

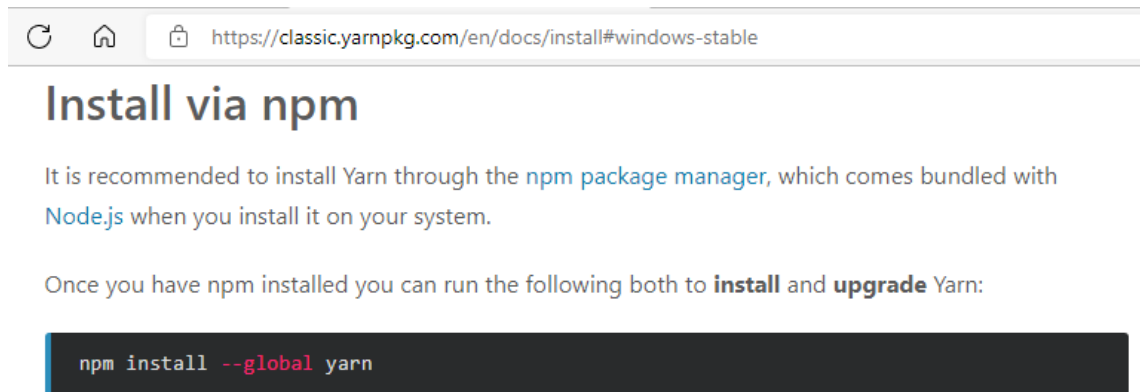
React.JS Fundamentos e Intermediário 15/12/2021

Iniciaremos agora a navegação com React Router. Para isso construiremos uma página com um menu lateral.

O yarn tem se apresentado mais moderno que o npm.

Para instalar o yarn, é preciso ter o Node já instalado, digite o código:

```
npm install --global yarn
```



Iniciaremos um novo projeto na pasta curso-react com o comando:

```
yarn create react-app navegacao
```

```
Alexandre@DESKTOP-78RVV7R MINGW64 /c/curso-react
$ yarn create react-app navegacao
```

Vamos entrar na pasta do projeto navegacao e deixar disponível o sistemas de rotas dinâmicas na nossa aplicação web, digitando o comando:

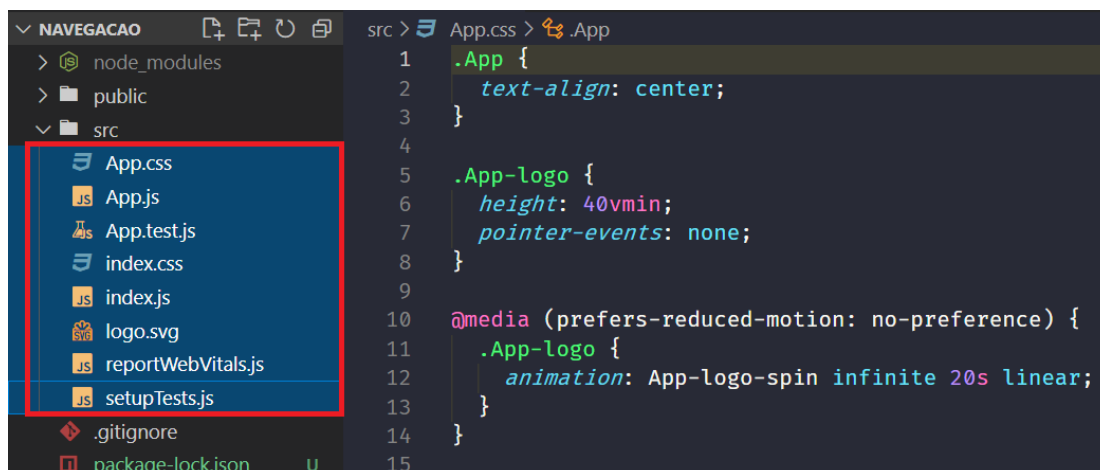
```
npm i react-router-dom
```

```
MINGW64:/c/curso-react/navegacao
Alexandre@DESKTOP-78RVV7R MINGW64 /c/curso-react/navegacao (main)
$ npm i react-router-dom
added 3 packages, changed 76 packages, and audited 1230 packages in 17s
163 packages are looking for funding
run `npm fund` for details
```

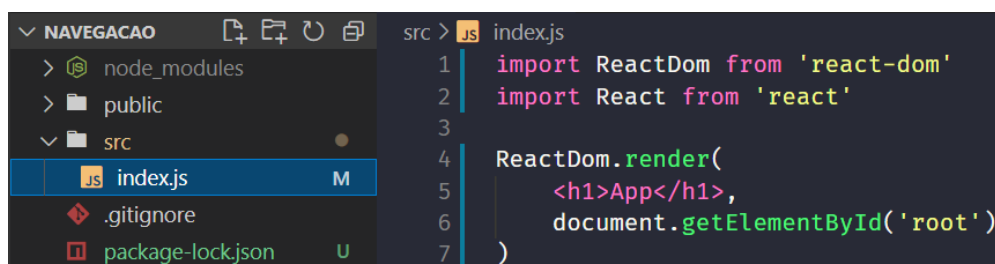
Abriremos nosso projeto no VSCode com o comando code .

```
MINGW64:/c/curso-react/navegacao
Alexandre@DESKTOP-78RVV7R MINGW64 /c/curso-react/navegacao (main)
$ code .
```

Apagaremos esses arquivos, pois iniciaremos nossa aplicação do zero.



E criar o arquivo index.js com o código abaixo:



E para iniciar a nossa aplicação no servidor:

yarn start

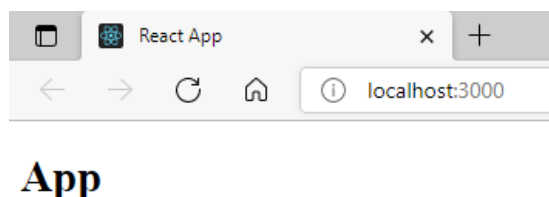
```
Alexandre@DESKTOP-78RVV7R MINGW64 /c/curso-react/navegacao (main)
$ yarn start
```

Ou

npm start

```
Alexandre@DESKTOP-78RVV7R MINGW64 /c/curso-react/navegacao (main)
$ npm start
```

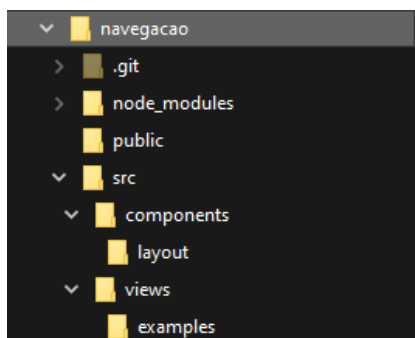
Nossa aplicação funcionando.



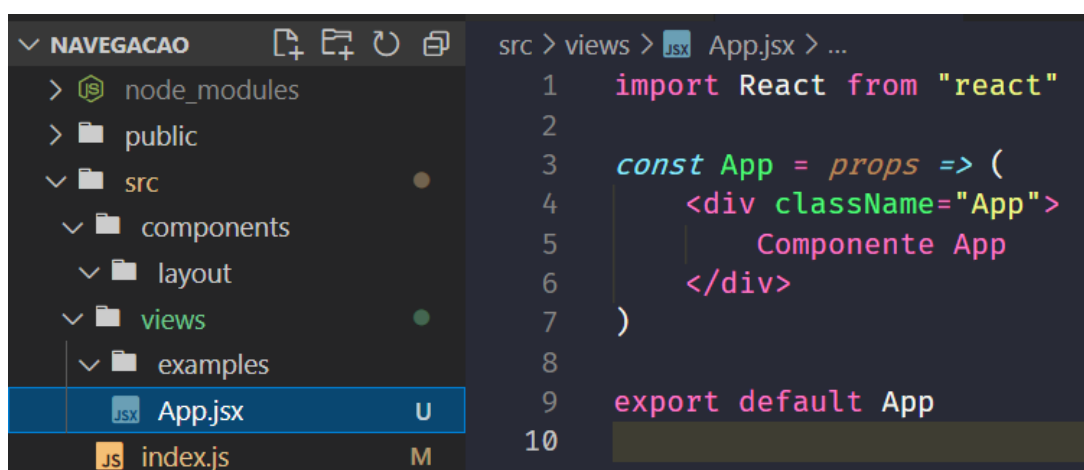
Criaremos a seguinte estrutura:

Pastas src > components > layout.

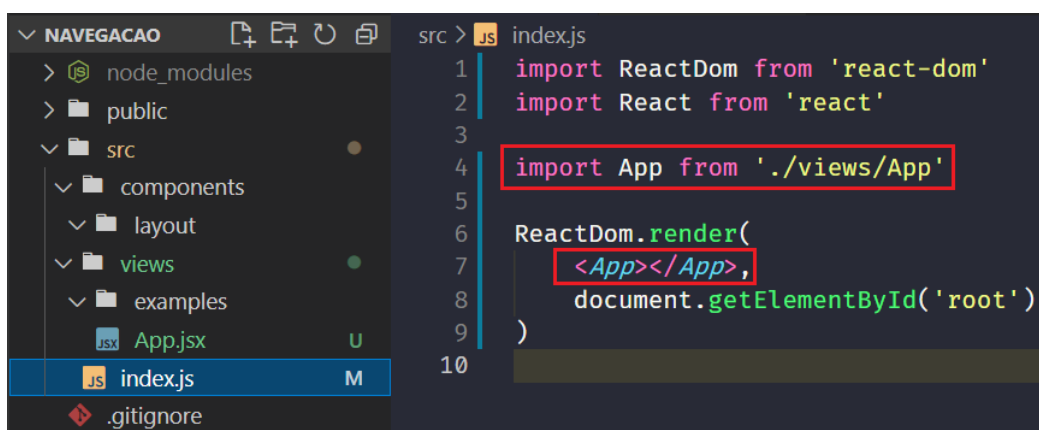
Pastas src > views > examples



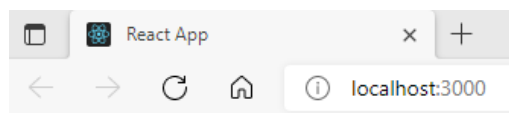
Dentro da pasta view criaremos o arquivo App.jsx que armazenará todas as nossas páginas;



Vamos importar o arquivo App.jsx no arquivo index.js e inserir o componente App.

