

# **UC: Projetos Assunto: Small Projects – 33 a 40**

# **UC: Projetos**

## **Assunto:**

### **Small Projects**

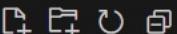
### **P33 – Quiz**



EXPLORER

...

P33 - QUIZ



estilos.css

quiz.html

JS quizjs



&gt; OUTLINE

&gt; TIMELINE



⊗ 0 △ 0 ⌂ 0

5 quiz.html X

5 quiz.html &gt; ...

```
1  <!DOCTYPE html>
2  <!-- A declaração DOCTYPE informa ao navegador a versão do
3      HTML que a página está usando. Neste caso, especifica
4      HTML5, a versão mais recente. -->
5
6  <html lang="pt">
7  <!-- A tag <html> é a raiz do documento HTML. O atributo 'lang="pt"' 
8      especifica que o idioma principal do documento é o português. -->
9
10 <head>
11 <!-- A tag <head> contém metadados (informações sobre dados) que não
12     são exibidos diretamente na página web, mas são usados pelo navegador. -->
13
14 <meta charset="UTF-8">
15 <!-- A tag <meta> com o atributo charset="UTF-8" especifica a codificação
16     de caracteres usada no documento, que é UTF-8. Esta codificação inclui a
17     maioria dos caracteres de todos os idiomas do mundo, garantindo que o
18     texto seja exibido corretamente. -->
19
20 <meta name="viewport" content="width=device-width, initial-scale=1.0">
21 <!-- Outra tag <meta>, desta vez configurando a viewport (área de
22     visualização) da página.
23     'width=device-width' faz a largura da página igualar a largura do
24     dispositivo (útil para dispositivos móveis), e 'initial-scale=1.0'
25     define o nível de zoom inicial quando a página é carregada. -->
26
27 <title>Quiz sobre Animais</title>
28 <!-- A tag <title> define o título da página, que é mostrado na aba do
29     navegador. Este título é também usado por mecanismos de busca e
30     quando a página é marcada como favorita. -->
```

EXPLORER ...

P33 - QUIZ

- estilos.css
- quiz.html
- JS quizjs

quiz.html

quiz.html > html > head

```
31 <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;500&display=swap" rel="stylesheet">
32     <!-- A tag &lt;link&gt; é usada aqui para incluir uma folha de estilo externa.
33         Esta particularmente importa a fonte 'Roboto' do Google Fonts, nas
34         espessuras 400 e 500. 'display=swap' faz com que o texto seja exibido
35         com uma fonte de reserva até que 'Roboto' seja baixada, o que melhora o
36         desempenho de carregamento percebido. --&gt;
37
38
39
40     &lt;link rel="stylesheet" href="estilos.css"&gt;
41     <!-- Link para uma folha de estilo externa chamada 'estilos.css'.
42         Este arquivo contém as regras de estilo CSS para a página. --&gt;
43
44 &lt;/head&gt;
45
46 &lt;body&gt;
47
48     &lt;div class="container"&gt;
49         <!-- Inicia uma &lt;div&gt; com a classe 'container'. Este é o elemento
50             contêiner principal que agrupa todos os componentes do quiz.
51             O estilo da classe 'container' geralmente inclui formatação para
52             centralizar, adicionar sombras, e definir largura e margens
53             para o conteúdo principal. --&gt;
54
55     &lt;div id="pergunta" class="pergunta"&gt;&lt;/div&gt;
56     <!-- Elemento &lt;div&gt; com ID 'pergunta' e classe 'pergunta', usado para
57             exibir a pergunta atual do quiz.
58             A classe 'pergunta' controla a estilização, como tamanho
59             da fonte e margem. --&gt;
60</pre>

2



OUTLINE



TIMELINE



x 0 ▲ 0 ⌂ 0



Ln 31, Col 1 Spaces: 6 UTF-8 CRLF HTML ⚡ Go Live Reload


```

EXPLORER ...

P33 - QUIZ

- estilos.css
- quiz.html
- JS quiz.js

quiz.html

quiz.html > html > body > div.container > div#opcoes.opcoes

```
61 <div id="opcoes" class="opcoes"></div>
62 <!-- Elemento <div> com ID 'opcoes' e classe 'opcoes', onde as opções de
63 | resposta serão inseridas dinamicamente via JavaScript.
64 | A classe 'opcoes' pode ser usada para estilizar as opções, como a
65 | formatação dos botões de escolha ou espaçamento. -->
66
67 <button id="submeter">Submeter Resposta</button>
68 <!-- Elemento <button> com ID 'submeter', usado para submeter a
69 | resposta selecionada pelo usuário.
70 | Este botão iniciará a verificação da resposta através de um evento
71 | de clique que é configurado no JavaScript. -->
72
73 <div id="resultado"></div>
74 <!-- Elemento <div> com ID 'resultado', utilizado para exibir o
75 | resultado do quiz, como a pontuação final ou mensagens de feedback.
76 | A estilização pode ser controlada para destacar essa informação,
77 | dependendo do resultado do usuário no quiz. -->
78
79 </div>
80
81 <script src="quiz.js"></script>
82 <!-- A tag <script> com o atributo src apontando para 'quiz.js'.
83 | Este script contém a lógica do quiz,
84 | incluindo a criação de perguntas, a captura de respostas, e a
85 | exibição dos resultados. É carregado no final do documento
86 | para garantir que todos os elementos HTML sejam carregados
87 | antes da execução do script. -->
88
89 </body>
90 </html>
```

OUTLINE 1

TIMELINE

Ln 61, Col 47 Spaces: 6 UTF-8 CRLF HTML Go Live Reload



EXPLORER

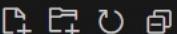
...

estilos.css X

&gt; □ ...



P33 - QUIZ



estilos.css

quiz.html

quizjs



&gt; OUTLINE

&gt; TIMELINE

estilos.css &gt; body

```
1 body {  
2     /* Seleciona o elemento <body> que envolve  
3         todo o conteúdo visível da página. */  
4  
5     font-family: 'Roboto', sans-serif;  
6     /* Define a família de fontes para o corpo do documento  
7         como 'Roboto', com um fallback para qualquer sans-serif  
8         caso 'Roboto' não esteja disponível. */  
9  
10    margin: 40px;  
11    /* Aplica uma margem de 40 pixels em todos os lados do corpo do  
12       documento, criando espaço entre o conteúdo e as bordas da  
13       janela do navegador. */  
14  
15    background-color: #f0f0f0;  
16    /* Define a cor de fundo do corpo do documento como um cinza  
17       claro (#f0f0f0), proporcionando um fundo suave que é  
18       fácil nos olhos. */  
19  
20}  
21  
22 .container {  
23     /* Seleciona todos os elementos com a classe 'container'. Essa  
24         classe é usada para envolver seções específicas de  
25         conteúdo, como o quiz. */  
26  
27     max-width: 600px;  
28     /* Define a largura máxima para o elemento 'container' como 600  
29         pixels. Isso ajuda a evitar que o conteúdo fique muito  
30         esticado em telas maiores. */
```

EXPLORER ...

