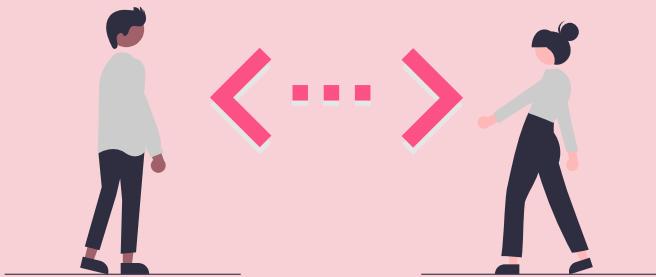


Programa Trilhas Inova 2B

Documentação - Back-End de um formulário

Amanda Mikely Abreu Macêdo
Gabriela Torres de Queiroz
10 de abril de 2025
Maranhão - Brasil



Sumário

1. Introdução.....	2
1.1. Visão Geral.....	2
1.2. Hipótese.....	2
1.3. Objetivos.....	2
1.3.1. Objetivo Geral.....	2
1.3.2. Objetivos Específicos.....	2
2. Desenvolvimento.....	4
2.1. Tecnologias utilizadas.....	5
2.2. Etapas do Processo.....	6
2.3. Sistema.....	6
2.4. Cadastro do Usuário.....	7
2.5. Resultados das Páginas Web: Formulário.....	9
3. Considerações Finais.....	10
4. Apêndice.....	11
4.1. Anexo dos Códigos em HTML e CSS.....	11
4.2. Anexo das Páginas Web: Formulário.....	33

1. Introdução

O projeto consiste em um formulário de inscrição desenvolvido para o programa "Trilhas Inova 2B". O objetivo principal é fornecer uma interface intuitiva e acessível para que os usuários possam se inscrever nas suas respectivas trilhas de aprendizagem, fornecendo informações pessoais, endereço e selecionando sua área de interesse. O formulário foi projetado para ser visualmente atraente, funcional e adaptável.

1.1. Visão Geral

O formulário é dividido em seções claras e organizadas, cada uma responsável por coletar um tipo específico de informação: **Informações Pessoais, Endereço Residencial, Trilhas de Aprendizagem e os Termos e Condições**. O design é moderno, com cores suaves e contrastantes, tipografia legível e elementos visuais que guiam o usuário ao longo do processo de preenchimento.

Este formulário tem como público-alvo:

- Jovens e adultos participantes do Programa Trilhas Inova 2B;
- Organizadores do programa que gerenciam as inscrições e avaliam os candidatos.

1.2. Hipótese

A implementação de um sistema de inscrição digital, pode otimizar o processo de registro e seleção de candidatos, reduzindo erros no preenchimento de dados, facilitando a inscrição de jovens e adultos interessados em formação na área de tecnologia e proporcionando uma experiência mais intuitiva e eficiente para os usuários.

1.3. Objetivos

1.3.1. Objetivo Geral

O projeto foi desenvolvido para otimizar e digitalizar o processo de inscrição do Programa Trilhas Inova, promovido pelo Governo do Maranhão em parceria com a SECTI e a FAPEMA. O sistema visa facilitar o registro e seleção de candidatos, reduzindo erros e tornando o processo mais acessível e eficiente.

1.3.2. Objetivos Específicos

1. Digitalizar o Processo de Inscrição:

- Substituir processos manuais ou baseados em papel por um sistema online, reduzindo a necessidade de intervenção humana e minimizando erros;
- Permitir que os candidatos se inscrevam de qualquer lugar, desde que tenham acesso à internet.

2. Reduzir Erros no Preenchimento de Dados:

- Implementar validações automáticas para garantir que os campos obrigatórios sejam preenchidos corretamente (ex.: CPF, e-mail, telefone);
- Fornecer feedback visual em tempo real para orientar o usuário durante o preenchimento do formulário.

3. Facilitar a Seleção de Trilhas de Aprendizado:

- Oferecer uma interface intuitiva para que os candidatos possam escolher a trilha desejada (Front-end, Back-end, Jogos, Design, Ciência de Dados);
- Garantir que a seleção seja clara e fácil de entender, mesmo para usuários com pouca experiência em tecnologia.

4. Simplificar o Envio de Documentação:

- Permitir o upload de documentos de identidade diretamente no sistema, eliminando a necessidade de envio físico ou por e-mail;
- Garantir que o processo de upload seja simples e acessível, com instruções claras para o usuário.

5. Melhorar a Experiência do Usuário:

- Criar uma interface amigável e responsiva, que funcione bem em diferentes dispositivos (desktop, tablet, mobile);
- Fornecer um fluxo de inscrição claro e linear, sem etapas desnecessárias ou complicadas.

6. Promover a Inclusão Digital:

- Garantir que o sistema seja acessível para usuários com diferentes níveis de conhecimento tecnológico;
- Fornecer instruções claras e suporte para candidatos que possam ter dificuldades com o processo de inscrição online.

2. Desenvolvimento

O projeto foi desenvolvido no software Visual Studio Code, utilizando a linguagem de marcação de texto Hypertext Markup Language (HTML), a linguagem de estilo Cascading Style Sheets (CSS) e a linguagem de programação *JavaScript* para a implementação das funcionalidades dinâmicas do sistema. O ambiente de elaboração foi organizado em pastas, e o desenvolvimento ocorreu de forma sequencial e estruturada. O uso do *JavaScript* possibilitou a criação de validações interativas, manipulação de elementos na interface e uma melhor experiência para o usuário. Foi utilizada a plataforma GitHub para controle de versão e armazenamento do repositório. Confira a seguir o diagrama de classes do sistema:

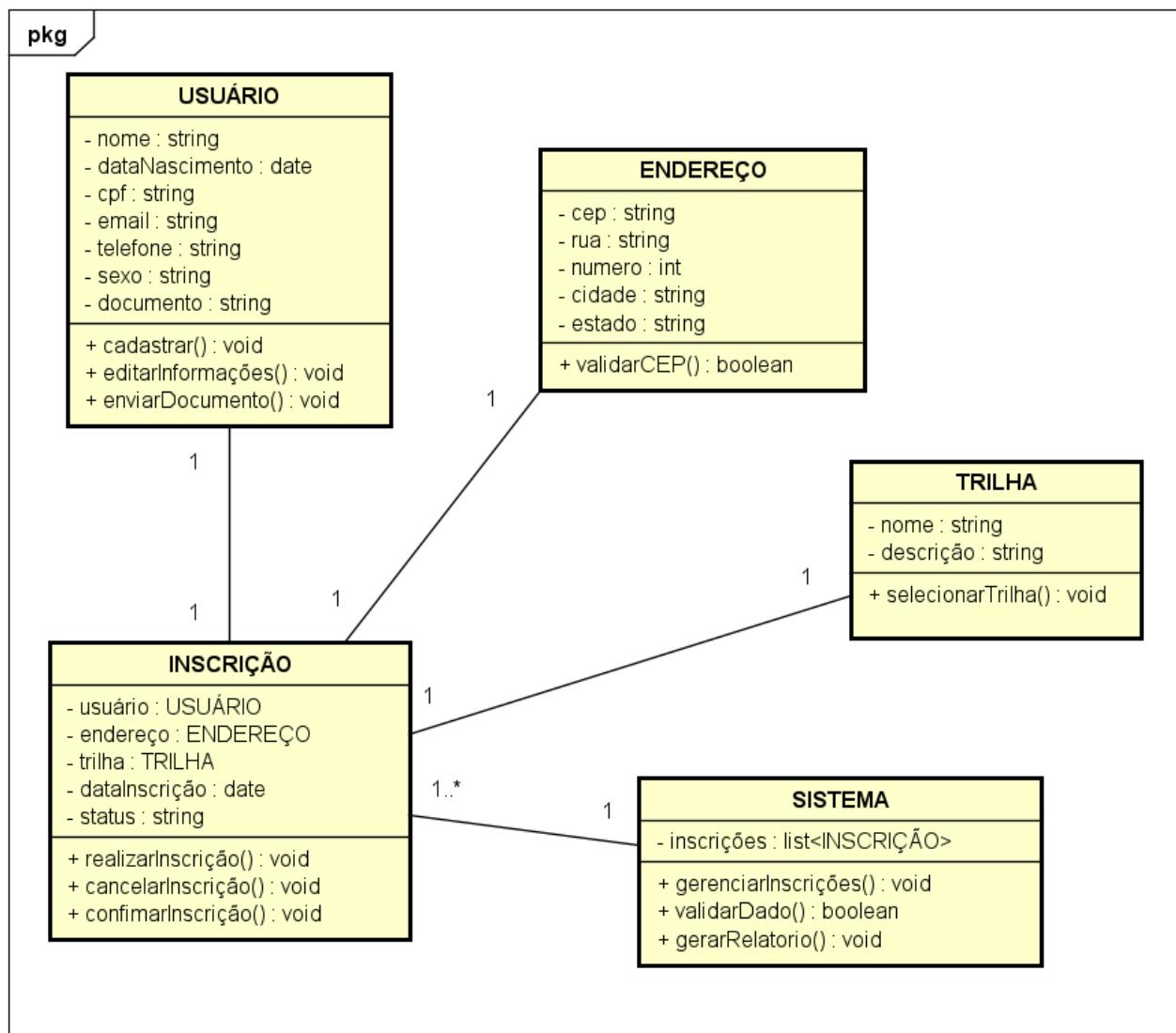


Figura 1: Diagrama de classes do sistema. Fonte: autoria própria.

O sistema é um **formulário de inscrição** desenvolvido para o programa "Trilhas Inova 2B". Ele permite que os usuários se inscrevam em uma das trilhas de aprendizagem disponíveis, fornecendo informações pessoais, endereço e concordando com os termos e condições. Confira o fluxograma do nosso sistema:

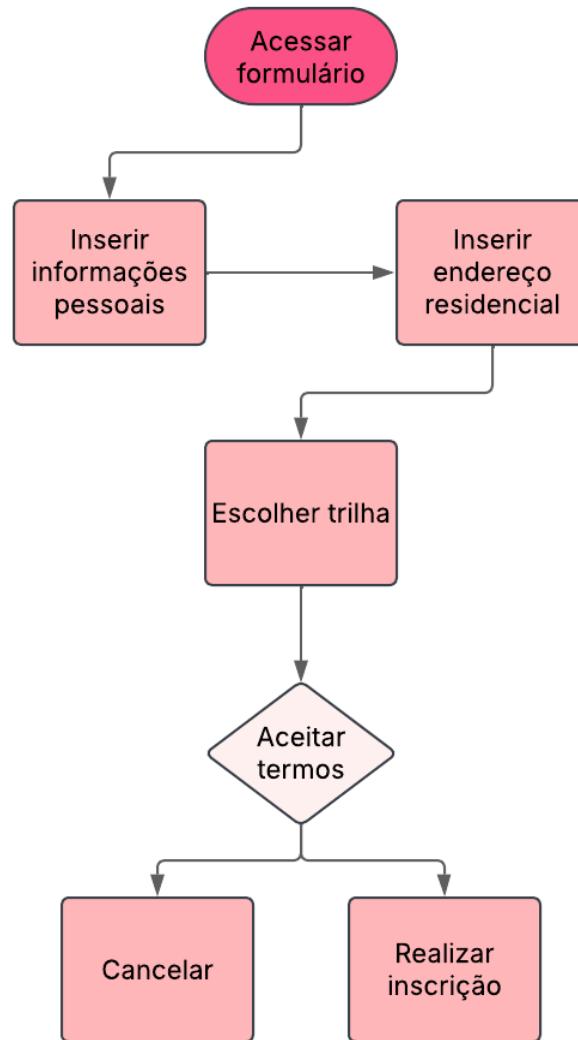


Figura 2: Fluxograma do sistema. Fonte: autoria própria.

2.1. Tecnologias utilizadas

O desenvolvimento do formulário de inscrição para o programa "Trilhas Inova 2B" utilizou uma combinação de tecnologias, sendo estas: **Visual Studio Code**, **Hypertext Markup Language (HTML)**, **Cascading Style Sheets (CSS)**, **JavaScript** e **Github**.

2.2. Etapas do Processo

1. Definição dos Requisitos:

- Identificação das necessidades do usuário e dos dados que queremos coletar;
- Definição dos campos obrigatórios e das trilhas de aprendizagem.

2. Projeto da Interface:

- Definição da paleta de cores, tipografia e elementos visuais a serem utilizados.

3. Implementação do Código:

- Desenvolvimento do HTML para estruturar o Formulário e CSS para estilização;
- JavaScript para validação de dados e armazenamento.

4. Testes e Validação:

- Testes de usabilidade, garantindo que o formulário seja intuitivo e funcional.

5. Documentação:

- Elaboração da documentação do sistema e instruções de uso.

2.3. Sistema

Este é o código do sistema, ele serve como ponto de entrada para o sistema, nele há a opção de realizar login, criar uma nova conta e fazer o login utilizando nome de usuário e senha.

- ➔ [Código HTML do Sistema \(parte 1\)](#)
- ➔ [Código HTML do Sistema \(parte 2\)](#)
- ➔ [Código HTML do Sistema \(parte 3\)](#)
- ➔ [Código HTML do Sistema \(parte 4\)](#)
- ➔ [Código HTML do Sistema \(parte 5\)](#)
- ➔ [Código HTML do Sistema \(parte 6\)](#)
- ➔ [Código HTML do Sistema \(parte 7\)](#)

Este é código CSS que define a estilização do nosso Sistema, inclui cores, fontes, layout e posicionamento dos elementos.

- ➔ [Código CSS do Sistema \(parte 1\)](#)
- ➔ [Código CSS do Sistema \(parte 2\)](#)
- ➔ [Código CSS do Sistema \(parte 3\)](#)
- ➔ [Código CSS do Sistema \(parte 4\)](#)

- [Código CSS do Sistema \(parte 5\)](#)
- [Código CSS do Sistema \(parte 6\)](#)
- [Código CSS do Sistema \(parte 7\)](#)
- [Código CSS do Sistema \(parte 8\)](#)
- [Código CSS do Sistema \(parte 9\)](#)

2.4. Cadastro do Usuário

O cadastro do usuário é implementado no arquivo index.html e a validação é feita no scrip.js, onde o usuário preenche suas informações pessoais. Este é o código do JavaScript do Sistema.