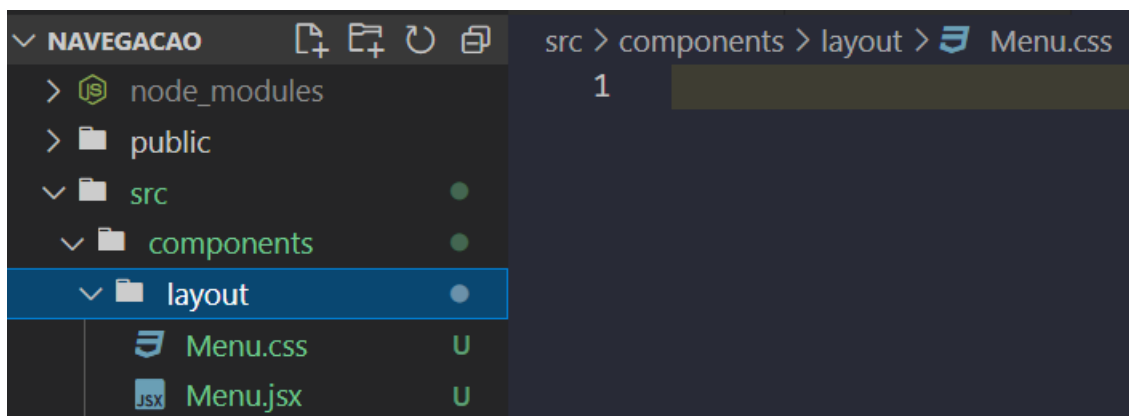


Na nossa aplicação aparecerá.

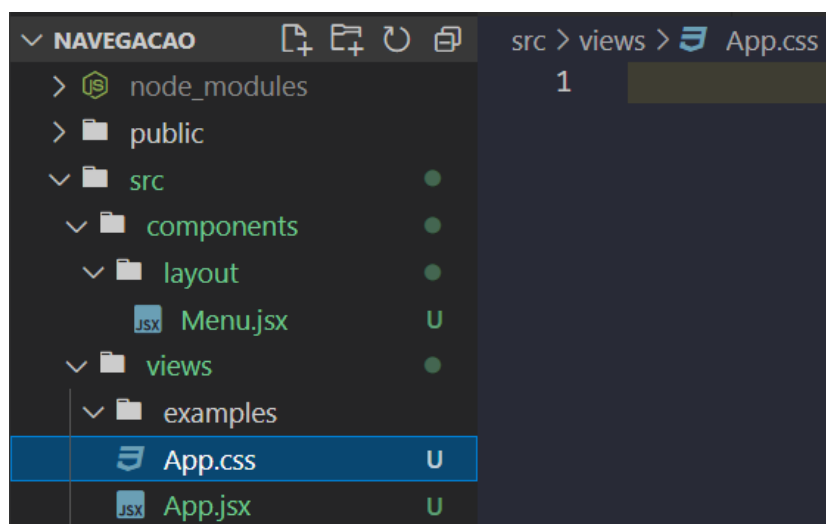


Componente App

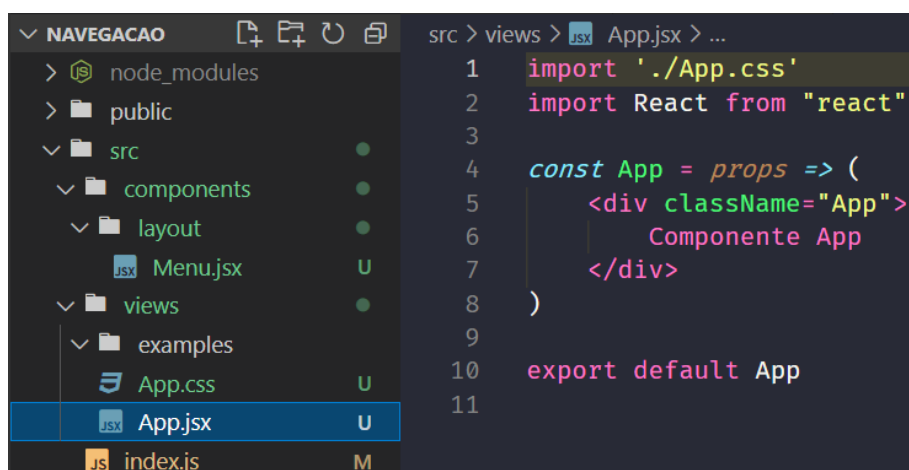
Na pasta layout definiremos o componente Menu.jsx e Menu.css



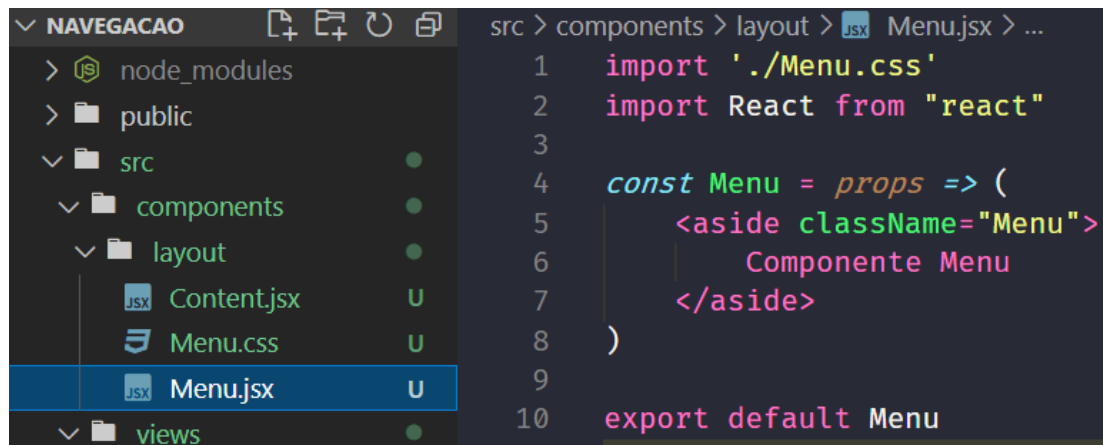
Na pasta views estilizaremos com App.css



Faremos o import do App.css no arquivo App.jsx

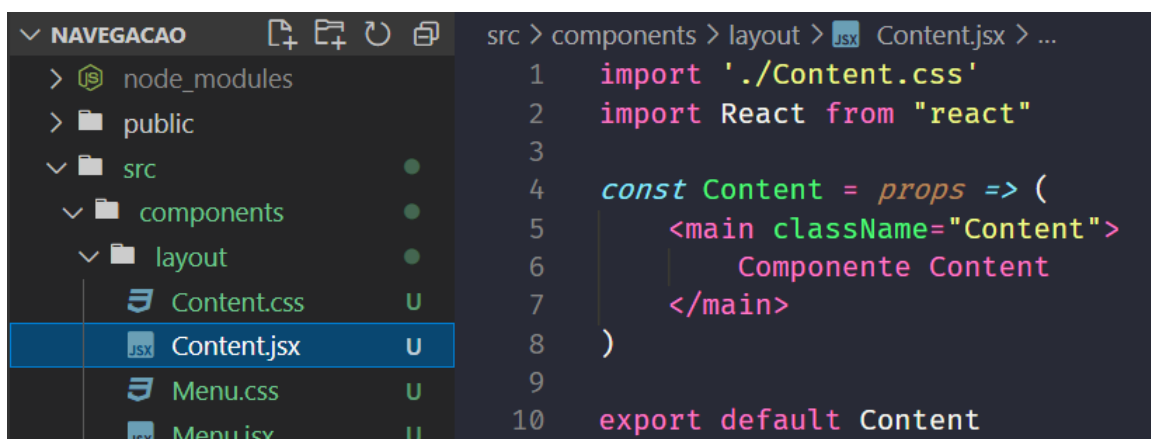


No arquivo Menu.jsx teremos a mesma estrutura do App.jsx



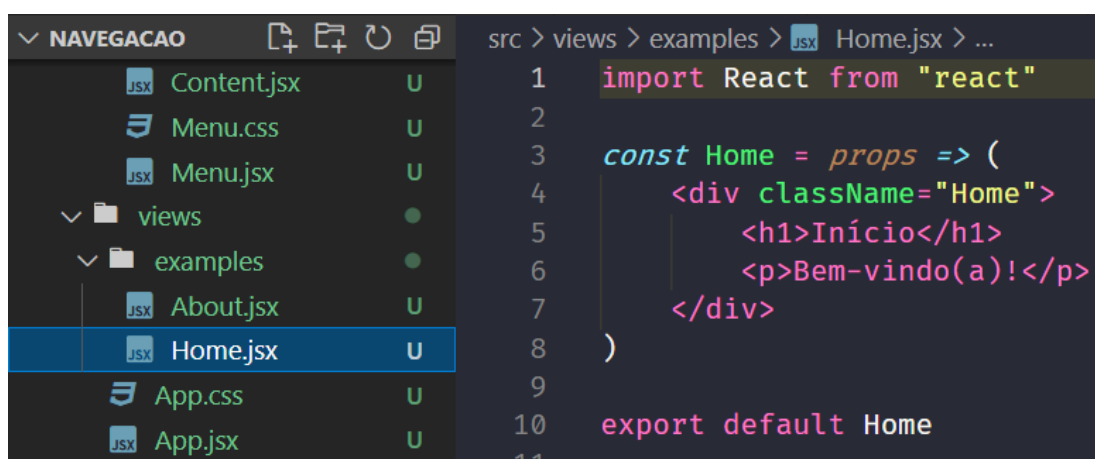
```
src > components > layout > Menu.jsx > ...
1  import './Menu.css'
2  import React from "react"
3
4  const Menu = props => (
5    <aside className="Menu">
6      Componente Menu
7    </aside>
8  )
9
10 export default Menu
```

Na pasta layout criaremos os componentes Content.css, Content.jsx e aproveitar a estrutura do menu.



```
src > components > layout > Content.jsx > ...
1  import './Content.css'
2  import React from "react"
3
4  const Content = props => (
5    <main className="Content">
6      Componente Content
7    </main>
8  )
9
10 export default Content
```

