



FRONT-END DESIGN ENGINEERING

[React – API – CRUD]



[aula 22]

[Atualizado em: 22/04/2024]

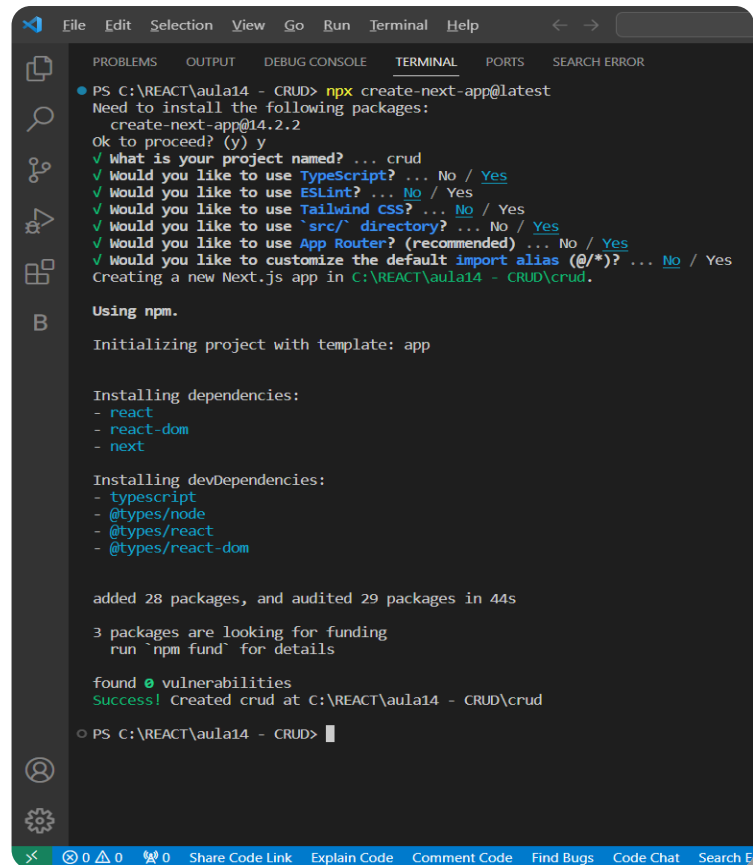


REACT – API – CRUD

REACT – API – CRUD

REACT – ESTILIZAÇÃO – STYLED COMPONENTS

✓ Vamos criar nossa aplicação



```
PS C:\REACT\aula14 - CRUD> npx create-next-app@latest
Need to install the following packages:
  create-next-app@14.2.2
Ok to proceed? (y) y
✓ What is your project named? ... crud
✓ Would you like to use TypeScript? ... No / Yes
✓ Would you like to use ESLint? ... No / Yes
✓ Would you like to use Tailwind CSS? ... No / Yes
✓ Would you like to use `src/` directory? ... No / Yes
✓ Would you like to use App Router? (recommended) ... No / Yes
✓ Would you like to customize the default import alias (@/*)? ... No / Yes
Creating a new Next.js app in C:\REACT\aula14 - CRUD\crud.

Using npm.

Initializing project with template: app

Installing dependencies:
- react
- react-dom
- next

Installing devDependencies:
- typescript
- @types/node
- @types/react
- @types/react-dom

added 28 packages, and audited 29 packages in 44s

3 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
Success! Created crud at C:\REACT\aula14 - CRUD\crud

PS C:\REACT\aula14 - CRUD>
```

REACT – API – CRUD

- ✓ Após a criação da aplicação, vamos instalar as seguintes bibliotecas
 - ✓ styled-components
 - ✓ npm install styled-components
 - ✓ react-icons
 - ✓ npm install react-icons
 - ✓ http-proxy-middle-ware
 - ✓ npm install http-proxy-middleware

REACT – API – CRUD

- ✓ Na aula de hoje vamos consumir a API criada em JAVA.
- ✓ A primeira coisa que precisamos fazer, para evitar erros de permissão de acesso (CORS) é colocar a URL da nossa API Java no package.json da seguinte forma:

REACT – API – CRUD

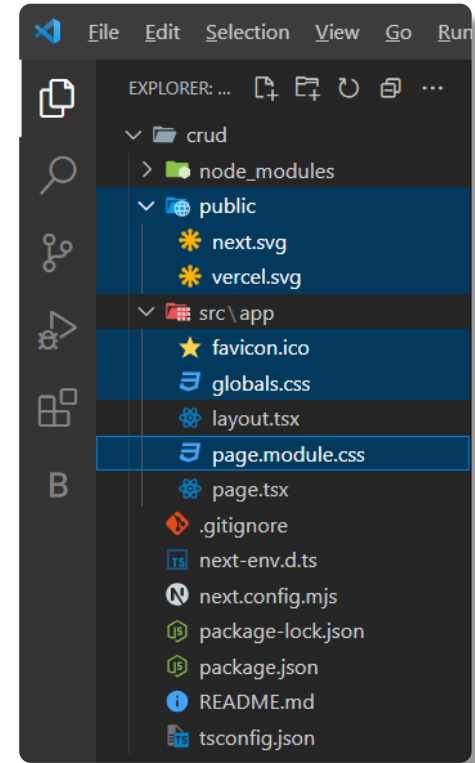
package.json

**Rode sua API Java e copie o endereço principal, a URL dela.
Não coloque os paths internos.**

Bibliotecas que foram instaladas.

