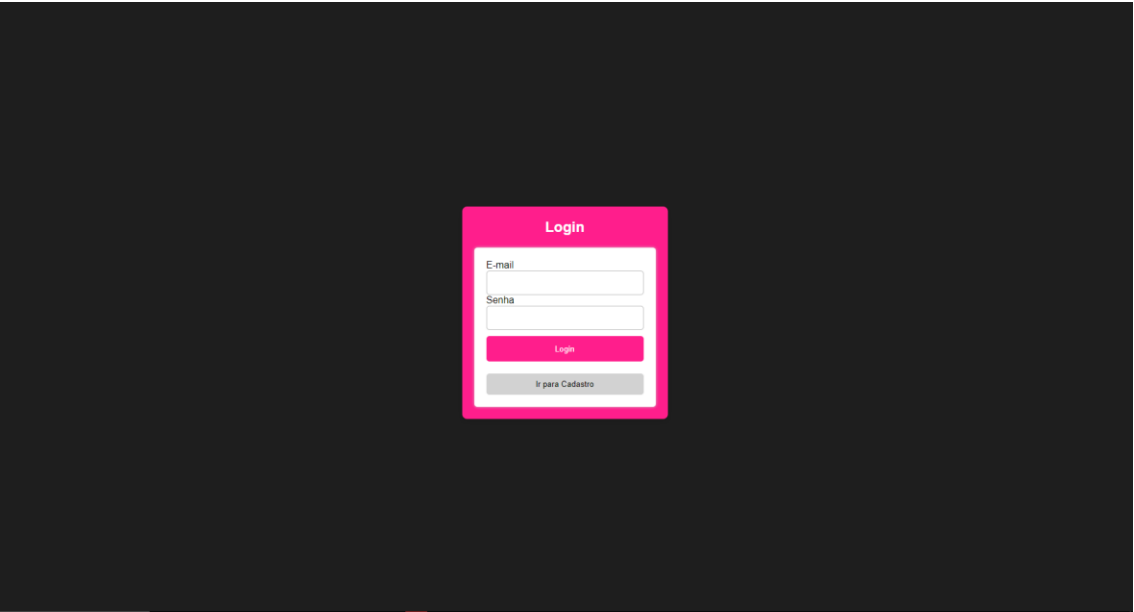


Nome: Artur Martins

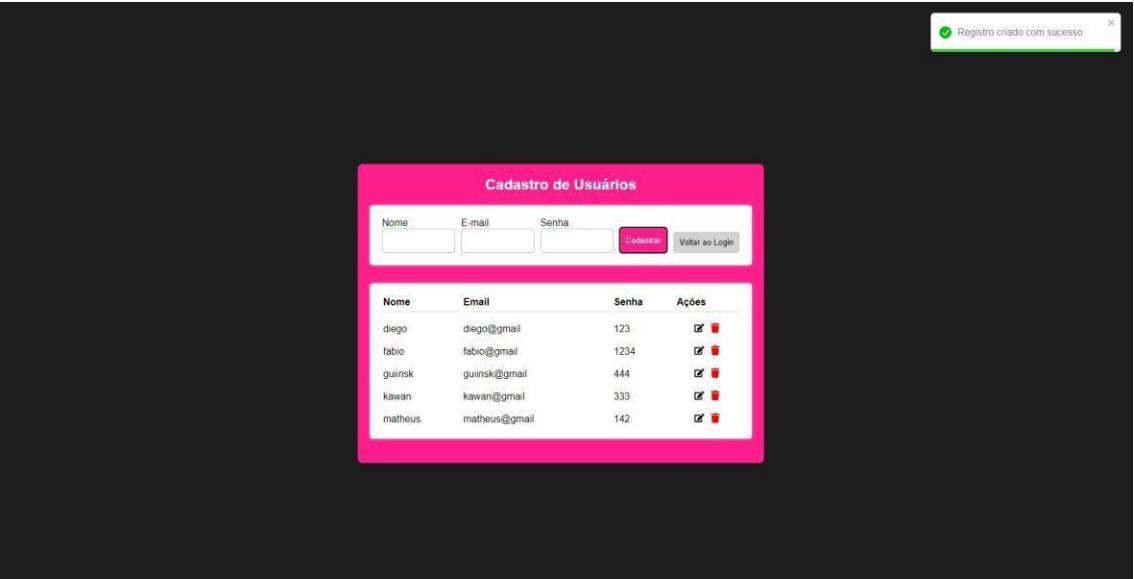
Tela de Login



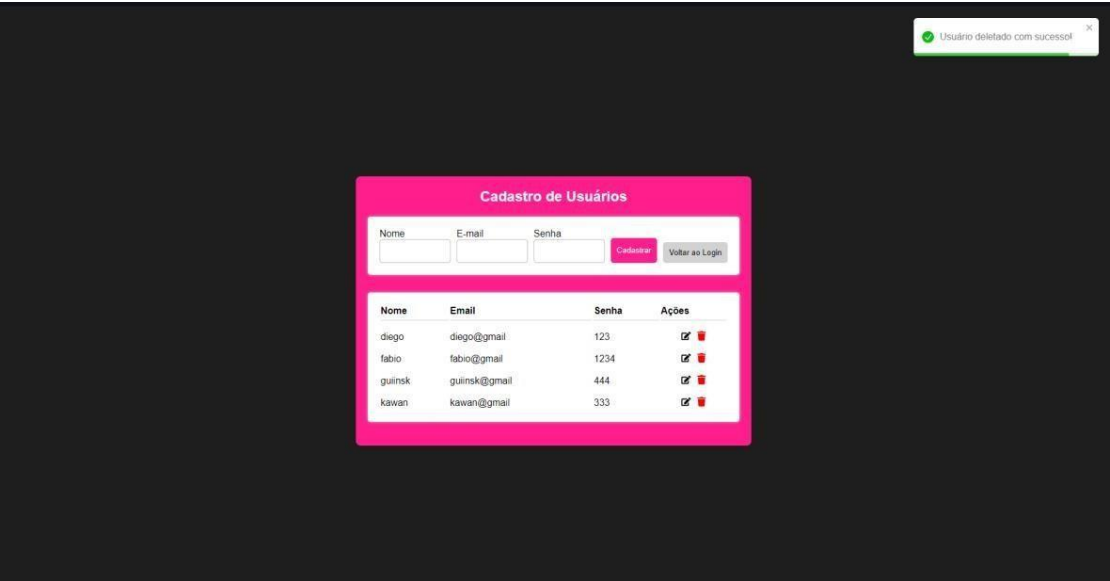
Cadastro de Usuários



Cadastro de Novo Usuário



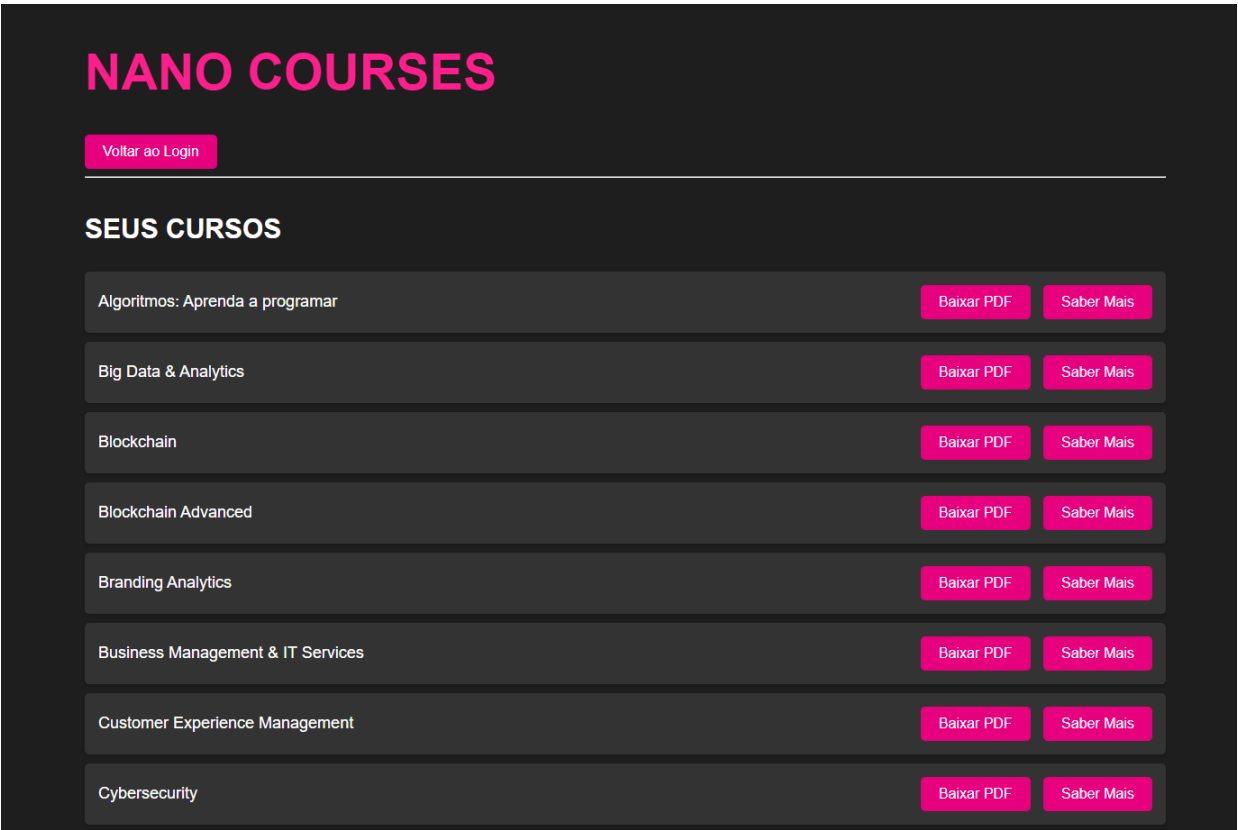
Deletando Usuário



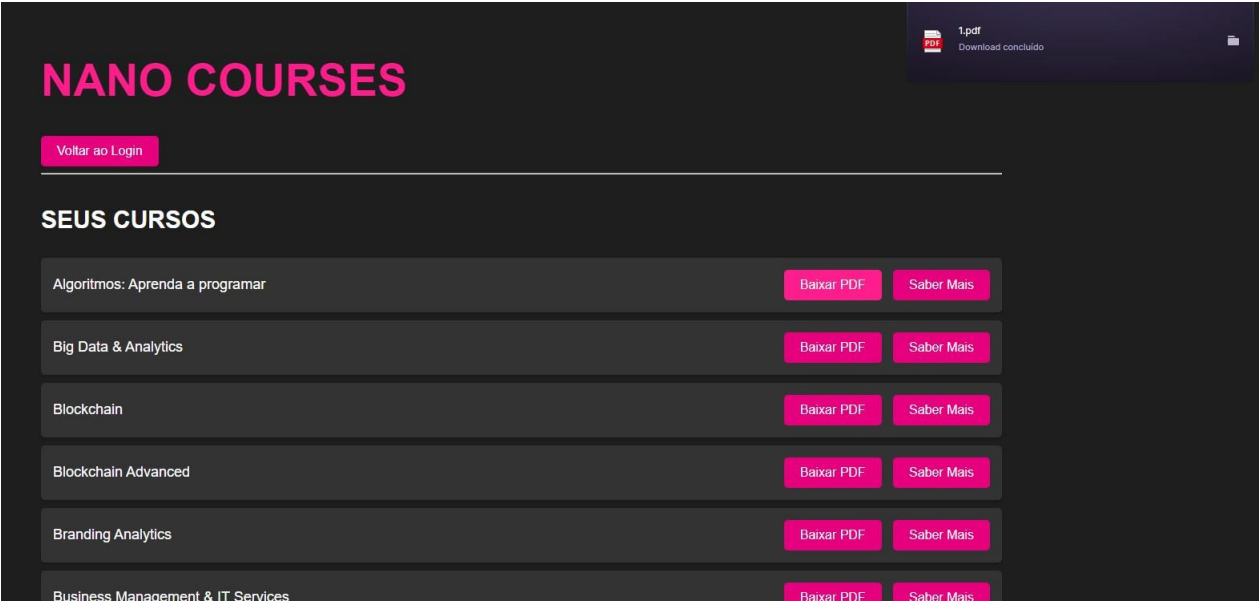
Editando Usuário



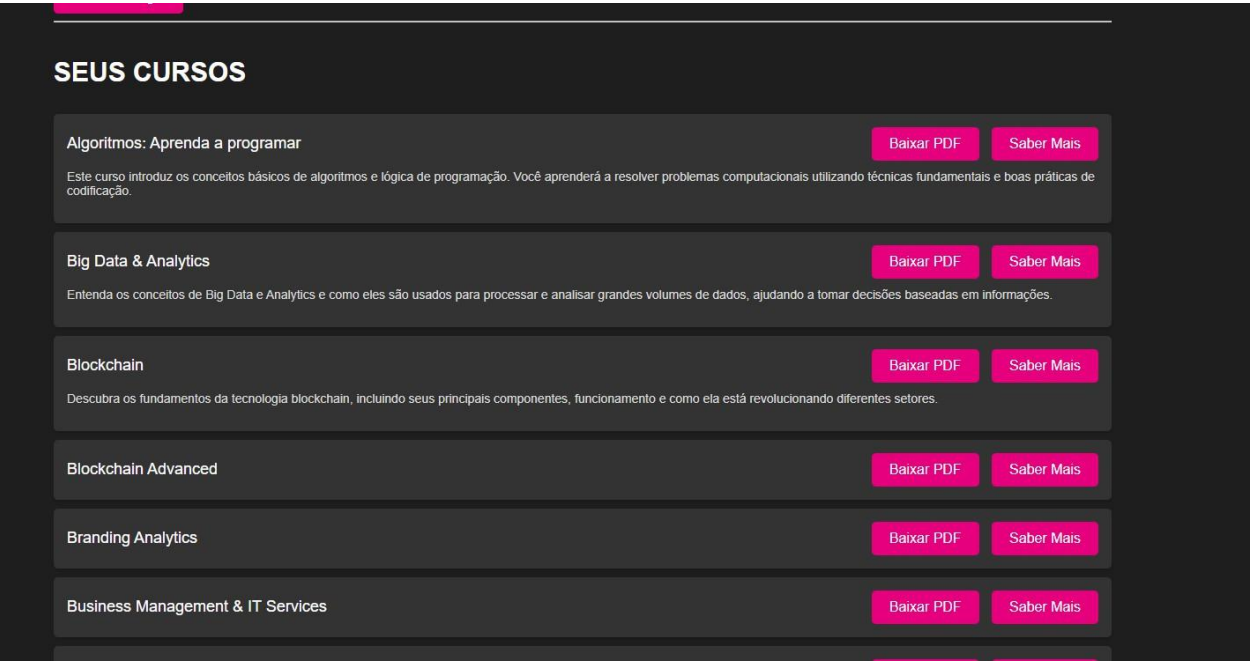
Ao logar é direcionado a tela de cursos



Ao pressionar os botões “Baixar PDF” será feito o download do PDF do curso.



Ao pressionar os botões “Baixar Saiba Mais” irá aparecer um breve resumo sobre o curso.



Controllers/usuarios.js

```
1 import { db } from "../db.js";
2
3 // Função para processar o login
4 export const login = (req, res) => {
5   const { email, senha } = req.body;
6
7   // Verifica no banco de dados se o usuário existe
8   const q = "SELECT * FROM usuarios WHERE email = ? AND senha = ?";
9   db.query(q, [email, senha], (err, results) => {
10     if (err) {
11       return res.status(500).json({ message: "Erro interno no servidor", error: err });
12     }
13
14     if (results.length > 0) {
15       // login bem-sucedido
16       return res.status(200).json({ message: "login bem-sucedido!", user: results[0] });
17     } else {
18       // Credenciais inválidas
19       return res.status(401).json({ message: "Credenciais inválidas!" });
20     }
21   });
22 };
23
24 // Função para listar os usuários
25 export const getusuarios = (_, res) => {