- [Código JavaScript do Sistema \(parte 1\)](#)
- [Código JavaScript do Sistema \(parte 2\)](#)
- [Código JavaScript do Sistema \(parte 3\)](#)
- [Código JavaScript do Sistema \(parte 4\)](#)
- [Código JavaScript do Sistema \(parte 5\)](#)
- [Código JavaScript do Sistema \(parte 6\)](#)
- [Código JavaScript do Sistema \(parte 7\)](#)
- [Código JavaScript do Sistema \(parte 8\)](#)
- [Código JavaScript do Sistema \(parte 9\)](#)

Funcionalidades:

- **Estrutura e Inicialização:** O código começa selecionando elementos do DOM usando `document.getElementById` e `document.querySelector`. Isso permite que o código interaja com os elementos da interface do usuário, como botões, campos de entrada e seções do formulário.
- **Validação de Dados:** O código inclui várias funções de validação para garantir que os dados inseridos pelos usuários estejam corretos. Exemplos incluem:
 1. `validateUsername`: Verifica se o nome de usuário contém apenas letras e números.
 2. `validateFullName`: Verifica se o nome completo contém apenas letras e espaços.
 3. `validateEmail`: Valida o formato do email.
 4. `validatePassword`: Verifica se a senha tem pelo menos 8 caracteres.

5. *validateCPF*: Valida se o CPF contém 11 dígitos.
6. *validateTelefone*: Valida se o telefone tem entre 10 e 11 dígitos.
7. *validateTrilha*: Verifica se uma trilha foi selecionada.

- ➔ **Manipulação de Formulários:** O código permite que os usuários naveguem entre diferentes etapas do formulário de cadastro. Isso é feito através de botões de navegação (*nextToStep*, *prevToStep*, *backToStep*) que mostram ou escondem as seções do formulário conforme necessário.
- ➔ **Coleta e Armazenamento de Dados:** Funções como *saveFormData*, *getFormData*, *loadFormData* e *setFormData* são utilizadas para coletar dados dos formulários, armazená-los no *localStorage*, e carregar dados salvos quando necessário. Isso é útil para manter os dados do usuário durante a navegação entre etapas.
- ➔ **Exibição de Erros:** O código inclui funções para mostrar e limpar mensagens de erro (*showError*, *clearError*, *clearAllErrors*). Essas funções ajudam a fornecer feedback ao usuário em tempo real sobre entradas inválidas.
- ➔ **Feedback Visual:** A função *showFeedbackModal* exibe um modal para informar ao usuário que o registro foi realizado com sucesso. Esta interação melhora a experiência do usuário.
- ➔ **Integração com a API de CEP:** O sistema permite que os usuários insiram um CEP, e ao perder o foco do campo, uma requisição é feita à API ViaCEP para preencher automaticamente os campos de endereço, como rua, bairro, cidade e estado.
- ➔ **Múltiplas Etapas de Cadastro:** O cadastro é dividido em quatro etapas, cada uma com seus próprios campos. Isso torna o processo menos intimidador e mais organizado para o usuário.
- ➔ **Login e Logout:** O código inclui funcionalidades para login e logout de usuários. Ao fazer login, o sistema verifica as credenciais e, se forem válidas, exibe uma tela de boas-vindas. O logout esconde a tela de boas-vindas e exibe novamente os formulários de cadastro e login.
- ➔ **Persistência de Dados:** Os dados dos usuários e dos formulários são armazenados no *localStorage*, permitindo que as informações sejam retidas mesmo após o fechamento do navegador, oferecendo uma experiência mais contínua ao usuário.
- ➔ **Design Responsivo:** O código inclui ajustes de altura de rolagem para que o conteúdo do formulário se ajuste corretamente ao espaço disponível, melhorando a acessibilidade e a usabilidade em diferentes dispositivos.

2.5. Resultados das Páginas Web: Formulário

→ [**Página Web do Formulário \(Parte 1\)**](#)

→ [**Página Web do Formulário \(Parte 2\)**](#)

3. Considerações Finais

O desenvolvimento do sistema de cadastro para o Programa Trilhas Inova resultou em uma interface intuitiva e um processo estruturado, visando minimizar erros e otimizar a experiência do usuário. Os desafios enfrentados, especialmente na parte visual e de validação, foram superados através de pesquisa e ajustes meticulosos no código. Com a implementação de futuras melhorias, o sistema poderá se tornar ainda mais robusto, permitindo um gerenciamento dinâmico e eficaz das inscrições.

O projeto possui uma base sólida, um design atraente e funcionalidades essenciais. O layout apresenta uma responsividade ideal, com uma distribuição equilibrada de elementos e cores suaves que criam uma identidade visual coesa e moderna. A utilização de ícones e imagens torna a interface mais intuitiva e amigável, facilitando a navegação e a compreensão do formulário. Além disso, o formulário inclui uma validação abrangente, onde os campos obrigatórios são destacados para garantir que todas as informações essenciais sejam preenchidas antes do envio. Isso ajuda a evitar que o usuário deixe de inserir dados importantes inadvertidamente..

No entanto, há espaços para melhorias que elevem a qualidade do projeto, especialmente em validação e feedback visual. Em termos de interatividade, o projeto poderia se beneficiar com a inclusão de JavaScript para adicionar máscaras para o campo de nome completo, tornar único o nome de usuário e fazer uma melhor checagem do email. Implementar essas sugestões tornará o formulário mais robusto, acessível e completo para o usuário e gerentes.

4. Apêndice

4.1. Anexo dos Códigos em HTML e CSS

```
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
7      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css">
8      <link rel="icon" href="/assets/img/icon-lista.png" type="image/png">
9      <link rel="stylesheet" href="/assets/css/style.css">
10
11     <title>Programa Trilhas Inova 2B</title>
12 </head>
13
14 <body>
15     <div class="container" id="container">
16
17         <div class="form-container sign-in">
18             <form>
19                 <h1>Entrar</h1>
20                 <h4>Faça login usando seu nome de usuário e senha</h4>
21
22                 <input type="text" placeholder="Seu nome de usuário" name="username" required>
23
24                 <div class="password-wrapper">
25                     <input type="password" placeholder="Sua senha" name="password" id="login-password" required>
26                     <span class="toggle-password" id="toggle-login-password">
27                         <i class="fa-solid fa-eye"></i>
28                     </span>
29                 </div>
30
31                 <p>Esqueceu sua senha? <a href="#">Clique aqui</a></p>
32                 <button type="submit">Entrar</button>
33
34                 <span>ou faça login usando as redes sociais</span>
35                 <div class="social-icons">
36                     <a href="#" class="icon"><i class="fa-brands fa-google-plus-g"></i></a>
37                     <a href="#" class="icon"><i class="fa-brands fa-facebook-f"></i></a>
38                     <a href="#" class="icon"><i class="fa-brands fa-github"></i></a>
39                     <a href="#" class="icon"><i class="fa-brands fa-linkedin-in"></i></a>
40                 </div>
41             </form>
42         </div>
43
44         <div class="form-container sign-up">
45             <form id="signup-step1" class="signup-form" style="display: flex;">
46                 <h1>Criar uma conta</h1>
47                 <span>Crie um nome de usuário e senha para se registrar</span>
48             </form>
49         </div>
```

Figura 3: Código HTML do formulário (parte 1).

```
index.html
2 <html lang="pt-BR">
14 <body>
15     <div class="container" id="container">
16         <div class="form-container sign-up">
17             <form id="signup-step1" class="signup-form" style="display: flex;">
18                 <h1>Criar uma conta</h1>
19                 <span>Crie um nome de usuário e senha para se registrar</span>
20
21                 <input type="text" placeholder="Nome de usuário" required>
22                 <span class="error-message"></span>
23
24                 <input type="email" placeholder="Email" required>
25                 <span class="error-message" id="email-error"></span>
26
27                 <input type="password" placeholder="Senha" required>
28                 <span class="error-message" id="password-error"></span>
29
30                 <button type="button" id="next-to-step2">Próximo</button>
31             </form>
32
33             <form id="signup-step2" class="signup-form" style="display: none;">
34                 <div class="form-scroll-content">
35                     <h1>Informações Pessoais</h1>
36                     <span>Complete suas informações pessoais</span>
37
38                     <input type="text" placeholder="Nome Completo" required>
39                     <span class="error-message"></span>
40
41                     <input type="text" placeholder="CPF" max="11" id="cpf" required>
42                     <span class="error-message"></span>
43
44                     <input type="tel" placeholder="Celular" required>
45                     <span class="error-message"></span>
46
47                     <input type="number" placeholder="Idade" min="16" max="120" required>
48                     <span class="error-message"></span>
49
50                     <select required>
51                         <option value="">Selecione o sexo</option>
52                         <option value="masculino">Masculino</option>
53                         <option value="feminino">Feminino</option>
54                         <option value="outro">Outro</option>
55                     </select>
56
57                     <div class="input-id">
58                         <label for="identidade">Documento de identidade</label>
59                         <div class="upload-container">
60                             <input type="file" id="identidade" name="identidade" accept="image/*" required>
61                         </div>
62                     </div>
63             </form>
64         </div>
65     </div>
66 
```

Figura 4: Código HTML do formulário (parte 2).

```

1 index.html > <body>
2   <html lang="pt-BR">
3     <body>
4       <div class="container" id="container">
5         <div class="form-container sign-up">
6           <form id="signup-step2" class="signup-form" style="display: none;">
7             <input type="text" id="outro" name="outro" value="Outro" required>
8             <option value="outro">Outro</option>
9             </select>
10
11           <div class="input-id">
12             <label for="identidade">Documento de identidade</label>
13             <div class="upload-container">
14               <input type="file" id="identidade" name="identidade" accept="image/*" required>
15               <label for="identidade" class="upload-label">
16                 
17                 <p>Clique aqui para selecionar o arquivo</p>
18               </label>
19             </div>
20           </div>
21         </div>
22       </form>
23
24       <form id="signup-step3" class="signup-form" style="display: none;">
25         <div class="form-scroll-content">
26           <h1>Endereço Residencial</h1>
27           <span>Informe seu endereço completo</span>
28
29           <input type="text" id="cep" placeholder="CEP" required>
30           <span class="error-message"></span>
31
32           <input type="text" id="rua" placeholder="Rua" required disabled>
33           <input type="text" id="numero" placeholder="Número" required>
34           <span class="error-message"></span>
35
36           <input type="text" id="complemento" placeholder="Complemento (opcional)">
37           <input type="text" id="bairro" placeholder="Bairro" required disabled>
38           <input type="text" id="cidade" placeholder="Cidade" required disabled>
39
40           <select id="estado" required disabled>
41             <option value="">Selecione o estado</option>
42             <option value="AC">Acre</option>
43             <option value="AL">Alagoas</option>
44             <option value="AP">Amapá</option>
45             <option value="AM">Amazonas</option>
46             <option value="RR">Roraima</option>
47             <option value="RO">Rondônia</option>
48             <option value="MT">Mato Grosso</option>
49             <option value="MS">Mato Grosso do Sul</option>
50             <option value="GO">Goiás</option>
51             <option value="TO">Tocantins</option>
52             <option value="PI">Piauí</option>
53             <option value="CE">Ceará</option>
54             <option value="RN">Rio Grande do Norte</option>
55             <option value="PB">Paraíba</option>
56             <option value="PE">Pernambuco</option>
57             <option value="AL">Sergipe</option>
58             <option value="BA">Bahia</option>
59             <option value="ES">Espírito Santo</option>
60             <option value="MG">Minas Gerais</option>
61             <option value="SP">São Paulo</option>
62             <option value="PR">Paraná</option>
63             <option value="SC">Santa Catarina</option>
64             <option value="RS">Rio Grande do Sul</option>
65           </select>
66         </div>
67       </form>
68
69       <form id="signup-step4" class="signup-form" style="display: none;">
70         <div class="form-scroll-content">
71           <h1>Informações Pessoais</h1>
72           <span>Por favor, preencha as informações necessárias</span>
73
74           <input type="text" id="nome" placeholder="Nome" required>
75           <span class="error-message"></span>
76
77           <input type="text" id="sobrenome" placeholder="Sobrenome" required>
78           <span class="error-message"></span>
79
80           <input type="text" id="email" placeholder="Email" required>
81           <span class="error-message"></span>
82
83           <input type="text" id="senha" placeholder="Senha" required>
84           <span class="error-message"></span>
85
86           <input type="text" id="confirmar-senha" placeholder="Confirmar Senha" required>
87           <span class="error-message"></span>
88
89           <input type="checkbox" id="aceitar-termos" required>
90           <label for="aceitar-termos">Eu aceito os termos e condições</label>
91
92         </div>
93       </form>
94
95       <div class="form-buttons">
96         <button type="button" id="prev-to-step1">Voltar</button>
97         <button type="button" id="next-to-step3">Próximo</button>
98       </div>
99     </div>
100   </div>
101 </body>
102 </html>

```

Figura 5: Código HTML do formulário (parte 3).

Documentação - Sistema Interativo de Inscrição - Trilha Back-End

```
 2  <html lang="pt-BR">
14 <body>
15   <div class="container" id="container">
16     <div class="form-container sign-up">
17       <form id="signup-step3" class="signup-form" style="display: none;">
18         <option value="TO">Tocantins</option>
19       </select>
20     </div>
21     <div class="form-buttons">
22       <button type="button" id="prev-to-step2">Voltar</button>
23       <button type="button" id="next-to-step4">Próximo</button>
24     </div>
25   </form>
26
27   <form id="signup-step4" class="signup-form" style="display: none;">
28     <div class="form-scroll-content">
29       <h1>Seleção de Trilha:</h1>
30       <span>Escolha sua trilha de aprendizado e aceite os termos para finalizar seu cadastro</span>
31
32       <div class="trilha-options" required>
33         <h3>Selecione sua trilha:</h3>
34         <div class="trilha-options">
35           <div class="option">
36             <input type="radio" id="frontend" name="trilha" value="frontend" required>
37             <label for="frontend">
38               
39               Programação Front-end
40             </label>
41           </div>
42           <div class="option">
43             <input type="radio" id="backend" name="trilha" value="backend" required>
44             <label for="backend">
45               
46               Programação Back-end
47             </label>
48           </div>
49           <div class="option">
50             <input type="radio" id="jogos" name="trilha" value="jogos" required>
51             <label for="jogos">
52               
53               Programação de Jogos
54             </label>
55           </div>
56           <div class="option">
57             <input type="radio" id="design" name="trilha" value="design" required>
58             <label for="design">
59               
60               Design e Experiência
61           </div>
62         </div>
63       </div>
64     </div>
65
66     <div class="terms-container">
67       <input type="checkbox" id="terms" required>
68       <label class="text">Declaro que li e concordo com os <strong>Termos e Condições</strong>
69       e com a <strong>Política de Privacidade</strong>.</label>
70     </div>
71     <span id="terms-error" class="error-message">Aceite os termos para continuar</span>
72   </div>
73
74   <div class="form-buttons">
75     <button type="button" id="prev-to-step3">Voltar</button>
76     <button type="submit" id="finalizar-cadastro">Finalizar Cadastro</button>
77   </div>
78
79 </div>
```

Figura 6: Código HTML do formulário (parte 4)

```
 2  <html lang="pt-BR">
14 <body>
15   <div class="container" id="container">
16     <div class="form-container sign-up">
17       <form id="signup-step4" class="signup-form" style="display: none;">
18         <input type="radio" id="jogos" name="trilha" value="jogos" required>
19         <label for="jogos">
20           
21           Programação de Jogos
22         </label>
23       </div>
24       <div class="option">
25         <input type="radio" id="design" name="trilha" value="design" required>
26         <label for="design">
27           
28           Design e Experiência
29         </label>
30       </div>
31       <div class="option">
32         <input type="radio" id="dados" name="trilha" value="dados" required>
33         <label for="dados">
34           
35           Ciência de Dados
36         </label>
37       </div>
38       <span id="trilha-error" class="error-message">Por favor, selecione uma trilha</span>
39     </div>
40
41     <div class="terms-container">
42       <input type="checkbox" id="terms" required>
43       <label class="text">Declaro que li e concordo com os <strong>Termos e Condições</strong>
44       e com a <strong>Política de Privacidade</strong>.</label>
45     </div>
46     <span id="terms-error" class="error-message">Aceite os termos para continuar</span>
47   </div>
48
49   <div class="form-buttons">
50     <button type="button" id="prev-to-step3">Voltar</button>
51     <button type="submit" id="finalizar-cadastro">Finalizar Cadastro</button>
52   </div>
53
54 </div>
```

Figura 7: Código HTML do formulário (parte 5).