P33 - QUIZ

- estilos.css
- quiz.html
- quiz.js

estilos.css X

estilos.css > .container

```
31 margin: auto;
32 /* Define as margens laterais para 'auto', o que centraliza o
33 | 'container' horizontalmente dentro de seu elemento pai. */
34 |
35
36 background-color: #ffffff;
37 /* Define a cor de fundo do elemento 'container' como branco (#ffffff),
38 | proporcionando um contraste claro com o texto e outros
39 | elementos dentro dele. */
40
41 padding: 20px;
42 /* Aplica um preenchimento de 20 pixels em todas as direções
43 | dentro do 'container', aumentando o espaço entre o conteúdo e
44 | as bordas do elemento. */
45
46 border-radius: 10px;
47 /* Aplica um raio de borda de 10 pixels nos cantos do 'container',
48 | suavizando-os para uma aparência mais amigável e menos rígida. */
49
50 box-shadow: 0 2px 4px rgba(0,0,0,0.9);
51 /* Aplica uma sombra leve abaixo do 'container'. A sombra se
52 | move 0 pixels para a direita e 2 pixels para baixo do 'container',
53 | com um desfoque de 4 pixels e uma cor preta com 90% de
54 | opacidade (rgba(0,0,0,0.9)), criando uma sensação de profundidade. */
55
56 }
57
58 .pergunta {
59 /* Seleciona todos os elementos com a classe 'pergunta'. */
60 }
```

Ln 31, Col 1 Spaces: 6 UTF-8 CRLF CSS Go Live Reload

EXPLORER ...

P33 - QUIZ

- estilos.css
- quiz.html
- quiz.js

estilos.css X

estilos.css > .pergunta

```
1  font-size: 24px;
2  /* Define o tamanho da fonte para 24 pixels, garantindo que o
3   texto da pergunta seja grande e claro o suficiente para
4   ser facilmente lido. */
5
6  margin-bottom: 20px;
7  /* Aplica uma margem de 20 pixels na parte inferior do elemento.
8   Isso ajuda a separar visualmente a pergunta das opções de
9   resposta abaixo dela. */
10
11 }
12
13 .opcoes label {
14  /* Seleciona todos os elementos <label> dentro de elementos
15   com a classe 'opcoes'. */
16
17  display: block;
18  /* Faz com que cada <label> se comporte como um bloco, ocupando a
19   largura total de seu elemento pai. Isso garante que cada opção
20   de resposta esteja em sua própria linha. */
21
22  margin: 10px 0;
23  /* Aplica uma margem de 10 pixels acima e abaixo de cada <label>,
24   separando visualmente cada opção de resposta. */
25
26  cursor: pointer;
27  /* Muda o cursor para um ponteiro quando está sobre os <label>,
28   indicando que eles são clicáveis. */
29
30 }
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90 }
```

Ln 61, Col 8 Spaces: 6 UTF-8 CRLF CSS Go Live Reload

EXPLORER ...

P33 - QUIZ

- estilos.css
- quiz.html
- quiz.js

estilos.css X

estilos.css > ...

```
91
92 .opcoes input[type="radio"] {
93     /* Seleciona todos os elementos <input> do tipo 'radio'
94     | dentro de elementos com a classe 'opcoes'. */
95
96     margin-right: 10px;
97     /* Aplica uma margem de 10 pixels à direita de cada <input> tipo
98     | 'radio'. Isso adiciona espaço entre o botão de opção e o
99     | texto da opção, melhorando a legibilidade e a estética. */
100
101 }
102
103 button {
104     /* Seleciona todos os elementos <button>. */
105
106     display: block;
107     /* Define o botão para ser um bloco, o que significa
108     | que ele ocupará a largura total do seu contêiner pai, a
109     | menos que a largura seja definida de outra forma. */
110
111     width: 100%;
112     /* Ajusta a largura do botão para cobrir 100% da largura do seu
113     | contêiner pai, estendendo-se totalmente de um lado ao outro. */
114
115     padding: 10px;
116     /* Adiciona 10 pixels de preenchimento dentro do botão em todas as
117     | direções, aumentando a área clicável e melhorando a estética. */
118
119     margin-top: 20px;
120     /* Aplica uma margem de 20 pixels apenas na parte superior do botão.
```

Ln 91, Col 1 Spaces: 6 UTF-8 CRLF CSS ⚙ Go Live Reload

EXPLORER ...

P33 - QUIZ

- estilos.css
- quiz.html
- quiz.js

estilos.css X

estilos.css > button

```
121     criando espaço entre ele e qualquer conteúdo acima. */
122
123     font-size: 18px;
124     /* Define o tamanho da fonte do texto dentro do botão para 18 pixels,
125        fazendo com que seja facilmente legível. */
126
127     color: white;
128     /* Define a cor do texto dentro do botão para branco, oferecendo um
129        contraste alto com o fundo azul, melhorando a legibilidade. */
130
131     background-color: #007bff;
132     /* Define a cor de fundo do botão para um azul brilhante (#007bff),
133        tornando-o visualmente distinto e atraente. */
134
135     border: none;
136     /* Remove qualquer borda do botão, mantendo um design
137        limpo e moderno. */
138
139     border-radius: 5px;
140     /* Arredonda os cantos do botão com um raio de 5 pixels,
141        suavizando a aparência geral. */
142
143     cursor: pointer;
144     /* Altera o cursor para um ponteiro quando ele está sobre o
145        botão, indicando que o botão é clicável. */
146
147 }
148
149 button:hover {
150     /* Seleciona o estado de hover do botão, ativado quando o
```

Ln 121, Col 19 Spaces: 6 UTF-8 CRLF CSS Go Live Reload



EXPLORER

...

estilos.css X



P33 - QUIZ

estilos.css

quiz.html

JS quiz.js



&gt; OUTLINE

&gt; TIMELINE



estilos.css &gt; button:hover

```
151 |     usuário passa o mouse sobre ele. */
152 |
153     background-color: #032141;
154     /* Muda a cor de fundo do botão para um azul mais escuro (#032141)
155     | durante o hover, fornecendo um feedback visual imediato. */
156 }
157 |
158 |
159 #resultado {
160     /* Seleciona o elemento com o ID 'resultado'. */
161 |
162     font-size: 22px;
163     /* Define o tamanho da fonte para 22 pixels, tornando o texto grande
164     | o suficiente para ser destacado como um elemento importante na página. */
165 |
166     margin-top: 20px;
167     /* Aplica uma margem de 20 pixels apenas na parte superior do elemento,
168     | criando espaço entre ele e qualquer conteúdo acima. */
169 |
170 }
```

