

**ANHANGUERA EDUCACIONAL**  
**SCRN (ASA NORTE) 708/709 - BLOCO B, S/N, ASA NORTE - BRASÍLIA – DF**

**SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE**  
**SISTEMAS**

**AUTOR(ES):**  
**LUCAS LOPES RIBEIRO**

**DESENVOLVIMENTO EM JAVASCRIPT**  
**Construção de Função para Validação de Campo de E-mail em Formulário HTML com**  
**JavaScript na plataforma Playcode**

**BRASÍLIA**  
**2025**  
**Lucas Lopes Ribeiro**

## **DESENVOLVIMENTO EM JAVASCRIPT**

Construção de Função para Validação de Campo de E-mail em Formulário HTML com  
JavaScript na plataforma Playcode

Portifólio de Desenvolvimento em JavaScript apresentado ao(s)  
tutor(es): André Martins e Vinicius Mendes Gomes da Silva, do  
curso de Tecnologia em Análise e Desenvolvimento de Sistemas  
da instituição Anhanguera Educacional.

**BRASÍLIA**

**2025**

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>4</b>
<b>2 DESENVOLVIMENTO.....</b>	<b>5</b>
<b>3 CONCLUSÃO.....</b>	<b>12</b>

## **1 INTRODUÇÃO**

Este trabalho tem como objetivo criar um formulário de validação de e-mail em que o usuário pode interagir com a página, que por sua vez retorna validando o preenchimento.

Foi utilizado um código que combina as linguagens Html, Css e JavaScript e a plataforma utilizada para realização do projeto o PlayCode.

## 2 DESENVOLVIMENTO

Seguindo a orientação do portfólio, o código foi desenvolvido em duas etapas principais. A estrutura Html onde a base do formulário foi construída incluindo um campo de entrada para o e-mail e um botão para a validação. Essa interface foi aprimorada com estilos utilizando CSS.

Logo em seguida, a validação com JavaScript em que uma função foi desenvolvida para verificar o padrão do e-mail inserido, utilizando a expressão (RegEx). Que impede que o formulário seja submetido com dados inválidos e fornece feedback visual ao usuário.

Na estrutura do código Html contém o formulário com um campo de entrada do tipo "email" e um botão. A marcação utiliza atributos como "**required**", garantindo que o preenchimento seja obrigatório. Conforme imagem abaixo:



```
index.html x
94     }
95   </style>
96 </head>
97 <body>
98   <h1 class="title">Atividade do Portfólio de Desenvolvimento em JavaScript</h1>
99   <div class="container">
100     <h2>Formulário de Validação de E-mail</h2>
101     .....<form id="emailForm">
102       .....<input type="email" id="email" placeholder="Digite seu e-mail" required>
103       .....<button type="button" onclick="validateEmail()">Validar</button>
104     .....</form>
105     <p id="message" class="message"></p>
106   </div>
107
108   <script>
```

Web View x

O estilo Css foi aplicado para melhorar a experiência visual do formulário. No trabalho em questão foi utilizado o gradiente no fundo da página (de azul ao branco) e cores de indicação de validação, com verde claro para e-mails válidos e vermelho claro para inválidos. É possível analisar na imagem abaixo.

```

index.html x
55     border-radius: 5px;
56     font-size: 16px;
57     transition: background-color 0.3s ease; /* Transição para alteração de cor */
58 }
59
60 ..input.valid {
61     background-color: #d4edda; /* Verde claro para indicar sucesso */
62 }
63
64 input.invalid {
65     background-color: #f8d7da; /* Vermelho claro para indicar erro */
66 }
67