26   const q = "SELECT * FROM usuarios ORDER BY nome";
27   db.query(q, (err, data) => {
28     if (err) return res.status(500).json({ message: "Erro ao buscar usuários", error: err });
29     return res.status(200).json(data);
30   });
31 };
32
33 // Função para adicionar um novo usuário
34 export const addusuarios = (req, res) => {
35   console.log("Dados recebidos:", req.body); // Log para ver os dados recebidos
36
37   const q = "INSERT INTO usuarios ('nome', 'email', 'senha') VALUES (?)";
38   const values = [req.body.nome, req.body.email, req.body.senha];
39
40   db.query(q, [values], (err) => {
41     if (err) {
42       console.error("Erro ao inserir no banco de dados:", err); // log do erro
43       return res.status(500).json({ message: "Erro ao criar usuário", error: err });
44     }
45     return res.status(200).json("Registro criado com sucesso");
46   });
47 };
48
```

```

48
49 // Função para atualizar um usuário
50 export const updateusuarios = (req, res) => {
51   const q = "UPDATE usuarios SET `nome` = ?, `email` = ?, `senha` = ? WHERE `codigo` = ?";
52   const values = [req.body.nome, req.body.email, req.body.senha];
53
54   db.query(q, [...values, req.params.codigo], (err) => {
55     if (err) {
56       return res.status(500).json({ message: "Erro ao atualizar usuário", error: err });
57     }
58     return res.status(200).json("Registro atualizado com sucesso");
59   });
60 };
61
62 // Função para deletar um usuário