Documentação - Sistema Interativo de Inscrição - Trilha Back-End

```
index.html ④ num.
2   <html lang="pt-BR">
14  <body>
15    <div class="container" id="container">
16
17      <div class="toggle-container">
18        <div class="toggle">
19          <div class="toggle-panel toggle-left">
20            <h1>Olá!</h1>
21            <p>Já possui uma conta? Clique no botão abaixo para entrar na sua conta.</p>
22            <button class="hidden" id="login">Entrar</button>
23          </div>
24
25          <div class="toggle-panel toggle-left-two">
26            <h1>Complete seu cadastro!</h1>
27            <p>Termine de preencher as informações para registrar sua conta.</p>
28            <button class="hidden" id="back-to-step1">Voltar</button>
29          </div>
30
31          <div class="toggle-panel toggle-left-three">
32            <h1>Falta pouco!</h1>
33            <p>Preencha seu endereço para prosseguir.</p>
34            <button class="hidden" id="back-to-step2">Voltar</button>
35          </div>
36
37          <div class="toggle-panel toggle-left-four">
38            <h1>Quase lá!</h1>
39            <p>Escolha sua trilha de aprendizado e aceite os termos para finalizar.</p>
40            <button class="hidden" id="back-to-step3">Voltar</button>
41          </div>
42
43          <div class="toggle-panel toggle-right">
44            <h1>Seja bem-vindo(a)!</h1>
45            <p>Não possui uma conta? Clique no botão abaixo para se registrar.</p>
46            <button class="hidden" id="register">Criar conta</button>
47          </div>
48
49        </div>
50      </div>
51
52      <div id="feedback-modal" class="feedback-modal">
53        <div class="feedback-content">
54          <span class="close-feedback">&times;</span>
55          <h2>Cadastro realizado com sucesso!</h2>
56          <p>Como foi sua experiência com o nosso formulário de cadastro?</p>
57
58          <form id="feedback-form">
59            <div class="rating">
60              <input type="radio" id="star5" name="rating" value="5" />
61              <label for="star5">★</label>
62            </div>
63
64          </form>
65        </div>
66      </div>
67
68
69
70
```

Figura 8: Código HTML do formulário (parte 6)

```
index.html ④ num.
2   <html lang="pt-BR">
14  <body>
15    <div class="container" id="container">
16      <div class="toggle-container">
17        <div class="toggle">
18
19          <div class="toggle-panel toggle-right">
20            <h1>Seja bem-vindo(a)!</h1>
21            <p>Não possui uma conta? Clique no botão abaixo para se registrar.</p>
22            <button class="hidden" id="register">Criar conta</button>
23          </div>
24
25        </div>
26      </div>
27
28      <div id="feedback-modal" class="feedback-modal">
29        <div class="feedback-content">
30          <span class="close-feedback">&times;</span>
31          <h2>Cadastro realizado com sucesso!</h2>
32          <p>Como foi sua experiência com o nosso formulário de cadastro?</p>
33
34          <form id="feedback-form">
35            <div class="rating">
36              <input type="radio" id="star5" name="rating" value="5" />
37              <label for="star5">★</label>
38              <input type="radio" id="star4" name="rating" value="4" />
39              <label for="star4">★</label>
40              <input type="radio" id="star3" name="rating" value="3" />
41              <label for="star3">★</label>
42              <input type="radio" id="star2" name="rating" value="2" />
43              <label for="star2">★</label>
44              <input type="radio" id="star1" name="rating" value="1" required />
45              <label for="star1">★</label>
46            </div>
47
48            <textare id="feedback-comment" placeholder="Deixe seu comentário (opcional)..."/>
49
50            <button type="submit">Enviar Feedback</button>
51
52          </form>
53
54        </div>
55      </div>
56
57      <script src="script.js"></script>
58
59    </body>
60  </html>
```

Figura 9: Código HTML do formulário (parte 7).

Documentação - Sistema Interativo de Inscrição - Trilha Back-End

The screenshot shows the Visual Studio Code interface with the file 'style.css' open. The code is a CSS stylesheet for a registration form. It includes imports for Google Fonts (Montserrat) and a font-face rule for the 'number' input type. The code defines styles for various elements like inputs, buttons, and containers, using flexbox and CSS variables. The interface shows the code editor with syntax highlighting and a sidebar with file navigation.

```

1 @import url('https://fonts.googleapis.com/css?family=Montserrat:wght@300;400;500;600;700&display=swap');
2
3 *{
4   margin: 0;
5   padding: 0;
6   box-sizing: border-box;
7   font-family: 'Montserrat', sans-serif;
8 }
9
10 input[type="number"]::-webkit-inner-spin-button,
11 input[type="number"]::-webkit-outer-spin-button {
12   -webkit-appearance: none;
13   margin: 0;
14 }
15
16 input[type="number"] {
17   -moz-appearance: textfield;
18 }
19
20 body{
21   background-color: #c9daff;
22   background: linear-gradient(to right, #e2e2e2, #f893d1);
23   display: flex;
24   align-items: center;
25   justify-content: center;
26   flex-direction: column;
27   height: 100vh;
28 }
29
30 h4 {
31   font-weight: 400;
32   padding: 5px;
33   font-size: 14px;
34   text-align: justify;
35 }
36
37 .container{
38   background-color: #fff;
39   border-radius: 30px;
40   box-shadow: 0 8px 15px rgba(0, 0, 0, 0.35);
41   position: relative;
42   overflow: hidden;
43   width: 768px;
44   max-width: 100%;
45   min-height: 486px;
46 }
47
48 .container p{
49   font-size: 14px;
50 }

```

Figura 10: Código CSS do Sistema (Parte 1).

The screenshot shows the Visual Studio Code interface with the file 'style.css' open, continuing from Part 1. This section adds more complex styling for the sign-in and sign-up forms. It includes rules for the '.Form-container' class, which has absolute positioning, and the '.sign-in' and '.sign-up' classes, which have relative positioning. It also includes styles for the upload container, including a dashed border and a preview image. The code uses CSS transitions for element movement and z-index for layering. The interface shows the code editor with syntax highlighting and a sidebar with file navigation.

```

111 .Form-container{
112   position: absolute;
113   top: 0;
114   height: 100%;
115   transition: all 0.6s ease-in-out;
116 }
117
118 .sign-in{
119   left: 0;
120   width: 50%;
121   opacity: 0;
122   z-index: 2;
123 }
124 .container.active .sign-in{
125   transform: translateX(100%);
126 }
127
128 .sign-up{
129   left: 0;
130   width: 50%;
131   opacity: 0;
132   z-index: 1;
133 }
134
135 .container.active .sign-up{
136   transform: translateX(100%);
137   opacity: 1;
138   z-index: 5;
139   animation: move 0.6s;
140 }
141
142 .upload-container {
143   display: flex;
144   align-items: center;
145   justify-content: center;
146   border: 2px dashed #ccc;
147   border-radius: 10px;
148   padding: 20px;
149   width: 100%;
150   height: 100%;
151   text-align: center;
152   cursor: pointer;
153   transition: border 0.3s ease;
154 }
155
156 .upload-container:hover {
157   border-color: #f052b3;
158 }

```

Figura 11: Código CSS do Sistema (Parte 2).

Apêndice

Documentação - Sistema Interativo de Inscrição Trilha Back-End

The screenshot shows the Visual Studio Code interface with two tabs open: 'index.html' and 'style.css'. The 'style.css' tab is active, displaying a large block of CSS code. The code includes styles for various elements like input fields, social icons, and toggle containers, with some parts using keyframes for transitions. The interface includes a sidebar with file navigation, a top bar with tabs and icons, and a status bar at the bottom.

```
10 de abr 19:24
Desafio 3 - Trilha Back-End
File Edit Selection View Go Run Terminal Help
index.html style.css 1
assets > css > style.css > input[type="number"]
209 .input-id p {
210   color: #d3d3d1;
211 }
212
213
214
215 @keyframes move{
216   0%, 49.99%{
217     opacity: 0;
218     z-index: 1;
219   }
220   50%, 100%{
221     opacity: 1;
222     z-index: 5;
223   }
224 }
225
226 .social-icons{
227   margin: 0;
228 }
229
230 .social-icons a{
231   border: 1px solid #f052b3;
232   border-radius: 20%;
233   display: inline-flex;
234   justify-content: center;
235   align-items: center;
236   margin: 0 3px;
237   width: 40px;
238   height: 40px;
239 }
240
241 .toggle-container{
242   position: absolute;
243   top: 0;
244   left: 50%;
245   width: 50%;
246   height: 100%;
247   overflow: hidden;
248   transition: all 0.6s ease-in-out;
249   border-radius: 150px 0 0 100px;
250   z-index: 1000;
251 }
252
253 .container.active .toggle-container{
254   transform: translateX(-100%);
255   border-radius: 0 150px 100px 0;
256 }
257
258 .toggle{
259   background-color: #f052b3;
260   height: 100%;
261   background: linear-gradient(to right, #90cc00, #f052b3);
262   color: #fff;
263   position: relative;
264   left: -100px;
265   height: 100px;
266   width: 200px;
267   transform: translateX(0);
268   transition: all 0.6s ease-in-out;
269 }
270
271 .container.active .toggle{
272   transform: translateX(50%);
273 }
274
275 .toggle-panel{
276   position: absolute;
277   width: 50px;
278   height: 100px;
279   display: flex;
280   align-items: center;
281   justify-content: center;
282   flex-direction: column;
283   padding: 0 30px;
284   text-align: center;
285   top: 0;
286   transform: translateX(0);
287   transition: all 0.6s ease-in-out;
288 }
289
290 .toggle-left{
291   transform: translateX(-200px);
292 }
293
294 .container.active .toggle-left{
295   transform: translateX(0);
296 }
297
298 .toggle-right{
299   right: 0;
300   transform: translateX(0);
301 }
302
303 .container.active .toggle-right{
304   transform: translateX(200px);
305 }
```

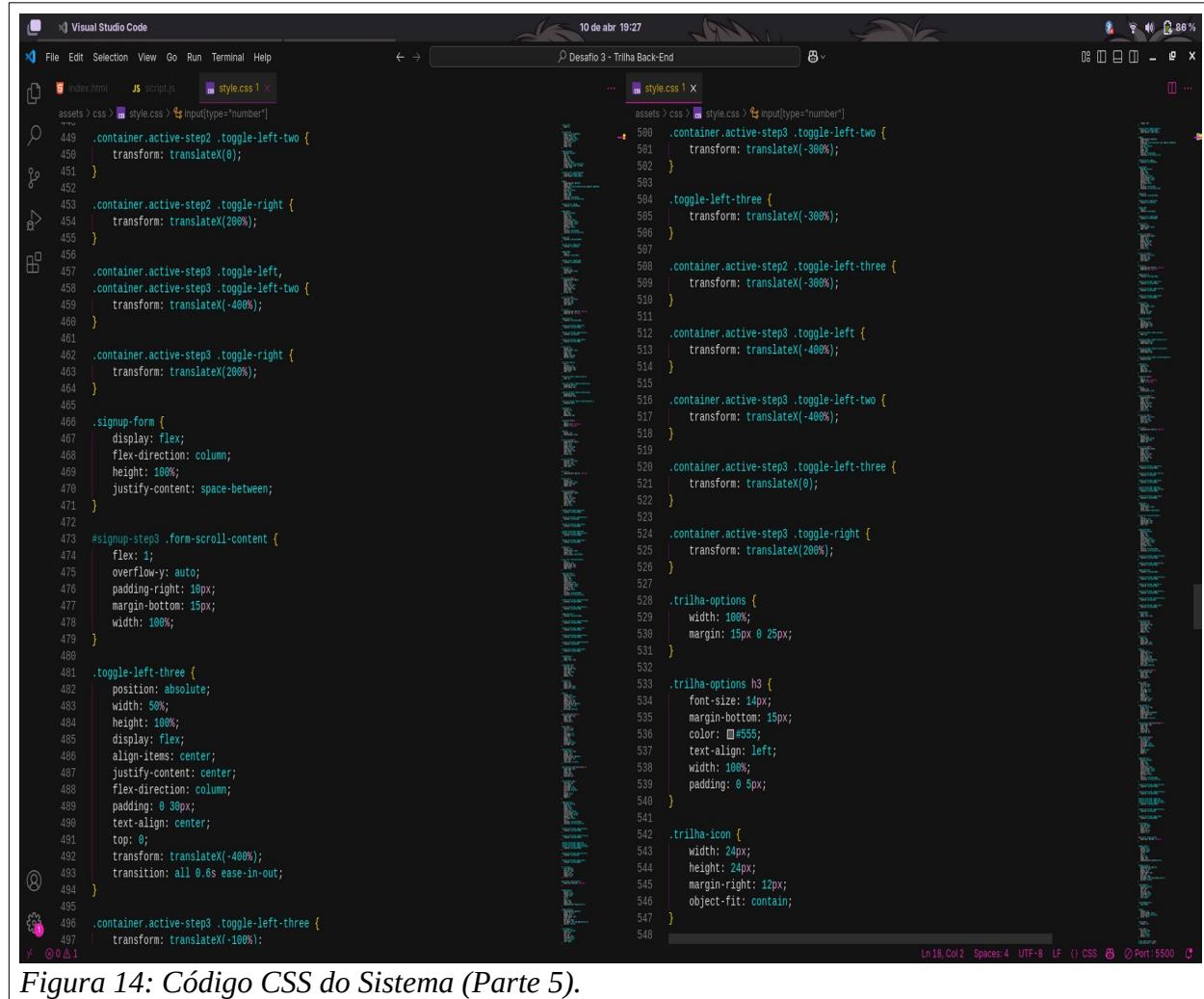
Figura 12: Código CSS do Sistema (Parte 3).