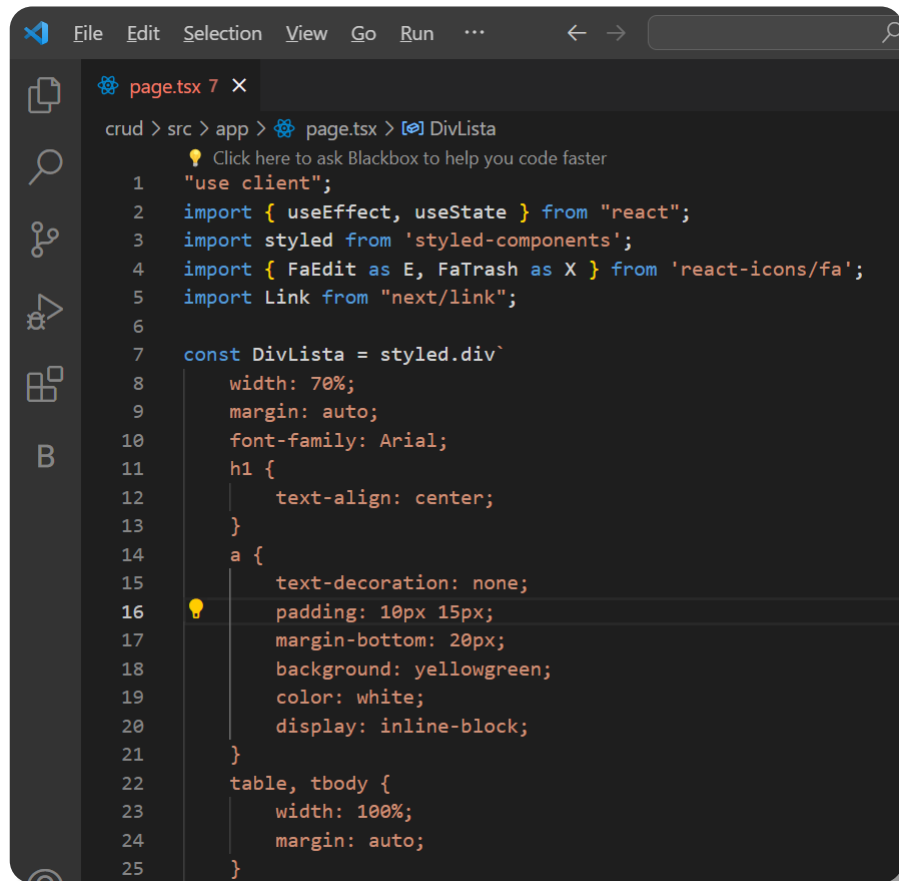
REACT – API – CRUD

✓ Vamos apagar os seguintes arquivos:



REACT – API – CRUD

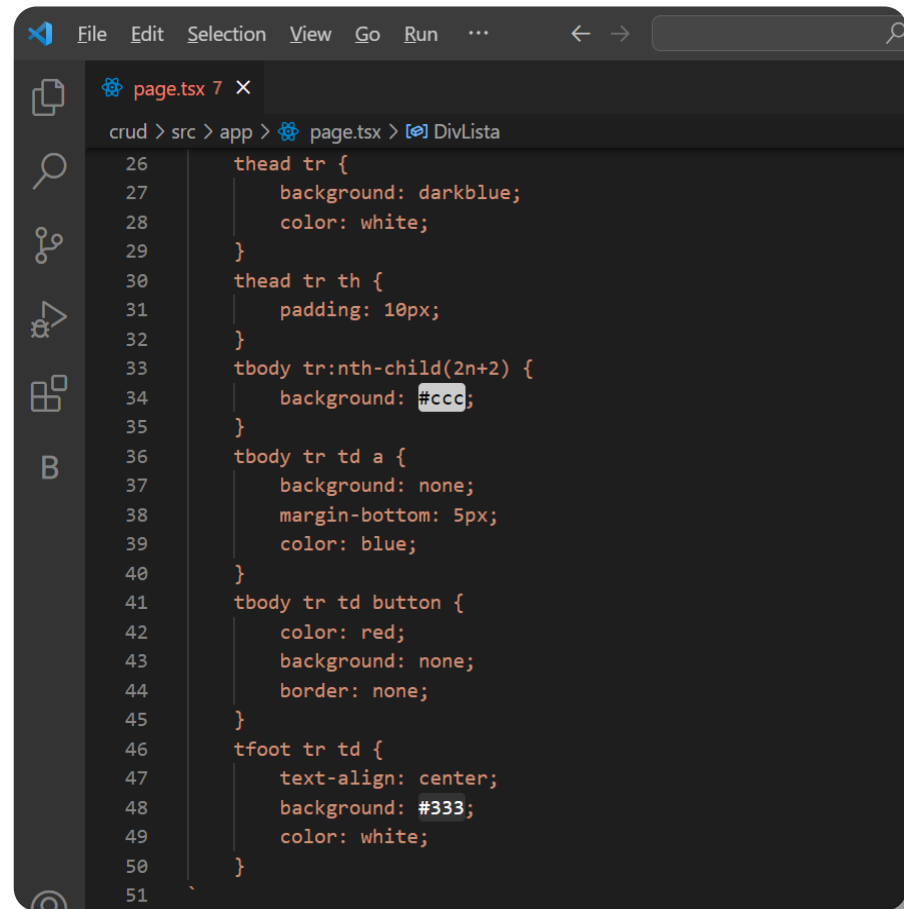
✓ Vamos modificar o arquivo page.tsx:



```
File Edit Selection View Go Run ...  
page.tsx 7 X  
crud > src > app > page.tsx > DivLista  
Click here to ask Blackbox to help you code faster  
1 "use client";  
2 import { useEffect, useState } from "react";  
3 import styled from 'styled-components';  
4 import { FaEdit as E, FaTrash as X } from 'react-icons/fa';  
5 import Link from "next/link";  
6  
7 const DivLista = styled.div`  
8   width: 70%;  
9   margin: auto;  
10  font-family: Arial;  
11  h1 {  
12    text-align: center;  
13  }  
14  a {  
15    text-decoration: none;  
16    padding: 10px 15px;  
17    margin-bottom: 20px;  
18    background: yellowgreen;  
19    color: white;  
20    display: inline-block;  
21  }  
22  table, tbody {  
23    width: 100%;  
24    margin: auto;  
25  }
```


