e dentro dele usaremos o nosso componente Routes, fazendo isso a nossa navegação já estará funcionando.



```
import './App.css';

import { BrowserRouter } from 'react-router-dom'
import Routes from './components/Routes/Routes'

function App() {
  return (
    <BrowserRouter>
      <Routes />
    </BrowserRouter>
  )
}

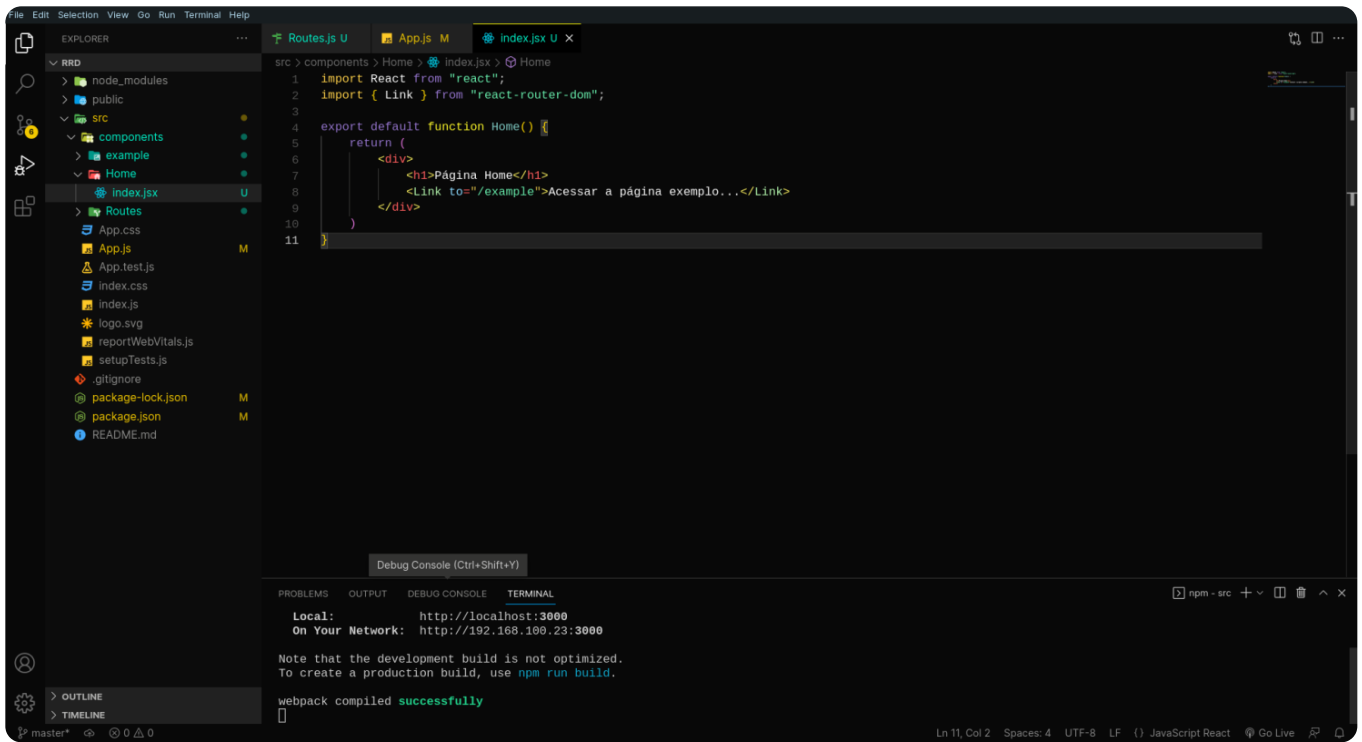
export default App;
```

JAVASCRIPT

Para fazer o nosso teste vamos adicionar dentro de cada página um Link para fazer o roteamento.

Para isso importamos o Link do React Router Dom e usamos a propriedade 'to' para informar a rota.