Documentação - Sistema Interativo de Inscrição - Trilha Back-End

```
10 de abr 19:26
Desafio 3 - Trilha Back-End
style.css 1 X
assets > css > style.css > input[type="number"]
352 #signup-step2 {
353   display: flex;
354   flex-direction: column;
355   width: 100%;
356   height: 100%;
357   padding: 20px 40px;
358   overflow: hidden;
359 }
360 .form-scroll-content {
361   flex: 1;
362   overflow-y: auto;
363   padding-right: 10px;
364   margin-bottom: 15px;
365   width: 100%;
366 }
367 .form-scroll-content::webkit-scrollbar {
368   width: 6px;
369 }
370 .form-scroll-content::webkit-scrollbar-track {
371   background: #f1f1f1;
372   border-radius: 10px;
373 }
374 .form-scroll-content::webkit-scrollbar-thumb {
375   background: #888;
376   border-radius: 10px;
377 }
378 .form-scroll-content::webkit-scrollbar-thumb:hover {
379   background: #555;
380 }
381 .form-buttons {
382   display: flex;
383   gap: 10px;
384   width: 100%;
385   padding-top: 10px;
386 }
387 .form-buttons button {
388   flex: 1;
389   padding: 10px 0 !important;
390   margin: 0 !important;
391 }
392 .sign-up {
393   399
394   400 .sign-up {
395     display: flex;
396     flex-direction: column;
397     401
398   }
399
400 .sign-up {
401   display: flex;
402   flex-direction: column;
403 }
404
405 #signup-step2 select {
406   background-color: #eeee;
407   border: none;
408   margin: 8px 0;
409   padding: 10px 15px;
410   font-size: 13px;
411   border-radius: 8px;
412   width: 100%;
413   outline: none;
414 }
415
416 #signup-step2 label {
417   align-self: flex-start;
418   font-size: 12px;
419   margin: 8px 0 4px;
420   color: #555;
421 }
422
423 #prev-step {
424   background-color: #e6c75d !important;
425 }
426
427 #signup-step3 {
428   display: flex;
429   flex-direction: column;
430   padding: 20px 40px;
431   height: 100%;
432 }
433
434 #signup-step3 select {
435   background-color: #eeee;
436   border: none;
437   margin: 8px 0;
438   padding: 10px 15px;
439   font-size: 13px;
440   border-radius: 8px;
441   width: 100%;
442   outline: none;
443 }
444
445 .container.active-step2 .toggle-left {
446   transform: translateX(-300%);
447 }
```

Figura 13: Código CSS do Sistema (Parte 4).

Documentação - Sistema Interativo de Inscrição - Trilha Back-End



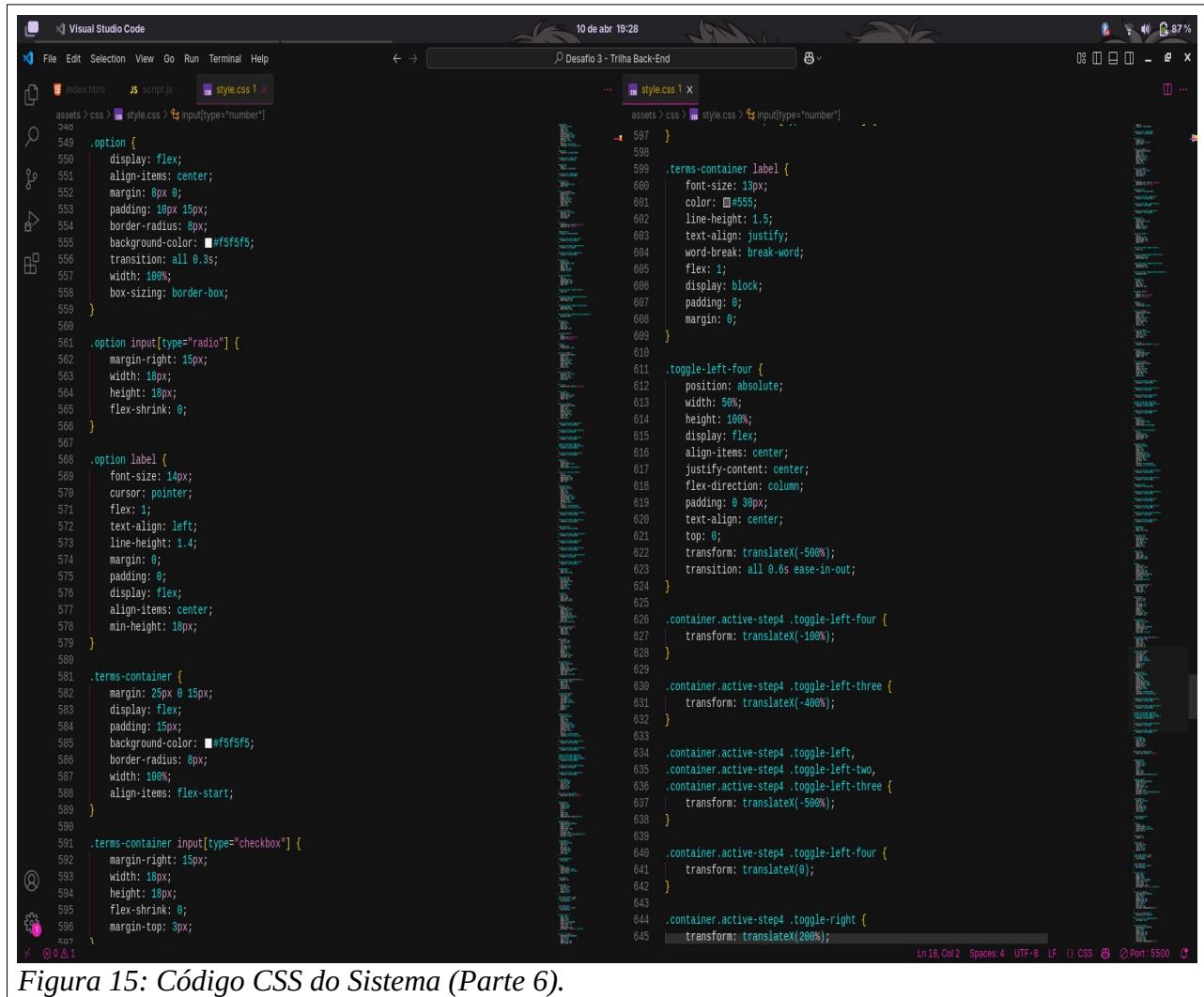
The screenshot shows a Visual Studio Code interface with the title bar "Visual Studio Code" and the date "10 de abr 19:27". The active tab is "style.css 1" with the file path "assets > css > style.css". The code editor displays a portion of the CSS file, specifically the styles for ".container.active-step3 .toggle-left-three" and ".trilha-options". The CSS uses transform: translateX() to manage the position of elements during transitions. The interface includes a sidebar with file navigation, a top bar with tabs and icons, and a status bar at the bottom.

```
449 .container.active-step2 .toggle-left-two {
450   transform: translateX(0);
451 }
452
453 .container.active-step2 .toggle-right {
454   transform: translateX(200%);
455 }
456
457 .container.active-step3 .toggle-left,
458 .container.active-step3 .toggle-left-two {
459   transform: translateX(-400%);
460 }
461
462 .container.active-step3 .toggle-right {
463   transform: translateX(200%);
464 }
465
466 .signup-form {
467   display: flex;
468   flex-direction: column;
469   height: 100%;
470   justify-content: space-between;
471 }
472
473 #signup-step3 .form-scroll-content {
474   flex: 1;
475   overflow-y: auto;
476   padding-right: 10px;
477   margin-bottom: 15px;
478   width: 100%;
479 }
480
481 .toggle-left-three {
482   position: absolute;
483   width: 50%;
484   height: 100%;
485   display: flex;
486   align-items: center;
487   justify-content: center;
488   flex-direction: column;
489   padding: 0 30px;
490   text-align: center;
491   top: 0;
492   transform: translateX(-400%);
493   transition: all 0.6s ease-in-out;
494 }
495
496 .container.active-step3 .toggle-left-three {
497   transform: translateX(-100%);
```

500 .container.active-step3 .toggle-left-two {
501 transform: translateX(-300%);
502 }
503
504 .toggle-left-three {
505 transform: translateX(-300%);
506 }
507
508 .container.active-step2 .toggle-left-three {
509 transform: translateX(-300%);
510 }
511
512 .container.active-step3 .toggle-left {
513 transform: translateX(-400%);
514 }
515
516 .container.active-step3 .toggle-left-two {
517 transform: translateX(-400%);
518 }
519
520 .container.active-step3 .toggle-left-three {
521 transform: translateX(0);
522 }
523
524 .container.active-step3 .toggle-right {
525 transform: translateX(200%);
526 }
527
528 .trilha-options {
529 width: 100%;
530 margin: 15px 0 25px;
531 }
532
533 .trilha-options h3 {
534 font-size: 14px;
535 margin-bottom: 15px;
536 color: #555;
537 text-align: left;
538 width: 100%;
539 padding: 0 5px;
540 }
541
542 .trilha-icon {
543 width: 24px;
544 height: 24px;
545 margin-right: 12px;
546 object-fit: contain;
547 }

Figura 14: Código CSS do Sistema (Parte 5).

Documentação - Sistema Interativo de Inscrição - Trilha Back-End

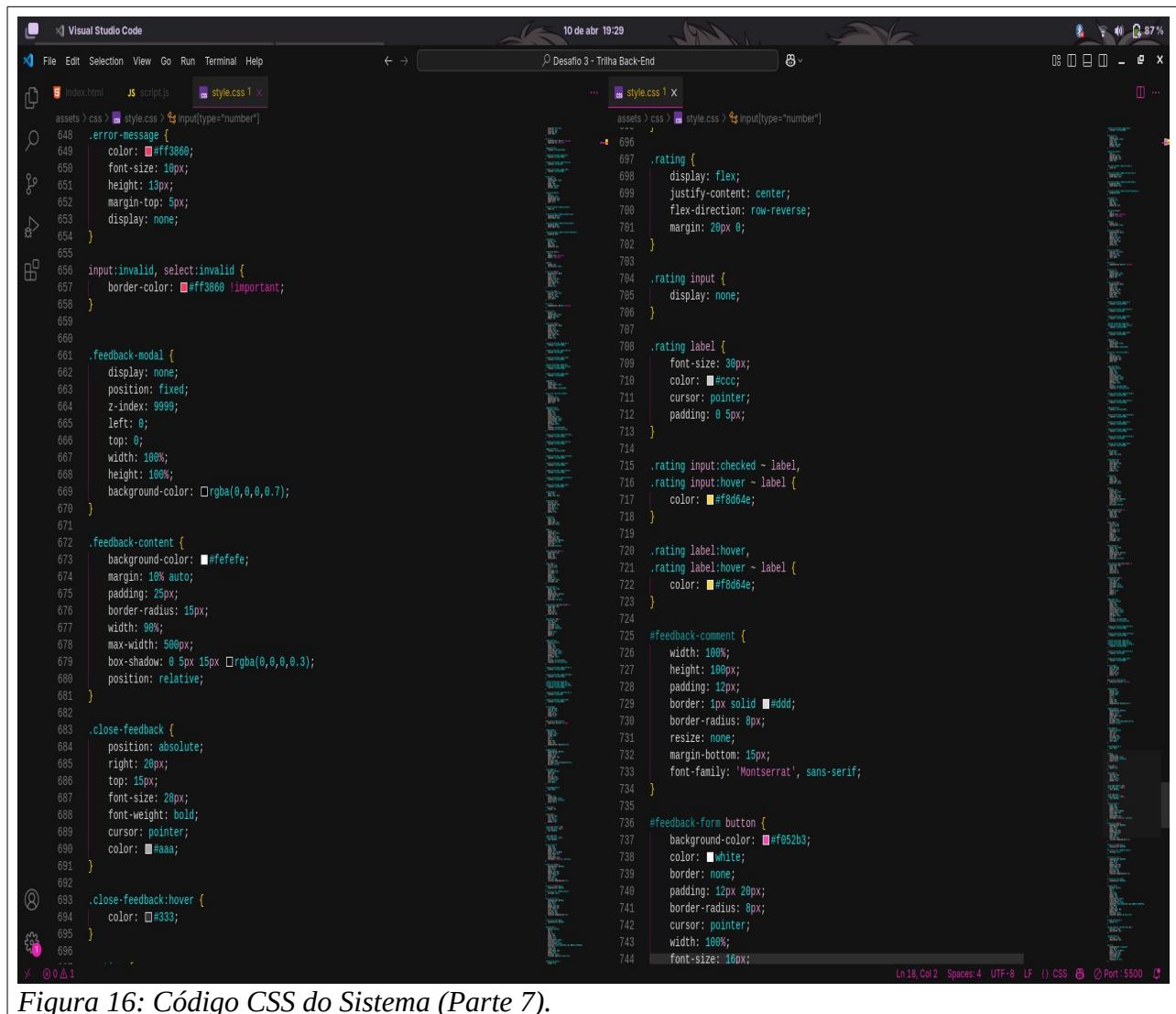


The screenshot shows the Visual Studio Code interface with two tabs open: 'style.css' and 'index.html'. The 'style.css' tab contains approximately 650 lines of CSS code, primarily for styling input fields and containers. The code includes rules for '.option' (flexible display), '.terms-container' (flex items), and various '.toggle-left' and '.container.active-step4' classes for toggle buttons. The 'index.html' tab is partially visible on the left. The status bar at the bottom right shows 'Ln18, Col2' and 'UTF-8'.

```
10 de abr 19:28
Desafio 3 - Trilha Back-End
File Edit Selection View Go Run Terminal Help
index.html JS script.js style.css 1 x
assets > css > style.css > input[type="number"]
549 .option {
550   display: flex;
551   align-items: center;
552   margin: 8px 0;
553   padding: 10px 15px;
554   border-radius: 8px;
555   background-color: #f5f5f5;
556   transition: all 0.3s;
557   width: 100%;
558   box-sizing: border-box;
559 }
560 .option input[type="radio"] {
561   margin-right: 15px;
562   width: 18px;
563   height: 18px;
564   flex-shrink: 0;
565 }
566 .option label {
567   font-size: 14px;
568   cursor: pointer;
569   flex: 1;
570   text-align: left;
571   line-height: 1.4;
572   margin: 0;
573   padding: 0;
574   display: flex;
575   align-items: center;
576   min-height: 18px;
577 }
578 .terms-container {
579   margin: 25px 0 15px;
580   display: flex;
581   padding: 15px;
582   background-color: #f5f5f5;
583   border-radius: 8px;
584   width: 100%;
585   align-items: flex-start;
586 }
587 .terms-container input[type="checkbox"] {
588   margin-right: 15px;
589   width: 18px;
590   height: 18px;
591   flex-shrink: 0;
592   margin-top: 3px;
593 }
594
595
596
597
598
599 .terms-container label {
600   font-size: 13px;
601   color: #555;
602   line-height: 1.5;
603   text-align: justify;
604   word-break: break-word;
605   flex: 1;
606   display: block;
607   padding: 0;
608   margin: 0;
609 }
610 .toggle-left-four {
611   position: absolute;
612   width: 50px;
613   height: 100px;
614   display: flex;
615   align-items: center;
616   justify-content: center;
617   flex-direction: column;
618   padding: 0 30px;
619   text-align: center;
620   top: 0;
621   transform: translateX(-50%);
622   transition: all 0.6s ease-in-out;
623 }
624
625 .container.active-step4 .toggle-left-four {
626   transform: translateX(-100%);
627 }
628
629 .container.active-step4 .toggle-left-three {
630   transform: translateX(-400%);
631 }
632
633 .container.active-step4 .toggle-left,
634 .container.active-step4 .toggle-left-two,
635 .container.active-step4 .toggle-left-three {
636   transform: translateX(-50%);
637 }
638
639
640 .container.active-step4 .toggle-left-four {
641   transform: translateX(0);
642 }
643
644 .container.active-step4 .toggle-right {
645   transform: translateX(200%);
```

Figura 15: Código CSS do Sistema (Parte 6).