```

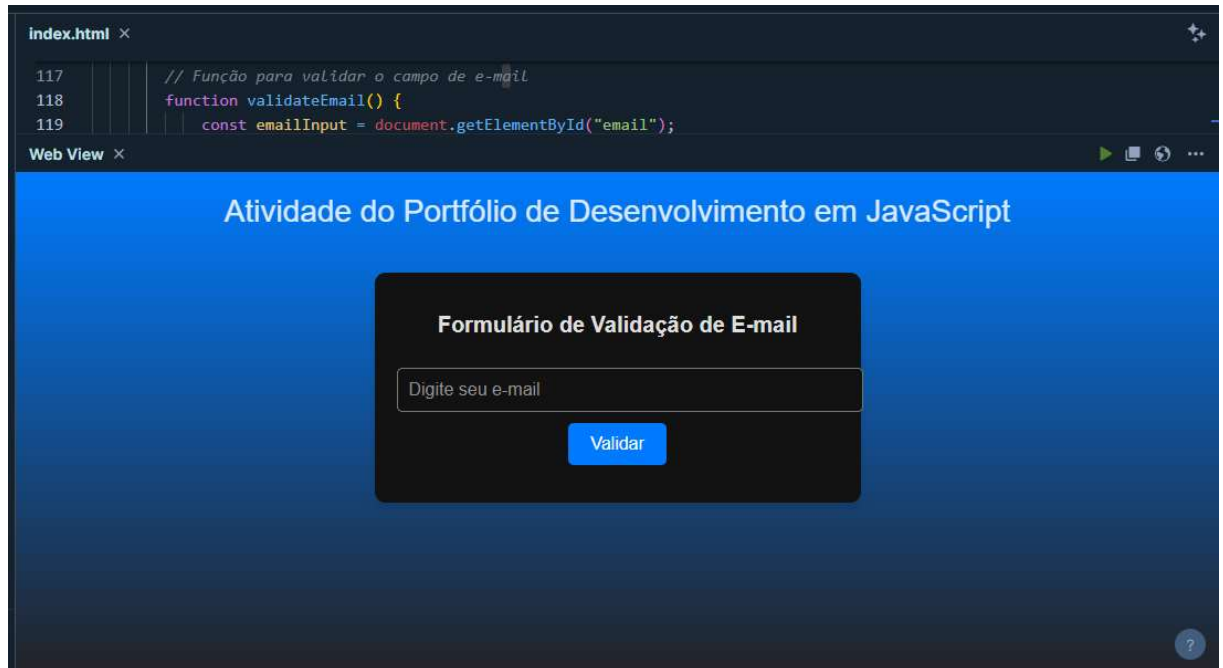
Na parte do código onde o JavaScript entra em questão, a função principal de validação utiliza RegEx para identificar padrões válidos de e-mail. Também foi implementado um listener para prevenir o comportamento padrão do formulário.

```

index.html x
117 // Função para validar o campo de e-mail
118 function validateEmail() {
119     const emailInput = document.getElementById("email");
120     const message = document.getElementById("message");
121     const emailPattern = /^[^ ]+@[^ ]+\.[a-z]{2,3}$/;
122
123     if (emailInput.value.match(emailPattern)) {
124         message.textContent = "E-mail válido!";
125         message.className = "message success";
126         emailInput.classList.add("valid");
127         emailInput.classList.remove("invalid");
128     } else {
129         message.textContent = "E-mail inválido. Certifique-se de que segue o padrão: exemplo@dominio.com";
130         message.className = "message error";
131         emailInput.classList.add("invalid");
132         emailInput.classList.remove("valid");
133     }
134 }
135 </script>
136 </body>

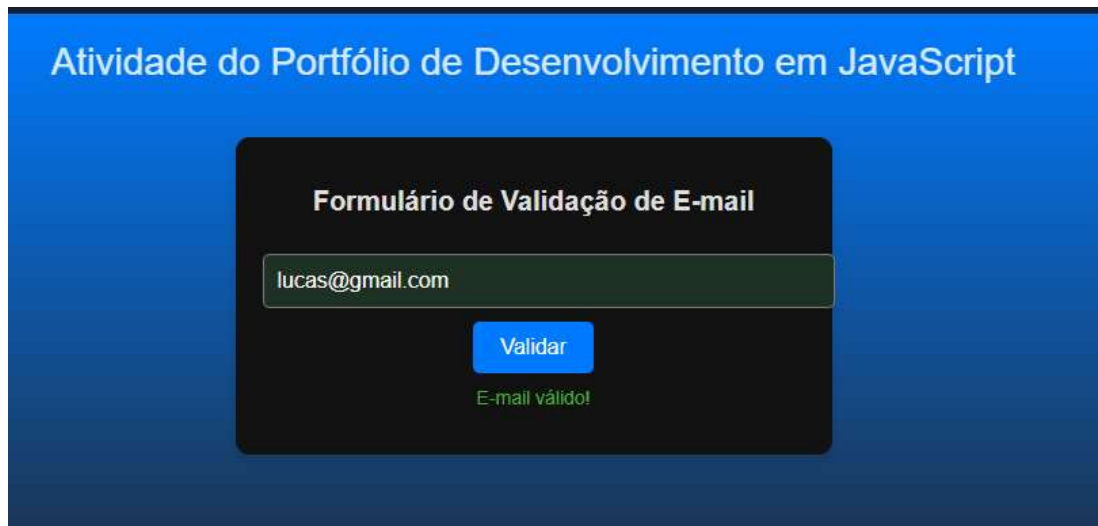
```

Em seguida, o código funciona ao inserir um e-mail no campo e ao clicar pressionando o “enter”, o código verifica o padrão de e-mail então exibe uma das mensagens de validação, retornando se está válido ou não. Durante isso, altera dinamicamente a cor de fundo do campo indicando visualmente o estado da validação.