63 export const deleteusuarios = (req, res) => {
64   const { codigo } = req.params;
65   console.log("Deletando usuário com código:", codigo); // Log para verificar o valor do código
66
67   // SQL para deletar o usuário com base no código fornecido
68   const q = "DELETE FROM usuarios WHERE `codigo` = ?";
69
70   // Realiza a consulta de exclusão no banco de dados
71   db.query(q, [codigo], (err, result) => {
72     if (err) {
73       console.error("Erro ao deletar no banco de dados:", err); // Log do erro
74       return res.status(500).json({ message: "Erro ao deletar usuário", error: err });
75     }
76
77     // Se nenhum usuário foi afetado pela deleção
78     if (result.affectedRows === 0) {
79       return res.status(404).json({ message: "Usuário não encontrado" });
80     }
81
82     // Se a deleção for bem-sucedida
83     return res.status(200).json({ message: "Usuário deletado com sucesso!" });
84   });
85 };
86

```

Routes/usuarios.js

```

1  import express from "express";
2  import {
3    getusuarios,
4    addusuarios,
5    updateusuarios,
6    deleteusuarios,
7    login // Importe a função de login
8  } from "../controllers/usuarios.js";
9
10 const router = express.Router();
11
12 router.get("/", getusuarios);
13 router.post("/", addusuarios); // Rota para adicionar um novo usuário
14 router.put("/:codigo", updateusuarios);
15 router.delete('/usuarios/:codigo', deleteusuarios);
16 router.post("/login", login);
17
18 export default router;
19