Documentação - Sistema Interativo de Inscrição - Trilha Back-End



The screenshot shows a Visual Studio Code interface with two tabs open: 'style.css' and 'style.css' (another instance). The left pane displays the CSS code for various UI components like error messages, feedback modals, and rating scales. The right pane shows the same code with some lines highlighted in yellow, indicating they are being edited or are part of the current selection.

```
assets > css > style.css > input[type="number"]  
648 .error-message {  
649     color: #ff3860;  
650     font-size: 10px;  
651     height: 13px;  
652     margin-top: 5px;  
653     display: none;  
654 }  
655  
656 input:invalid, select:invalid {  
657     border-color: #ff3860 !important;  
658 }  
659  
660 .feedback-modal {  
661     display: none;  
662     position: fixed;  
663     z-index: 9999;  
664     left: 0;  
665     top: 0;  
666     width: 100%;  
667     height: 100%;  
668     background-color: rgba(0,0,0,0.7);  
669 }  
670  
671 .feedback-content {  
672     background-color: #fefefe;  
673     margin: 10% auto;  
674     padding: 25px;  
675     border-radius: 15px;  
676     width: 90%;  
677     max-width: 500px;  
678     box-shadow: 0 5px 15px rgba(0,0,0,0.3);  
679     position: relative;  
680 }  
681  
682 .close-feedback {  
683     position: absolute;  
684     right: 20px;  
685     top: 15px;  
686     font-size: 28px;  
687     font-weight: bold;  
688     cursor: pointer;  
689     color: #aaa;  
690 }  
691  
692 .close-feedback:hover {  
693     color: #333;  
694 }  
695  
696 ...  
697  
698 .rating {  
699     display: flex;  
700     justify-content: center;  
701     flex-direction: row-reverse;  
702     margin: 20px 0;  
703 }  
704 .rating input {  
705     display: none;  
706 }  
707  
708 .rating label {  
709     font-size: 30px;  
710     color: #ccc;  
711     cursor: pointer;  
712     padding: 0 5px;  
713 }  
714  
715 .rating input:checked ~ label,  
716 .rating input:hover ~ label {  
717     color: #f8d64e;  
718 }  
719  
720 .rating label:hover,  
721 .rating label:hover ~ label {  
722     color: #f8d64e;  
723 }  
724  
725 #feedback-comment {  
726     width: 100%;  
727     height: 100px;  
728     padding: 12px;  
729     border: 1px solid #ddd;  
730     border-radius: 8px;  
731     resize: none;  
732     margin-bottom: 15px;  
733     font-family: 'Montserrat', sans-serif;  
734 }  
735  
736 #feedback-form button {  
737     background-color: #f052b3;  
738     color: white;  
739     border: none;  
740     padding: 12px 20px;  
741     border-radius: 8px;  
742     cursor: pointer;  
743     width: 100%;  
744     font-size: 16px;
```

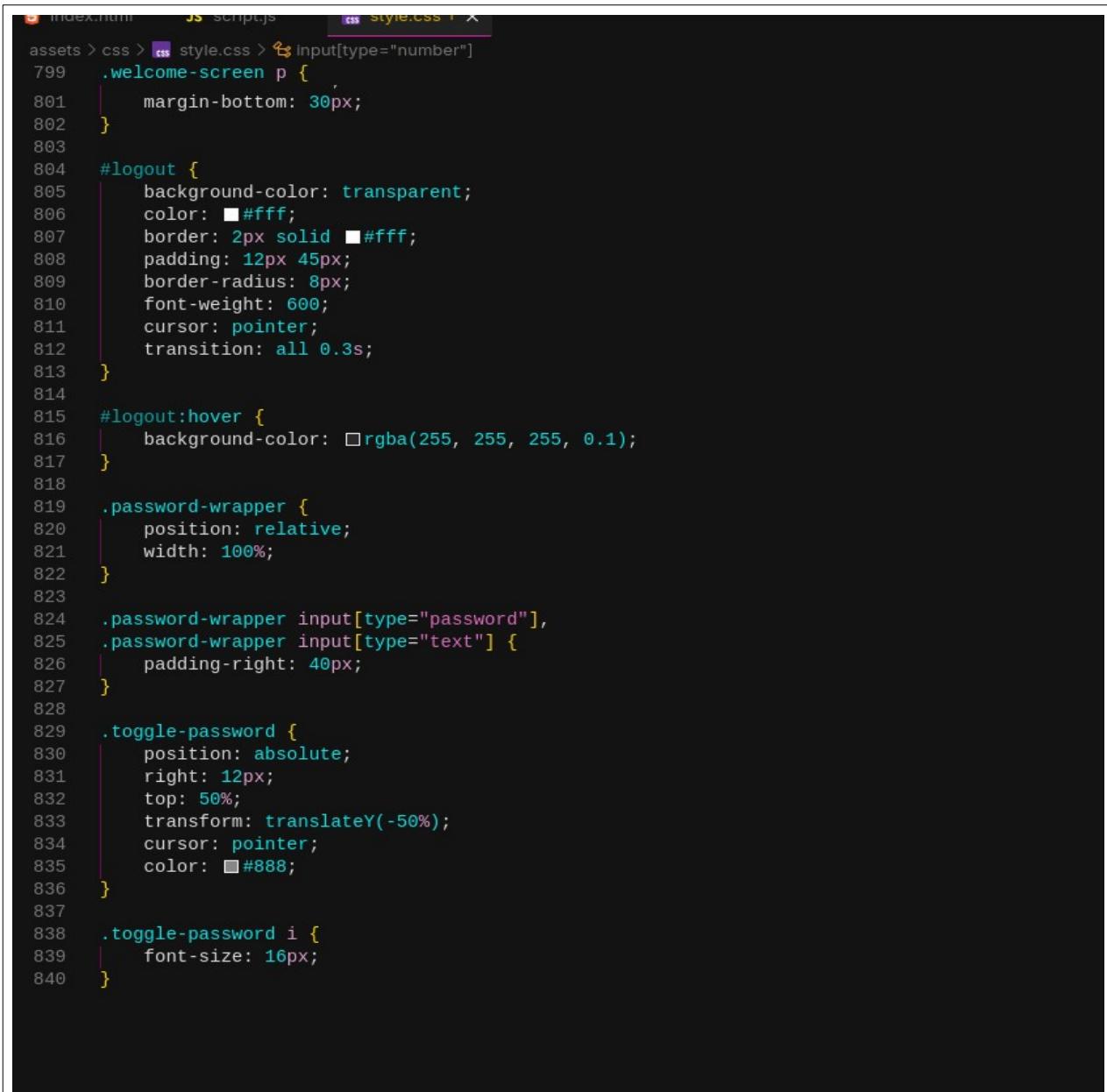
Figura 16: Código CSS do Sistema (Parte 7).

Documentação - Sistema Interativo de Inscrição - Trilha Back-End

```
10 de abr 19:29
Desafio 3 - Trilha Back-End

File Edit Selection View Go Run Terminal Help
index.html style.css
style.css
assets > css > style.css > input[type="number"]
736 #feedback-form button {
737   background-color: #f052b3;
738   color: white;
739   border: none;
740   padding: 12px 28px;
741   border-radius: 8px;
742   cursor: pointer;
743   width: 100%;
744   font-size: 16px;
745   transition: background-color 0.3s;
746 }
747
748 #feedback-form button:hover {
749   background-color: #d0459b;
750 }
751 #confirmation-modal .feedback-content {
752   text-align: center;
753 }
754
755 #confirmation-ok-btn {
756   background-color: #f052b3;
757   color: white;
758   border: none;
759   padding: 12px 38px;
760   border-radius: 8px;
761   cursor: pointer;
762   font-size: 16px;
763   margin-top: 28px;
764   transition: background-color 0.3s;
765 }
766
767 #confirmation-ok-btn:hover {
768   background-color: #d0459b;
769 }
770
771 .welcome-screen {
772   position: absolute;
773   top: 0;
774   left: 0;
775   width: 100%;
776   height: 100%;
777   display: none;
778   flex-direction: column;
779   align-items: center;
780   justify-content: center;
781   background: linear-gradient(to right, #905cc0, #f052b3);
782   color: white;
783   z-index: 100;
784   border-radius: 30px;
785   padding: 40px;
786   text-align: center;
787   transition: opacity 0.6s ease-in-out;
788 }
789
790 .welcome-screen.active {
791   display: flex;
792 }
793
794 .welcome-screen h1, .welcome-screen span {
795   font-size: 2.5rem;
796   margin-bottom: 28px;
797 }
798
799 .welcome-screen p {
800   font-size: 1.2rem;
801   margin-bottom: 38px;
802 }
803
804 #logout {
805   background-color: transparent;
806   color: white;
807   border: 2px solid white;
808   padding: 12px 45px;
809   border-radius: 8px;
810   font-weight: bold;
811   cursor: pointer;
812   transition: all 0.3s;
813 }
814
815 #logout:hover {
816   background-color: rgba(255, 255, 255, 0.1);
817 }
818
819 .password-wrapper {
```

Figura 17: Código CSS do Sistema (Parte 8).

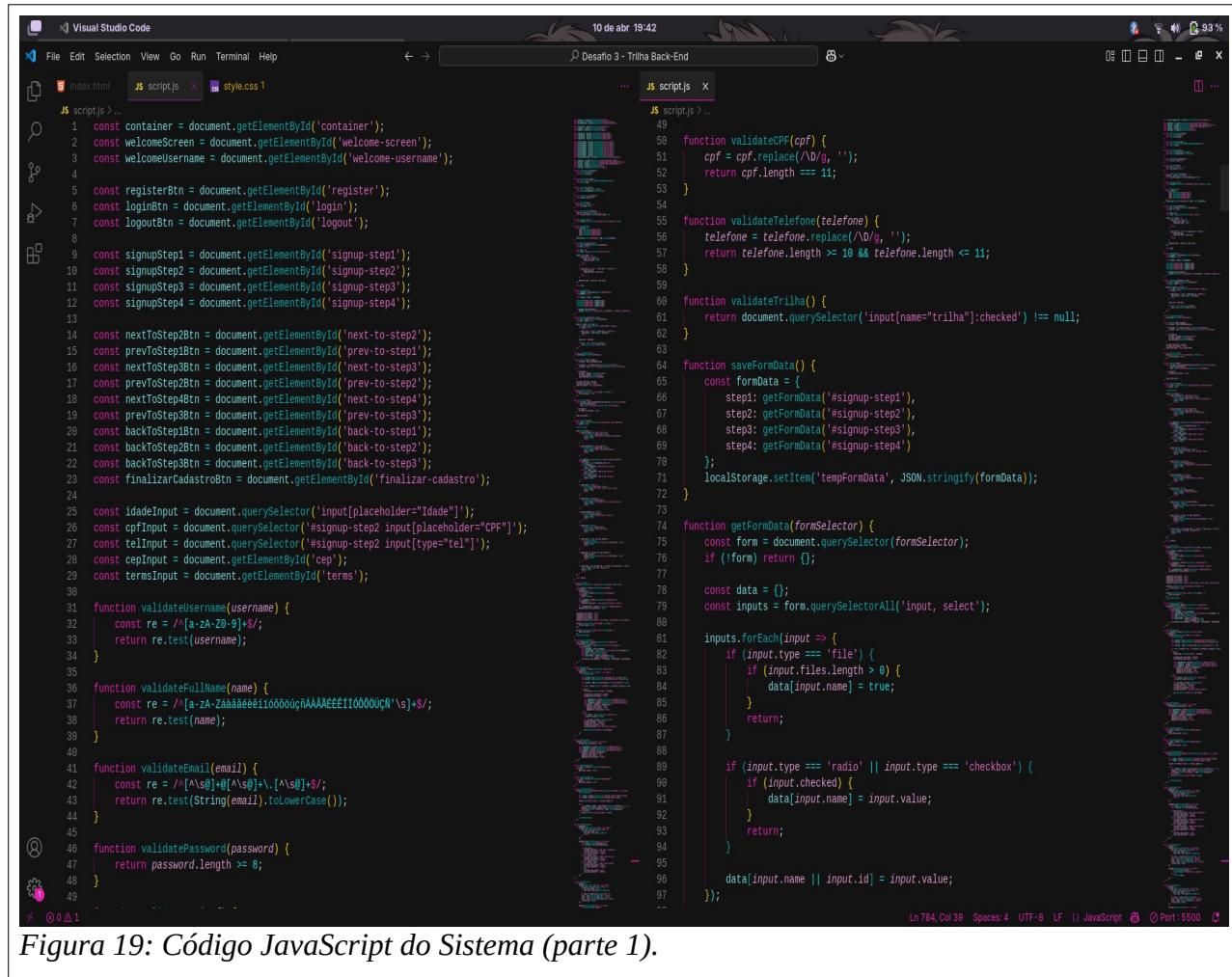


The screenshot shows a code editor with several tabs at the top: 'Index.html', 'script.js', 'style.css', and 'assets'. The 'style.css' tab is active, displaying the following CSS code:

```
assets > css > style.css > input[type="number"]
799 .welcome-screen p {
800   margin-bottom: 30px;
801 }
802
803 #logout {
804   background-color: transparent;
805   color: #fff;
806   border: 2px solid #fff;
807   padding: 12px 45px;
808   border-radius: 8px;
809   font-weight: 600;
810   cursor: pointer;
811   transition: all 0.3s;
812 }
813
814 #logout:hover {
815   background-color: rgba(255, 255, 255, 0.1);
816 }
817
818 .password-wrapper {
819   position: relative;
820   width: 100%;
821 }
822
823 .password-wrapper input[type="password"],
824 .password-wrapper input[type="text"] {
825   padding-right: 40px;
826 }
827
828 .toggle-password {
829   position: absolute;
830   right: 12px;
831   top: 50%;
832   transform: translateY(-50%);
833   cursor: pointer;
834   color: #888;
835 }
836
837 .toggle-password i {
838   font-size: 16px;
839 }
```

Figura 18: Código CSS do Sistema (Parte 9).