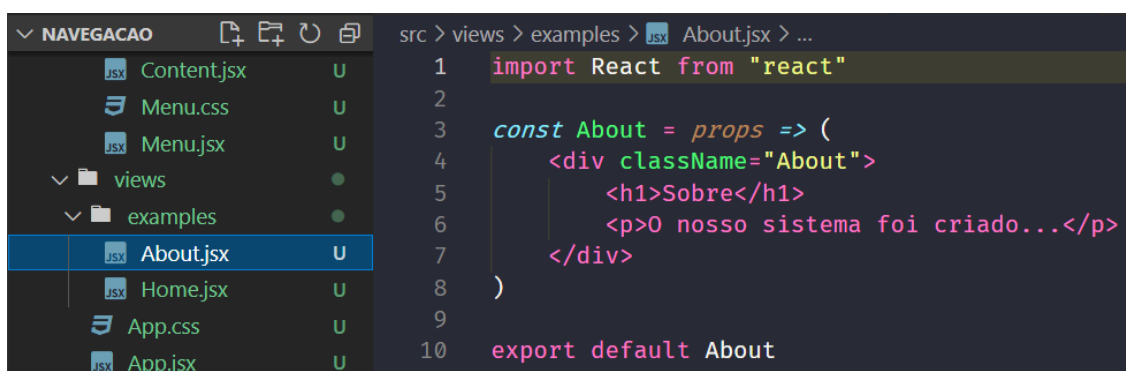
Na pasta Home.jsx criaremos a estrutura abaixo.



```
src > views > examples > Home.jsx > ...
1  import React from "react"
2
3  const Home = props => (
4    <div className="Home">
5      <h1>Início</h1>
6      <p>Bem-vindo(a)!</p>
7    </div>
8  )
9
10 export default Home
```

No arquivo About.jsx criaremos o arquivo com a frase:

O nosso sistema foi criado...



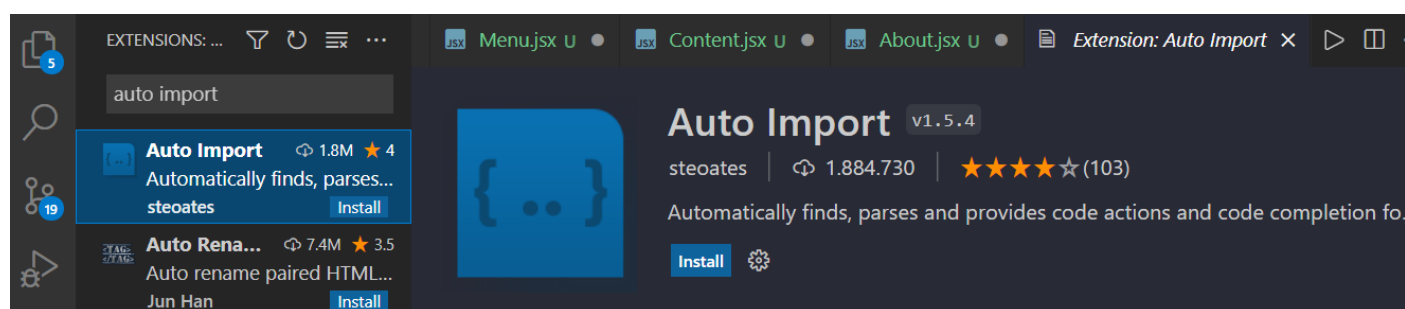
```

src > views > examples > About.jsx > ...
1  import React from "react"
2
3  const About = props => (
4    <div className="About">
5      <h1>Sobre</h1>
6      <p>O nosso sistema foi criado...</p>
7    </div>
8  )
9
10 export default About

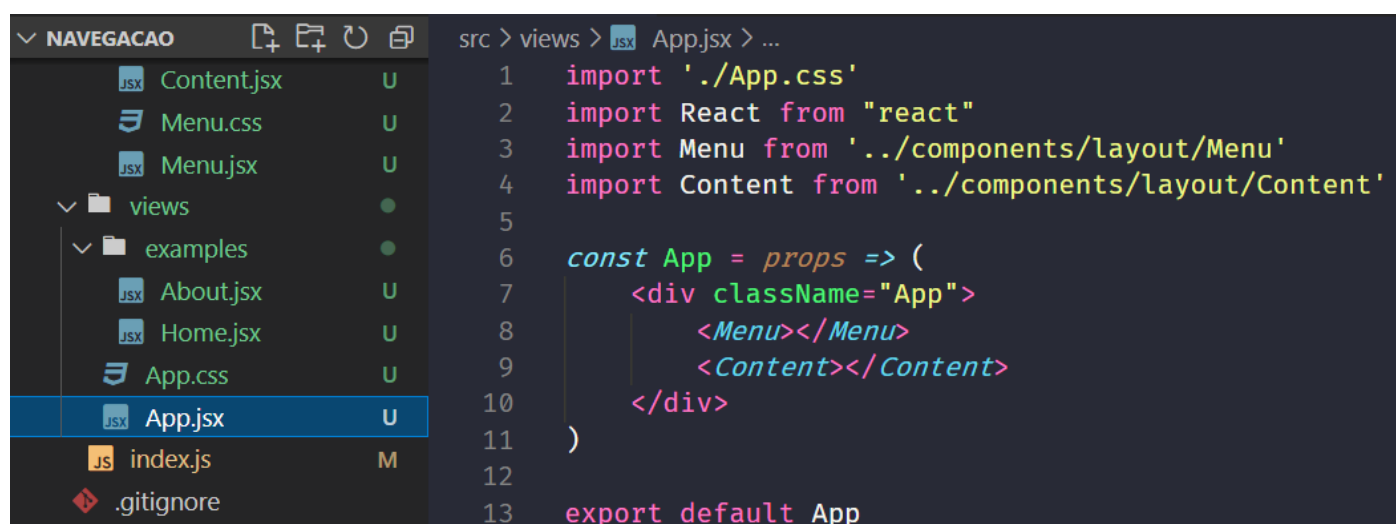
```

Faremos a alteração no componente App.jsx para referenciarmos tanto para o componente Menu como para o componente Content.

Existe um plugin Auto Import que pode ser muito útil.



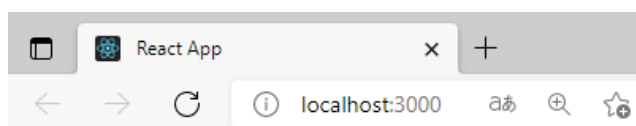
No arquivo App.jsx faremos as alterações.



```

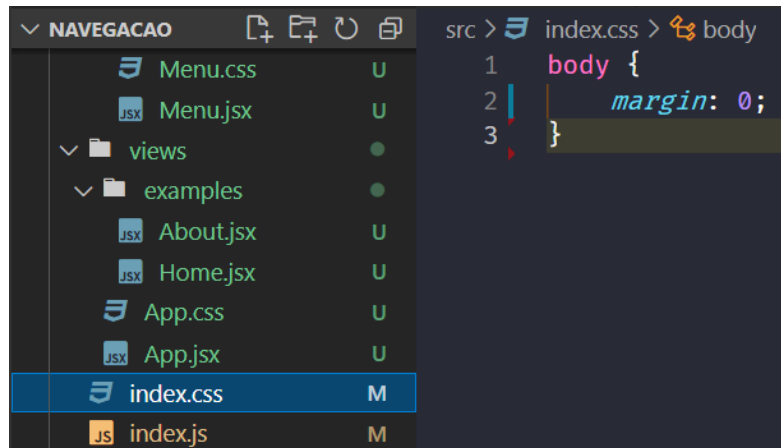
src > views > App.jsx > ...
1  import './App.css'
2  import React from "react"
3  import Menu from '../components/layout/Menu'
4  import Content from '../components/layout/Content'
5
6  const App = props => (
7    <div className="App">
8      <Menu></Menu>
9      <Content></Content>
10   </div>
11 )
12
13 export default App

```



Componente Menu
Componente Content

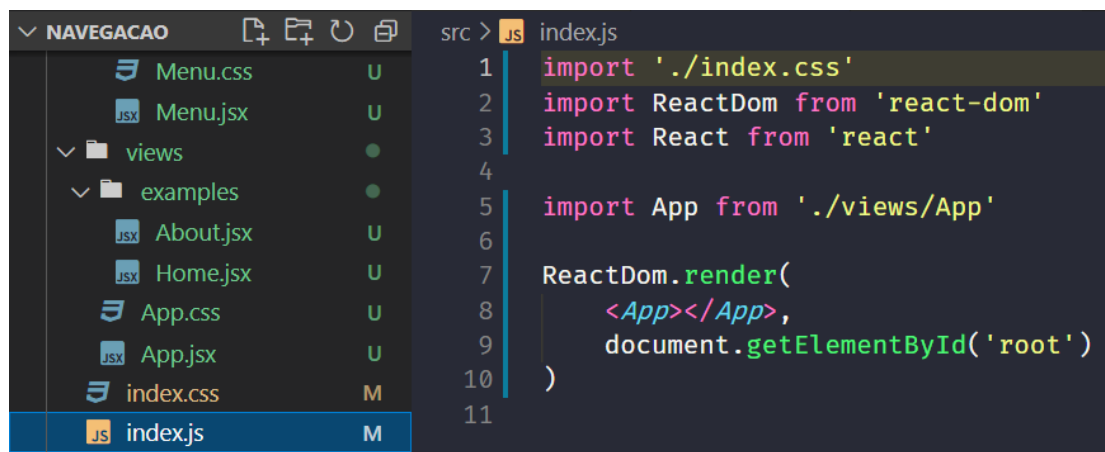
Aplicaremos os estilos nas páginas começando pelo index.css



The screenshot shows the VS Code interface. On the left, the 'NAVEGACAO' (Navigation) sidebar lists the project files: Menu.css, Menu.jsx, a 'views' folder, an 'examples' folder, About.jsx, Home.jsx, App.css, App.jsx, index.css (selected), and index.js. The main editor area shows the 'index.css' file with the following content:

```
src > index.css > body
1  body {
2      margin: 0;
3  }
```

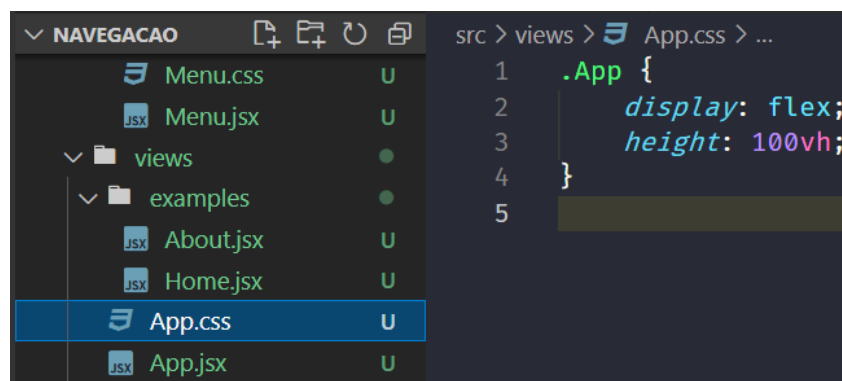
E fazer a referência no arquivo index.js com o import.