EXPLORER ... JS quiz.js X

JS quiz.js > ...

```
1 // Adiciona um ouvinte de evento ao documento que
2 // espera pelo evento 'DOMContentLoaded'.
3 // Este evento é disparado quando todo o conteúdo do HTML
4 // foi completamente carregado e analisado.
5 document.addEventListener('DOMContentLoaded', () => {
6
7     // Define uma constante 'perguntas', que é um array de objetos.
8     // Cada objeto representa uma pergunta do quiz.
9     const perguntas = [
10
11         // Cada objeto contém três propriedades: 'pergunta', 'opcoes', e 'resposta'.
12         // 'pergunta' é uma string que contém a pergunta a ser apresentada ao usuário.
13         { pergunta: "Qual é o maior animal terrestre?", opcoes: ["Elefante africano", "Rinoceronte branco", "Girafa", "Urso polar"], resposta: "Elefante africano" },
14         { pergunta: "Qual animal é conhecido como o 'Rei da Selva'?", opcoes: ["Tigre", "Leão", "Elefante", "Gorila"], resposta: "Leão" },
15         { pergunta: "Qual destes animais é um mamífero aquático?", opcoes: ["Tubarão", "Baleia-azul", "Polvo", "Camarão"], resposta: "Baleia-azul" },
16         { pergunta: "Quantas vidas diz-se que um gato tem?", opcoes: ["1", "5", "7", "9"], resposta: "7" },
17         { pergunta: "Qual é o animal mais rápido do mundo?", opcoes: ["Falcão peregrino", "Leopardo", "Guepardo", "Leão"], resposta: "Falcão peregrino" },
18         { pergunta: "Qual desses animais é um réptil?", opcoes: ["Sapo", "Salamandra", "Cobra", "Baleia"], resposta: "Cobra" },
19         { pergunta: "Qual é o maior animal do mundo?", opcoes: ["Elefante africano", "Baleia-azul", "Girafa", "Tubarão-branco"], resposta: "Baleia-azul" },
20         { pergunta: "O que é uma panda gigante?", opcoes: ["Carnívoro", "Herbívoro", "Onívoro", "Insetívoro"], resposta: "Herbívoro" },
21         { pergunta: "Qual destes animais é conhecido por ter uma excelente memória?", opcoes: ["Elefante", "Cachorro", "Gato", "Peixe"], resposta: "Elefante" },
22         { pergunta: "Que animal é o símbolo nacional da Austrália?", opcoes: ["Canguru", "Koala", "Emu", "Dingo"] }]
```



EXPLORER

...

JS quiz.js

X

P33 - QUIZ

estilos.css

quiz.html

JS quiz.js



&gt; OUTLINE

&gt; TIMELINE

```
JS quiz.js > document.addEventListener('DOMContentLoaded') callback > perguntas > resposta
    resposta: "Canguru" }

23 ];
24
25 // Esta constante 'perguntas' agora armazena todas as perguntas do
26 // quiz. Quando o código é executado, ele pode usar este array para
27 // apresentar perguntas e verificar respostas dinamicamente.
28
29 // Declara e inicializa uma variável 'perguntaAtual' com o valor 0.
30 // Esta variável é usada para rastrear o índice da
31 // pergunta atual no array de perguntas.
32 let perguntaAtual = 0;
33
34 // Declara e inicializa uma variável 'pontuacao' com o valor 0.
35 // Esta variável armazena a pontuação do usuário
36 // conforme ele responde corretamente às perguntas.
37 let pontuacao = 0;
38
39 // Declara e inicializa um array 'respostasUsuario'.
40 // Este array será usado para armazenar as respostas
41 // dadas pelo usuário ao longo do quiz.
42 let respostasUsuario = [];
43
44 // Declara uma constante 'elementoPergunta' e a
45 // associa ao elemento HTML com o ID 'pergunta'.
46 // Esta constante será usada para referenciar e
47 // manipular o elemento que exibe a pergunta no quiz.
48 const elementoPergunta = document.getElementById('pergunta');
49
50 // Declara uma constante 'elementoOpcoes' e a
51 // associa ao elemento HTML com o ID 'opcoes'.
```



EXPLORER

...

JS quiz.js

X

P33 - QUIZ

estilos.css

quiz.html

JS quiz.js



&gt; OUTLINE

&gt; TIMELINE

JS quiz.js &gt; ⌂ document.addEventListener('DOMContentLoaded') callback

```
52 // Esta constante será usada para referenciar e
53 // manipular o elemento que contém as opções de resposta do quiz.
54 const elementoOpcoes = document.getElementById('opcoes');

55
56 // Declara uma constante 'elementoResultado' e a
57 // associa ao elemento HTML com o ID 'resultado'.
58 // Esta constante será usada para referenciar e
59 // manipular o elemento que exibe os resultados do quiz.
60 const elementoResultado = document.getElementById('resultado');

61
62 // Declara uma constante 'botaoSubmeter' e a associa ao
63 // elemento HTML com o ID 'submeter'.
64 // Esta constante será usada para referenciar e manipular o
65 // botão que o usuário clica para submeter suas respostas.
66 const botaoSubmeter = document.getElementById('submeter');

67
68
69 // Define a função 'mostrarPergunta' que aceita um
70 // objeto 'pergunta' como parâmetro.
71 function mostrarPergunta(pergunta) {

72
73     // Define o texto do elemento 'elementoPergunta' para
74     // ser o texto da pergunta atual.
75     // 'pergunta.pergunta' acessa a propriedade 'pergunta' do
76     // objeto 'pergunta' passado para a função.
77     elementoPergunta.textContent = pergunta.pergunta;

78
79     // Limpa o conteúdo interno de 'elementoOpcoes' para garantir que
80     // não haja opções de perguntas anteriores sendo exibidas.
81     elementoOpcoes.innerHTML = '';
```