```

db.js

```
1  import mysql from "mysql";
2  export const db = mysql.createConnection({
3    host: "localhost",
4    user: "root",
5    password: "",
6    database: "bd_usuarios",
7  });
```

index.js

```
1  import express from "express";
2  import usuariosRoutes from "../routes/usuarios.js";
3  import cors from "cors";
4  //Para utilizar o modo json para alterações como
5  // Post e Put
6  const app = express();
7  app.use(express.json());
8  app.use(cors());
9  app.use("/", usuariosRoutes);
10 //Escutar a porta 8800
11 app.listen(8800);
12
```

Package.json

```
1  {
2    "name": "backend_node",
3    "version": "1.0.0",
4    "main": "index.js",
5    "type": "module",
6    "scripts": {
7      "start": "nodemon index.js"
8    },
9    "keywords": [],
10   "author": "",
11   "license": "ISC",
12   "dependencies": {
13     "cors": "^2.8.5",
14     "express": "^4.21.0",
15     "mysql": "^2.18.1",
16     "nodemon": "^3.1.7"
17   },
18   "description": ""
19 }
20
```

Index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1" />
6     <link rel="stylesheet" href="/frontend_react/public/css/style.css">
7     <title>CADASTRO</title>
8   </head>
9   <body>
10     <noscript>You need to enable JavaScript to run this app.</noscript>
11     <div id="root"></div>
12   </body>
13 </html>
```

principal.html

```

1 <!DOCTYPE html
2 <html lang="pt-BR">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Lista de Cursos</title>
7 <style>
8     body {
9         font-family: Arial, sans-serif;
10        background-color: #f1e1e1;
11        color: #fff;
12        margin: 0;
13        padding: 0;
14    }
15
16    .container {
17        width: 100%;
18        max-width: 1200px;
19        margin: 20px auto;
20        padding: 20px;
21    }
22
23    .principal {
24        color: #ffe8c1;
25        font-size: 56px;
26    }
27
28    .title {
29        font-size: 30px;
30        color: #fff;
31        margin-bottom: 30px;
32    }
33
34    .back-button {
35        background-color: #e6007e;
36        color: #fff;
37        border: none;
38        padding: 10px 20px;
39        font-size: 16px;
40        border-radius: 5px;
41        cursor: pointer;
42        transition: background-color 0.3s;
43    }
44
45    .course-list {
46        display: flex;
47        flex-direction: column;
48        gap: 10px;
49    }
50
51    .course-item {
52        display: flex;
53        flex-direction: column;
54        background-color: #333;
55        padding: 15px;
56        border-radius: 5px;
57        box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2);
58    }

```



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

    .course-item span {
        font-size: 18px;
        flex: 1;
    }

    .pdf-button, .info-button {
        background-color: #fe690e;
        color: #ffff;
        border: none;
        padding: 10px 20px;
        font-size: 16px;
        border-radius: 5px;
        cursor: pointer;
        margin-left: 10px;
        transition: background-color 0.3s;
    }

    .pdf-button:hover, .info-button:hover {
        background-color: #ffe80c;
    }

    .pdf-button:focus, .info-button:focus {
        outline: none;
    }

    .course-description {
        font-size: 14px;
        margin-top: 10px;
        color: #888;
    }
}

</style>
</head>
<body>
    <div class="container">
        <div class="header">
            <h1 class="principal">NANO COURSES</h1>
            <button class="back-button" onclick="window.location.href='login.html'">Voltar ao Login</button>
        </div>
        <hr>
        <h2 class="title">SEUS CURSOS</h2>
        <div class="course-list">
            <!-- Cursos -->
            <div class="course-item">
                <div class="course-header">
                    <span>Algoritmos: Aprenda a programar</span>
                    <div>
                        <button class="pdf-button" onclick="downloadPDF('PDFS/1.pdf')">Baixar PDF</button>
                        <button class="info-button" onclick="showDescription(this, 'Este curso introduz os conceitos básicos de algoritmos e lógica de programação. Você aprenderá a resolver problemas computacionais utilizando técnicas fundamentais e boas práticas de codificação.')">
                            Saber Mais
                        </button>
                    </div>
                </div>
                <p class="course-description" style="display: none;"></p>
            </div>
            <div class="course-item">
                <div class="course-header">
                    <span>Big Data & Analytics</span>
                    <div>
                        <button class="pdf-button" onclick="downloadPDF('PDFS/2.pdf')">Baixar PDF</button>
                        <button class="info-button" onclick="showDescription(this, 'Entenda os conceitos de Big Data e Analytics e como eles são usados para processar e analisar grandes volumes de dados, ajudando a tomar decisões baseadas em informações.')">
                            Saber Mais
                        </button>
                    </div>
                </div>
                <p class="course-description" style="display: none;"></p>
            </div>
            <div class="course-item">
                <div class="course-header">
                    <span>Blockchain</span>
                    <div>
                        <button class="pdf-button" onclick="downloadPDF('PDFS/3.pdf')">Baixar PDF</button>
                        <button class="info-button" onclick="showDescription(this, 'Descubra os fundamentos da tecnologia blockchain, incluindo seus principais componentes, funcionamento e como ela está revolucionando diferentes setores.')">
                            Saber Mais
                        </button>
                    </div>
                </div>
                <p class="course-description" style="display: none;"></p>
            </div>
            <div class="course-item">
                <div class="course-header">
                    <span>Blockchain Advanced</span>
                    <div>
                        <button class="pdf-button" onclick="downloadPDF('PDFS/4.pdf')">Baixar PDF</button>
                        <button class="info-button" onclick="showDescription(this, 'Avance nos conhecimentos de blockchain e explore conceitos avançados como contratos inteligentes, redes privadas e aplicações práticas.')">
                            Saber Mais
                        </button>
                    </div>
                </div>
                <p class="course-description" style="display: none;"></p>
            </div>
            <!-- Adicione os outros cursos seguindo o mesmo padrão -->
            <div class="course-item">
                <div class="course-header">
                    <span>Branding Analytics</span>
                    <div>
                        <button class="pdf-button" onclick="downloadPDF('PDFS/5.pdf')">Baixar PDF</button>
                        <button class="info-button" onclick="showDescription(this, 'Aprenda como usar dados analíticos para fortalecer marcas e criar campanhas de marketing mais eficazes.')">
                            Saber Mais
                        </button>
                    </div>
                </div>
                <p class="course-description" style="display: none;"></p>
            </div>
        </div>
    </div>
```

```
15         </button>
16     </div>
17 </div>
18 <hr>
19 <p class="course-description" style="display: none;"></p>
20 </div>
21 <div class="course-item">
22     <div class="course-header">
23         <span>Big Data & Analytics</span>
24         <div>
25             <button class="pdf-button" onclick="downloadPDF('PDFS/2.pdf')">Baixar PDF</button>
26             <button class="info-button" onclick="showDescription(this, 'Entenda os conceitos de Big Data e Analytics e como eles são usados para processar e analisar grandes volumes de dados, ajudando a tomar decisões baseadas em informações.')">
27                 Saber Mais
28             </button>
29         </div>
30     </div>
31     <p class="course-description" style="display: none;"></p>
32 </div>
33 <div class="course-item">
34     <div class="course-header">
35         <span>Blockchain</span>
36         <div>
37             <button class="pdf-button" onclick="downloadPDF('PDFS/3.pdf')">Baixar PDF</button>
38             <button class="info-button" onclick="showDescription(this, 'Descubra os fundamentos da tecnologia blockchain, incluindo seus principais componentes, funcionamento e como ela está revolucionando diferentes setores.')">
39                 Saber Mais
40             </button>
41         </div>
42     </div>
43     <p class="course-description" style="display: none;"></p>
44 </div>
45 <div class="course-item">
46     <div class="course-header">
47         <span>Blockchain Advanced</span>
48         <div>
49             <button class="pdf-button" onclick="downloadPDF('PDFS/4.pdf')">Baixar PDF</button>
50             <button class="info-button" onclick="showDescription(this, 'Avance nos conhecimentos de blockchain e explore conceitos avançados como contratos inteligentes, redes privadas e aplicações práticas.')">
51                 Saber Mais
52             </button>
53         </div>
54     </div>
55     <p class="course-description" style="display: none;"></p>
56 </div>
57 <!-- Adicione os outros cursos seguindo o mesmo padrão -->
58 <div class="course-item">
59     <div class="course-header">
60         <span>Branding Analytics</span>
61         <div>
62             <button class="pdf-button" onclick="downloadPDF('PDFS/5.pdf')">Baixar PDF</button>
63             <button class="info-button" onclick="showDescription(this, 'Aprenda como usar dados analíticos para fortalecer marcas e criar campanhas de marketing mais eficazes.')">
64                 Saber Mais
65             </button>
66         </div>
67     </div>
68     <p class="course-description" style="display: none;"></p>
69 </div>
```

```

107 <div class="course-item">
170 <div class="course-header">
171 <span>Business Management & IT Services</span>
172 <div>
173 <button class="pdf-button" onclick="downloadPDF('PDFS/6.pdf')">Baixar PDF</button>
174 <button class="info-button" onclick="showDescription(this, 'Explore estratégias de gestão de negócios e a integração de serviços de TI para alcançar melhores resultados.')">
175 Saber Mais
176 </button>
177 </div>
178 </div>
179 <p class="course-description" style="display: none;"></p>
180 </div>
181
182 <div class="course-item">
183 <div class="course-header">
184 <span>Customer Experience Management</span>
185 <div>
186 <button class="pdf-button" onclick="downloadPDF('PDFS/7.pdf')">Baixar PDF</button>
187 <button class="info-button" onclick="showDescription(this, 'Descubra como melhorar a experiência do cliente em todos os pontos de contato, criando fidelidade e satisfação.')">
188 Saber Mais
189 </button>
190 </div>
191 </div>
192 <p class="course-description" style="display: none;"></p>
193 </div>
194
195 <div class="course-item">
196 <div class="course-header">
197 <span>Cybersecurity</span>
198 <div>
199 <button class="pdf-button" onclick="downloadPDF('PDFS/8.pdf')">Baixar PDF</button>
200 <button class="info-button" onclick="showDescription(this, 'Entenda os fundamentos da segurança cibernética e como proteger sistemas e dados contra ameaças digitais.')">
201 Saber Mais
202 </button>
203 </div>
204 </div>
205 <p class="course-description" style="display: none;"></p>
206 </div>
207
208 <div class="course-item">
209 <div class="course-header">
210 <span>Data Driven Decision</span>
211 <div>
212 <button class="pdf-button" onclick="downloadPDF('PDFS/9.pdf')">Baixar PDF</button>
213 <button class="info-button" onclick="showDescription(this, 'Aprenda a tomar decisões estratégicas com base em análise de dados e insights valiosos.')">
214 Saber Mais
215 </button>
216 </div>
217 </div>
218 <p class="course-description" style="display: none;"></p>
219 </div>
220
221 <div class="course-item">
222 <div class="course-header">
223