The screenshot shows the VS Code interface. On the left, the 'NAVEGACAO' sidebar is the same as in the previous image, but 'index.js' is now selected. The main editor area shows the 'index.js' file with the following content:

```
src > index.js
1  import './index.css'
2  import ReactDOM from 'react-dom'
3  import React from 'react'
4
5  import App from './views/App'
6
7  ReactDOM.render(
8      <App></App>,
9      document.getElementById('root')
10 )
11
```

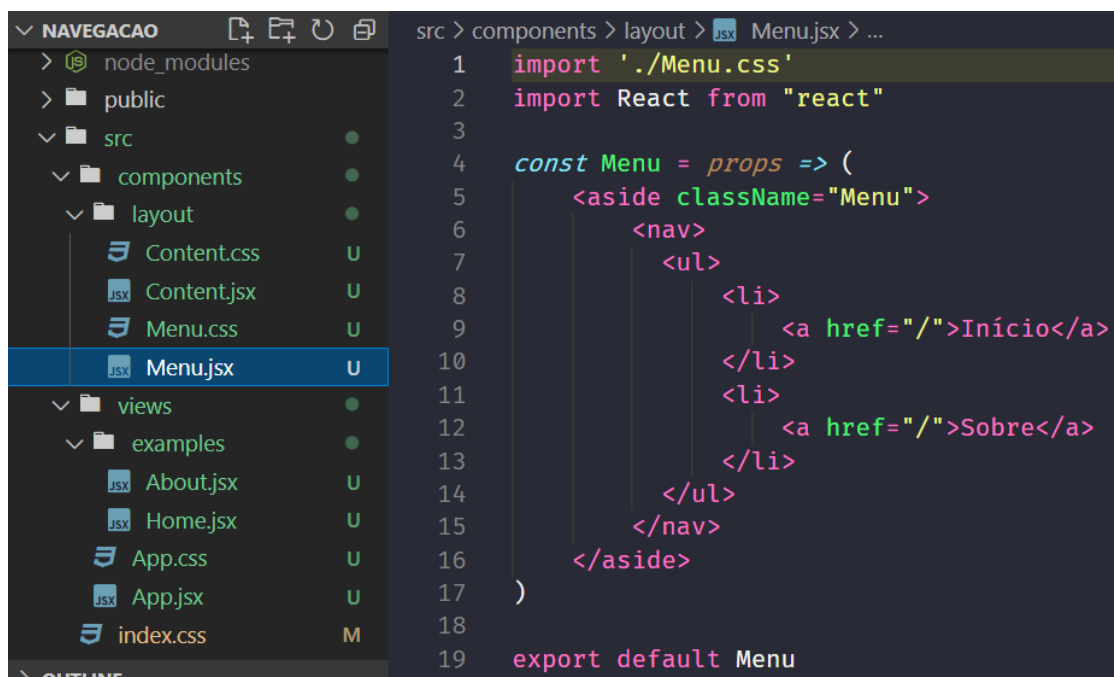
No arquivo App.css colcoaremos o flex box e altura de 100 % com 100vh



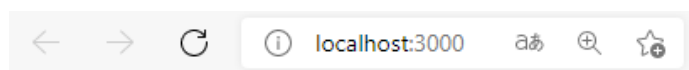
The screenshot shows the VS Code interface. On the left, the 'NAVEGACAO' sidebar shows 'App.css' selected under the 'views' folder. The main editor area shows the 'App.css' file with the following content:

```
src > views > App.css > ...
1  .App {
2      display: flex;
3      height: 100vh;
4  }
5
```

No menu faremos uma tabela de navegação com `<nav>` e utilizar a lista com `` e ``.



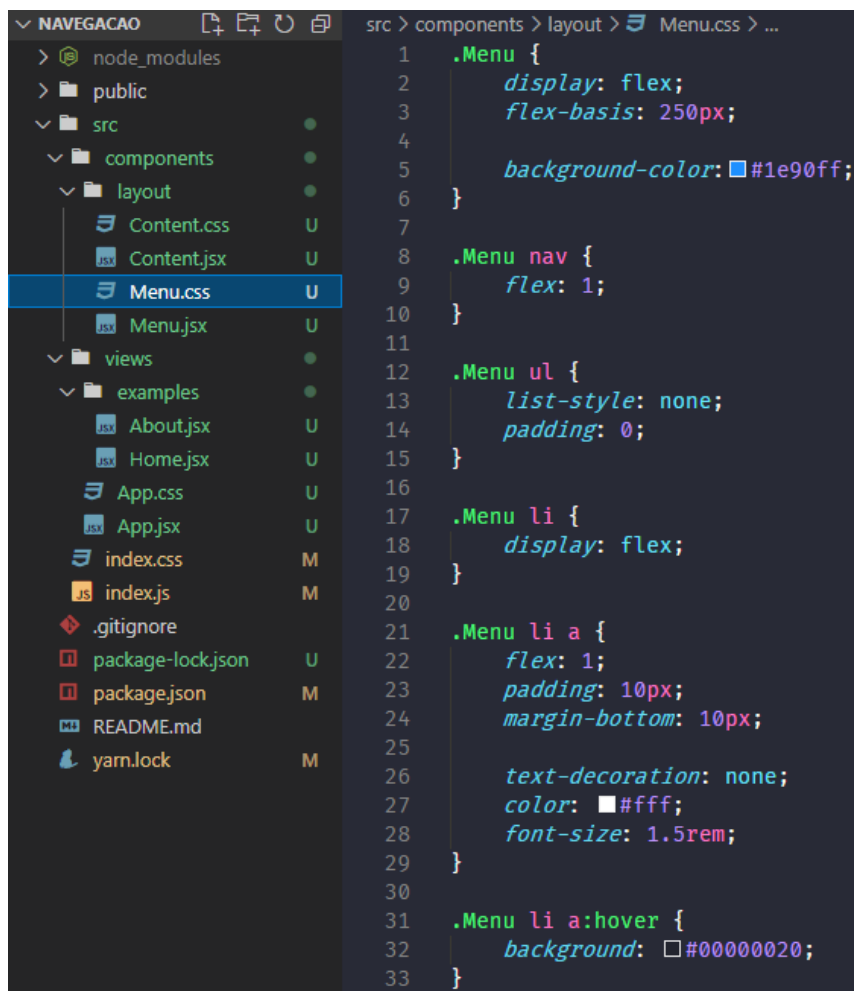
```
src > components > layout > Menu.jsx > ...
1  import './Menu.css'
2  import React from "react"
3
4  const Menu = props => (
5    <aside className="Menu">
6      <nav>
7        <ul>
8          <li>
9            <a href="/">Início</a>
10         </li>
11         <li>
12           <a href="/">Sobre</a>
13         </li>
14       </ul>
15     </nav>
16   </aside>
17 )
18
19 export default Menu
```



Componente Content

- [Início](#)
- [Sobre](#)

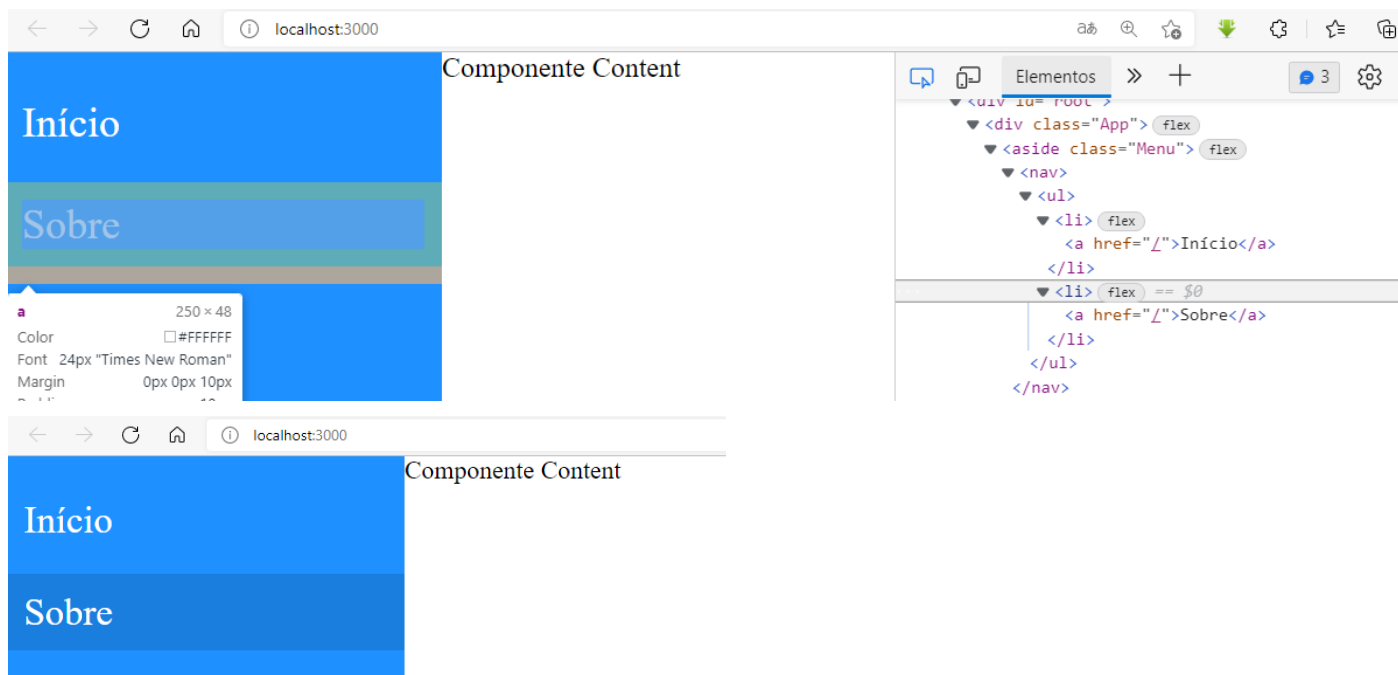
Utilizaremos o menu conforme abaixo:



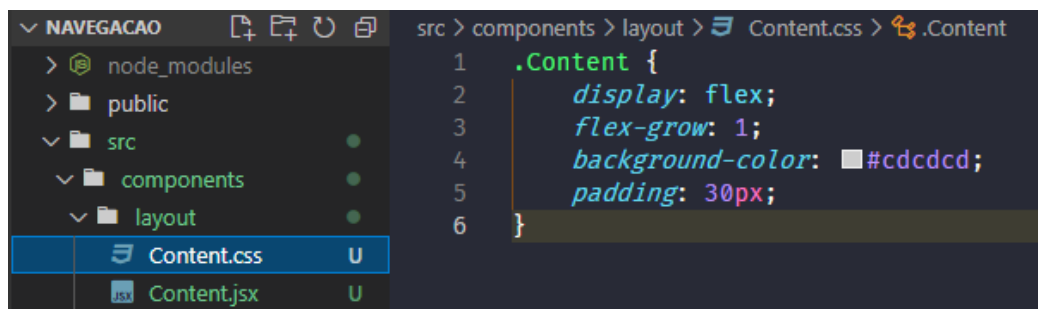
The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like node_modules, public, src, components, layout, views, and examples. The src/components/layout directory is expanded, showing files like Content.css, Content.jsx, Menu.css, Menu.jsx, App.css, App.jsx, index.css, index.js, .gitignore, package-lock.json, package.json, README.md, and yarn.lock. The Menu.css file is selected and its content is displayed in the code editor.

```

1  .Menu {
2      display: flex;
3      flex-basis: 250px;
4
5      background-color: #1e90ff;
6  }
7
8  .Menu nav {
9      flex: 1;
10 }
11
12 .Menu ul {
13     list-style: none;
14     padding: 0;
15 }
16
17 .Menu li {
18     display: flex;
19 }
20
21 .Menu li a {
22     flex: 1;
23     padding: 10px;
24     margin-bottom: 10px;
25
26     text-decoration: none;
27     color: #fff;
28     font-size: 1.5rem;
29 }
30
31 .Menu li a:hover {
32     background: #00000020;
33 }
  
```

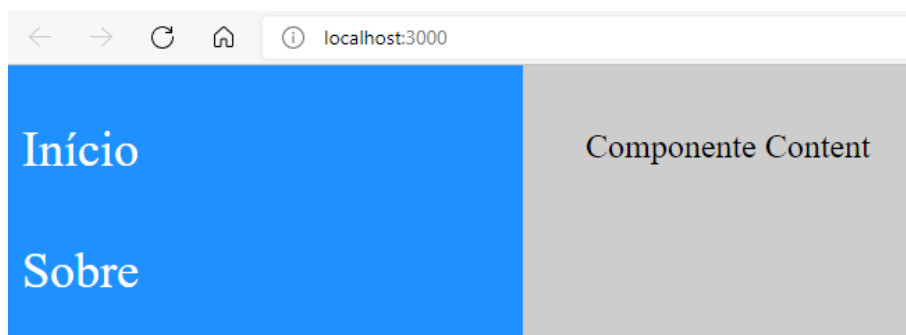


Em relação ao conteúdo Content.css



```

1  .Content {
2      display: flex;
3      flex-grow: 1;
4      background-color: #cdcdcd;
5      padding: 30px;
6  }
  
```



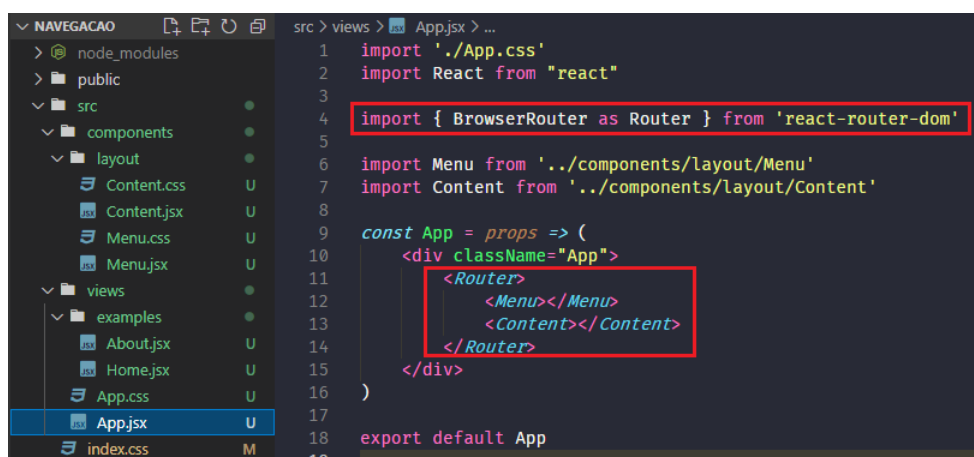
Iremos agora importar o React Router Dom para fazer a navegação que é navegar sobre os componentes de Início e Sobre.

A documentação se encontra em

<https://reactrouter.com/docs/en/v6/getting-started/overview>

A ideia é envolver o menu com as tags de navegação e conteúdo:

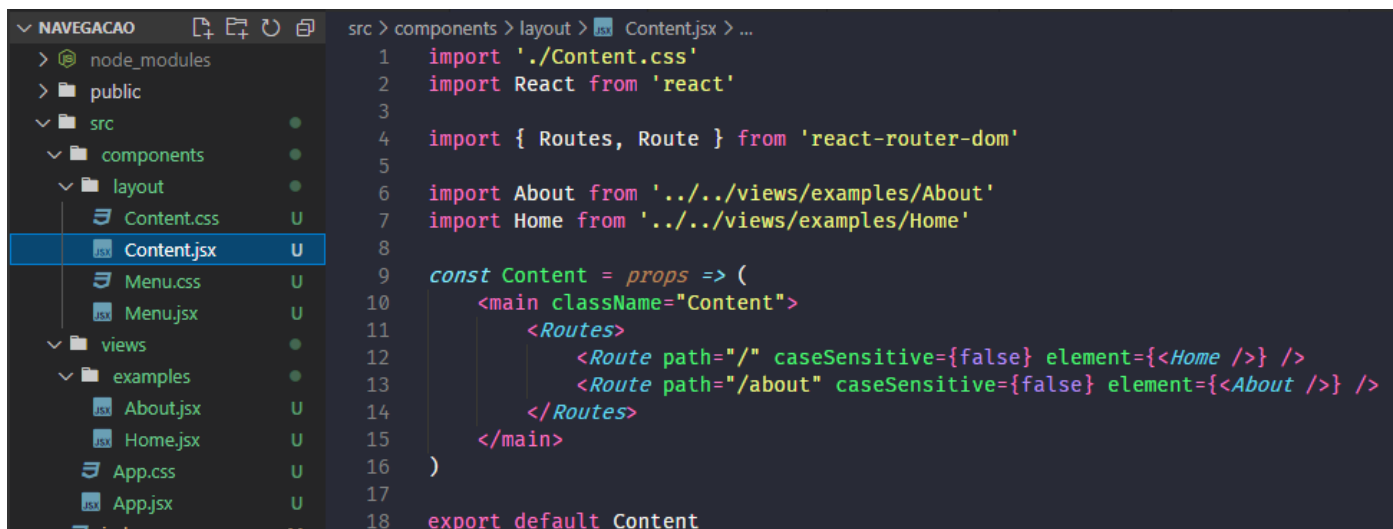
Faremos o import do BrowserRouter e envolver o menu e o conteúdo com o `<Router>` no arquivo App.jsx



```

1  import './App.css'
2  import React from "react"
3
4  import { BrowserRouter as Router } from 'react-router-dom'
5
6  import Menu from '../components/layout/Menu'
7  import Content from '../components/layout/Content'
8
9  const App = props => (
10     <div className="App">
11         <Router>
12             <Menu></Menu>
13             <Content></Content>
14         </Router>
15     </div>
16 )
17
18 export default App
  
```

Fizemos a importação de Routes, Route

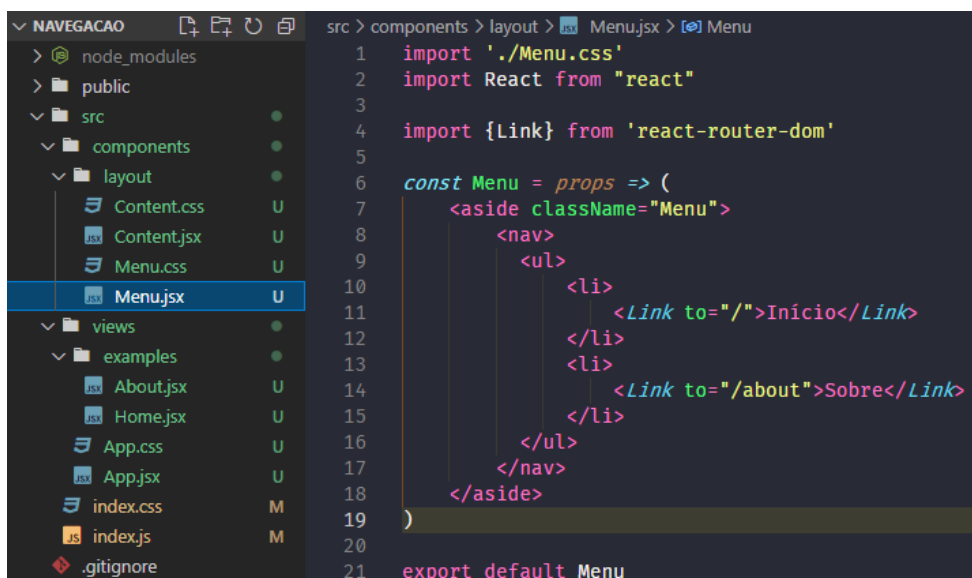


```

src > components > layout > Content.jsx > ...
1  import './Content.css'
2  import React from 'react'
3
4  import { Routes, Route } from 'react-router-dom'
5
6  import About from '../views/examples/About'
7  import Home from '../views/examples/Home'
8
9  const Content = props => (
10     <main className="Content">
11         <Routes>
12             <Route path="/" caseSensitive={false} element={<Home />} />
13             <Route path="/about" caseSensitive={false} element={<About />} />
14         </Routes>
15     </main>
16 )
17
18 export default Content