EXPLORER

... JS quiz.js X

P33 - QUIZ

- estilos.css
- quiz.html
- JS quiz.js

2

OUTLINE

TIMELINE

```
JS quiz.js > document.addEventListener('DOMContentLoaded') callback > mostrarPergunta
82
83     // Itera sobre o array 'opcoes' do objeto 'pergunta'. Para
84     // cada 'opcao' no array, executa a função abaixo.
85     pergunta.opcoes.forEach(opcao => {
86
87         // Cria um novo elemento <label> e o armazena na constante 'label'.
88         const label = document.createElement('label');
89
90         // Cria um novo elemento <input> do tipo 'radio' e o armazena na constante 'radio'.
91         const radio = document.createElement('input');
92
93         // Define o tipo do elemento 'radio' para 'radio',
94         // fazendo com que seja um botão de opção.
95         radio.type = 'radio';
96
97         // Define o nome do grupo de botões de opção para 'opcao',
98         // garantindo que o usuário possa selecionar apenas uma opção.
99         radio.name = 'opcao';
100
101        // Define o valor do botão de opção para ser a opção atual do loop.
102        radio.value = opcao;
103
104        // Adiciona o botão de opção 'radio' ao elemento 'label'.
105        label.appendChild(radio);
106
107        // Cria um nó de texto com a opção atual e adiciona esse
108        // nó ao elemento 'label'.
109        label.appendChild(document.createTextNode(opcao));
110
111        // Adiciona o elemento 'label' completo ao 'elementoOpcoes'.
```

## EXPLORER JS quiz.js X

P33 - QUIZ

estilos.css

quiz.html

JS quiz.js

JS quiz.js &gt; document.addEventListener('DOMContentLoaded') callback &gt; mostrarPergunta &gt; pergunta.opcoes.forEach() callback

```
112     // fazendo com que ele seja exibido na página.  
113     elementoOpcoes.appendChild(label);  
114  
115 };  
116 }  
117  
118 // Define a função 'verificarResposta' para verificar a  
119 // resposta escolhida pelo usuário.  
120 function verificarResposta() {  
121  
122     // Busca no documento HTML o primeiro elemento 'input' do  
123     // tipo 'radio' que está marcado e pertence ao grupo 'opcao'.  
124     const opcaoSelecionada = document.querySelector('input[name="opcao"]:checked');  
125  
126     // Verifica se nenhuma opção foi selecionada.  
127     if (!opcaoSelecionada) {  
128  
129         // Exibe um alerta para o usuário pedindo para  
130         // que selecione uma opção.  
131         alert('Por favor, selecione uma opção!');  
132  
133         // Encerra a função prematuramente para evitar que o  
134         // código subsequente seja executado.  
135         return;  
136  
137     }  
138  
139     // Adiciona o valor da opção selecionada ao array  
140     // 'respostasUsuario', armazenando o histórico de respostas do usuário.  
141     respostasUsuario.push(opcaoSelecionada.value);
```



&gt; OUTLINE

&gt; TIMELINE

EXPLORER

... JS quiz.js X

P33 - QUIZ

- estilos.css
- quiz.html
- JS quiz.js

2

OUTLINE

TIMELINE

```
JS quiz.js > document.addEventListener('DOMContentLoaded') callback > verificarResposta
142
143     // Verifica se o valor da opção selecionada é igual à
144     // resposta correta da pergunta atual.
145     if (opcaoSelecionada.value === perguntas[perguntaAtual].resposta) {
146
147         // Incrementa a variável 'pontuacao' por um, indicando
148         // que o usuário acertou a resposta.
149         // pontuacao = pontuacao + 1
150         pontuacao++;
151
152     }
153
154     // Incrementa a variável 'perguntaAtual' para
155     // mover para a próxima pergunta.
156     perguntaAtual++;
157
158     // Verifica se ainda há perguntas restantes no array 'perguntas'.
159     if (perguntaAtual < perguntas.length) {
160
161         // Se houver, chama a função 'mostrarPergunta' com a próxima pergunta.
162         mostrarPergunta(perguntas[perguntaAtual]);
163
164     } else {
165
166         // Se não houver mais perguntas, chama a função 'mostrarResultado'
167         // para exibir os resultados finais do quiz.
168         mostrarResultado();
169
170     }
171 }
```



EXPLORER

...

JS quiz.js X



P33 - QUIZ

estilos.css

quiz.html

JS quiz.js



&gt; OUTLINE

&gt; TIMELINE

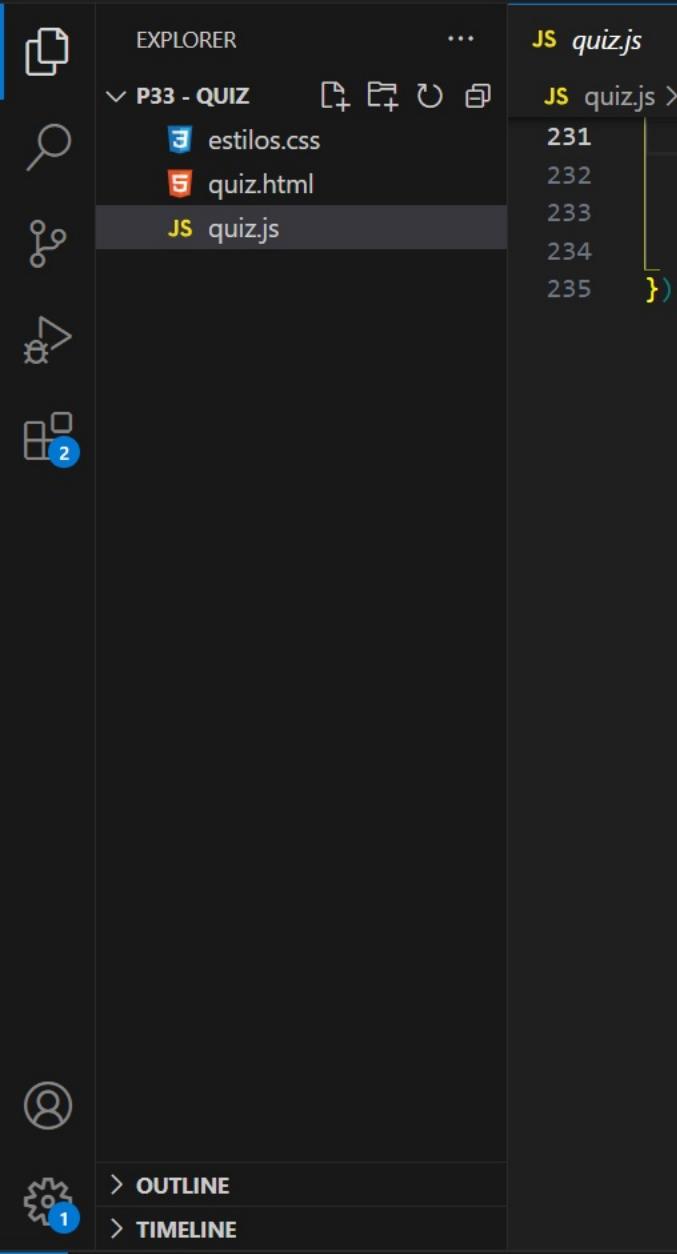
JS quiz.js &gt; ⚒ document.addEventListener('DOMContentLoaded') callback

```
172  
173  
174  
175 // Define a função 'mostrarResultado' que será chamada ao  
176 // final do quiz para exibir os resultados.  
177 function mostrarResultado() {  
178  
179     // Oculta o elemento que exibe a pergunta atual do quiz.  
180     elementoPergunta.style.display = 'none';  
181  
182     // Oculta o elemento que exibe as opções de resposta.  
183     elementoOpcoes.style.display = 'none';  
184  
185     // Oculta o botão de submissão.  
186     botaoSubmeter.style.display = 'none';  
187  
188     // Calcula a porcentagem de pontuação do usuário baseada no  
189     // número de respostas corretas e no total de perguntas.  
190     const porcentagemPontuacao = (pontuacao / perguntas.length) * 100;  
191  
192     // Determina a mensagem de resultado com base na pontuação do  
193     // usuário. Se 70% ou mais, considera aprovado; caso contrário, reprovado.  
194     let resultadoHTML = porcentagemPontuacao >= 70 ?  
195         `Parabéns! Você foi aprovado com ${pontuacao} de ${perguntas.length} acertos.` :  
196         `Você foi reprovado. Você acertou ${pontuacao} de ${perguntas.length}.`;  
197  
198     // Adiciona uma seção para detalhar todas as perguntas e respostas.  
199     resultadoHTML += '<br><br><h2>Respostas:</h2>';  
200  
201     // Itera sobre cada pergunta no array 'perguntas', incluindo o
```

