

Documentação do Sistema de Login e Validação de Usuário

**Ester Lopes
Maria Isabel
Gabriel Santos**

**Rio de Janeiro
2024**

SUMÁRIO

1. Introdução.....	3
2. Estrutura do Sistema.....	3
3. Código HTML - Página de Login.....	3
4. Código CSS - Estilização da Página.....	4
5. Fluxograma.....	11
6. Algoritmo em Portugol.....	12
7. Conclusão.....	13

1. INTRODUÇÃO

A documentação descreve a implementação de um sistema de login baseado em HTML, CSS e PHP, com foco na autenticação de usuários. O sistema verifica as credenciais de login de um usuário comparando o e-mail e a senha fornecidos pelo usuário com um conjunto de e-mails e senhas armazenados em um array no servidor.

2. ESTRUTURA DO SISTEMA

- HTML: Responsável pela estrutura da página de login e exibição de formulários.
- CSS: Define o estilo visual da página.
- PHP: Realiza a verificação de credenciais e gera mensagens de sucesso ou erro com base no resultado.

3. CÓDIGO HTML - PÁGINA DE LOGIN

O arquivo HTML contém a estrutura básica do formulário de login e algumas configurações de estilo e metadados. Abaixo, detalhamos as seções do código.

```

<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tela de Login</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="pagina">
    <form method="POST" action="usuarios.php" class="formLogin">
      <h1>Login</h1>
      <p>Digite seus dados nos campos abaixo para acessar.</p>
      <label for="email">Seu E-mail</label>
      <input type="email" placeholder="Digite seu e-mail" autofocus="true" name="email" id="email" required>
      <label for="password">Sua Senha</label>
      <input type="password" placeholder="Digite sua senha" name="senha" id="senha" required>
      <input type="submit" value="Entrar" class="btn" />
    </div>
  </form>
</body>
</html>

```

- **<form method="POST" action="usuarios.php" class="formLogin">**: O formulário é enviado para o arquivo PHP **usuarios.php** utilizando o método POST, passando as variáveis **email** e **senha** para o processamento.
- **<input type="email">**: Campo de entrada para o e-mail do usuário. Atributo **required** indica que o campo é obrigatório.
- **<input type="password">**: Campo de entrada para a senha do usuário.
- **<input type="submit">**: Botão de envio do formulário.

4. CÓDIGO CSS - ESTILIZAÇÃO DA PÁGINA



O CSS é responsável por fornecer o estilo visual da página, como o layout, cores e efeitos de interatividade. Abaixo, detalhamos as principais regras de estilo.

```

@font-face {
  font-family: 'Coolvetica';
  src: url(coolvetica\ rg.otf);
}

```

- **@font-face:** Esta regra permite importar uma fonte personalizada que será utilizada no estilo da página. Aqui, ela está declarando que a fonte **Coolvetica** será usada na página.
- **font-family: 'Coolvetica';** Este é o nome que será usado para aplicar a fonte em outros elementos da página.
- **src: url(coolvetica\ rg.otf);** Define a localização do arquivo da fonte (**coolvetica rg.otf**). Este arquivo de fonte deve estar disponível no diretório especificado para ser carregado corretamente.

```
body {  
  margin: 0;  
  padding: 0;  
  color:  rgb(67, 107, 194);  
  background-color:  pink;  
}
```

- **margin: 0; padding: 0;** Remove a margem e o preenchimento padrão do corpo da página, garantindo que o conteúdo ocupe toda a área da tela.
- **color: rgb(67, 107, 194);** Define a cor do texto do corpo da página como um tom de azul (RGB: 67, 107, 194).
- **background-color: pink;** Define a cor de fundo da página como rosa.

```
.pagina {  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  align-content: center;  
  justify-content: center;  
  width: 100%;  
  height: 100vh;  
}
```

- **display: flex;** Define que o layout da **.pagina** será baseado no modelo **Flexbox**, que facilita o alinhamento de elementos.
- **flex-direction: column;** Organiza os itens dentro do contêiner de forma vertical.
- **align-items: center; align-content: center;** Alinha os itens no eixo horizontal (centrado).
- **justify-content: center;** Alinha os itens no eixo vertical (centrado).
- **width: 100%; height: 100vh;** Define a largura da página como 100% e a altura como 100% da altura da janela de visualização (viewport).

```
.formLogin {
  font-family: 'Coolvetica';
  display: flex;
  flex-direction: column;
  background-color: #fff;
  border-radius: 7px;
  padding: 40px;
  box-shadow: 10px 10px 40px rgba(0, 0, 0, 0.4);
  gap: 5px;
}
```