Documentação - Sistema Interativo de Inscrição - Trilha Back-End



```
1 const container = document.getElementById('container');
2 const welcomeScreen = document.getElementById('welcome-screen');
3 const welcomeUsername = document.getElementById('welcome-username');
4
5 const registerBtn = document.getElementById('register');
6 const loginBtn = document.getElementById('login');
7 const logoutBtn = document.getElementById('logout');
8
9 const signupStep1 = document.getElementById('signup-step1');
10 const signupStep2 = document.getElementById('signup-step2');
11 const signupStep3 = document.getElementById('signup-step3');
12 const signupStep4 = document.getElementById('signup-step4');
13
14 const nextToStep2Btn = document.getElementById('next-to-step2');
15 const prevToStep1Btn = document.getElementById('prev-to-step1');
16 const nextToStep3Btn = document.getElementById('next-to-step3');
17 const prevToStep2Btn = document.getElementById('prev-to-step2');
18 const nextToStep4Btn = document.getElementById('next-to-step4');
19 const prevToStep3Btn = document.getElementById('prev-to-step3');
20 const backToStep1Btn = document.getElementById('back-to-step1');
21 const backToStep2Btn = document.getElementById('back-to-step2');
22 const backToStep3Btn = document.getElementById('back-to-step3');
23 const finalizarCadastroBtn = document.getElementById('finalizar-cadastro');
24
25 const idadeInput = document.querySelector('input[placeholder="Idade"]');
26 const cpfInput = document.querySelector('#signup-step2 input[placeholder="CPF"]');
27 const telInput = document.querySelector('#signup-step2 input[type="tel"]');
28 const cepInput = document.getElementById('cep');
29 const termsInput = document.getElementById('terms');
30
31 function validateUsername(username) {
32   const re = /^[a-zA-Z0-9]+$/;
33   return re.test(username);
34 }
35
36 function validateFullName(name) {
37   const re = /^[a-zA-Zááééííóóúúçç]{3,10}[a-zA-Zááééííóóúúçç]{0,10}([a-zA-Zááééííóóúúçç]{0,10})+$/;
38   return re.test(name);
39 }
40
41 function validateEmail(email) {
42   const re = /^[^@]+@[^\s]+\.\w+\w+$/;
43   return re.test(String(email).toLowerCase());
44 }
45
46 function validatePassword(password) {
47   return password.length > 8;
48 }
49
50 function validateCPF(cpf) {
51   cpf = cpf.replace(/\D/g, '');
52   return cpf.length === 11;
53 }
54
55 function validateTelefone(telefone) {
56   telefone = telefone.replace(/\D/g, '');
57   return telefone.length >= 10 && telefone.length <= 11;
58 }
59
60 function validateTrilha() {
61   return document.querySelector('input[name="trilha":checked"]') != null;
62 }
63
64 function saveFormData() {
65   const formData = {
66     step1: getFormData('#signup-step1'),
67     step2: getFormData('#signup-step2'),
68     step3: getFormData('#signup-step3'),
69     step4: getFormData('#signup-step4')
70   };
71   localStorage.setItem('tempFormData', JSON.stringify(formData));
72 }
73
74 function getFormData(formSelector) {
75   const form = document.querySelector(formSelector);
76   if (!form) return {};
77
78   const data = {};
79   const inputs = form.querySelectorAll('input, select');
80
81   inputs.forEach(input => {
82     if (input.type === 'file') {
83       if (input.files.length > 0) {
84         data[input.name] = true;
85       }
86       return;
87     }
88
89     if (input.type === 'radio' || input.type === 'checkbox') {
90       if (input.checked) {
91         data[input.name] = input.value;
92       }
93       return;
94     }
95
96     data[input.name] = input.value;
97   });
98 }
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
```

Figura 19: Código JavaScript do Sistema (parte 1).

Documentação - Sistema Interativo de Inscrição - Trilha Back-End

The screenshot shows a Visual Studio Code interface with two tabs open: 'script.js' and 'style.css'. The 'script.js' tab contains the following code:

```
function getFormData(selector) {
  inputs.forEach(input => {
    data[input.name || input.id] = input.value;
  });
  return data;
}

function loadFormData() {
  const savedData = localStorage.getItem('tempFormData');
  if (!savedData) return;

  const formData = JSON.parse(savedData);

  setFormData('#signup-step1', formData.step1);
  setFormData('#signup-step2', formData.step2);
  setFormData('#signup-step3', formData.step3);
  setFormData('#signup-step4', formData.step4);

  if (formData.step2.identidade) {
    document.querySelector('.upload-label p').textContent = 'Documento carregado (pré)';
  }
}

function setFormData(selector, data) {
  const form = document.querySelector(selector);
  if (!form || !data) return;

  Object.keys(data).forEach(key => {
    const input = form.querySelector(`[name="${key}"]`, `[${key}]`);
    if (!input) return;

    if (input.type === 'radio' || input.type === 'checkbox') {
      input.checked = (input.value === data[key]);
      return;
    }

    input.value = data[key];
  });

  if(['cpf', 'cep', 'tel'].includes(key)) {
    input.dispatchEvent(new Event('input'));
  }
}

function clearSavedFormData() {
  localStorage.removeItem('tempFormData');
}

function showError(input, message) {
  let errorSpan = input.nextElementSibling;

  if (!errorSpan || !errorSpan.classList.contains('error-message')) {
    errorSpan = input.closest('.form-group')?.querySelector('.error-message');
  }

  if (!errorSpan) {
    errorSpan = document.createElement('span');
    errorSpan.className = 'error-message';
    input.parentNode.appendChild(errorSpan);
  }

  errorSpan.textContent = message;
  errorSpan.style.display = 'block';
  input.style.borderColor = '1px solid #ff3060';
}

function validateForm(form) {
  const inputs = form.querySelectorAll('input[required], select[required]');
  let isValid = true;

  clearAllErrors();

  inputs.forEach(input => {
    if (input.placeholder === 'Nome de usuário') {
      if (!validateUsername(input.value.trim())) {
        showError(input, 'Apenas letras e números são permitidos');
        isValid = false;
        return;
      }
    }
  });
}

function validateUsername(username) {
  const regex = /^[a-zA-Z0-9]+$/;
  return regex.test(username);
}
```

The 'style.css' tab is partially visible on the right.

Figura 20: Código JavaScript do Sistema (parte 2).

Documentação - Sistema Interativo de Inscrição - Trilha Back-End

The screenshot shows two tabs of a JavaScript file named 'script.js' in a Visual Studio Code editor. The left tab contains the first part of the code, and the right tab contains the second part. Both tabs have line numbers from 179 to 286. The code implements a validation function for various input types including text, date, email, password, and file uploads. It includes logic for validating full names, telephone numbers, age ranges, and CPF numbers. It also handles file inputs and checkbox requirements. The code uses functions like validateForm, validateName, validateCPF, validateEmail, validatePassword, validateTrilha, and validateTelefone. It includes error handling and UI updates for invalid inputs.

```
function validateForm(form) {
    inputs.forEach(input => {
        if (input.placeholder === 'Nome Completo') {
            if (!validateFullName(input.value.trim())) {
                showError(input, 'Apenas letras são permitidas');
                isValid = false;
                return;
            }
        }

        const idadeField = form.querySelector('input[placeholder="Idade"]');
        if (idadeField) {
            const idade = parseInt(idadeField.value) || 0;
            if (!isValidIdade()) {
                showError(idadeField, 'Idade deve ser um número');
                isValid = false;
            } else if (idade < 18) {
                showError(idadeField, 'Idade mínima: 18 anos');
                isValid = false;
            } else if (idade > 120) {
                showError(idadeField, 'Idade máxima: 120 anos');
                isValid = false;
            }
        }

        if (input.type === 'email') {
            if (!validateEmail(input.value.trim())) {
                showError(input, 'Por favor, insira um email válido');
                isValid = false;
                return;
            }
        }

        if (input.type === 'password') {
            if (!validatePassword(input.value)) {
                showError(input, 'A senha deve ter no mínimo 8 caracteres');
                isValid = false;
                return;
            }
        }

        if (input.placeholder === 'CPF') {
            if (!validateCPF(input.value)) {
                showError(input, 'CPF deve conter 11 dígitos');
                isValid = false;
                return;
            }
        }
    })
}

function validateForm(form) {
    inputs.forEach(input => {
        if (input.type === 'tel') {
            if (!validateTelefone(input.value)) {
                showError(input, 'Telefone inválido');
                isValid = false;
                return;
            }
        }

        if (input.type === 'radio' && input.required) {
            if (!validateTrilha()) {
                document.getElementById('trilha-error').style.display = 'block';
                isValid = false;
                return;
            }
        }

        if (input.type === 'checkbox' && input.required) {
            if (!input.checked) {
                document.getElementById('terms-error').style.display = 'block';
                isValid = false;
                return;
            }
        }

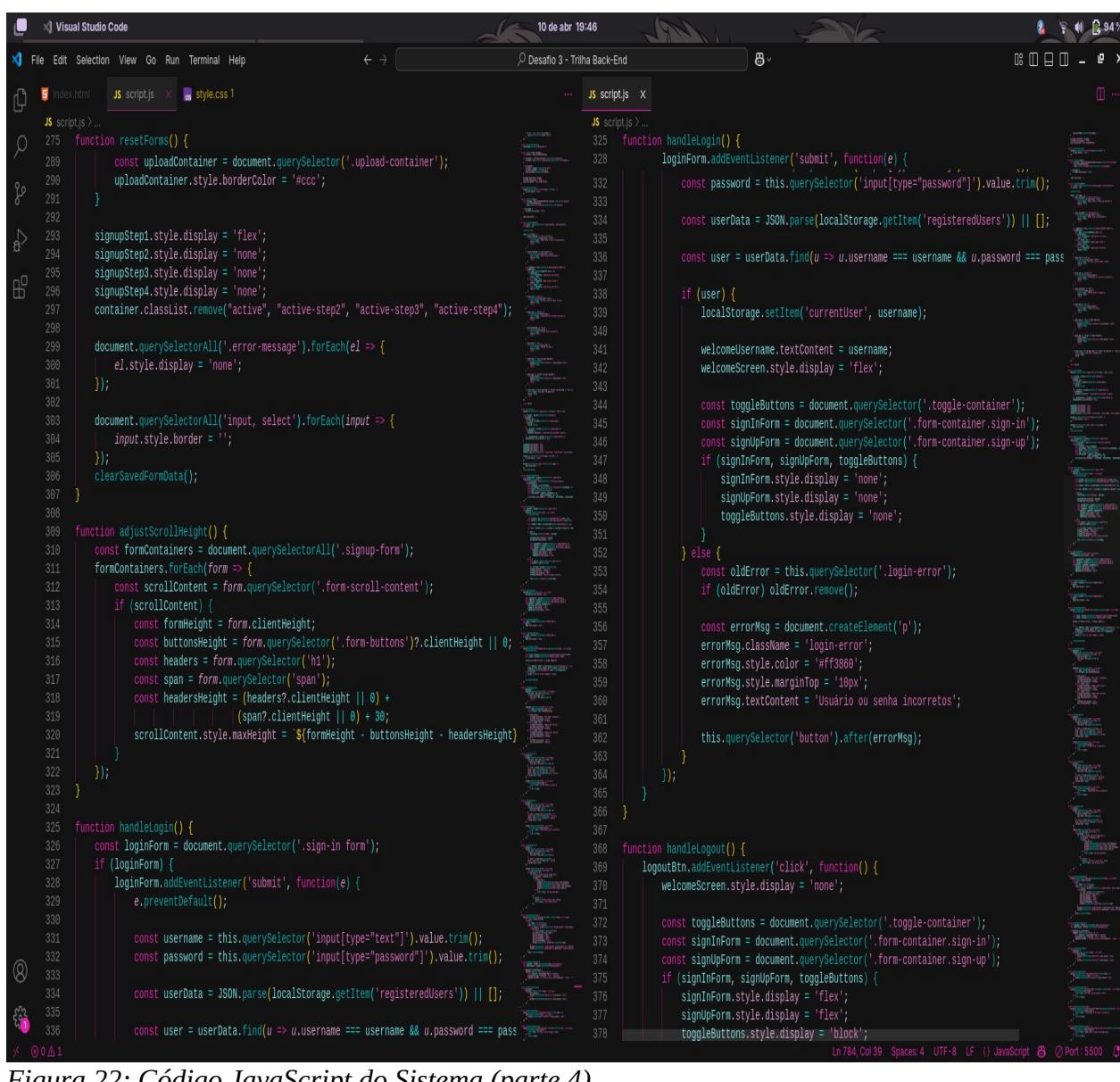
        if (!input.value.trim() && input.type !== 'checkbox' && input.type !== 'radio') {
            showError(input, 'Este campo é obrigatório');
            isValid = false;
            return;
        }
    });
    return isValid;
}

function resetForms() {
    document.querySelectorAll('.signup-form').forEach(form => form.reset());
}

const fileInput = document.getElementById('identidade');
if (fileInput) {
    fileInput.value = '';
    const preview = document.querySelector('.preview-image');
    if (preview) {
        preview.remove();
    }
    const uploadText = document.querySelector('.upload-label p');
    if (uploadText) {
```

Figura 21: Código JavaScript do Sistema (parte 3).

Documentação - Sistema Interativo de Inscrição - Trilha Back-End



The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows files: index.html, script.js, and style.css.
- Editor:** Displays the script.js file content.
- Status Bar:** Shows the date (10 de abr 19:46), battery level (94%), and other system information.