EXPLORER ... JS quiz.js X

JS quiz.js > document.addEventListener('DOMContentLoaded') callback > mostrarResultado

```
202     // índice da pergunta para referência.  
203     perguntas.forEach((pergunta, index) => {  
204         // Para cada pergunta, adiciona a pergunta e a resposta correta ao HTML de resultado.  
205         resultadoHTML += `<p><strong>Pergunta ${index + 1}:</strong> ${pergunta.pergunta}<br>`;  
206         resultadoHTML += `<strong>Resposta Correta:</strong> ${pergunta.resposta}<br>`;  
207         // Verifica se o usuário respondeu à pergunta e adiciona sua resposta ou  
208         // indica que não foi respondida.  
209         resultadoHTML += `<strong>Sua Resposta:</strong> ${respostasUsuario[index] ? respostasUsuario[index] : 'Não  
210             respondida'}</p>`;  
211     });  
212     // Define o HTML do elemento 'elementoResultado' para a string  
213     // 'resultadoHTML', que contém todos os detalhes dos resultados.  
214     elementoResultado.innerHTML = resultadoHTML;  
215     //  
216     // Chama a função 'mostrarPergunta' com a pergunta atual do  
217     // array 'perguntas' usando a variável 'perguntaAtual' como índice.  
218     // 'perguntas[perguntaAtual]' acessa o objeto de pergunta atual no  
219     // array 'perguntas', que contém a pergunta e as opções a serem exibidas.  
220     mostrarPergunta(perguntas[perguntaAtual]);  
221     //  
222     // Adiciona um ouvinte de eventos ao botão 'botaoSubmeter'.  
223     // Esse ouvinte 'click' chama a função 'verificarResposta' sempre que o botão é clicado.  
224     // A função 'verificarResposta' é responsável por verificar se a  
225     // resposta fornecida pelo usuário é correta.  
226     //  
227     //  
228     //  
229     //  
230     //
```



JS quiz.js

```
JS quiz.js > document.addEventListener('DOMContentLoaded') callback
231     // resposta selecionada pelo usuário é correta ou não.
232     botaoSubmeter.addEventListener('click', verificarResposta);
233
234
235 };
```

# **UC: Projetos Assunto: Small Projects P34 – Valor por Extenso**

EXPLORER

... □

✓ P34 - VALOR POR EXTENSO

JS converter.js

estilos.css

index.html

2

OUTLINE

TIMELINE

1

index.html X

index.html > ...

```
1  <!DOCTYPE html>
2  <!-- A linha acima declara o tipo de documento como HTML5,
3      que é a versão atual do HTML. -->
4
5  <html lang="pt-br">
6  <!-- A tag <html> define o início do documento HTML. O atributo 'lang'
7      especifica a língua do documento, neste caso, Português
8      do Brasil (pt-br). -->
9
10 <head>
11 <!-- A tag <head> contém metadados (informações sobre os dados)
12     que não são exibidos diretamente na página web. -->
13
14 <meta charset="UTF-8">
15 <!-- A tag <meta> com o atributo 'charset' define a codificação de
16     caracteres usada pelo documento HTML, neste caso, UTF-8, que
17     inclui a maioria dos caracteres de todos os idiomas escritos. -->
18
19 <title>Valor por Extenso</title>
20 <!-- A tag <title> define o título da página, que aparece na aba do navegador. -->
21
22 <link rel="stylesheet" href="estilos.css">
23 <!-- A tag <link> é usada aqui para vincular uma folha de
24     estilos externa ao documento HTML.
25     'rel' especifica o relacionamento entre o documento atual e o
26     recurso vinculado, neste caso, uma folha de estilos.
27     'href' especifica o caminho para a folha de estilos. -->
28
29 </head>
30
```

EXPLORER ...

P34 - VALOR POR EXTENSO

- JS converter.js
- estilos.css
- index.html

2

OUTLINE

TIMELINE 1

index.html X

index.html > html > body

```
31 <body>
32     <!-- A tag <body> contém todo o conteúdo da página que é
33         visível para o usuário. -->
34
35     <div class="container">
36         <!-- A tag <div> é usada para agrupar conteúdo, facilitando o
37             estilo ou script do grupo como uma unidade.
38             O atributo 'class' é usado para especificar que o div pertence à
39                 classe 'container', que pode ser usada para estilização
40                     específica definida na folha de estilos. -->
41
42     <h1>Converta Número para Reais por Extenso</h1>
43     <!-- A tag <h1> define um cabeçalho de nível um, que é geralmente
44         usado para o título principal ou mais importante da página. -->
45
46     <input type="number" id="numeroInput" placeholder="Insira um número" step="0.01">
47     <!-- A tag <input> é usada para criar um elemento de entrada de dados.
48         'type="number"' especifica que a entrada deve ser um número.
49         'id' fornece um identificador único para o elemento, que pode ser
50             usado para referenciá-lo em scripts.
51         'placeholder' fornece um texto de dica que descreve a expectativa de
52             entrada no campo de entrada.
53         'step' define o intervalo de números válidos que podem ser
54             inseridos, neste caso, permitindo valores decimais até duas casas. -->
55
56     <button onclick="converterParaExtenso()">Converter</button>
57     <!-- A tag <button> cria um botão clicável.
58         'onclick' é um atributo de evento que especifica que a
59             função JavaScript 'converterParaExtenso()' deve ser
60                 executada quando o botão é clicado. -->
```

Ln 31, Col 6 Spaces: 4 UTF-8 CRLF HTML Ø Port: 5500 Reload 🔍



EXPLORER

...