- **font-family: 'Coolvetica';** Aplica a fonte **Coolvetica** ao formulário.
- **display: flex; flex-direction: column;** Utiliza o layout Flexbox e organiza os itens do formulário verticalmente.
- **background-color: #fff;** Define o fundo do formulário como branco.
- **border-radius: 7px;** Aplica bordas arredondadas ao formulário com raio de 7px.
- **padding: 40px;** Adiciona 40px de espaço interno ao redor do conteúdo do formulário.
- **box-shadow: 10px 10px 40px rgba(0, 0, 0, 0.4);** Aplica uma sombra ao redor do formulário para dar um efeito de profundidade, com deslocamento de 10px nas direções X e Y e um desfoque de 40px.
- **gap: 5px;** Define um espaçamento de 5px entre os elementos dentro do formulário.

```
.areaLogin img {  
  width: 420px;  
}
```

- **width: 420px;**: Define a largura da imagem na área de login como 420px.

```
.formLogin h1 {  
  padding: 0;  
  margin: 0;  
  font-weight: 500;  
  font-size: 2.3em;  
}
```

- **padding: 0; margin: 0;**: Remove o preenchimento e a margem padrão do título (<h1>).
- **font-weight: 500;**: Define a espessura do texto como 500 (valor médio).
- **font-size: 2.3em;**: Define o tamanho da fonte como 2.3 vezes o tamanho da fonte padrão.

```
.formLogin p {  
  display: inline-block;  
  font-size: 14px;  
  color: #555;  
  margin-bottom: 25px;  
}
```

- **display: inline-block;**: Faz com que o parágrafo se comporte como um bloco, mas ainda permita que ele se comporte de forma semelhante a um elemento inline.
- **font-size: 14px;**: Define o tamanho da fonte para 14px.
- **color: #555;**: Define a cor do texto como um cinza escuro.

- **margin-bottom: 25px;**: Adiciona um espaço de 25px abaixo do parágrafo.

```
.formLogin input {  
  padding: 15px;  
  font-size: 14px;  
  border: 1px solid #ccc;  
  margin-bottom: 20px;  
  margin-top: 5px;  
  border-radius: 4px;  
  transition: all linear 160ms;  
  outline: none;  
}
```

- **padding: 15px;**: Adiciona 15px de preenchimento dentro dos campos de entrada.
- **font-size: 14px;**: Define o tamanho da fonte dentro dos campos de entrada para 14px.
- **border: 1px solid #ccc;**: Define a borda do campo com 1px de espessura e cor cinza claro.
- **margin-bottom: 20px; margin-top: 5px;**: Adiciona um espaço de 20px abaixo do campo e 5px acima.
- **border-radius: 4px;**: Aplica bordas arredondadas aos campos de entrada.
- **transition: all linear 160ms;**: Aplica uma transição suave de 160ms para todas as alterações de estilo.
- **outline: none;**: Remove a borda azul padrão que aparece ao clicar no campo de entrada (em alguns navegadores).

```
.formLogin input:focus {  
  border: 1px solid pink;  
}
```

- **border: 1px solid pink;**: Quando um campo de entrada é focado (clicado ou selecionado), a borda muda para a cor rosa.


```
.formLogin label {  
    font-size: 14px;  
    font-weight: 600;  
}
```

- **font-size: 14px;**: Define o tamanho da fonte do rótulo para 14px.
- **font-weight: 600;**: Define o peso da fonte para 600, deixando o texto mais grosso.

```
.btn {  
    background-color: rgb(224, 141, 155);  
    color: #fff;  
    font-size: 14px;  
    font-weight: 600;  
    border: none !important;  
    transition: all linear 160ms;  
    cursor: pointer;  
    margin: 0 !important;  
}
```