```

```

224 <span>Design Thinking</span>
225 <div>
226 <button class="pdf-button" onclick="downloadPDF('PDFS/10.pdf')">Baixar PDF</button>
227 <button class="info-button" onclick="showDescription(this, 'Descubra como o Design Thinking pode ajudar a resolver problemas complexos de forma criativa e colaborativa.')">
228 Saber Mais
229 </button>
230 </div>
231 </div>
232 <p class="course-description" style="display: none;"></p>
233 </div>
234
235 <div class="course-item">
236 <div class="course-header">
237 <span>DevOps & Agile Culture</span>
238 <div>
239 <button class="pdf-button" onclick="downloadPDF('PDFS/11.pdf')">Baixar PDF</button>
240 <button class="info-button" onclick="showDescription(this, 'Entenda como o DevOps e a cultura ágil podem melhorar a entrega de software e a colaboração entre equipes.')">
241 Saber Mais
242 </button>
243 </div>
244 </div>
245 <p class="course-description" style="display: none;"></p>
246 </div>
247
248 <div class="course-item">
249 <div class="course-header">
250 <span>Dominando IoT</span>
251 <div>
252 <button class="pdf-button" onclick="downloadPDF('PDFS/12.pdf')">Baixar PDF</button>
253 <button class="info-button" onclick="showDescription(this, 'Aprenda como a Internet das Coisas (IoT) está conectando dispositivos e criando novas oportunidades de inovação.')">
254 Saber Mais
255 </button>
256 </div>
257 </div>
258 <p class="course-description" style="display: none;"></p>
259 </div>
260
261 <div class="course-item">
262 <div class="course-header">
263 <span>Future-Proof Organizations</span>
264 <div>
265 <button class="pdf-button" onclick="downloadPDF('PDFS/13.pdf')">Baixar PDF</button>
266 <button class="info-button" onclick="showDescription(this, 'Explore estratégias para tornar organizações resilientes e prontas para o futuro.')">
267 Saber Mais
268 </button>
269 </div>
270 </div>
271 <p class="course-description" style="display: none;"></p>
272 </div>
273
274 <div class="course-item">
275 <div class="course-header">
276 <span>Game Design</span>