## P34 - VALOR POR EXTENSO

JS converter.js

CSS estilos.css

index.html



&gt; OUTLINE

&gt; TIMELINE

index.html X

index.html &gt; html &gt; body &gt; div.container

```
61
62    <p id="resultado"></p>
63    <!-- A tag &lt;p&gt; é usada para definir um parágrafo.
64        'id' fornece um identificador único para o parágrafo, permitindo
65        que seu conteúdo seja manipulado dinamicamente via JavaScript. --&gt;
66
67    &lt;/div&gt;
68
69    &lt;script src="converter.js"&gt;&lt;/script&gt;
70    <!-- A tag &lt;script&gt; é usada para incorporar ou referenciar um script
71        externo, neste caso, 'converter.js'.
72        'src' especifica o caminho para o arquivo de script. --&gt;
73
74    &lt;/body&gt;
75    &lt;/html&gt;</pre>
```

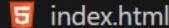
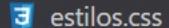
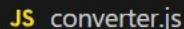


EXPLORER

•



✓ P34 - VALOR POR EXTERNO



> OUTLINE

> TIMELINE



> TIMELINE

EXPLORER

...

P34 - VALOR POR EXTEÑO

JS converter.js

estilos.css

index.html

2

...

OUTLINE

1

TIMELINE

### estilos.css

```
estilos.css > body

31     |     dentro do <body> sejam alinhados e distribuídos de forma
32     |     mais eficiente e com layouts responsivos. */
33
34     justify-content: center;
35     /* 'justify-content: center' alinha horizontalmente os itens
36     |     filhos (elementos dentro do <body>) ao centro.
37     |     Essa propriedade só funciona porque 'display' está definido como 'flex'. */
38
39     align-items: center;
40     /* 'align-items: center' alinha verticalmente os itens filhos ao centro.
41     |     Assim como 'justify-content', esta propriedade também depende
42     |     do 'display: flex' para funcionar. */
43
44     height: 100vh;
45     /* 'height' define a altura do elemento <body>.
46     |     '100vh' significa 100% da altura da janela de visualização (viewport),
47     |     garantindo que o <body> ocupe toda a altura visível da
48     |     tela do dispositivo. */
49
50 }
51
52 .container {
53
54     background: white;
55     /* 'background' define a cor de fundo do elemento que
56     |     tem a classe 'container'.
57     |     Aqui, o valor 'white' especifica que a cor de fundo
58     |     será branca. */
59
60     padding: 20px;
```

EXPLORER ...

✓ P34 - VALOR POR EXTOENO

JS converter.js

estilos.css

index.html

estilos.css X

estilos.css > .container

```
61  /* 'padding' define a margem interna do elemento. Isso é o espaço
62  | entre o conteúdo interno do elemento e suas bordas.
63  | '20px' aplica um espaço uniforme de 20 pixels em todas as
64  | direções (topo, direita, base, esquerda) dentro do elemento. */
65
66  border-radius: 8px;
67  /* 'border-radius' arredonda os cantos do elemento.
68  | '8px' significa que os cantos do elemento serão arredondados com um
69  | raio de 8 pixels, dando ao elemento uma aparência mais
70  | suave e menos quadrada. */
71
72  box-shadow: 0 4px 8px rgba(0,0,0,0.1);
73  /* 'box-shadow' adiciona uma sombra ao redor do elemento.
74  | - O primeiro valor '0' é o deslocamento horizontal da sombra.
75  | - O segundo valor '4px' é o deslocamento vertical.
76  | - O terceiro valor '8px' é o raio do desfoque da sombra.
77  | - 'rgba(0,0,0,0.1)' define a cor da sombra. Aqui, a cor é
78  |     preta ('rgb(0,0,0)') com uma opacidade de 10% ('0.1'),
79  |     tornando a sombra sutil. */
80
81  text-align: center;
82  /* 'text-align: center' alinha o texto dentro do elemento ao centro.
83  | Isso significa que todo o texto dentro do elemento com a
84  | classe 'container' será centralizado horizontalmente. */
85
86 }
87
88 input, button {
89
90  padding: 10px;
```



EXPLORER

...

## P34 - VALOR POR EXTENSO

JS converter.js

estilos.css

index.html



&gt; OUTLINE

&gt; TIMELINE

x

0 ▲ 0 ⌂ 0

Ln 91, Col 2 Spaces: 3 UTF-8 CRLF CSS Ø Port : 5500 Reload

estilos.css X

estilos.css &gt; input

```
91  /* 'padding' define a margem interna dos elementos, que é o
92   |     espaço entre o conteúdo interno e as bordas do elemento.
93   |     '10px' aplica um espaço uniforme de 10 pixels em todas as direções
94   |     dentro dos elementos `input` e `button`.
95   |     Isso faz com que o conteúdo interno desses elementos (texto do botão
96   |     ou texto dentro do campo de entrada) tenha um espaço adicional
97   |     ao redor, tornando-os mais fáceis de ler e clicar. */
98
99  margin-top: 10px;
100 /* 'margin-top' define a margem superior para os
101  |     elementos `input` e `button`.
102  |     '10px' coloca uma separação de 10 pixels entre o topo desses
103  |     elementos e quaisquer elementos acima deles.
104  |     Isso ajuda a espaçar verticalmente os controles de forma
105  |     uniforme na interface do usuário. */
106
107 border-radius: 5px;
108 /* 'border-radius' arredonda os cantos dos elementos.
109  |     '5px' significa que os cantos dos elementos `input` e `button`
110  |     serão arredondados com um raio de 5 pixels,
111  |     o que dá a esses elementos uma aparência mais suave e moderna. */
112
113 border: 1px solid #ccc;
114 /* 'border' define a borda dos elementos.
115  |     - '1px' define a espessura da borda como um pixel.
116  |     - 'solid' define o estilo da borda como sólido (sem interrupções ou traços).
117  |     - '#ccc' (cinza claro) define a cor da borda. Esta cor é sutil e
118  |     é comumente usada em designs para dar um contorno discreto que
119  |     se destaca contra uma variedade de fundos sem ser muito dominante. */
120
```



EXPLORER

...