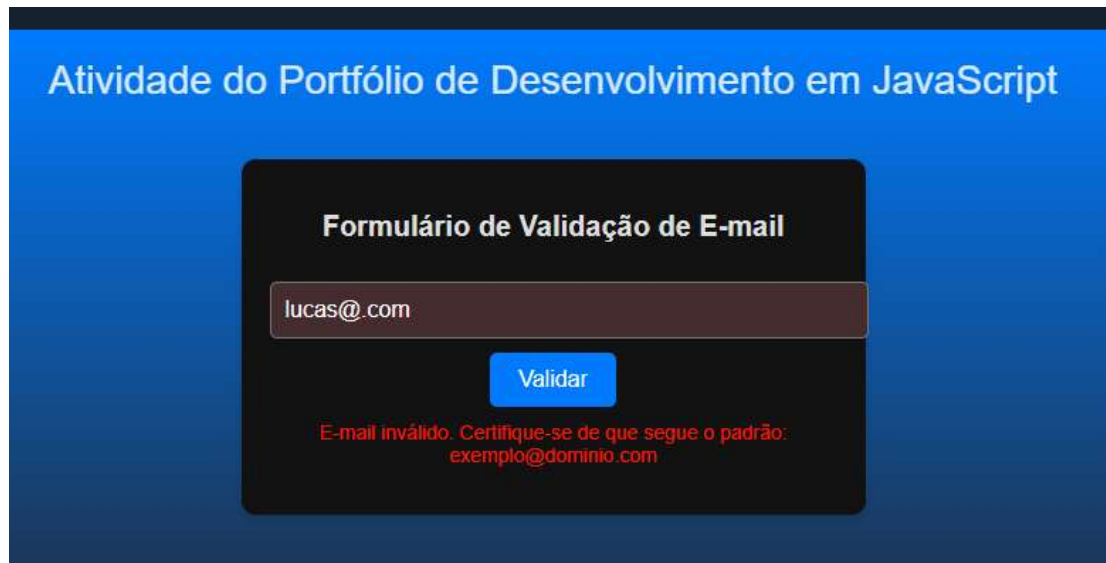


Agora, será demonstrado a interface do editor PlayCode juntamente com o resultado do código, ou seja, o site em funcionamento:

O preenchimento do formulário com um e-mail válido:



O preenchimento do formulário com um e-mail inválido:



Chegando ao fim da conclusão e apresentação onde as três linguagens são combinadas e apresentam bom funcionamento, será apresentado o código que foi desenvolvido para este trabalho:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Formulário de Validação</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background: linear-gradient(to bottom, #007BFF, #f4f4f9);
      color: #333;
      margin: 0;
      padding: 0;
      display: flex;
      flex-direction: column;
      align-items: center;
      height: 100vh; }
  
```



```

h1.title {
    font-size: 28px;
    margin-top: 20px;
    text-align: center;
    color: #ffffff;
    opacity: 0.85;
    font-weight: normal;}
.container {
    background: #fff;
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    text-align: center;
    max-width: 400px;
    width: 100%;
    margin-top: 20px;}
h2 {font-size: 20px;color: #333}
input[type="email"] {
    width: 100%;
    padding: 10px;
    margin: 10px 0;
    border: 1px solid #ccc;
    border-radius: 5px;
    font-size: 16px;
    transition: background-color 0.3s ease; }
input.valid {
    background-color: #d4edda; /* Verde claro para indicar sucesso */
}
input.invalid {
    background-color: #f8d7da; /* Vermelho claro para indicar erro */
}
button {
    background-color: #007BFF;
    color: white;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    font-size: 16px;
}

```

```

    button:hover {
        background-color: #0056b3;
    }
    /* Mensagem de erro ou sucesso */
    .message {
        margin-top: 10px;
        font-size: 14px;
    }
    .error {
        color: red;
    }
    .success {
        color: green;
    }
</style>
</head>
<body>
    <h1 class="title">Atividade do Portfólio de Desenvolvimento em JavaScript</h1>
    <div class="container">
        <h2>Formulário de Validação de E-mail</h2>
        <form id="emailForm">
            <input type="email" id="email" placeholder="Digite seu e-mail" required>
            <button type="button" onclick="validateEmail()">Validar</button>
        </form>
        <p id="message" class="message"></p>
    </div>

    <script>
        // Prevenir o comportamento padrão do formulário
        const form = document.getElementById("emailForm");
        form.addEventListener("submit", (e) => {
            e.preventDefault(); // para evitar que o formulário recarregue a página
            validateEmail(); //função de validação
        });

        // validando o campo de e-mail
        function validateEmail() {
            const emailInput = document.getElementById("email");
            const message = document.getElementById("message");
            const emailPattern = /^[^ ]+@[^ ]+\.[a-z]{2,3}$/;

```

```
        if (emailInput.value.match(emailPattern)) {  
            message.textContent = "E-mail válido!";  
            message.className = "message success";  
            emailInput.classList.add("valid");  
            emailInput.classList.remove("invalid");  
        } else {  
            message.textContent = "E-mail inválido. Certifique-se de que segue o  
padrão: exemplo@dominio.com";  
            message.className = "message error";  
            emailInput.classList.add("invalid");  
            emailInput.classList.remove("valid");  
        }  
    }  
</script>  
</body>  
</html>
```

### 3 CONCLUSÃO

O projeto demonstrou como construir uma função simples para validação de campo de e-mail em um formulário HTML, utilizando JavaScript. Durante o desenvolvimento, foram aplicadas técnicas de estruturação visual com CSS e implementação de lógica de validação com expressões regulares.

Por fim neste trabalho foi desenvolvida a interface funcional e responsiva do formulário e validação do preenchimento, entrega de feedback ao usuário por meio de mensagens e mudanças visuais e a compreensão da integração entre HTML, CSS e JavaScript.