```

```
277         <div>
278             <button class="pdf-button" onclick="downloadPDF('PDFS/14.pdf')">Baixar PDF</button>
279             <button class="info-button" onclick="showDescription(this, 'Descubra os fundamentos do design de jogos, incluindo mecânicas, narrativa e experiência do jogador.')">
280                 Saber Mais
281             </button>
282         </div>
283     </div>
284     <p class="course-description" style="display: none;"></p>
285 </div>
286
287 <div class="course-item">
288     <div class="course-header">
289         <span>Governança de TI</span>
290         <div>
291             <button class="pdf-button" onclick="downloadPDF('PDFS/15.pdf')">Baixar PDF</button>
292             <button class="info-button" onclick="showDescription(this, 'Aprenda a alinhar TI às estratégias empresariais para alcançar maior eficiência e controle.')">
293                 Saber Mais
294             </button>
295         </div>
296     </div>
297     <p class="course-description" style="display: none;"></p>
298 </div>
299 <div class="course-item">
300     <div class="course-header">
301         <span>Inovação Disruptiva</span>
302         <div>
303             <button class="pdf-button" onclick="downloadPDF('PDFS/16.pdf')">Baixar PDF</button>
304             <button class="info-button" onclick="showDescription(this, 'Entenda como a inovação disruptiva pode transformar mercados e criar novas oportunidades de negócios.')">
305                 Saber Mais
306             </button>
307         </div>
308     </div>
309     <p class="course-description" style="display: none;"></p>
310 </div>
311
312 <div class="course-item">
313     <div class="course-header">
314         <span>Inteligência Artificial e Computacional</span>
315         <div>
316             <button class="pdf-button" onclick="downloadPDF('PDFS/17.pdf')">Baixar PDF</button>
317             <button class="info-button" onclick="showDescription(this, 'Descubra os fundamentos da inteligência artificial e como ela está sendo aplicada em diferentes indústrias.')">
318                 Saber Mais
319             </button>
320         </div>
321     </div>
322     <p class="course-description" style="display: none;"></p>
323 </div>
324
325 <div class="course-item">
326     <div class="course-header">
327         <span>Leadership and Corporate Accountability</span>
328         <div>
329             <button class="pdf-button" onclick="downloadPDF('PDFS/18.pdf')">Baixar PDF</button>
```

```
330             <button class="info-button" onclick="showDescription(this, 'Aprenda como liderar com responsabilidade e ética no ambiente corporativo, garantindo decisões justas e sustentáveis.')">
331                 Saber Mais
332             </button>
333         </div>
334     </div>
335     <p class="course-description" style="display: none;"></p>
336 </div>
337
338 <div class="course-item">
339     <div class="course-header">
340         <span>Programação em Banco de Dados</span>
341         <div>
342             <button class="pdf-button" onclick="downloadPDF('PDFS/19.pdf')">Baixar PDF</button>
343             <button class="info-button" onclick="showDescription(this, 'Explore os conceitos de programação em banco de dados, incluindo consultas SQL, modelagem e gestão eficiente de dados.')">
344                 Saber Mais
345             </button>
346         </div>
347     </div>
348     <p class="course-description" style="display: none;"></p>
349 </div>
350
351 <div class="course-item">
352     <div class="course-header">
353         <span>Social Media Planning</span>
354         <div>
355             <button class="pdf-button" onclick="downloadPDF('PDFS/20.pdf')">Baixar PDF</button>
356             <button class="info-button" onclick="showDescription(this, 'Aprenda a planejar e executar campanhas em redes sociais para maximizar engajamento e alcance.')">
357                 Saber Mais
358             </button>
359         </div>
360     </div>
361     <p class="course-description" style="display: none;"></p>
362 </div>
363
364 <div class="course-item">
365     <div class="course-header">
366         <span>User Experience</span>
367         <div>
368             <button class="pdf-button" onclick="downloadPDF('PDFS/21.pdf')">Baixar PDF</button>
369             <button class="info-button" onclick="showDescription(this, 'Descubra como criar experiências excepcionais para os usuários, alinhando design, usabilidade e satisfação.')">
370                 Saber Mais
371             </button>
372         </div>
373     </div>
374     <p class="course-description" style="display: none;"></p>
375 </div>
376
377 </div>
378 </div>
```

```

379
380     <script>
381         function downloadPDF(pdfURL) {
382             const a = document.createElement('a');
383             a.href = pdfURL;
384             a.download = pdfURL.split('/').pop();
385             document.body.appendChild(a);
386             a.click();
387             document.body.removeChild(a);
388         }
389
390         function showDescription(button, description) {
391             const descriptionElement = button.parentElement.parentElement.nextElementSibling;
392             if (descriptionElement.style.display === 'none') {
393                 descriptionElement.style.display = 'block';
394                 descriptionElement.textContent = description;
395             } else {
396                 descriptionElement.style.display = 'none';
397             }
398         }
399     </script>
400 </body>
401 </html>
402

```

Form.js

```

1  import React, { useEffect, useRef } from "react";
2  import styled from "styled-components";
3  import axios from "axios";
4  import { toast } from "react-toastify";
5
6  const FormContainer = styled.form`
7      display: flex;
8      align-items: flex-end;
9      gap: 10px;
10     flex-wrap: wrap;
11     background-color: #fff;
12     padding: 20px;
13     box-shadow: 0px 0px 5px #ccc;
14     border-radius: 5px;
15 `;
16
17 const InputArea = styled.div`
18     display: flex;
19     flex-direction: column;
20 `;
21
22 const Input = styled.input`
23     width: 120px;
24     padding: 0 10px;
25     border: 1px solid #bbb;
26     border-radius: 5px;
27     height: 40px;
28 `;
29
30 const Button = styled.button`
31     padding: 10px;
32     cursor: pointer;
33     border-radius: 5px;
34     border: none;
35     background-color: #ff1e8c;
36     color: white;
37     height: 42px;
38 `;
39
40 const SwitchButton = styled.button`
41     margin-top: 10px;
42     padding: 10px;
43     cursor: pointer;
44     border-radius: 5px;
45     background-color: #d2d2d2;
46     color: black;
47     border: none;
48 `;
49
50 const Form = ({ getusuarios, onEdit, setOnEdit, switchToLogin }) => {