## P34 - VALOR POR EXTENSO

JS converter.js

estilos.css

index.html



estilos.css X

estilos.css &gt; input

```
121 | outline: none;
122 | /* 'outline' define o contorno que aparece ao redor de elementos interativos
123 | | quando são focados (por exemplo, quando um usuário clica ou usa a
124 | | tecla Tab para navegar até um elemento).
125 | | 'none' remove completamente esse contorno.
126 | | Atenção: remover o contorno pode impactar a acessibilidade, pois ele
127 | | indica visualmente o foco. É recomendável proporcionar uma
128 | | alternativa visual clara para usuários que dependem dessa
129 | | indicação para navegar. */
130 |
131 }
132
133 button {
134
135     cursor: pointer;
136     /* 'cursor: pointer' altera a forma do cursor do mouse para um
137     | ponteiro (típico ícone de mão), que é comumente usado
138     | para indicar que um elemento é clicável.
139     | Isso melhora a usabilidade, pois os usuários podem identificar
140     | visualmente que o elemento é interativo. */
141
142     background-color: #007bff;
143     /* 'background-color' define a cor de fundo do botão.
144     | '#007bff' é um azul claro vibrante, o que dá ao botão uma
145     | | aparência distinta e atraente, facilitando sua identificação
146     | | como um elemento clicável. */
147
148     color: white;
149     /* 'color' define a cor do texto dentro do botão.
150     | 'white' faz com que o texto dentro do botão seja branco, criando
```





EXPLORER

...

estilos.css X

D □ ...



P34 - VALOR POR EXTE

JS converter.js

estilos.css

index.html



&gt; OUTLINE

&gt; TIMELINE



0 ▲ 0 ⌂ 0

Ln 181, Col 3 Spaces: 3 UTF-8 CRLF CSS ⚙ Port: 5500 Reload



EXPLORER

...

JS converterjs X

▷ ⏪ ...

P34 - VALOR P...



JS converter.js

estilos.css

index.html



&gt; OUTLINE

&gt; TIMELINE



⊗ 0 △ 0 ⌂ 0

```
1 // Declaração da função principal que converte um número inteiro em texto por extenso.
2 function numeroPorExtenso(numero) {
3
4     // Arrays com as palavras correspondentes para cada dígito, dezena, centena, etc.
5     const unidades = ["zero", "Um", "Dois", "Três", "Quatro", "Cinco", "Seis", "Sete", "Oito", "Nove"];
6     const dezenas = ["", "Dez", "Vinte", "Trinta", "Quarenta", "Cinquenta", "Sessenta", "Setenta", "Oitenta", "Noventa"];
7     const especiais = ["Onze", "Doze", "Treze", "Quatorze", "Quinze", "Desesete", "Desesseis", "Dezoito", "Dezenove"];
8     const centenas = ["", "Cento", "Duzentos", "Trezentos", "Quatrocientos", "Quinhentos", "Seiscentos", "Setecentos",
9         "Oitocentos", "Novecentos"];
10    const classes = [[[], []], [" Mil", " Mil"], [" Milhão", " Milhões"], [" Bilhão", " Bilhões"], [" Trilhão", " Trilhões"]];
11
12    // Subfunção que lida com a conversão de cada parte do número,
13    // dependendo da sua grandeza.
14    function converterParte(num) {
15
16        // Se o número tem apenas um dígito.
17        if (num < 10) return unidades[num];
18
19        // Caso especial para o número 10.
20        if (num === 10) return "Dez";
21
22        // Se o número está entre 11 e 19, usa-se o array de
23        // números 'especiais'.
24        if (num < 20) return especiais[num - 11];
25
26        // Se o número é uma dezena ou uma combinação de dezenas com unidades.
27        if (num < 100) {
28
29            // Obtém a dezena do número.
```



EXPLORER

...

JS converter.js X

## P34 - VALOR POR EXTENSO

JS converter.js

estilos.css

index.html



&gt; OUTLINE

&gt; TIMELINE

JS converter.js &gt; numeroPorExtenso &gt; converterParte

```
29     let dec = Math.floor(num / 10); // Math.floor(num / 10) divide o número por 10 e arredonda para baixo,  
30                                         // resultando no número de dezenas completas no número original.  
31                                         // Por exemplo, 87 dividido por 10 é 8.7, que arredondado para baixo dá 8.  
32  
33     // Obtém a unidade do número.  
34     let uni = num % 10;           // num % 10 retorna o resto da divisão do número por 10,  
35                                         // o que equivale à parte das unidades do número.  
36                                         // Por exemplo, o resto de 87 dividido por 10 é 7.  
37  
38     // Combina a dezena e a unidade com a conjunção "e", se necessário.  
39     // Verifica se há unidades para adicionar ao texto.  
40     return dezenas[dec] + (uni ? " e " + unidades[uni] : "");  
41  
42     // dezenas[dec] acessa o array de dezenas para obter a palavra  
43     // correspondente ao número de dezenas.  
44     // Se 'uni' não for zero, adiciona " e " seguido pela palavra  
45     // para a unidade correspondente.  
46     // Se 'uni' for zero, apenas retorna a palavra da dezena.  
47 }  
48  
49  
50  
51     // Se o número é uma centena ou uma combinação de centenas com dezenas ou unidades.  
52     let cen = Math.floor(num / 100); // Math.floor(num / 100) divide o número por 100 e arredonda para baixo,  
53                                         // resultando no número de centenas completas no número original.  
54                                         // Por exemplo, 456 dividido por 100 é 4.56, que arredondado para baixo dá 4.  
55  
56     let resto = num % 100;        // num % 100 retorna o resto da divisão do número por 100,  
57                                         // o que equivale à parte que consiste em dezenas e unidades.  
58                                         // Por exemplo, o resto de 456 dividido por 100 é 56.
```

EXPLORER ... JS converter.js X

P34 - VALOR POR EXTEÑO JS converter.js > numeroPorExtenso > converterParte

```
59 // Combina a centena com o resto do número, usando a conjunção "e", se necessário.  
60 return (cen === 1 && resto === 0 ? "Cem" : centenas[cen]) + (resto ? " e " + converterParte(resto) : "");  
61  
62 // A expressão (cen === 1 && resto === 0 ? "Cem" : centenas[cen])  
63 // verifica se o número de centenas é 1 e não há resto.  
64 // Se verdadeiro, retorna "Cem" (a forma correta de escrever 100 em  
65 // português, sem a forma plural "Cento").  
66 // Caso contrário, acessa o array de centenas para obter a palavra  
67 // correspondente ao número de centenas.  
68 // Se 'resto' não for zero, adiciona " e " seguido pelo resultado  
69 // da função converterParte aplicada ao resto.  
70 // Isso permite decompor corretamente o resto em dezenas e unidades, se necessário.  
71  
72 }  
73  
74 // A função principal retorna o resultado final convertendo o  
75 // número absoluto (ignora sinais negativos).  
76 return formatarNumero(Math.abs(numero));  
77  
78  
79  
80 // Define a função que formata números inteiros grandes em texto por extenso.  
81 // Esta função é responsável por converter um número inteiro dado (num) em sua  
82 // representação por extenso em palavras, segmentando o  
83 // número em grupos de até três dígitos.  
84 function formatarNumero(num) {  
85  
86 // Verifica inicialmente se o número fornecido é zero.  
87 // Se for zero, a função retorna imediatamente a  
88 // string "Zero", pois não há necessidade
```



EXPLORER

...