The script.js code is as follows:

```
function resetForms() {
    const uploadContainer = document.querySelector('.upload-container');
    uploadContainer.style.borderColor = '#ccc';
}

signupStep1.style.display = 'flex';
signupStep2.style.display = 'none';
signupStep3.style.display = 'none';
signupStep4.style.display = 'none';
container.classList.remove('active', 'active-step2', 'active-step3', 'active-step4');

document.querySelectorAll('.error-message').forEach(el => {
    el.style.display = 'none';
});

document.querySelectorAll('input, select').forEach(input => {
    input.style.border = '';
});
clearSavedFormData();
}

function adjustScrollHeight() {
    const formContainers = document.querySelectorAll('.signup-form');
    formContainers.forEach(form => {
        const scrollContent = form.querySelector('.form-scroll-content');
        if (scrollContent) {
            const formHeight = form.clientHeight;
            const buttonsHeight = form.querySelector('.form-buttons')?.clientHeight || 0;
            const headers = form.querySelector('h1');
            const span = form.querySelector('span');
            const headersHeight = (headers.clientHeight || 0) +
                (span.clientHeight || 0) + 30;
            scrollContent.style.maxHeight = `${formHeight - buttonsHeight - headersHeight}`;
        }
    });
}

function handleLogin() {
    const loginForm = document.querySelector('.sign-in form');
    if (loginForm) {
        loginForm.addEventListener('submit', function(e) {
            e.preventDefault();

            const username = this.querySelector('input[type="text"]').value.trim();
            const password = this.querySelector('input[type="password"]').value.trim();

            const userData = JSON.parse(localStorage.getItem('registeredUsers')) || [];
            const user = userData.find(u => u.username === username && u.password === pass);
        });
    }
}

function handleLogout() {
    logoutBtn.addEventListener('click', function() {
        welcomeScreen.style.display = 'none';

        const toggleButtons = document.querySelector('.toggle-container');
        const signInForm = document.querySelector('.form-container.sign-in');
        const signUpForm = document.querySelector('.form-container.sign-up');
        if (signInForm, signUpForm, toggleButtons) {
            signInForm.style.display = 'flex';
            signUpForm.style.display = 'flex';
            toggleButtons.style.display = 'block';
        }
    });
}
```

Figura 22: Código JavaScript do Sistema (parte 4).

Documentação - Sistema Interativo de Inscrição - Trilha Back-End

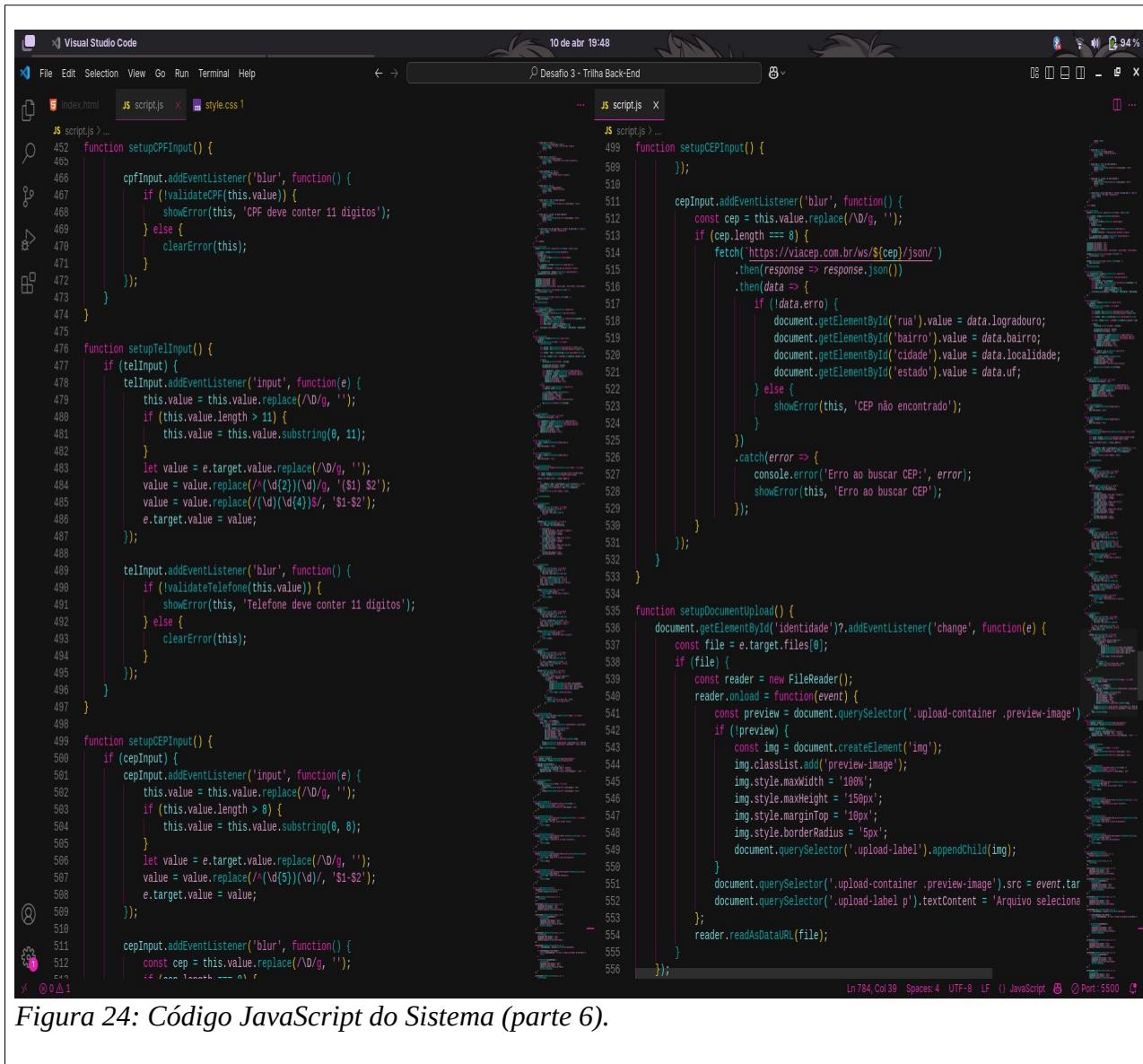
The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows files: index.html, script.js, and style.css.
- Editor:** The main area displays the script.js file with code numbered from 368 to 471. The code handles user interactions like logout, form toggling, and feedback submission, and performs validation on age and CPF inputs.
- Terminal:** Shows the date and time: 10 de abr 10:47.
- Status Bar:** Shows the file name: Desafio 3 - Trilha Back-End, line count: Ln 784, column count: Col 39, and encoding: UTF-8.

```
368     function handleLogout() {
369         logoutBtn.addEventListener('click', function() {
370             const toggleButtons = document.querySelector('.toggle-container');
371             const signInForm = document.querySelector('.form-container.signIn');
372             const signUpForm = document.querySelector('.form-container.signUp');
373             if (signInForm, signUpForm, toggleButtons) {
374                 signInForm.style.display = 'flex';
375                 signUpForm.style.display = 'flex';
376                 toggleButtons.style.display = 'block';
377             }
378         });
379     }
380 }
381
382 function showFeedbackModal() {
383     const modal = document.getElementById('feedback-modal');
384     if (modal) {
385         modal.style.display = 'block';
386     }
387 }
388
389 function closeFeedbackModal() {
390     const modal = document.getElementById('feedback-modal');
391     if (modal) {
392         modal.style.display = 'none';
393     }
394 }
395
396 function setupFeedbackForm() {
397     document.getElementById('feedback-form')?.addEventListener('submit', function(e) {
398         e.preventDefault();
399
400         const rating = document.querySelector('input[name="rating"]':checked)? .value;
401         const comment = document.getElementById('feedback-comment').value;
402
403         console.log('Feedback enviado:', { rating, comment });
404
405         try {
406             const feedbacks = JSON.parse(localStorage.getItem('formFeedbacks') || '[]');
407             feedbacks.push({ rating, comment, date: new Date().toISOString() });
408             localStorage.setItem('formFeedbacks', JSON.stringify(feedbacks));
409         } catch (error) {
410             console.error('Erro ao salvar feedback:', error);
411         }
412
413         closeFeedbackModal();
414     });
415 }
416
417
418 function setupIdadeInput() {
419     if (idadeInput) {
420         idadeInput.addEventListener('input', function(e) {
421             this.value = this.value.replace(/\D/g, '');
422             if (this.value.length > 3) {
423                 this.value = this.value.slice(0, 3);
424             }
425         });
426     }
427
428     idadeInput.addEventListener('blur', function() {
429         const idade = parseInt(this.value) || 0;
430         const errorSpan = this.nextElementSibling;
431
432         if (!this.value) {
433             errorSpan.textContent = 'Este campo é obrigatório';
434             errorSpan.style.display = 'block';
435             this.style.borderColor = '#ff3860';
436         } else if (idade < 16) {
437             errorSpan.textContent = 'Idade mínima: 16 anos';
438             errorSpan.style.display = 'block';
439             this.style.borderColor = '#ff3860';
440         } else if (idade > 120) {
441             errorSpan.textContent = 'Idade máxima: 120 anos';
442             errorSpan.style.display = 'block';
443             this.style.borderColor = '#ff3860';
444         } else {
445             errorSpan.style.display = 'none';
446             this.style.borderColor = '#09c372';
447         }
448     });
449 }
450
451
452 function setupCPFInput() {
453     if (cpfInput) {
454         cpfInput.addEventListener('input', function(e) {
455             this.value = this.value.replace(/\D/g, '');
456             if (this.value.length > 11) {
457                 this.value = this.value.substring(0, 11);
458             }
459             let value = e.target.value.replace(/\D/g, '');
460             value = value.replace('/(\d{3})\(\d{2}\)', '$1.$2');
461             value = value.replace('/(\d{3})\(\d{2}\)\(\d{3}\)', '$1.$2.$3');
462             value = value.replace('/(\d{1,2})$/', '$1-$2');
463             e.target.value = value;
464         });
465     }
466 }
```

Figura 23: Código JavaScript do Sistema (parte 5).

Documentação - Sistema Interativo de Inscrição - Trilha Back-End



The screenshot shows a Visual Studio Code interface with two tabs open: 'script.js' and 'script.js'. The left tab contains code for validating CPF and TEL inputs. The right tab contains code for validating CEP and setting up document upload functionality. Both tabs include validation logic using regular expressions and event listeners.

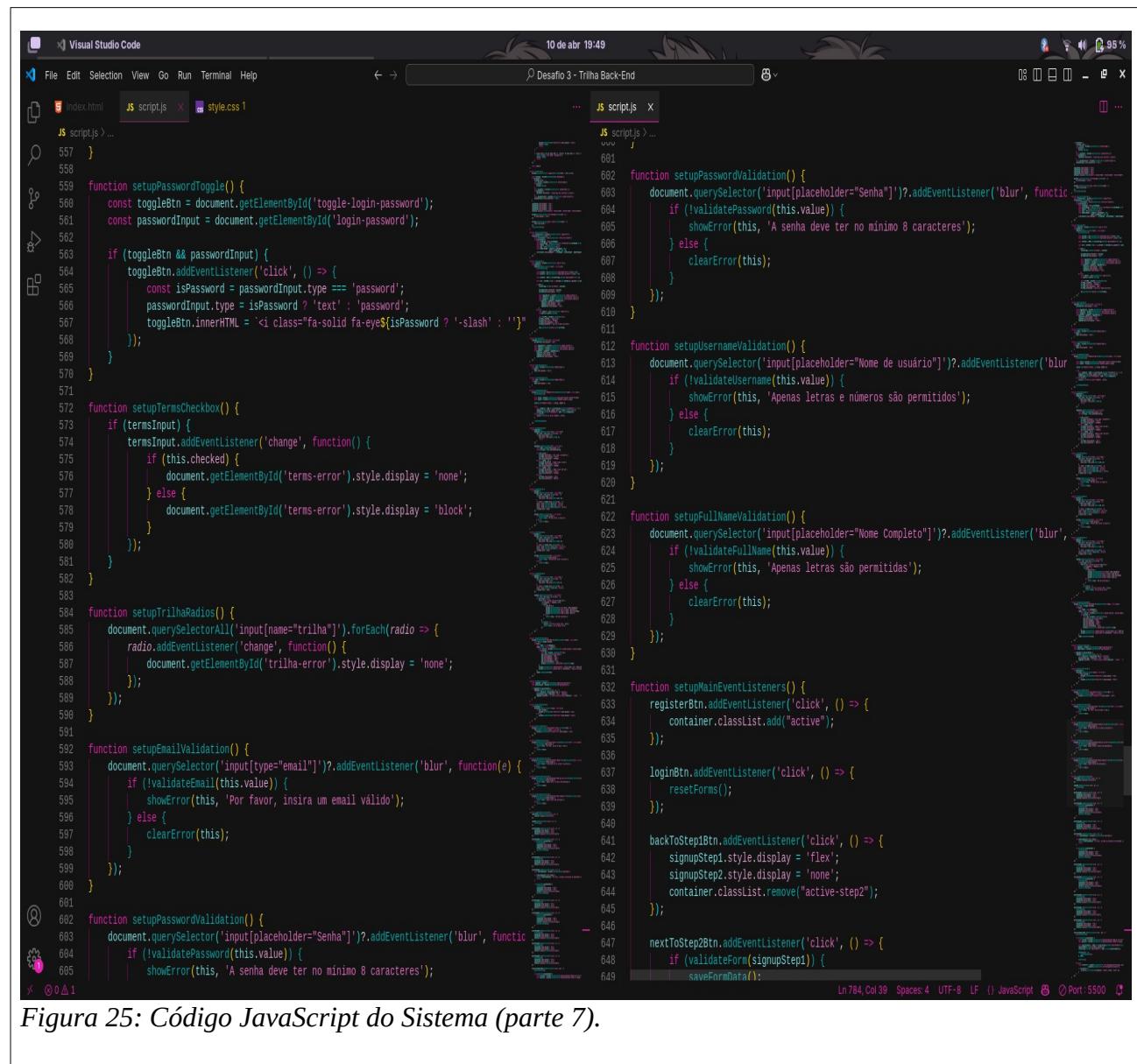
```

    // Left Tab (script.js)
    452 function setupCPFInput() {
    453   cpfInput.addEventListener('blur', function() {
    454     if (!validateCPF(this.value)) {
    455       showError(this, 'CPF deve conter 11 digitos');
    456     } else {
    457       clearError(this);
    458     }
    459   });
    460 }
    461
    462 function setupTelInput() {
    463   if (telInput) {
    464     telInput.addEventListener('input', function(e) {
    465       this.value = this.value.replace(/\D/g, '');
    466       if (this.value.length > 11) {
    467         this.value = this.value.substring(0, 11);
    468       }
    469       let value = e.target.value.replace(/\D/g, '');
    470       value = value.replace(/(\d{2})(\d)/g, '($1)$2');
    471       value = value.replace(/(\d{4})$/ , '$1-$2');
    472       e.target.value = value;
    473     });
    474   }
    475 }
    476
    477 function setupCEPInput() {
    478   if (cepInput) {
    479     cepInput.addEventListener('input', function(e) {
    480       this.value = this.value.replace(/\D/g, '');
    481       if (this.value.length > 8) {
    482         this.value = this.value.substring(0, 8);
    483       }
    484       let value = e.target.value.replace(/\D/g, '');
    485       value = value.replace(/(\d{5})(\d)/g, '$1-$2');
    486       e.target.value = value;
    487     });
    488   }
    489 }

    // Right Tab (script.js)
    499 function setupCEPInput() {
    500   cepInput.addEventListener('blur', function() {
    501     const cep = this.value.replace(/\D/g, '');
    502     if (cep.length === 8) {
    503       fetch(`https://viacep.com.br/ws/${cep}/json/`)
    504         .then(response => response.json())
    505         .then(data => {
    506           if (!data.erro) {
    507             document.getElementById('rua').value = data.logradouro;
    508             document.getElementById('bairro').value = data.bairro;
    509             document.getElementById('cidade').value = data.localidade;
    510             document.getElementById('estado').value = data.uf;
    511           } else {
    512             showError(this, 'CEP não encontrado');
    513           }
    514         })
    515         .catch(error => {
    516           console.error('Erro ao buscar CEP:', error);
    517           showError(this, 'Erro ao buscar CEP');
    518         });
    519     });
    520   }
    521
    522
    523
    524
    525
    526
    527
    528
    529
    530
    531
    532
    533
    534
    535
    536
    537
    538
    539
    540
    541
    542
    543
    544
    545
    546
    547
    548
    549
    550
    551
    552
    553
    554
    555
    556
  
```

Figura 24: Código JavaScript do Sistema (parte 6).