```



```

50 const Form = ({ getusuarios, onEdit, setOnEdit, switchToLogin }) => {
51
52   useEffect(() => {
53     // Carregar os dados sempre que o formulário for visível ou quando onEdit mudar
54     if (onEdit) {
55       const usuarios = ref.current;
56       usuarios.nome.value = onEdit.nome;
57       usuarios.email.value = onEdit.email;
58       usuarios.senha.value = onEdit.senha;
59     } else {
60       getusuarios(); // Puxar os dados sempre que o form estiver visível
61     }
62   }, [onEdit, getusuarios]); // Dependência para garantir a atualização sempre que onEdit ou getusuarios mudar
63
64   const handleSubmit = async (e) => {
65     e.preventDefault();
66
67     const usuarios = ref.current;
68
69     if (!usuarios.nome.value || !usuarios.email.value || !usuarios.senha.value) {
70       return toast.warn("Preencha todos os campos");
71     }
72
73     if (onEdit) {
74       await axios
75         .put("http://localhost:8800/" + onEdit.codigo, {
76           nome: usuarios.nome.value,
77           email: usuarios.email.value,
78           senha: usuarios.senha.value,
79         })
80         .then(({ data }) => toast.success(data))
81         .catch(({ data }) => toast.error(data));
82     } else {
83       await axios
84         .post("http://localhost:8800", {
85           nome: usuarios.nome.value,
86           email: usuarios.email.value,
87           senha: usuarios.senha.value,
88         })
89         .then(({ data }) => toast.success(data))
90         .catch(({ data }) => toast.error(data));
91     }
92     usuarios.nome.value = "";
93     usuarios.email.value = "";
94     usuarios.senha.value = "";
95
96     setOnEdit(null);
97     getusuarios();
98   });
99
100

```

```

100
101   return (
102     <FormContainer ref={ref} onSubmit={handleSubmit}>
103       <InputArea>
104         <label>Nome</label>
105         <Input type="text" name="nome" />
106       </InputArea>
107       <InputArea>
108         <label>E-mail</label>
109         <Input type="email" name="email" />
110       </InputArea>
111       <InputArea>
112         <label>Senha</label>
113         <Input type="password" name="senha" />
114       </InputArea>
115       <Button type="submit">{onEdit ? "Editar" : "Cadastrar"}</Button>
116       <SwitchButton onClick={switchToLogin}>Voltar ao Login</SwitchButton>
117     </FormContainer>
118   );
119 };
120
121 export default Form;
122

```

Grid.js

```
1 import React from "react";
2 import axios from "axios";
3 import styled from "styled-components";
4 import { FaTrash, FaEdit } from "react-icons/fa";
5 import { toast } from "react-toastify";
6
7 const Table = styled.table`
8   width: 100%;
9   background-color: #fff;
10  padding: 20px;
11  box-shadow: 0px 0px 5px #ccc;
12  border-radius: 5px;
13  max-width: 1000px;
14  margin: 20px auto;
15  word-break: break-all;
16 `;
17
18 export const Thead = styled.thead``;
19 export const Tbody = styled.tbody``;
20 export const Tr = styled.tr``;
21 export const Th = styled.th`
22   text-align: start;
23   border-bottom: inset;
24   padding-bottom: 5px;
25 `;
26 export const Td = styled.td`
27   padding-top: 15px;
28   text-align: ${(props) => (props.alignCenter ? "center" : "start")};
29   width: ${(props) => (props.width ? props.width : "auto")};
30 `;
31
32 const Grid = ({ usuarios, setusuarios, setOnEdit }) => {
33   // Função para deletar um usuário
34   const handleDelete = async (codigo) => {
35     try {
36       console.log("Iniciando a deleção do usuário com código:", codigo);
37
38       // Enviar requisição DELETE para a API
39       const response = await axios.delete(`http://localhost:8800/usuarios/${codigo}`);
40       console.log("Resposta do servidor:", response);
41
42       // Verifica se a resposta foi bem-sucedida
43       if (response.status === 200) {
44         // Atualiza a lista de usuários após deletar
45         const newUsuarios = usuarios.filter((usuariosItem) => usuariosItem.codigo !== codigo);
46         setusuarios(newUsuarios);
47
48         // Exibe mensagem de sucesso
49         toast.success("Usuário deletado com sucesso!");
50       }
51     } catch (error) {
52       console.error("Erro ao deletar usuário:", error);
53       toast.error(
54         error.response
55         ? `Erro ao deletar usuário: ${error.response.data.message || error.response.data}`
56         : `Erro: ${error.message}`
57       );
58     }
59   };
60
61   return (
62     <Table>
63       <Thead>
64         <Tr>
65           <Th>Nome</Th>
66           <Th>Email</Th>
67           <Th>Senha</Th>
68           <Th>Ações</Th>
69         </Tr>
70       </Thead>
71       <Tbody>
72         {usuarios.map((usuariosItem) => (
73           <Tr key={usuariosItem.codigo}>
74             <Td>{usuariosItem.nome}</Td>
75             <Td>{usuariosItem.email}</Td>
76             <Td>{usuariosItem.senha}</Td>
77             <Td alignCenter>
78               <FaEdit onClick={() => setOnEdit(usuariosItem)} />
79               <FaTrash
80                 onClick={() => handleDelete(usuariosItem.codigo)}
81                 style={{ cursor: "pointer", color: "red", marginLeft: "10px" }} />
82             </Td>
83           </Tr>
84         ))}
85       </Tbody>
86     </Table>
87   );
88 };
89
90 export default Grid;
```