```

Faremos mais um ajuste no menu com a tag Link.



```

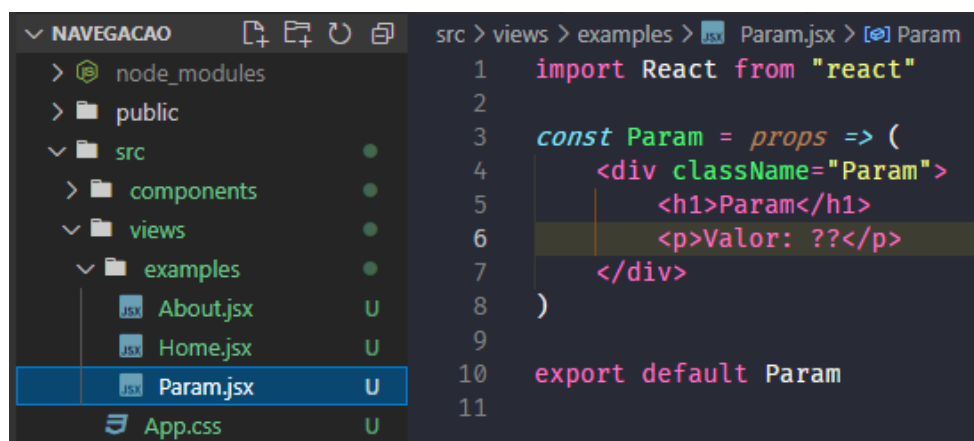
src > components > layout > Menu.jsx > Menu
1  import './Menu.css'
2  import React from "react"
3
4  import {Link} from 'react-router-dom'
5
6  const Menu = props => (
7     <aside className="Menu">
8         <nav>
9             <ul>
10                 <li>
11                     <Link to="/">Início</Link>
12                 </li>
13                 <li>
14                     <Link to="/about">Sobre</Link>
15                 </li>
16             </ul>
17         </nav>
18     </aside>
19 )
20
21 export default Menu

```



Navegação com parâmetro.

Vamos aprender agora a passar parâmetros pela URL, criando o arquivo Param.jsx na pasta examples. Usaremos a estrutura do arquivo Home.jsx

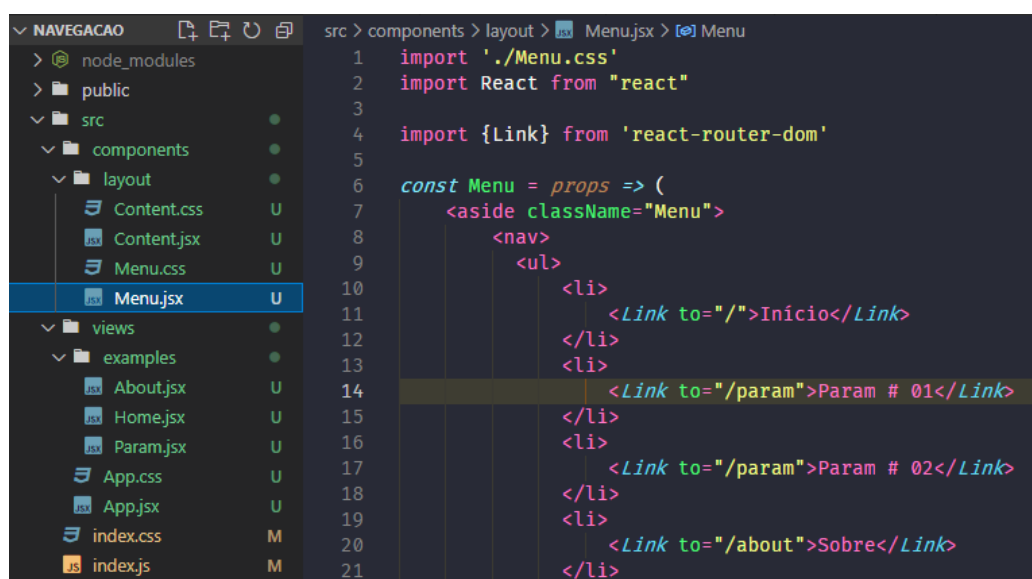


```

src > views > examples > Param.jsx
1  import React from "react"
2
3  const Param = props => (
4    <div className="Param">
5      <h1>Param</h1>
6      <p>Valor: ??</p>
7    </div>
8  )
9
10 export default Param
11

```

Precisamos referenciar esse componente na nossa aplicação em alterar dois arquivos: Menu e Conteúdo.



```

src > components > layout > Menu.jsx
1  import './Menu.css'
2  import React from "react"
3
4  import {Link} from 'react-router-dom'
5
6  const Menu = props => (
7    <aside className="Menu">
8      <nav>
9        <ul>
10         <li>
11           <Link to="/">Início</Link>
12         </li>
13         <li>
14           <Link to="/param">Param # 01</Link>
15         </li>
16         <li>
17           <Link to="/param">Param # 02</Link>
18         </li>
19         <li>
20           <Link to="/about">Sobre</Link>
21         </li>

```



```

src > components > layout > Content.jsx
1  import './Content.css'
2  import React from 'react'
3
4  import { Routes, Route } from 'react-router-dom'
5
6  import About from '../views/examples/About'
7  import Param from '../views/examples/Param'
8  import Home from '../views/examples/Home'
9
10 const Content = props => (
11   <main className="Content">
12     <Routes>
13       <Route path="/" caseSensitive={false} element={Home} />
14       <Route path="/param" caseSensitive={false} element={Param} />
15       <Route path="/about" caseSensitive={false} element={About} />
16     </Routes>
17   </main>
18 )
19
20 export default Content

```



Usamos o elemento `:id` que é uma variável podendo seu valor passar pela url.

```

src > components > layout > Content.jsx > Content
1  import './Content.css'
2  import React from 'react'
3
4  import { Routes, Route } from 'react-router-dom'
5
6  import About from '../../views/examples/About'
7  import Param from '../../views/examples/Param'
8  import Home from '../../views/examples/Home'
9
10 const Content = props => (
11   <main className="Content">
12     <Routes>
13       <Route path="/" caseSensitive={false} element={<Home />} />
14       <Route path="/param/:id" caseSensitive={false} element={<Param />} />
15       <Route path="/about" caseSensitive={false} element={<About />} />
16     </Routes>
17   </main>
18 )
19
20 export default Content
21

```

No arquivo Menu.jsx temos agora:

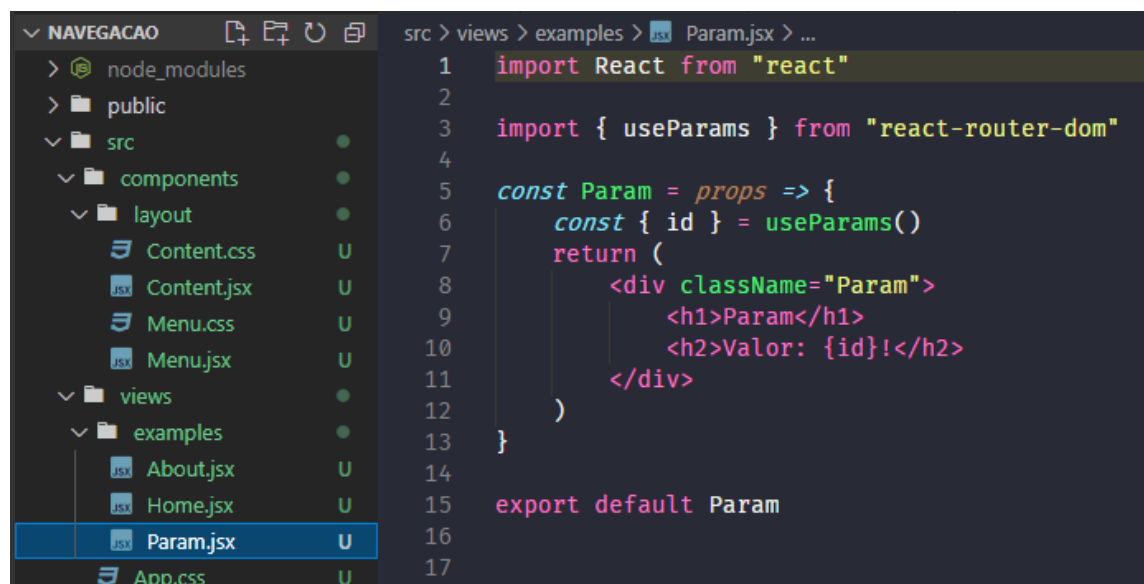
```

src > components > layout > Menu.jsx > Menu
1  import './Menu.css'
2  import React from "react"
3
4  import {Link} from 'react-router-dom'
5
6  const Menu = props => (
7    <aside className="Menu">
8      <nav>
9        <ul>
10         <li>
11           <Link to="/">Início</Link>
12         </li>
13         <li>
14           <Link to="/param/123">Param # 01</Link>
15         </li>
16         <li>
17           <Link to="/param/RECODE">Param # 02</Link>
18         </li>
19         <li>
20           <Link to="/about">Sobre</Link>
21         </li>
22       </ul>
23     </nav>
24   </aside>
25 )
26
27 export default Menu

```

Em Params.jsx podemos passar os parâmetros pelo id.