REACT – API – CRUD

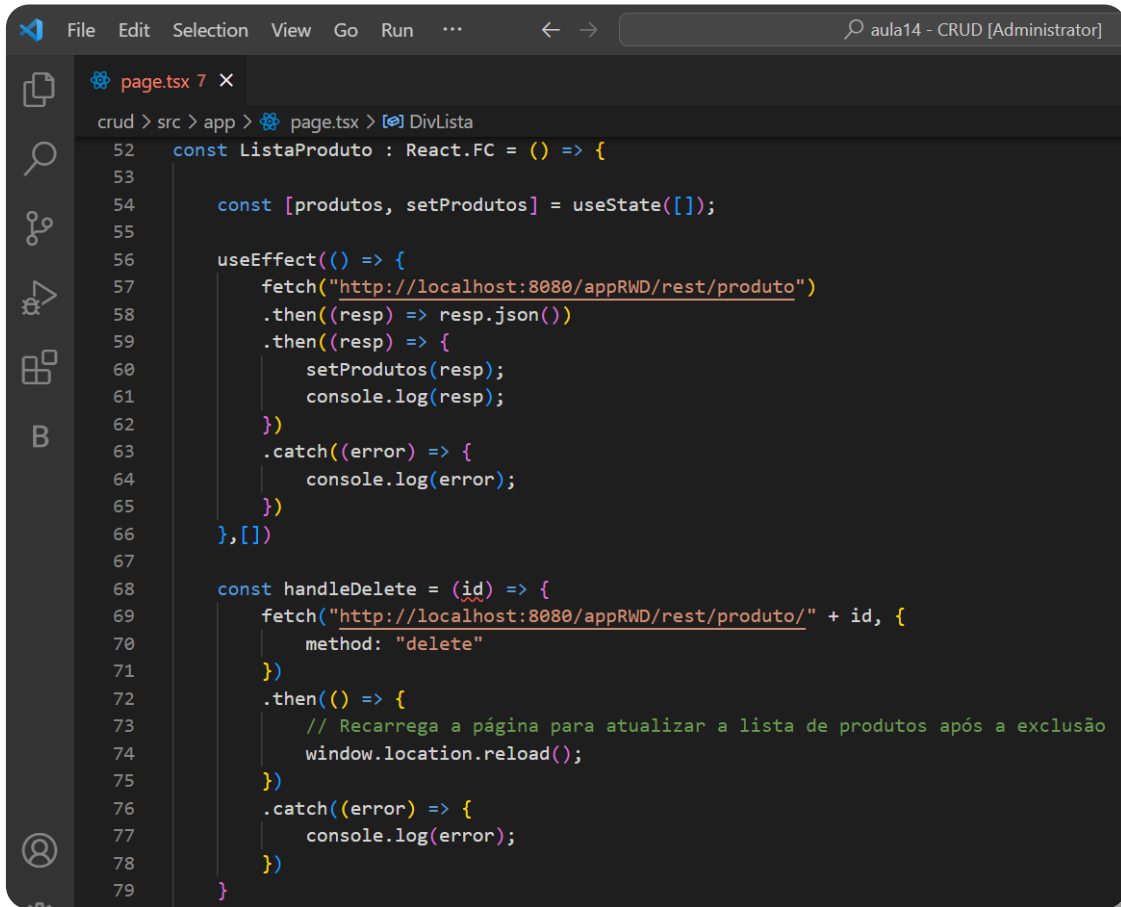
✓ Vamos modificar o arquivo page.tsx:



```
File Edit Selection View Go Run ...  
page.tsx 7 X  
crud > src > app > page.tsx > DivLista  
26   <thead> <tr> {  
27     <th> <div>  
28       background: darkblue;  
29       color: white;  
30     </div>  
31   </th> <th> <div>  
32     padding: 10px;  
33   </div>  
34 </thead> <tbody> <tr> <td> <div> <div>  
35   </div> <div>  
36     background: #ccc;  
37   </div>  
38 </td> <td> <div> <div>  
39   </div> <div>  
40     background: none;  
41     margin-bottom: 5px;  
42     color: blue;  
43   </div>  
44 </td> <td> <div> <div>  
45   </div> <div>  
46     color: red;  
47     background: none;  
48     border: none;  
49   </div>  
50 </td> <td> <div> <div>  
51   </div> <div>  
    text-align: center;  
    background: #333;  
    color: white;  
  </div>  
</td>  
</tr>  
</tbody>  
</table>
```

REACT – API – CRUD

- ✓ Vamos modificar o arquivo page.tsx:

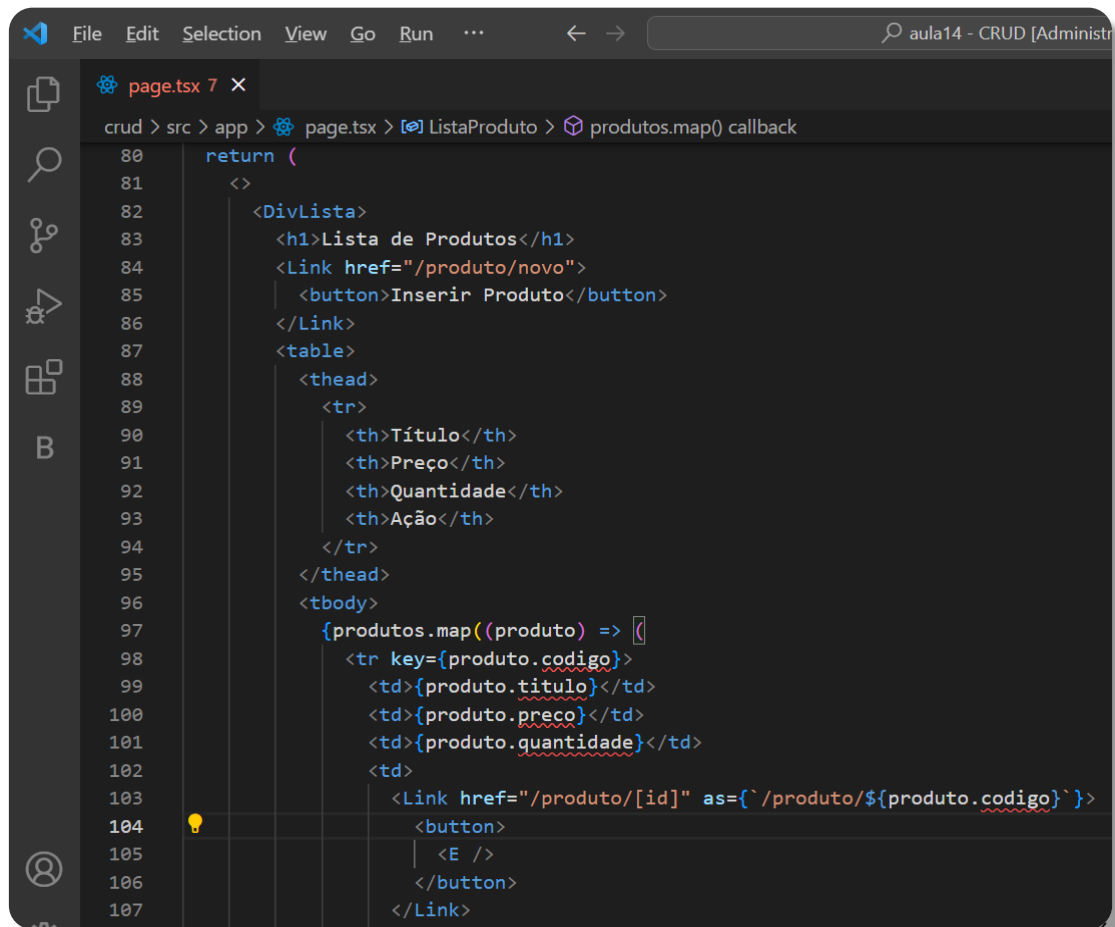


```
File Edit Selection View Go Run ... aula14 - CRUD [Administrator]

page.tsx 7 X
crud > src > app > page.tsx > DivLista
52 const ListaProduto : React.FC = () => {
53
54   const [produtos, setProdutos] = useState([]);
55
56   useEffect(() => {
57     fetch("http://localhost:8080/appRWD/rest/produto")
58       .then((resp) => resp.json())
59       .then((resp) => {
60         setProdutos(resp);
61         console.log(resp);
62       })
63       .catch((error) => {
64         console.log(error);
65       })
66   }, []);
67
68   const handleDelete = (id) => {
69     fetch("http://localhost:8080/appRWD/rest/produto/" + id, {
70       method: "delete"
71     })
72     .then(() => {
73       // Recarrega a página para atualizar a lista de produtos após a exclusão
74       window.location.reload();
75     })
76     .catch((error) => {
77       console.log(error);
78     })
79   }
```