Link da página Home

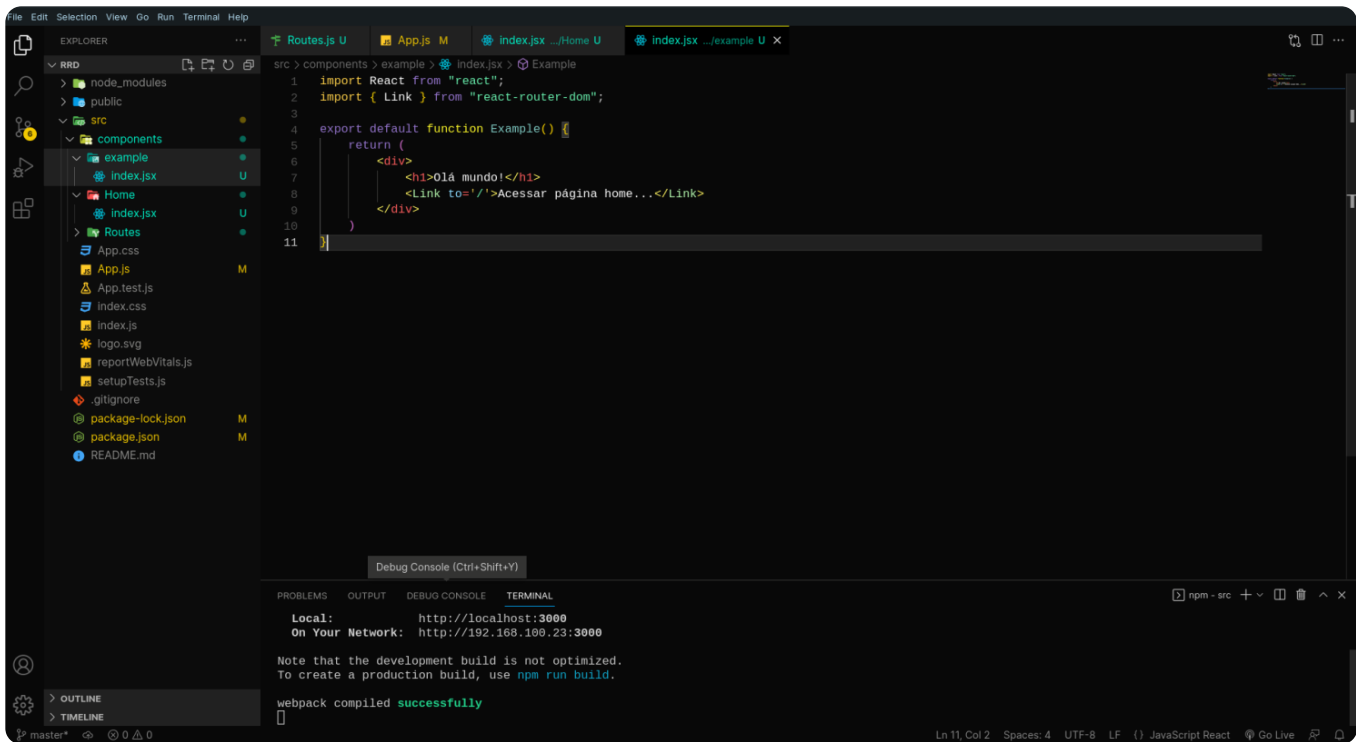


```
import React from "react";
import { Link } from "react-router-dom";

export default function Home() {
  return (
    <div>
      <h1>Página Home</h1>
      <Link to="/example">Acessar a página exemplo...</Link>
    </div>
  )
}
```