Usamos o useParams que é uma função Hook, onde podemos pegar id na forma de objeto e interpolar esse valor para ser exibido na tela.



```

src > views > examples > Param.jsx > ...
1  import React from "react"
2
3  import { useParams } from "react-router-dom"
4
5  const Param = props => {
6    const { id } = useParams()
7    return (
8      <div className="Param">
9        <h1>Param</h1>
10       <h2>Valor: {id}</h2>
11     </div>
12   )
13 }
14
15 export default Param
16
17

```

React Hooks

Hooks é uma adição que foi feita a partir do React 16.8 criando mais funcionalidades aos componentes funcionais.

Dentro do React temos componentes baseados em classe e componentes baseados em funções.

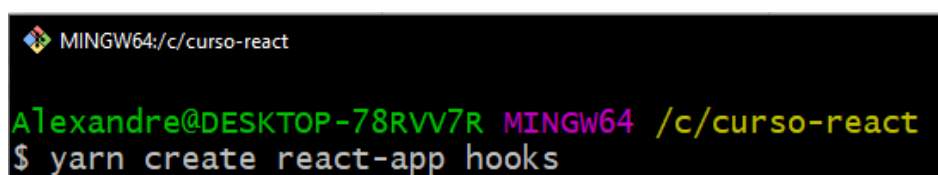
Os hooks nos ajudam a terem uma série de funcionalidades para trabalhar com mais funções dentro do React.

Vamos configurar nosso projeto, e existe o arquivo Aula-04-hooks.zip que será utilizado no nosso projeto, que nada mais é do que a pasta src com todos os arquivos que serão utilizados.

Apesar do React Router ter evoluído para a versão 6, utilizaremos a versão 5 pois trataremos aqui do React Hooks.

Na pasta curso-react, criaremos nosso projeto hooks com o comando:

yarn create react-app hooks



```

MINGW64:/c/curso-react
Alexandre@DESKTOP-78RVV7R MINGW64 /c/curso-react
$ yarn create react-app hooks

```

Vamos também utilizar o React-Router-Dom na versão 5.2.0.

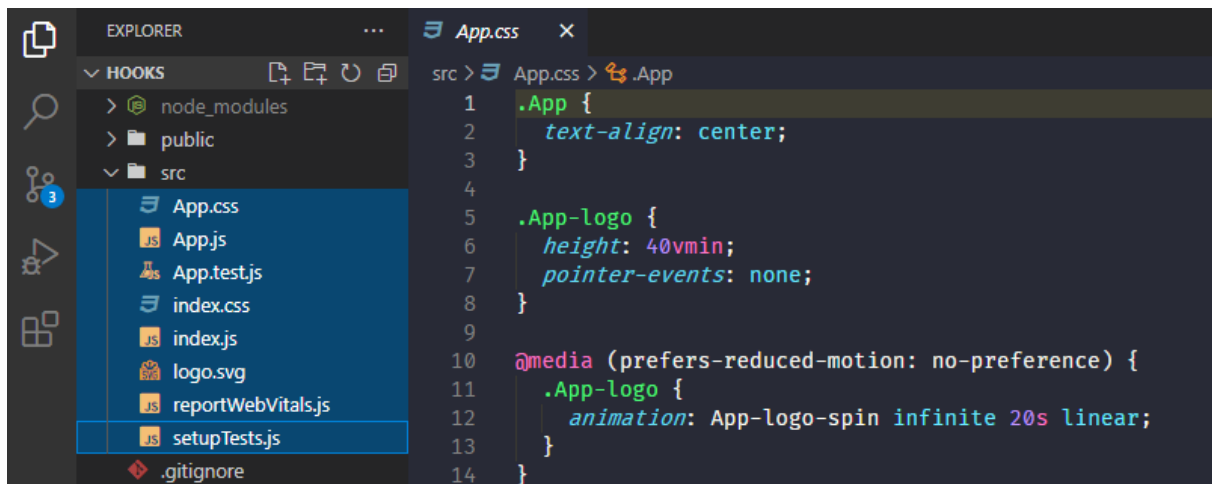
Para isso entraremos na pasta hooks e daremos o comando:

```
npm i react-router-dom@5.2.0
```

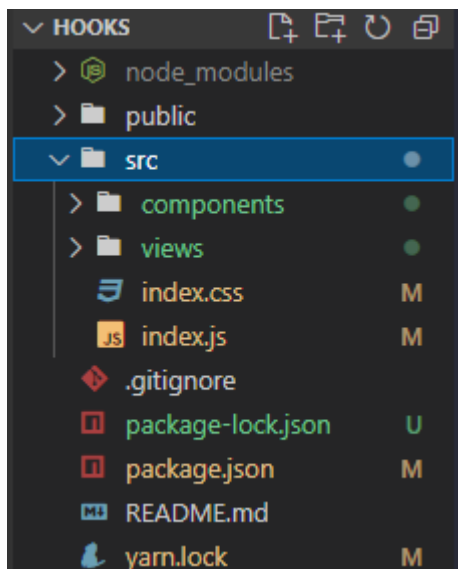
```
Alexandre@DESKTOP-78RVV7R MINGW64 /c/curso-react/hooks (main)
$ npm i react-router-dom@5.2.0
```

Iniciaremos o VSCode na pasta do projeto com o comando `code .`

Apagaremos o conteúdo da pasta `src` para colocarmos o conteúdo `Aula-04-hoks.zip`.



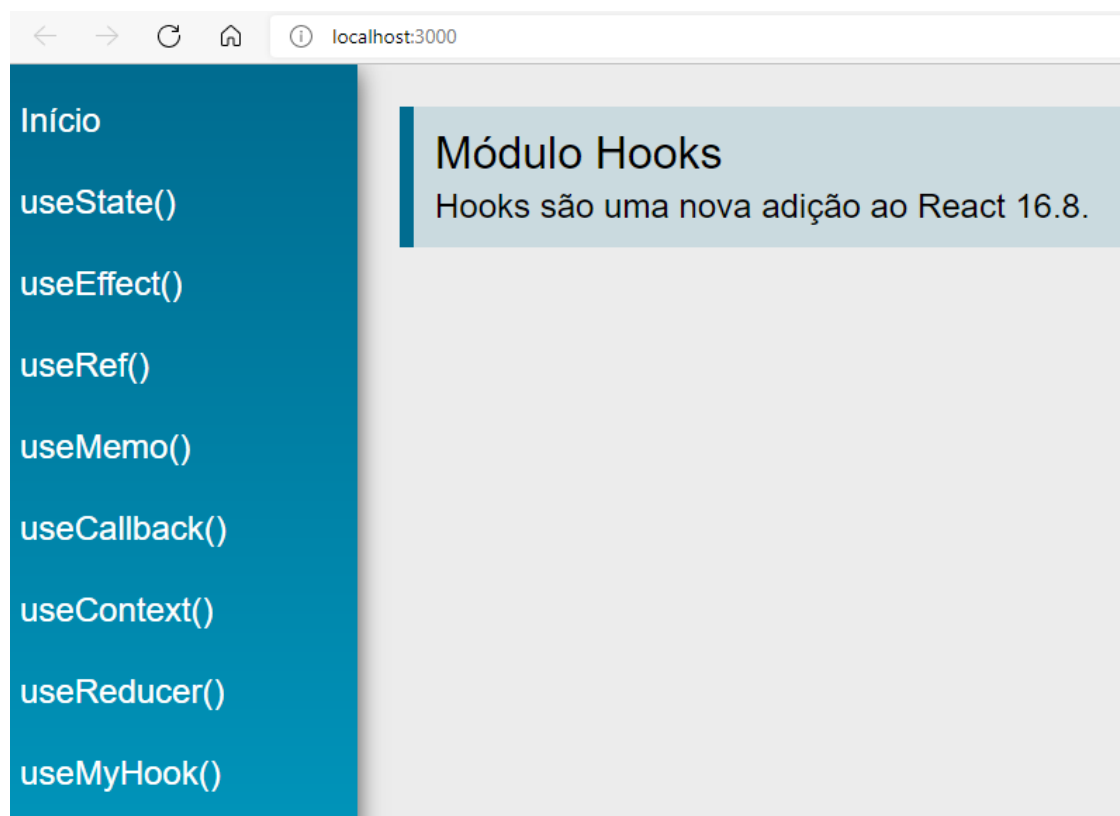
O conteúdo transferido tem esse aspecto.



Iniciaremos a nossa aplicação com o `yarn start` ou `npm start`

```
Alexandre@DESKTOP-78RVV7R MINGW64 /c/curso-react/hooks (main)
$ yarn start
```

Trabalharemos essas funções, apenas utilizaremos essa página para melhor visualizar os conceitos.



Iniciaremos com o `useState()` que é um dos mais usados.

```
UseState.jsx U X
src > views > examples > UseState.jsx > ...
1  import React from 'react'
2  import PageTitle from '../components/layout/PageTitle'
3
4  const UseState = (props) => {
5    return (
6      <div className="UseState">
7        <PageTitle
8          title="Hook UseState"
9          subtitle="Estado em componentes funcionais!"
10         />
11      </div>
12    )
13  }
14
15  export default UseState
```

Foi criado o `PageTitle` para estilizar o título e o subtítulo.

Primeira coisa é criar um estado dentro de um componente funcional, antes dos Hooks não havia essa possibilidade

Criaremos o count e o setCount que será responsável pelas alterações. Foi também criado SectionTitle para fazer as divisões entre exemplos. E adicionado dois botões.