REACT – API – CRUD

- ✓ Vamos modificar o arquivo page.tsx:



```
crud > src > app > page.tsx > ListaProduto > produtos.map() callback
80   return (
81     <>
82       <DivLista>
83         <h1>Lista de Produtos</h1>
84         <Link href="/produto/novo">
85           <button>Inserir Produto</button>
86         </Link>
87         <table>
88           <thead>
89             <tr>
90               <th>Título</th>
91               <th>Preço</th>
92               <th>Quantidade</th>
93               <th>Ação</th>
94             </tr>
95           </thead>
96           <tbody>
97             {produtos.map((produto) => (
98               <tr key={produto.codigo}>
99                 <td>{produto.titulo}</td>
100                <td>{produto.preco}</td>
101                <td>{produto.quantidade}</td>
102                <td>
103                  <Link href="/produto/[id]" as={` /produto/${produto.codigo}`}>
104                    <button>
105                      <E />
106                    </button>
107                  </Link>
```

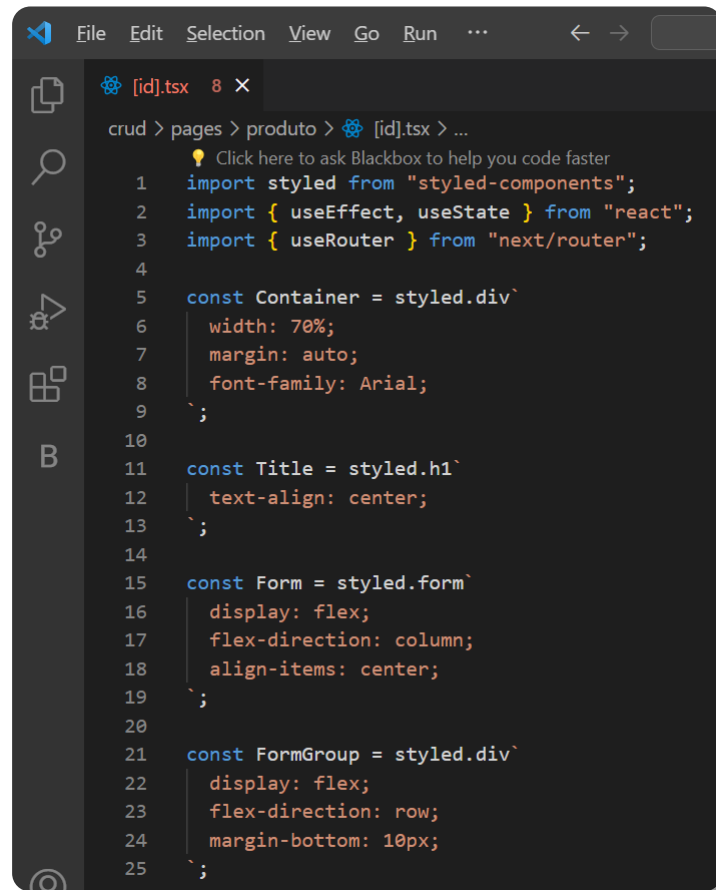
REACT – API – CRUD

- ✓ Vamos modificar o arquivo page.tsx:

```
108         <button
109             title="Excluir"
110             onClick={() => handleDelete(produto.codigo)}
111         >
112         </button>
113     </td>
114 </tr>
115 </tbody>
116 </tbody>
117 <tfoot>
118     <tr>
119         <td colspan={4}>Produtos do Servidor</td>
120     </tr>
121 </tfoot>
122 </table>
123 </DivLista>
124 </>
125 );
126 };
127
128
129 export default ListaProduto;
130
```