JAVASCRIPT

Link da página Ola mundo



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

export default function Example() {
  return (
    <div>
      <h1>Olá mundo!</h1>
      <Link to="/">Acessar página home...</Link>
    </div>
  )
}
```

JAVASCRIPT

Agora basta fazermos o teste...



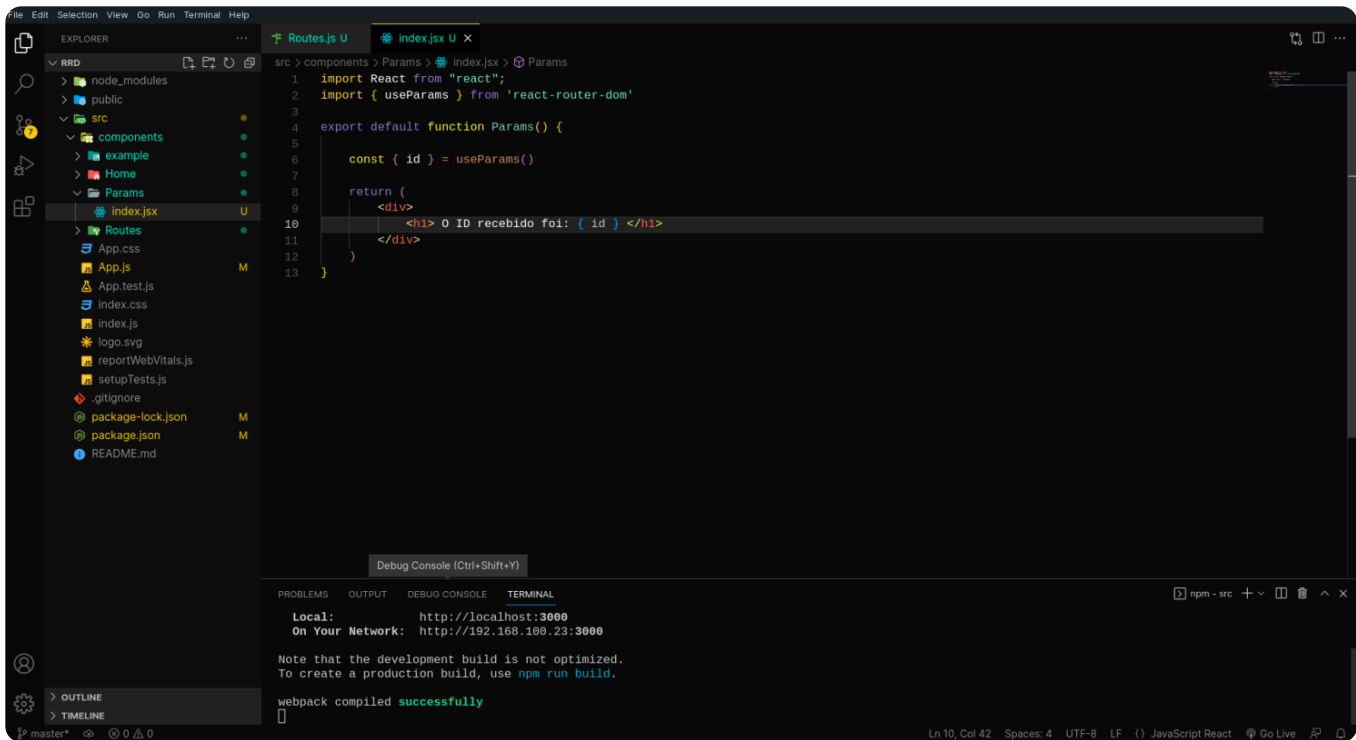
Página Home

[Acessar a página exemplo...](#)

Parâmetros via URL (bônus)

Podemos receber parâmetros através da nossa URL e redirecionar para páginas específicas através disso, para isso vamos usar atributos dinâmicos no Link e o useParams do React Router Dom, vamos lá!

Primeiramente vamos criar outra pasta dentro da nossa pasta components, eu irei chamar ela de Params, dentro dessa pasta iremos criar nosso arquivo index.jsx e vamos importar o useParams, após isso vamos atribuir o useParams a uma variável chamada id, e vamos retornar um h1 que exibe para nós o id recebido.