- **background-color: rgb(224, 141, 155);**: Define a cor de fundo do botão como um tom de rosa claro.
- **color: #fff;**: Define a cor do texto do botão como branco.
- **font-size: 14px;**: Define o tamanho da fonte do botão para 14px.
- **font-weight: 600;**: Define o peso da fonte do botão como 600.
- **border: none !important;**: Remove qualquer borda padrão do botão.
- **cursor: pointer;**: Define que o cursor se transformará em uma mãozinha quando passar sobre o botão.
- **transition: all linear 160ms;**: Aplica uma transição suave de 160ms para todas as alterações de estilo.
- **margin: 0 !important;**: Remove qualquer margem padrão do botão.

```
.btn:hover {  
    transform: scale(1.05);  
    background-color: rgb(224, 141, 155);  
}
```


- **transformer: scale(1.05);**: Amplia o botão em 5% durante o estado de hover.
- **background-color: rgb(224, 141, 155);**: Mantém a cor de fundo do botão durante o estado de hover.

```
.mensagem-container {  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
  align-items: center;  
  height: 100vh;  
  text-align: center;  
  margin: 0;  
}
```

- **display: flex; flex-direction: column;**: Define o layout como Flexbox e organiza os elementos verticalmente no contêiner.
- **justify-content: center; align-items: center;**: Centraliza os itens tanto horizontal quanto verticalmente dentro do contêiner.
- **height: 100vh;**: Define a altura do contêiner como 100% da altura da janela de visualização.
- **text-align: center;**: Centraliza o texto horizontalmente no contêiner.
- **margin: 0;**: Remove qualquer margem padrão do contêiner.

```
.mensagem {  
  font-family: 'Coolvetica';  
  font-size: 24px;  
  text-align: center;  
  margin-top: 0;  
}
```

- **font-family: 'Coolvetica';** Define a fonte da mensagem como
- **font-size: 24px;** Define o tamanho do texto da mensagem para 24px.
- **text-align: center;** Centraliza o texto horizontalmente.
- **margin-top: 0;** Remove qualquer margem superior padrão da mensagem.

```
.sucesso {  
  color:  #7ac567;  
}
```

- **color: #7ac567;** Define a cor do texto para um tom de verde, indicando sucesso.

```
.erro {  
  color:  #ff474c;  
}
```

- **color: #ff474c;** Define a cor do texto para um tom de vermelho, indicando erro.

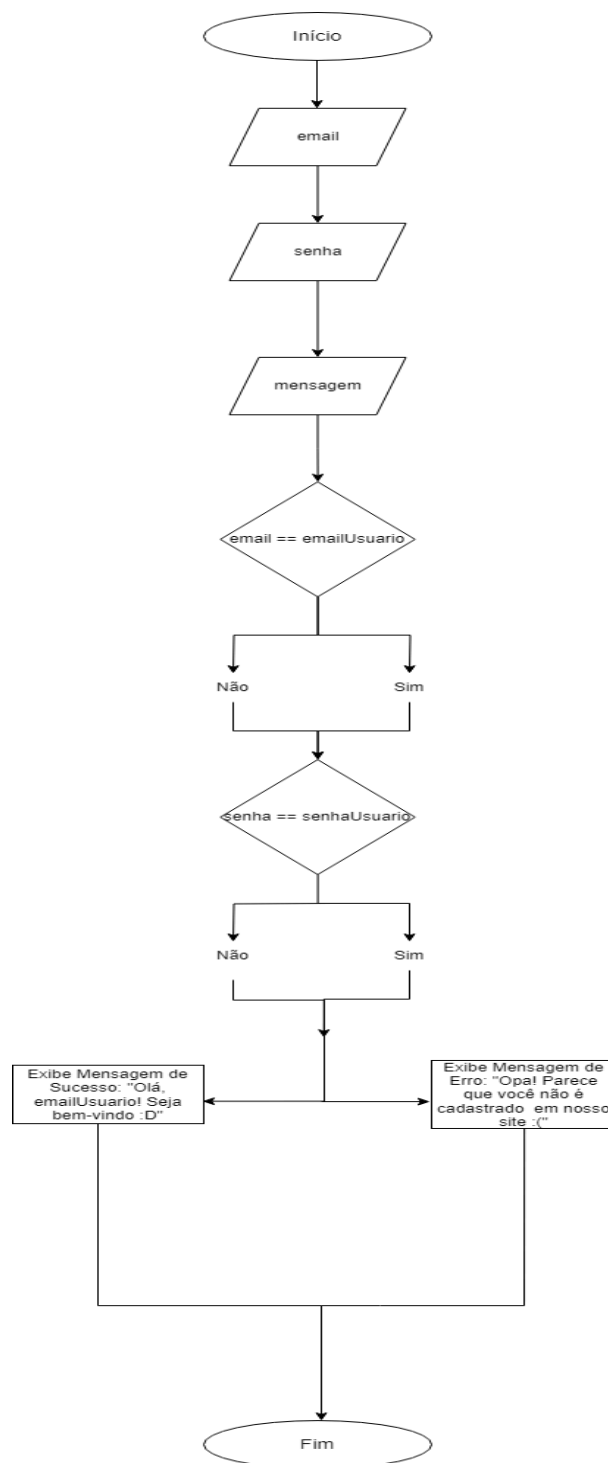
```
.imagem-bemvindo {  
  width: 200px;  
  height: auto;  
  margin-bottom: 20px;  
}
```