REACT – API – CRUD

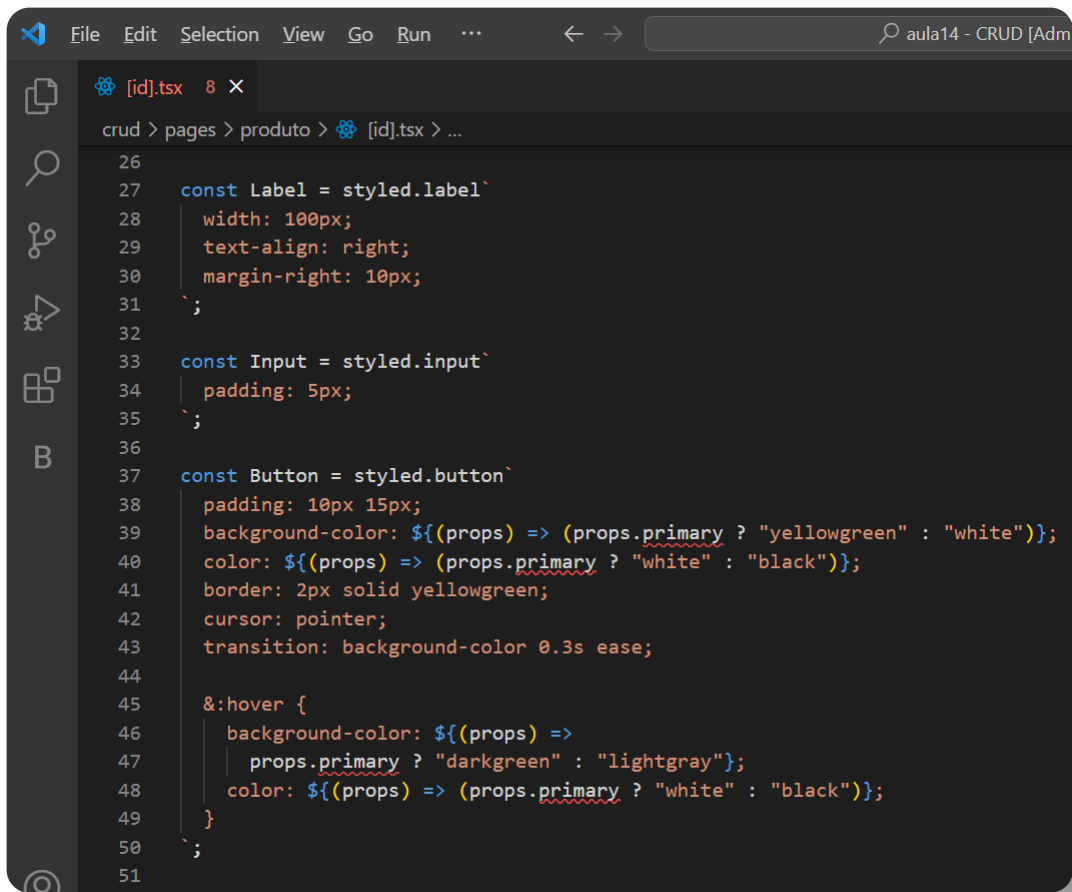
- ✓ Vamos criar o arquivo `pages/produto/[id].tsx`



```
[id].tsx 8 x
crud > pages > produto > [id].tsx > ...
Click here to ask Blackbox to help you code faster
1 import styled from "styled-components";
2 import { useEffect, useState } from "react";
3 import { useRouter } from "next/router";
4
5 const Container = styled.div`
6   width: 70%;
7   margin: auto;
8   font-family: Arial;
9 `;
10
11 const Title = styled.h1`
12   text-align: center;
13 `;
14
15 const Form = styled.form`
16   display: flex;
17   flex-direction: column;
18   align-items: center;
19 `;
20
21 const FormGroup = styled.div`
22   display: flex;
23   flex-direction: row;
24   margin-bottom: 10px;
25 `;
```

REACT – API – CRUD

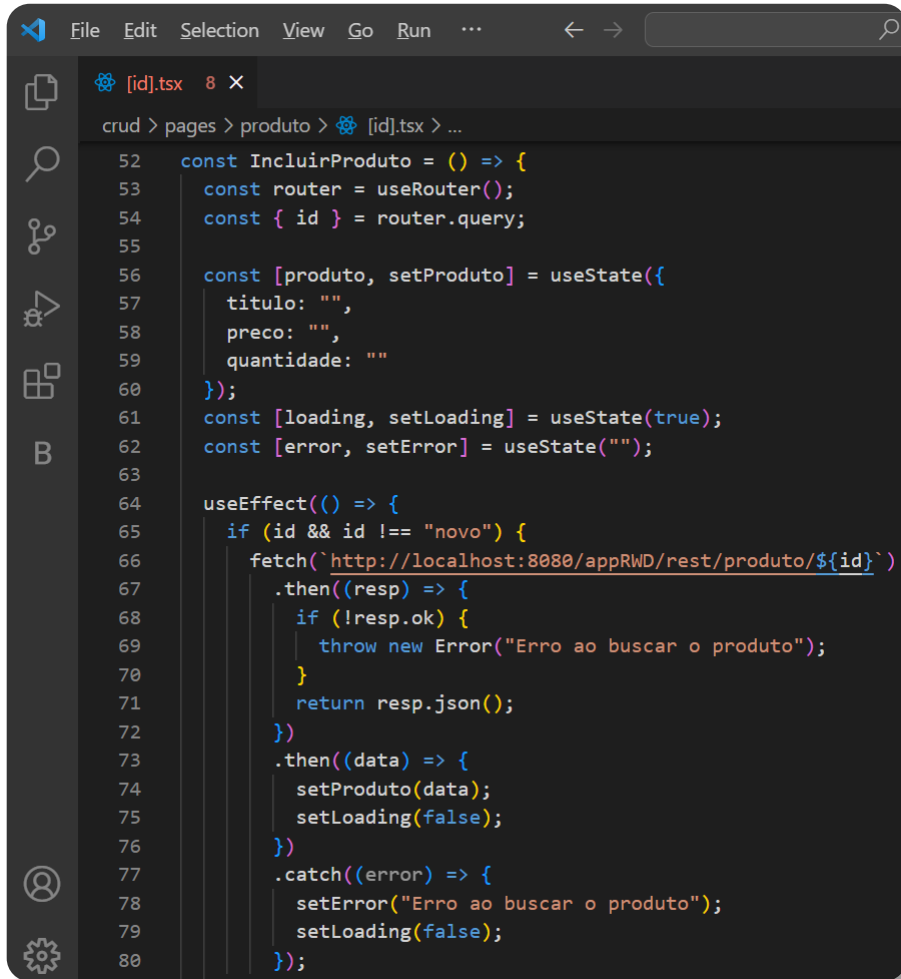
- ✓ Vamos criar o arquivo
pages/produto/[id].tsx



```
26
27  const Label = styled.label`
28    width: 100px;
29    text-align: right;
30    margin-right: 10px;
31  `;
32
33  const Input = styled.input`
34    padding: 5px;
35  `;
36
37  const Button = styled.button`
38    padding: 10px 15px;
39    background-color: ${(props) => (props.primary ? "yellowgreen" : "white")};
40    color: ${(props) => (props.primary ? "white" : "black")};
41    border: 2px solid yellowgreen;
42    cursor: pointer;
43    transition: background-color 0.3s ease;
44
45    &:hover {
46      background-color: ${(props) =>
47        props.primary ? "darkgreen" : "lightgray"};
48      color: ${(props) => (props.primary ? "white" : "black")};
49    }
50  `;
51
```