```
import React from "react";
import { useParams } from 'react-router-dom'

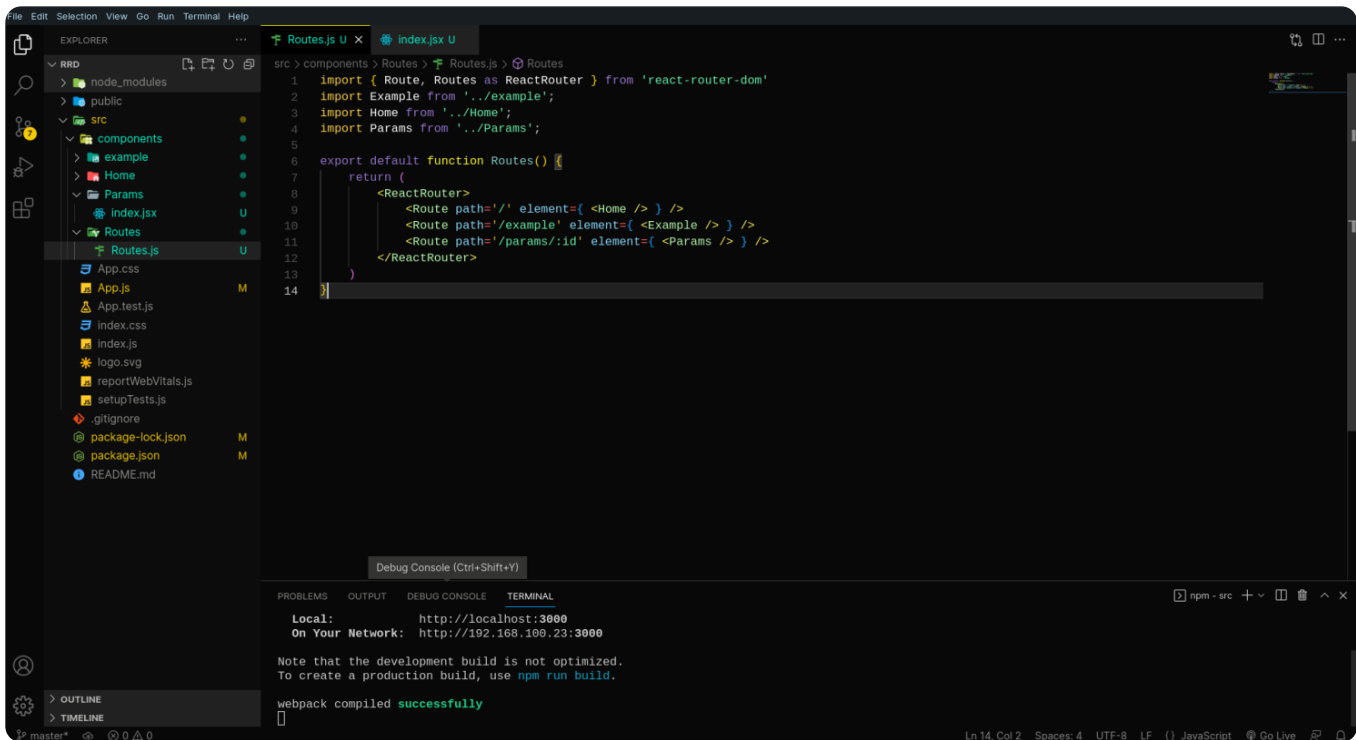
export default function Params() {

    const { id } = useParams()

    return (
        <div>
            <h1> 0 ID recebido foi: { id } </h1>
        </div>
    )
}
```

JAVASCRIPT

Depois de fazer isso vamos criar uma nova rota dentro do nosso componente Routes, vamos importar o nosso componente Params e então faremos essa nova rota usando um atributo não literal no nosso path, fazemos isso usando dois pontos e retornamos nessa rota o nosso componente Params.



```
import { Route, Routes as ReactRouter } from 'react-router-dom'
import Example from '../example';
import Home from '../Home';
import Params from '../Params';

export default function Routes() {
  return (
    <ReactRouter>
      <Route path="/" element={ <Home /> } />
      <Route path="/example" element={ <Example /> } />
      <Route path="/params/:id" element={ <Params /> } />
    </ReactRouter>
  )
}
```

JAVASCRIPT

Depois de ter seguido todos esses passos vamos testar:



Página Home

[Acessar a página exemplo...](#)

Dessa forma podemos fazer rotas mais complexas integrando nosso front-end com o Express por exemplo e fazendo comunicação entre eles dois.