Documentação - Sistema Interativo de Inscrição - Trilha Back-End



```
10 de abr 19:49
Desafio 3 - Trilha Back-End
File Edit Selection View Go Run Terminal Help
index.html JS script.js style.css 1
JS script.js > ...
557 }
558
559 function setupPasswordToggle() {
  const toggleBtn = document.getElementById('toggle-login-password');
  const passwordInput = document.getElementById('login-password');

  if (toggleBtn && passwordInput) {
    toggleBtn.addEventListener('click', () => {
      const isPassword = passwordInput.type === 'password';
      passwordInput.type = isPassword ? 'text' : 'password';
      toggleBtn.innerHTML = <i class="fa-solid fa-eye${isPassword ? '-slash' : ''}">
    });
  }
}
571
572 function setupTermsCheckbox() {
  if (termsInput) {
    termsInput.addEventListener('change', function() {
      if (this.checked) {
        document.getElementById('terms-error').style.display = 'none';
      } else {
        document.getElementById('terms-error').style.display = 'block';
      }
    });
  }
}
582
583 function setupTrilhaRadios() {
  document.querySelectorAll('input[name="trilha"]').forEach(radio => {
    radio.addEventListener('change', function() {
      document.getElementById('trilha-error').style.display = 'none';
    });
  });
}
591
592 function setupEmailValidation() {
  document.querySelector('input[type="email"]')?.addEventListener('blur', function(e) {
    if (!validateEmail(this.value)) {
      showError(this, 'Por favor, insira um email válido');
    } else {
      clearError(this);
    }
  });
}
599
600
601 function setupPasswordValidation() {
  document.querySelector('input[placeholder="Senha"]')?.addEventListener('blur', function() {
    if (!validatePassword(this.value)) {
      showError(this, 'A senha deve ter no mínimo 8 caracteres');
    }
  });
}
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
```

Figura 25: Código JavaScript do Sistema (parte 7).

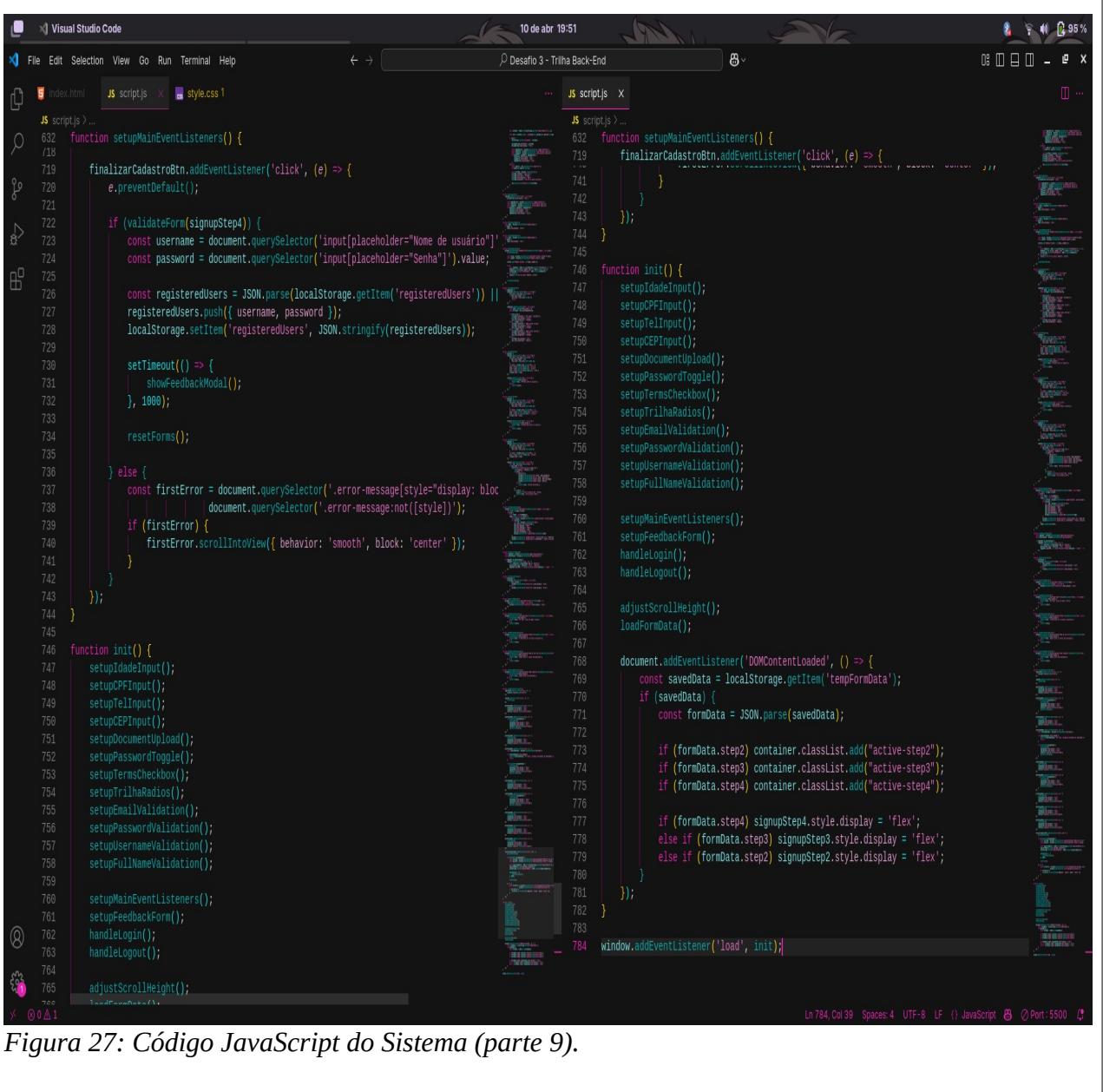
Documentação - Sistema Interativo de Inscrição - Trilha Back-End

```
10 de abr 19:50
Desafio 3 - Trilha Back-End
File Edit Selection View Go Run Terminal Help
index.html script.js style.css
...
JS script.js ...
632 function setupMainEventListeners() {
646
647 nextToStep2Btn.addEventListener('click', () => {
648 if (validateForm(signupStep1)) {
649 saveFormData();
650 signupStep1.style.display = 'none';
651 signupStep2.style.display = 'flex';
652 container.classList.add("active-step2");
653 adjustScrollHeight();
654 }
655 });
656
657 prevToStep1Btn.addEventListener('click', () => {
658 saveFormData();
659 signupStep1.style.display = 'flex';
660 signupStep2.style.display = 'none';
661 container.classList.remove("active-step2");
662 });
663
664 nextToStep3Btn.addEventListener('click', () => {
665 const identidadeInput = document.getElementById('identidade');
666
667 if (!identidadeInput.files.length) {
668 showError(identidadeInput, 'Por favor, carregue um documento de identidade.');
669 return;
670 }
671
672 if (validateForm(signupStep2)) {
673 saveFormData();
674 signupStep2.style.display = 'none';
675 signupStep3.style.display = 'flex';
676 container.classList.add("active-step3");
677 adjustScrollHeight();
678 }
679 });
680
681 prevToStep2Btn.addEventListener('click', () => {
682 saveFormData();
683 signupStep2.style.display = 'flex';
684 signupStep3.style.display = 'none';
685 container.classList.remove("active-step3");
686 });
687
688 nextToStep4Btn.addEventListener('click', () => {
689 if (validateForm(signupStep3)) {
690 saveFormData();
691 signupStep3.style.display = 'none';
692 signupStep4.style.display = 'flex';
693 container.classList.add("active-step4");
694 }
695 });
696
697 prevToStep3Btn.addEventListener('click', () => {
698 saveFormData();
699 signupStep3.style.display = 'flex';
700 signupStep4.style.display = 'none';
701 container.classList.remove("active-step4");
702 });
703
704 backToStep2Btn.addEventListener('click', () => {
705 saveFormData();
706 signupStep2.style.display = 'flex';
707 signupStep3.style.display = 'none';
708 container.classList.remove("active-step3");
709 });
710
711 backToStep3Btn.addEventListener('click', () => {
712 saveFormData();
713 signupStep3.style.display = 'flex';
714 signupStep4.style.display = 'none';
715 container.classList.remove("active-step4");
716 });
717
718 finalizarCadastroBtn.addEventListener('click', (e) => {
719 e.preventDefault();
720
721 if (validateForm(signupStep4)) {
722 const username = document.querySelector('input[placeholder="Nome de usuário"]');
723 const password = document.querySelector('input[placeholder="Senha"]').value;
724
725 const registeredUsers = JSON.parse(localStorage.getItem('registeredUsers'));
726 registeredUsers.push({username, password});
727 localStorage.setItem('registeredUsers', JSON.stringify(registeredUsers));
728
729 setTimeout(() => {
730 showFeedbackModal();
731 }, 1000);
732
733 resetForms();
734 }}
```

Ln 784, Col 39 Spaces: 4 UTF-8 () JavaScript 0 Port: 5500

Figura 26: Código JavaScript do Sistema (parte 8).

Documentação - Sistema Interativo de Inscrição - Trilha Back-End



```
10 de abr 19:51
Desafio 3 - Trilha Back-End
File Edit Selection View Go Run Terminal Help
index.html JS script.js style.css
JS script.js > ...
632 function setupMainEventListeners() {
633     finalizarCadastroBtn.addEventListener('click', (e) => {
634         e.preventDefault();
635
636         if (validateForm(signupStep4)) {
637             const username = document.querySelector('input[placeholder="Nome de usuário"]');
638             const password = document.querySelector('input[placeholder="Senha"]').value;
639
640             const registeredUsers = JSON.parse(localStorage.getItem('registeredUsers')) || [];
641             registeredUsers.push({username, password});
642             localStorage.setItem('registeredUsers', JSON.stringify(registeredUsers));
643
644             setTimeout(() => {
645                 showFeedbackModal();
646             }, 1000);
647
648             resetForms();
649
650         } else {
651             const firstError = document.querySelector('.error-message[style="display: block"]');
652             if (firstError) {
653                 firstError.scrollIntoView({behavior: 'smooth', block: 'center'});
654             }
655         }
656     });
657 }
658
659 function init() {
660     setupIdadeInput();
661     setupCPFInput();
662     setupTelInput();
663     setupCEPInput();
664     setupDocumentUpload();
665     setupPasswordToggle();
666     setupTermsCheckbox();
667     setupTrilhaRadios();
668     setupEmailValidation();
669     setupPasswordValidation();
670     setupUsernameValidation();
671     setupFullNameValidation();
672
673     setupMainEventListeners();
674     setupFeedbackForm();
675     handleLogin();
676     handleLogout();
677
678     adjustScrollHeight();
679
680     window.addEventListener('load', init);
681 }
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
```

Figura 27: Código JavaScript do Sistema (parte 9).

Documentação - Sistema Interativo de Inscrição - Trilha Back-End

4.2. Anexo das Páginas Web: Formulário



Figura 28: Página Web do Sistema (parte 1).

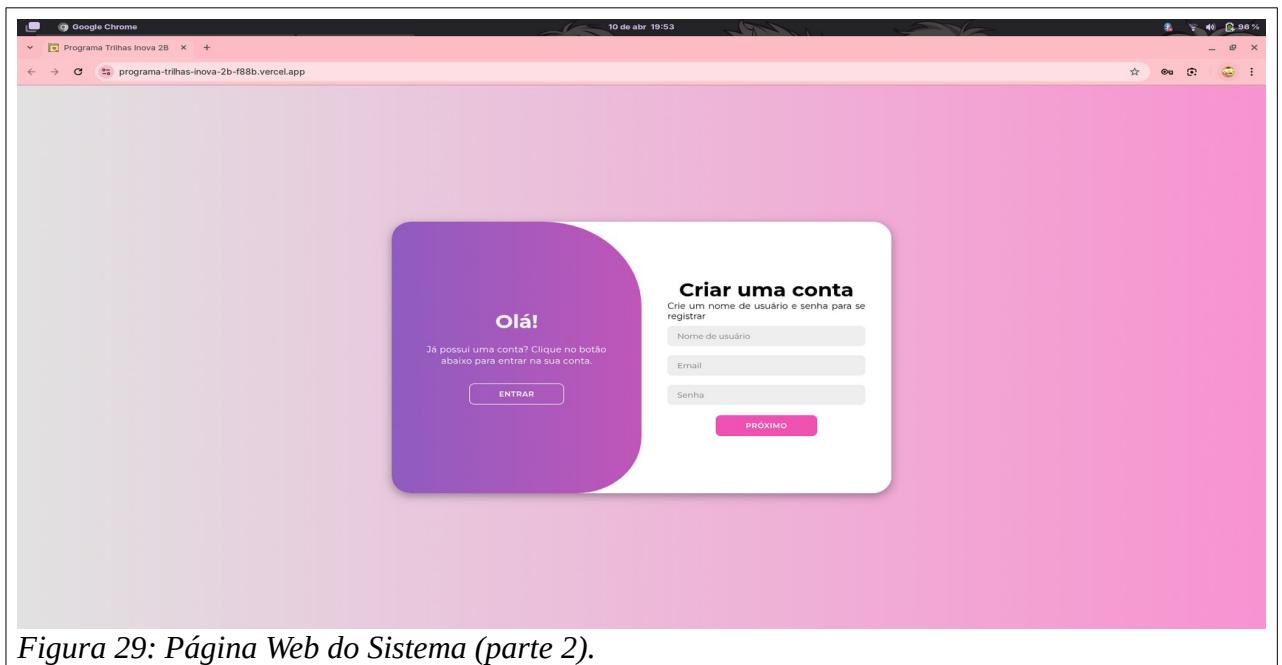


Figura 29: Página Web do Sistema (parte 2).