REACT – API – CRUD

- ✓ Vamos criar o arquivo
pages/produto/[id].tsx

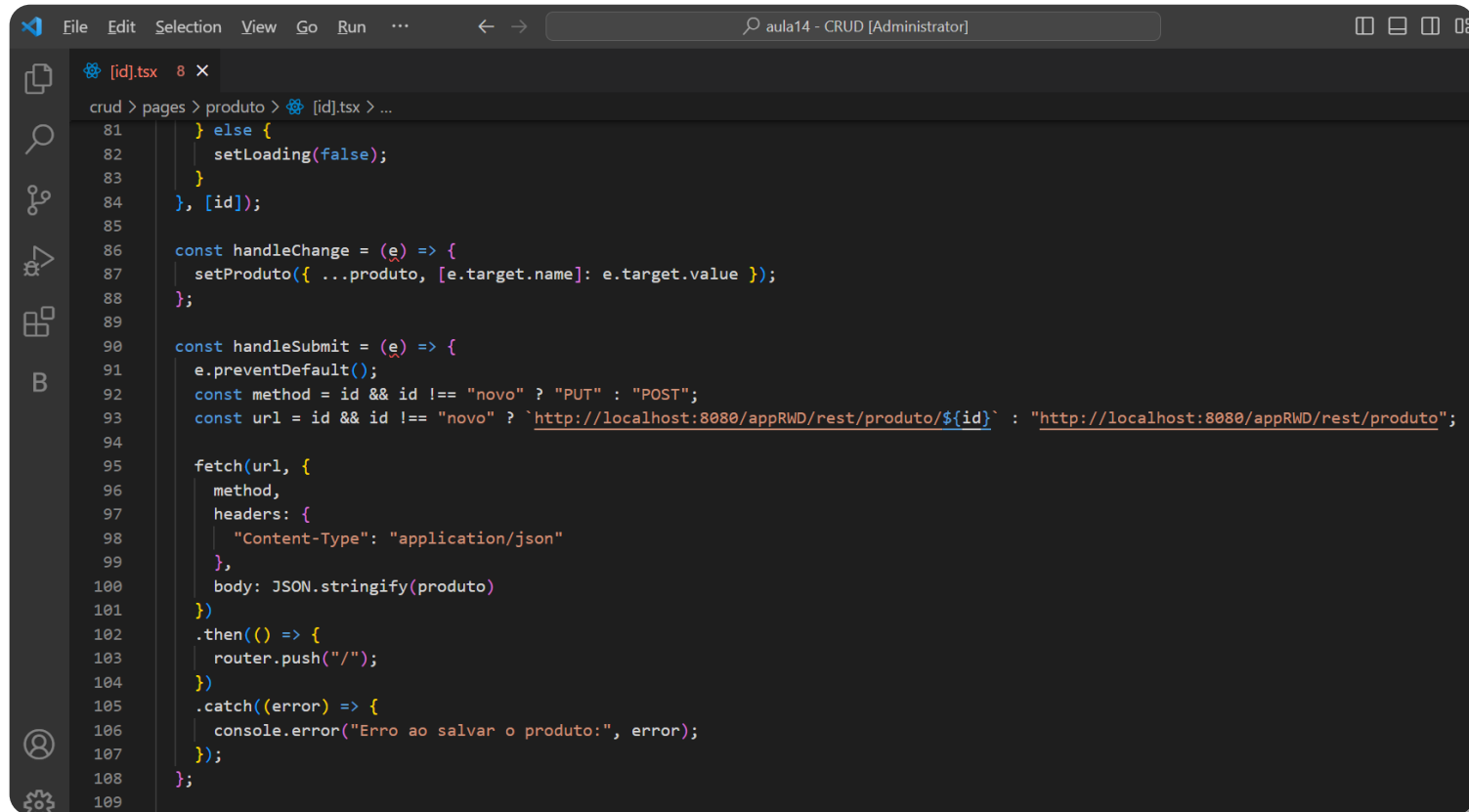


```
[id].tsx 8 X
crud > pages > produto > [id].tsx > ...

52  const IncluirProduto = () => {
53    const router = useRouter();
54    const { id } = router.query;
55
56    const [produto, setProduto] = useState({
57      titulo: "",
58      preco: "",
59      quantidade: ""
60    });
61    const [loading, setLoading] = useState(true);
62    const [error, setError] = useState("");
63
64    useEffect(() => {
65      if (id && id !== "novo") {
66        fetch(`http://localhost:8080/appRWD/rest/produto/${id}`)
67          .then((resp) => {
68            if (!resp.ok) {
69              throw new Error("Erro ao buscar o produto");
70            }
71            return resp.json();
72          })
73          .then((data) => {
74            setProduto(data);
75            setLoading(false);
76          })
77          .catch((error) => {
78            setError("Erro ao buscar o produto");
79            setLoading(false);
80          });
81      }
82    });
83  }
```

REACT – API – CRUD

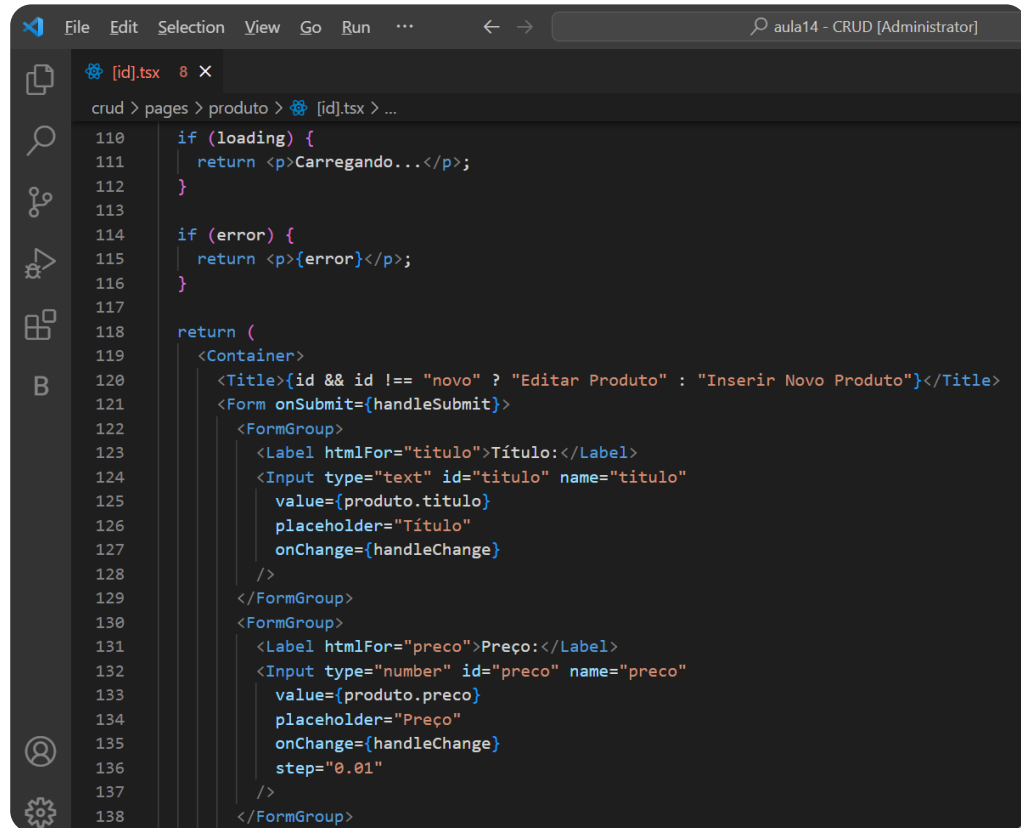
✓ Vamos criar o arquivo `pages/produto/[id].tsx`



```
[id].tsx 8 X
crud > pages > produto > [id].tsx > ...
81     } else {
82       setLoading(false);
83     }
84   }, [id]);
85
86   const handleChange = (e) => {
87     setProduto({ ...produto, [e.target.name]: e.target.value });
88   };
89
90   const handleSubmit = (e) => {
91     e.preventDefault();
92     const method = id && id !== "novo" ? "PUT" : "POST";
93     const url = id && id !== "novo" ? `http://localhost:8080/appRWD/rest/produto/${id}` : "http://localhost:8080/appRWD/rest/produto";
94
95     fetch(url, {
96       method,
97       headers: {
98         "Content-Type": "application/json"
99       },
100       body: JSON.stringify(produto)
101     })
102     .then(() => {
103       router.push("/");
104     })
105     .catch((error) => {
106       console.error("Erro ao salvar o produto:", error);
107     });
108   };
109
```