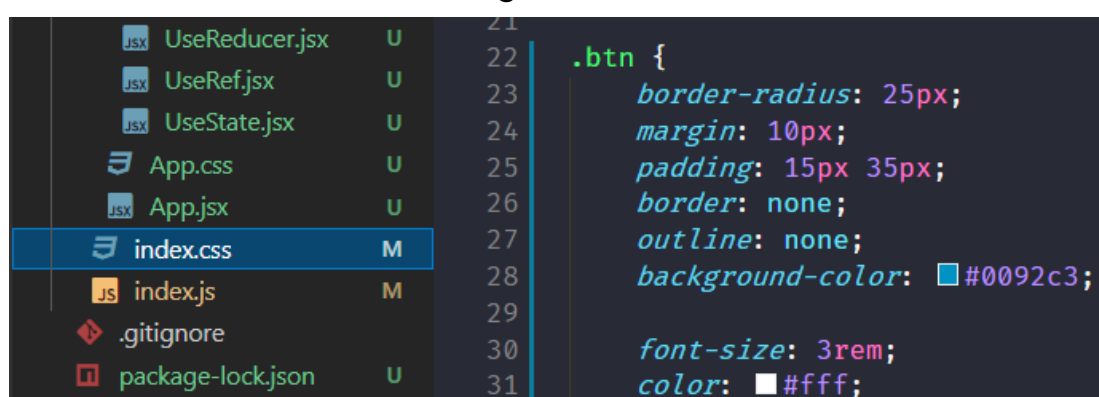
```
src > views > examples > JSX UseState.jsx > ...
1  import React, { useState } from 'react'
2  import PageTitle from '../components/layout/PageTitle'
3  import SectionTitle from '../components/layout/SectionTitle'
4
5  const UseState = (props) => {
6    const [count, setCount] = useState(0)
7
8    return (
9      <div className="UseState">
10        <PageTitle
11          title="Hook UseState"
12          subtitle="Estado em componentes funcionais!" />
13        <SectionTitle title="Exercício #01" />
14        <div className="center">
15          <span className="text">{count}</span>
16          <button className="btn">-1</button>
17          <button className="btn">+1</button>
18        </div>
19
20        <SectionTitle title="Exercício #02" />
21      </div>
22    )
23  }
24  export default UseState
```

Quando você digita **div.center** e dá um **<tab>** o VSCode constroe a div com a classe "center". Isso porque estamos dentro de um arquivo JSX

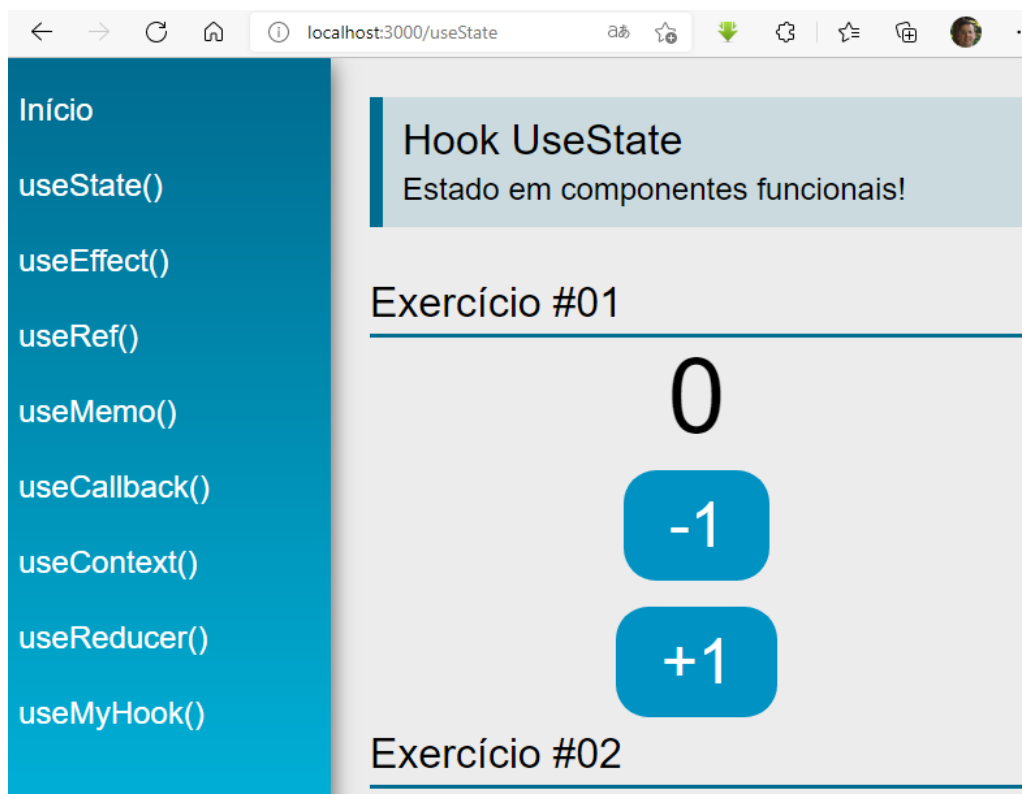


```
U 14 div.center
Emmet Abbreviation
x div.center
tle title="Exercício #02" />
<div className="center">|</div>
<div className="center"></div>
```

O botão foi estilizado com o código abaixo do index.css



```
JSX UseReducer.jsx U 21
JSX UseRef.jsx U 22
JSX UseState.jsx U 23
App.css U 24
App.jsx U 25
index.css M 26
index.js M 27
.gitignore 28
package-lock.json U 29
21
22 .btn {
23   border-radius: 25px;
24   margin: 10px;
25   padding: 15px 35px;
26   border: none;
27   outline: none;
28   background-color: #0092c3;
29
30   font-size: 3rem;
31   color: #fff;
```

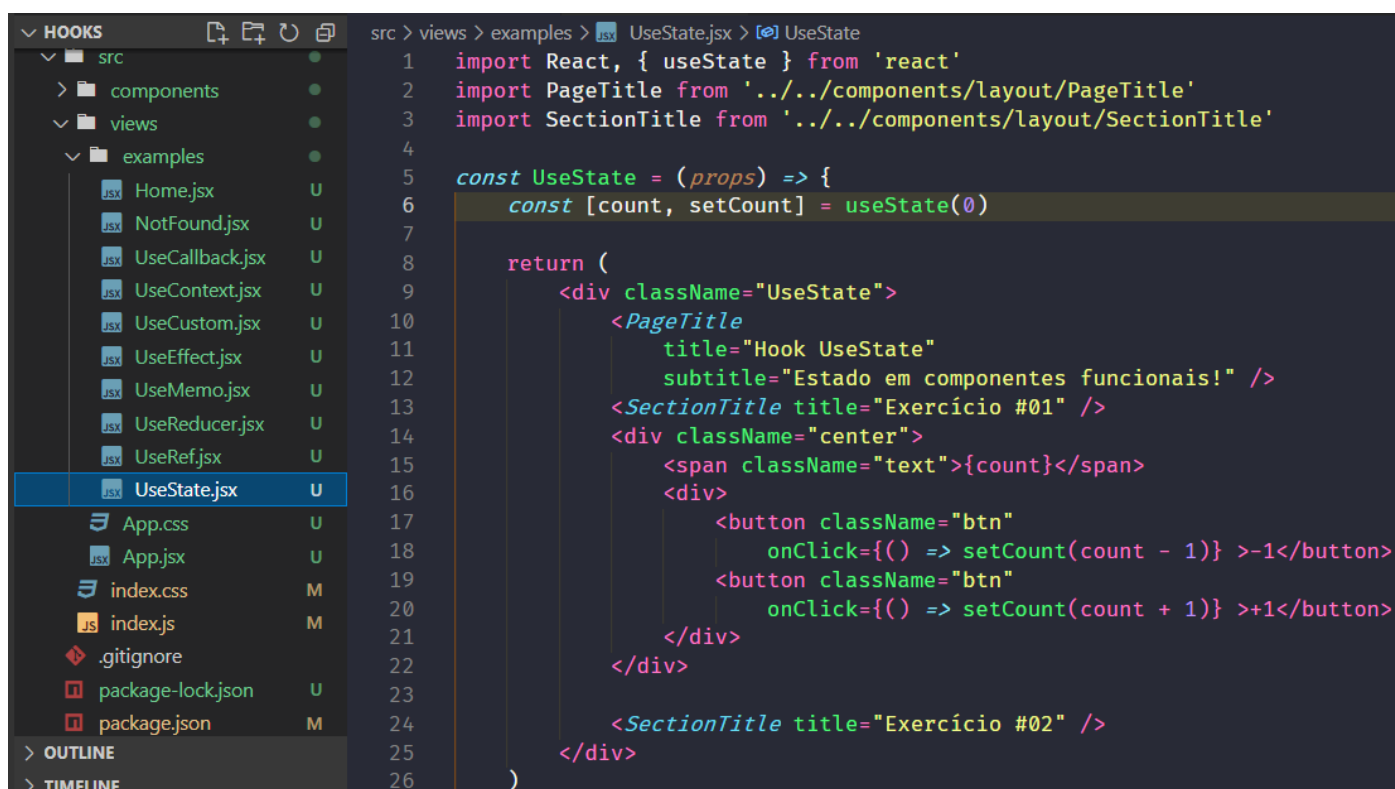


Agora vamos inserir uma função em cada botão.

O evento onClick quando acionado diminui ou aumenta uma unidade.

```
<button className="btn"
  onClick={() => setCount(count - 1)} >-1</button>
<button className="btn"
  onClick={() => setCount(count + 1)} >+1</button>
```

O estado do número é alterado a partir da função setCount.



Podemos ainda acrescentar outro botão para somar 10.

Nesse caso existe uma função que é passada como parâmetro para o `setCount` que adiciona e atualiza o valor.

```
src > views > examples > UseState.jsx > UseState
1  import React, { useState } from 'react'
2  import PageTitle from '../components/layout/PageTitle'
3  import SectionTitle from '../components/layout/SectionTitle'
4
5  const UseState = (props) => {
6    const [count, setCount] = useState(0)
7
8    return (
9      <div className="UseState">
10        <PageTitle
11          title="Hook UseState"
12          subtitle="Estado em componentes funcionais!" />
13        <SectionTitle title="Exercício #01" />
14        <div className="center">
15          <span className="text">{count}</span>
16          <div>
17            <button className="btn"
18              onClick={() => setCount(count - 1)} >-1</button>
19            <button className="btn"
20              onClick={() => setCount(count + 1)} >+1</button>
21            <button className="btn"
22              onClick={() => setCount(count => count + 10)} >+10</button>
23          </div>
24        </div>
25        <SectionTitle title="Exercício #02" />
26      </div>
27    )
28  }
29  export default UseState
30
```