- **width: 200px;** Define a largura da imagem como 200px.
- **height: auto;** Mantém a proporção da imagem ao ajustá-la.
- **margin-bottom: 20px;** Adiciona um espaço de 20px abaixo da imagem.

```
.imagem-erro {  
  width: 200px;  
  height: auto;  
  margin-bottom: 20px;  
}
```

- **width: 200px;** Define a largura da imagem como 200px (igual à de boas-vindas).
- **height: auto;** Mantém a proporção da imagem ao ajustá-la.
- **margin-bottom: 20px;** Adiciona um espaço de 20px abaixo da imagem.

5. FLUXOGRAMA



6. ALGORITMO EM PORTUGOL

caractere: emailUsuario, senhaUsuario;

caractere: contas[10][2];

inteiro: usuarioEncontrado, i;

contas[1][1] <- "usuario1@gmail.com"

contas[1][2] <- "senha1"

contas[2][1] <- "usuario2@gmail.com"

contas[2][2] <- "senha2"

contas[3][1] <- "usuario3@gmail.com"

contas[3][2] <- "senha3"

contas[4][1] <- "usuario4@gmail.com"

contas[4][2] <- "senha4"

contas[5][1] <- "usuario5@gmail.com"

contas[5][2] <- "senha5"

contas[6][1] <- "usuario6@gmail.com"

contas[6][2] <- "senha6"

contas[7][1] <- "usuario7@gmail.com"

contas[7][2] <- "senha7"

```
contas[8][1] <- "usuario8@gmail.com"
```

```
contas[8][2] <- "senha8"
```

```
contas[9][1] <- "usuario9@gmail.com"
```

```
contas[9][2] <- "senha9"
```

```
contas[10][1] <- "usuario10@gmail.com"
```

```
contas[10][2] <- "senha10"
```

```
escreva("Digite seu e-mail: ")
```

```
leia(emailUsuario)
```

```
escreva("Digite sua senha: ")
```

```
leia(senhaUsuario)
```

```
usuarioEncontrado <- 0
```

```
para i de 1 até 10 passo 1 faça
```

```
se contas[i][1] = emailUsuario e contas[i][2] = senhaUsuario então
```

```
    escreva("Olá, ", emailUsuario, "! Seja bem-vindo :D")
```

```
    usuarioEncontrado <- 1
```

```
pare
```

fim-se;

fim para.

se usuarioEncontrado = 0 então

 escreva("Opa! Parece que você não é cadastrado em nosso site
:(")

fim-se;

fim.

7. CONCLUSÃO

O sistema de login apresentado combina HTML, CSS e PHP de maneira simples, mas eficiente, para criar uma experiência funcional e confortável para o usuário. Ele permite que os dados sejam verificados com base em informações pré definidas, garantindo o retorno de mensagens claras, seja de sucesso ou de erro, dependendo do resultado.

O fluxograma ajuda a visualizar o funcionamento do sistema como um todo. Ele descreve passo a passo o processo, desde a entrada dos dados pelo usuário até a validação e exibição das mensagens finais. Essa representação facilita o entendimento e destaca os pontos principais da lógica, como as condições de verificação e o fluxo alternativo em caso de erro.

Já o algoritmo em Portugol traduz esse funcionamento de forma mais clara, servindo como um mapa lógico que orienta a implementação final.