REACT – API – CRUD

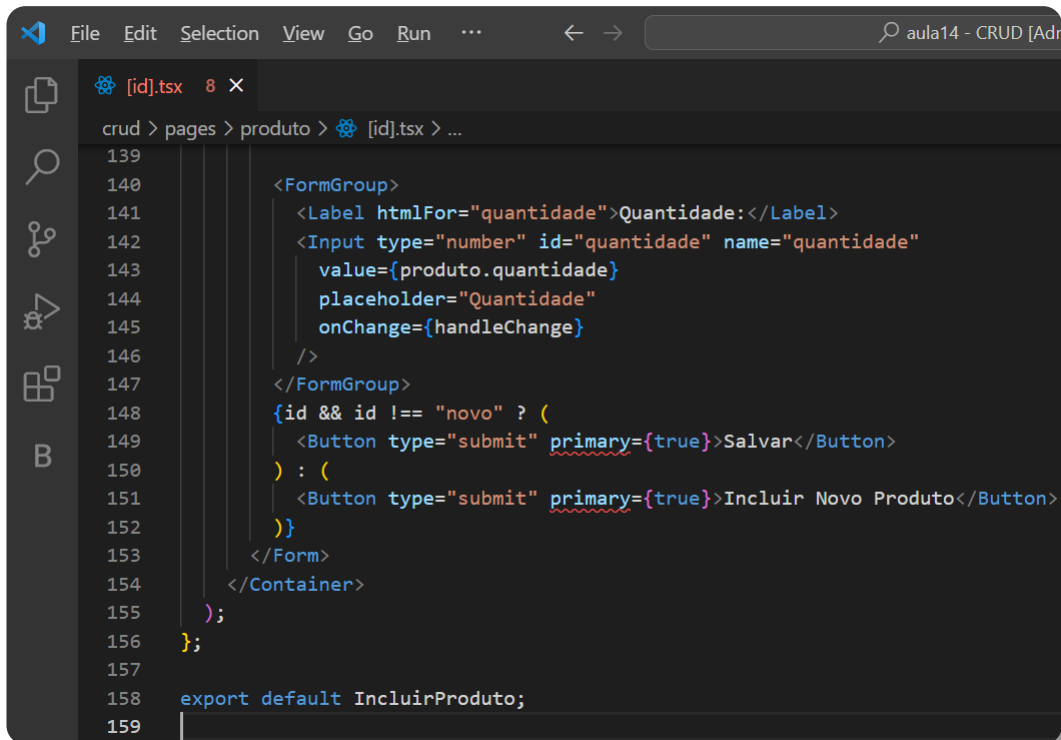
- ✓ Vamos criar o arquivo `pages/produto/[id].tsx`



```
110   if (loading) {
111     return <p>Carregando...</p>;
112   }
113
114   if (error) {
115     return <p>{error}</p>;
116   }
117
118   return (
119     <Container>
120       <Title>{id && id !== "novo" ? "Editar Produto" : "Inserir Novo Produto"}</Title>
121       <Form onSubmit={handleSubmit}>
122         <FormGroup>
123           <Label htmlFor="titulo">Título:</Label>
124           <Input type="text" id="titulo" name="titulo"
125             value={produto.titulo}
126             placeholder="Título"
127             onChange={handleChange}
128           />
129         </FormGroup>
130         <FormGroup>
131           <Label htmlFor="preco">Preço:</Label>
132           <Input type="number" id="preco" name="preco"
133             value={produto.preco}
134             placeholder="Preço"
135             onChange={handleChange}
136             step="0.01"
137           />
138         </FormGroup>
```

REACT – API – CRUD

- ✓ Vamos criar o arquivo `pages/produto/[id].tsx`



```
139
140
141   <FormGroup>
142     <Label htmlFor="quantidade">Quantidade:</Label>
143     <Input type="number" id="quantidade" name="quantidade"
144           value={produto.quantidade}
145           placeholder="Quantidade"
146           onChange={handleChange}
147     />
148   </FormGroup>
149   {id && id !== "novo" ? (
150     <Button type="submit" primary={true}>Salvar</Button>
151   ) : (
152     <Button type="submit" primary={true}>Incluir Novo Produto</Button>
153   )}
154 </Form>
155 </Container>
156 );
157 };
158
159 export default IncluirProduto;
```

REFERÊNCIAS BIBLIOGRÁFICAS

ANTONIO, C. Pro React: Build Complex Front-End Applications in a Composable Way With React. Apress, 2015.

BOSWELL, D; FOUCHER, T. The Art of Readable Code: Simple and Practical Techniques for Writing Better Code. Estados Unidos: O'Reilly Media, 2012.