LoginForm.js

```
1 import React, { useState } from "react";
2 import styled from "styled-components";
3 import axios from "axios";
4 import { toast } from "react-toastify";
5
6 const FormContainer = styled.div`
7   display: flex;
8   flex-direction: column;
9   gap: 10px;
10  padding: 20px;
11  background-color: #fff;
12  border-radius: 5px;
13  box-shadow: 0px 0px 5px #ccc;
14  width: 300px; /* Definindo uma largura fixa para centralizar os botões */
15  margin: 0 auto; /* Centralizando o formulário */
16 `;
17
18 const InputArea = styled.div`
19   display: flex;
20   flex-direction: column;
21 `;
22
23 const Input = styled.input`
24   width: 100%; /* Ajustando para 100% para que ocupe toda a largura do formulário */
25   padding: 0 10px;
26   border: 1px solid #bbb;
27   border-radius: 5px;
28   height: 40px;
29 `;
30
31 const Button = styled.button`
32   padding: 10px;
33   cursor: pointer;
34   border-radius: 5px;
35   background-color: #ff1e8c;
36   color: white;
37   height: 42px;
38   border: none;
39   width: 100%; /* Fazendo o botão de login ocupar toda a largura do formulário */
40   margin-top: 10px; /* Adicionando um espaço acima do botão */
41 `;
42
43 const SwitchButton = styled.button`
44   margin-top: 10px;
45   padding: 10px;
46   cursor: pointer;
47   border-radius: 5px;
48   background-color: #d2d2d2;
49   color: black;
50   border: none;
```

```
50   border: none;
51   width: 100%; /* Fazendo o botão de troca de tela ocupar toda a largura do formulário */
52 `;
53
54 const LoginForm = ({ setUser, switchToRegister }) => {
55   const [email, setEmail] = useState("");
56   const [senha, setSenha] = useState("");
57
58   // Função para pegar todos os usuários
59   const getUsuarios = async () => {
60     try {
61       const res = await axios.get("http://localhost:8080/usuarios");
62       console.log(res.data); // Aqui você pode fazer algo com os dados dos usuários
63       // Caso queira fazer algo com os dados recebidos, você pode armazená-los em um estado, por exemplo.
64     } catch (err) {
65       toast.error("Erro ao buscar usuários");
66     }
67   };
68
69   const handleLogin = async (e) => {
70     e.preventDefault();
71     try {
72       const res = await axios.post("http://localhost:8080/login", { email, senha });
73       toast.success(res.data.message);
74       setUser(res.data.user);
75
76       // Redireciona para o principal.html após o login bem-sucedido
77       window.location.href = "/principal.html";
78     } catch (err) {
79       toast.error(err.response?.data.message || "Erro ao fazer login");
80     }
81   };
82
83   return (
84     <FormContainer>
85       <form onSubmit={handleLogin}>
86         <InputArea>
87           <label>E-mail</label>
88           <Input
89             type="email"
90             value={email}
91             onChange={e => setEmail(e.target.value)}
92             required
93           />
94         </InputArea>
95         <InputArea>
96           <label>Senha</label>
97           <Input
```

```

96     <label>Senha</label>
97     <Input
98     |   type="password"
99     |   value={senha}
100    |   onChange={(e) => setSenha(e.target.value)}
101    |   required
102    |   />
103    </InputArea>
104    <Button type="submit">Login</Button>
105  </form>
106  <SwitchButton
107  |   onClick={() => {
108  |     switchToRegister(); // Chama a função de troca para a tela de cadastro
109  |   }}
110  |   >
111  |     Ir para Cadastro
112  |   </SwitchButton>
113 </FormContainer>
114 );
115 };
116
117 export default LoginForm;
118

```

Global.js

```

1  // Importar a biblioteca styled-components associando a createGlobalStyle
2  import { createGlobalStyle } from "styled-components";
3
4  // Criar uma constante com as definições de createGlobalStyle
5  const Global = createGlobalStyle`
6    * {
7      margin: 0;
8      padding: 0;
9      box-sizing: border-box; /* Para garantir que o padding e margin não aumentem o tamanho do elemento */
10     font-family: sans-serif;
11   }
12
13   body {
14     width: 100vw; // Para ocupar todo o espaço da tela
15     height: 100vh;
16     display: flex;
17     justify-content: center;
18     align-items: center; /* Centraliza verticalmente */
19     background-color: #1e1e1e; /* Cor de fundo da página */
20   }
21 `;
22
23 // O estilo definido no arquivo global.js deve ser exportada a constante
24 // global para que o resto do código-fonte tenha acesso
25 export default Global;
26

```


App.js

```
1 import React, { useState, useEffect } from "react";
2 import GlobalStyle from "../style/global";
3 import { toast, ToastContainer } from "react-toastify";
4 import "react-toastify/dist/ReactToastify.css";
5 import styled from "styled-components";
6 import Form from "../components/Form";
7 import Grid from "../components/Grid";
8 import LoginForm from "../components/LoginForm";
9 import axios from "axios";
10
11 const Container = styled.div`
12   width: 100%;
13   max-width: 800px;
14   margin-top: 20px;
15   display: flex;
16   flex-direction: column;
17   align-items: center;
18   gap: 10px;
19   background-color: #fff1e8c;
20   padding: 20px;
21   border-radius: 8px;
22   box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
23 `;
24
25 const Title = styled.h2`
26   color: #ffffff;
27   margin-bottom: 10px;
28   font-size: 1.5rem;
29   text-align: center;
30 `;
31
32 function App() {
33   const [usuarios, setusuarios] = useState([]);
34   const [onEdit, setOnEdit] = useState(null);
35   const [user, setUser] = useState(null); // Estado para o usuário logado
36   const [isLogin, setIsLogin] = useState(true); // Estado para controlar a tela de Login ou Cadastro
37
38   const getusuarios = async () => {
39     try {
40       const res = await axios.get("http://localhost:8800");
41       setusuarios(res.data.sort((a, b) => (a.nome > b.nome ? 1 : -1)));
42     } catch (error) {
43       toast.error(error);
44     }
45   };
46
47   useEffect(() => {
48     if (user) {
49       getusuarios();
50     }
51   }, [user]);
52
53   const switchToLogin = () => setIsLogin(true); // Alterna para o login
54   const switchToRegister = () => setIsLogin(false); // Alterna para o cadastro
55
56   return (
57     <>
58       <Container>
59         <Title>{isLogin ? "Login" : "Cadastro de Usuários"}</Title>
60         {isLogin ? (
61           <LoginForm
62             setUser={setUser}
63             switchToRegister={switchToRegister} // Passa a função para alternar para o cadastro
64           />
65         ) : (
66           <>
67             <Form
68               onEdit={onEdit}
69               setOnEdit={setOnEdit}
70               getusuarios={getusuarios}
71               switchToLogin={switchToLogin} // Passa a função para alternar para o login
72             />
73             <Grid usuarios={usuarios} setusuarios={setusuarios} setOnEdit={setOnEdit} />
74           </>
75         )}
76       </Container>
77       <ToastContainer autoClose={3000} />
78       <GlobalStyle />
79     </>
80   );
81 }
82
83 export default App;
84
```

Index.js

```
1  import React from "react";
2  import ReactDOM from "react-dom/client";
3  import App from "./App";
4  const root = ReactDOM.createRoot(document.getElementById("root"));
5  root.render(
6    <React.StrictMode>
7      <App />
8    </React.StrictMode>
9  );
10
```