BRITO, Robin Cris. Android Com Android Studio - Passo A Passo. Editora Ciência Moderna.

BUNA, S. React Succinctly. Estados Unidos: [s.n], 2016. Disponível em: <www.syncfusion.com/ebooks/reactjs_succinctly>. Acesso em: 12 de janeiro de 2023.

FACEBOOK (2019a). React: Getting Started. React Docs, 2019. Disponível em: <reactjs.org/docs/react-api.html>. Acesso em: 13 de janeiro de 2023.

FACEBOOK (2019b). React Without ES6. React Docs, 2019. Disponível em: <reactjs.org/docs/react-without-es6.html>. Acesso em: 10 de janeiro de 2023.

FACEBOOK (2019c). React Without JSX. React Docs, 2019. Disponível em: <reactjs.org/docs/react-without-jsx.html>. Acesso em: 10 de janeiro de 2023.

FREEMAN, Eric ROBSON, Elisabeth. Use a Cabeça! Programação em HTML5. Rio de Janeiro: Editora Alta Books, 2014

GACKENHEIMER, C. Introduction to React: Using React to Build scalable and efficient user interfaces.[s.i.]: Apress, 2015.

HUDSON, P. Hacking with React. 2016. Disponível em: <www.hackingwithreact.com/read/1/3/introduction-to-jsx>. Acesso em: 13 janeiro de 2023.

KOSTRZEWA, D. Is React.js the Best JavaScript Framework in 2018? 2018. Disponível em: <hackernoon.com/is-react-js-the-best-javascript-framework-in-2018-264a0eb373c8>. Acesso em: janeiro de 2023.

MARTIN, R. Clean Code: A Handbook of Agile Software Craftsmanship. Estados Unidos: Prentice Hall, 2009.

MDN WEB DOCS. Guia JavaScript. Disponível em <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide>>. Acessado em 29 de janeiro de 2023.

REFERÊNCIAS BIBLIOGRÁFICAS

NELSON, J. Learn React's Fundamentals Without the Buzzwords? 2018. Disponível em: <jamesknelson.com/learn-react-fundamentals-sans-buzzwords>. Acesso em: 12 janeiro de 2023.

NIELSEN, J. Response Times: The 3 Important Limits. 1993. Disponível em: <www.nngroup.com/articles/response-times-3-important-limits>. Acesso em: 10 janeiro de 2023.

O'REILLY, T. What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. 2005. Disponível em: <www.oreilly.com/pub/a/web2/archive/what-is-web-20.html#mememap>. Acesso em: 10 de janeiro de 2023.

PANDIT, N. What Is ReactJS and Why Should We Use It? 2018. Disponível em: <www.c-sharpcorner.com/article/what-and-why-reactjs>. Acesso em: 12 de janeiro de 2023.

RAUSCHMAYER, A. Speaking JavaScript: An In-Depth Guide for Programmers. Estados Unidos: O'Reilly Media, 2014.

REACTIVA. O arquivo package-lock.json. Disponível em: <<https://nodejs.reativa.dev/0020-package-lock-json/index>>. Acessado em 13 de janeiro de 2023.

_____. O guia do package.json. Disponível em: <<https://nodejs.reativa.dev/0019-package-json/index>>. Acessado em 13 de janeiro de 2023.

RICOY, L. Desmitificando React: Uma Reflexão para Iniciantes. 2018. Disponível em: <medium.com/trainingcenter/desmitificando-react-uma-reflex%C3%A3o-para-iniciantes-a57af90b6114>. Acesso em: 13 janeiro de 2023.

SILVA, Maurício Samy. Ajax com jQuery: requisições Ajax com a simplicidade de jQuery. São Paulo: Novatec Editora, 2009.

_____. Construindo Sites com CSS e XHTML. Sites Controlados por Folhas de Estilo em Cascata. São Paulo: Novatec, 2010.

_____. CSS3 - Desenvolva aplicações web profissionais com o uso dos poderosos recursos de estilização das CSS. São Paulo: Novatec Editora, 2010.

STACKOVERFLOW. Most Popular Technologies: Web Frameworks. Developer Survey Results, StackOverflow, 2019. Disponível em: <insights.stackoverflow.com/survey/2019#technology>. Acesso em: 13 de janeiro de 2023.

REFERÊNCIAS BIBLIOGRÁFICAS

W3C. HTML5 - A linguagem de marcação que revolucionou a web. São Paulo: Novatec Editora, 2010.

_____. A vocabulary and associated APIs for HTML and XHTML. Disponível em <<https://www.w3.org/TR/2018/SPSD-html5-20180327/>>. Acessado em 28 de abril de 2020, às 20h53min.

_____. Cascading Style Sheets, level 1. Disponível em <<https://www.w3.org/TR/2018/SPSD-CSS1-20180913/>>. Acessado em 28 de abril de 2020, às 21h58min.

_____. Cascading Style Sheets, level 2 Revision 2. Disponível em <<https://www.w3.org/TR/2016/WD-CSS22-20160412/>>. Acessado em 28 de abril de 2020, às 22h17min.

_____. Cascading Style Sheets, level 2. Disponível em <<https://www.w3.org/TR/2008/REC-CSS2-20080411/>>. Acessado em 28 de abril de 2020, às 22h03min.

_____. Cascading Style Sheets, level 3. Disponível em <<https://www.w3.org/TR/css-syntax-3/>>. Acessado em 28 de abril de 2020, às 22h18min.

_____. HTML 3.2 Reference Specification. Disponível em <<https://www.w3.org/TR/2018/SPSD-html32-20180315/>>. Acessado em 28 de abril de 2020, às 19h37min.

_____. HTML 4.0 Specification. Disponível em <<https://www.w3.org/TR/1998/REC-html40-19980424/>>. Acessado em 28 de abril de 2020, às 19h53min.

_____. HTML 4.01 Specification. Disponível em <<https://www.w3.org/TR/2018/SPSD-html401-20180327/>>. Acessado em 28 de abril de 2020, às 20h04min.

_____. Cascading Style Sheets, level 2 Revision 1. Disponível em <<https://www.w3.org/TR/CSS2/>>. Acessado em 28 de abril de 2020, às 22h13min.

WIKIPEDIA. JavaScript. Disponível em <<https://pt.wikipedia.org/wiki/JavaScript>>. Acessado em 29 de abril de 2020, às 10h.

- Dúvidas?
- Críticas?
 - Sugestões?
 - Ameaças?