JS converter.js X

▷ ⏪ ⏴ ...

## P34 - VALOR POR EXTENSO

JS converter.js

estilos.css

index.html



&gt; OUTLINE

&gt; TIMELINE

JS converter.js &gt; numeroPorExtenso &gt; formatarNumero

```
89 // de processar mais nada; zero é diretamente
90 // representado pela palavra "Zero".
91 if (num === 0) return "Zero";
92
93 // Inicializa um índice para acompanhar a "classe" do número.
94 // Classes referem-se a unidades, milhares, milhões, etc.
95 // O índice é usado para determinar o sufixo apropriado
96 // (mil, milhão, bilhão, etc.) para cada grupo de três dígitos.
97 let indice = 0;
98
99 // Cria um array vazio chamado 'partes'.
100 // Este array armazenará cada segmento do
101 // número já convertido em palavras.
102 // Por exemplo, 123456 se tornaria ["cento e vinte e
103 // três mil", "quatrocentos e cinquenta e seis"]
104 // antes de ser combinado.
105 let partes = [];
106
107 // Executa um loop enquanto ainda houver um número para processar.
108 // O loop continua até que todo o número seja
109 // decomposto em segmentos de três dígitos.
110 while (num > 0) {
111
112     // Calcula o pedaço do número tomando o resto da divisão por 1000.
113     // Isso isola os três últimos dígitos do número,
114     // permitindo processá-los separadamente.
115     let pedaco = num % 1000;
116
117     // Usa a função 'converterParte' para converter o
118     // segmento atual do número em texto.
```



⊗ 0 △ 0 ⌂ 0

Ln 89, Col 5 Spaces: 4 UTF-8 CRLF {} JavaScript Ø Port: 5500 Reload



EXPLORER

...

JS converter.js X

&gt; □ ...

## P34 - VALOR POR EXTENSO

JS converter.js

estilos.css

index.html



&gt; OUTLINE

&gt; TIMELINE

JS converter.js &gt; numeroPorExtenso &gt; formatarNumero

```
119      // 'converterParte' lida com números até 999,  
120      // transformando-os em sua representação por extenso.  
121      let prefixo = converterParte(pedaco);  
122  
123      // Verifica se o pedaço atual é maior que zero para processar.  
124      // Isso é necessário para evitar a adição de sufixos  
125      // desnecessários a segmentos que são zero,  
126      // o que poderia levar a representações  
127      // incorretas como "mil zero".  
128      if (pedaco > 0) {  
129  
130          // Seleciona o sufixo apropriado da lista de classes  
131          // com base no índice atual e no valor do pedaço.  
132          // O sufixo é escolhido entre a forma singular ou  
133          // plural dependendo do valor do segmento (pedaço > 1).  
134          let sufixo = classes[indice][pedaco > 1 ? 1 : 0];  
135  
136          // Adiciona o sufixo selecionado ao texto  
137          // convertido (prefixo).  
138          prefixo += sufixo;  
139  
140          // Verifica se o texto convertido (com o sufixo  
141          // adicionado) não está vazio após remover espaços.  
142          // Isso é útil para evitar a adição de strings  
143          // vazias ao array 'partes'.  
144          if (prefixo.trim() !== "") {  
145  
146              // Adiciona o prefixo ao início do array de partes.  
147              // 'unshift' é usado para inserir cada novo  
148              // segmento convertido na frente do array.
```

## EXPLORER

## P34 - VALOR POR EXTE

- JS converter.js
- estilos.css
- index.html

2

## JS converter.js

```
JS converter.js > numeroPorExtenso > formatarNumero
149     // garantindo que os segmentos sejam armazenados
150     // na ordem correta (milhares antes de unidades, etc.).
151     partes.unshift(prefixo);
152
153 }
154
155
156     // Divide o número original por 1000, descartando os
157     // três últimos dígitos já processados.
158     // Isso reduz o número para os próximos três dígitos a
159     // serem convertidos no próximo ciclo do loop.
160     num = Math.floor(num / 1000);
161
162     // Incrementa o índice para mover para a próxima classe
163     // numérica (de milhares para milhões, por exemplo).
164     indice++;
165
166 }
167
168     // Junta todas as partes convertidas em uma string final.
169     // 'join' é usado com " e " como separador, combinando todos
170     // os segmentos em uma representação contínua do número.
171     // A expressão regular no 'replace' remove espaços extras,
172     // garantindo uma formatação limpa do texto final.
173     return partes.length > 0 ? partes.join(" e ").replace(/\s+/g, ' ').trim() : "Zero";
174
175 }
176
177     return formatarNumero(Math.abs(numero));
178 }
```

## &gt; OUTLINE

## &gt; TIMELINE



EXPLORER

...

JS converter.js X

&gt; □ ...

## P34 - VALOR POR EXTENO

JS converter.js

estilos.css

index.html



&gt; OUTLINE

&gt; TIMELINE

JS converter.js &gt; numeroPorExtenso

```
179 }  
180  
181 // Define a função que irá converter o valor numérico de um input  
182 // para sua representação por extenso em reais e centavos.  
183 function converterParaExtenso() {  
184  
185     // Extrai o número do campo de input, convertendo-o de string para float.  
186     let numero = parseFloat(document.getElementById('numeroInput').value);  
187  
188     // Utiliza Math.floor para arredondar o número para baixo e  
189     // obter a parte inteira dos reais.  
190     let reais = Math.floor(numero);  
191  
192     // Calcula os centavos subtraindo os reais do número original e  
193     // multiplicando por 100, depois arredondando com Math.round.  
194     let centavos = Math.round((numero - reais) * 100);  
195  
196     // Converte a parte dos reais para texto por extenso, usando a função 'numeroPorExtenso'.  
197     // Adiciona a palavra 'Real' ou 'Reais' dependendo da quantidade.  
198     let extensoReais = numeroPorExtenso(reais) + (reais === 1 ? ' Real' : ' Reais');  
199  
200     // Converte os centavos para texto por extenso, se houver algum centavo.  
201     // Adiciona a palavra 'Centavo' ou 'Centavos' dependendo da quantidade.  
202     let extensoCentavos = centavos > 0 ? numeroPorExtenso(centavos) + (centavos === 1 ? ' Centavo' : ' Centavos') : '';  
203  
204     // Inicia a formação do resultado final, começando com o valor por extenso dos reais.  
205     let resultado = extensoReais;  
206  
207     // Se existirem centavos, adiciona à string de resultado, precedendo com ' e '.  
208     if (extensoCentavos) resultado += ' e ' + extensoCentavos;
```



EXPLORER

...

JS converter.js X

▷ □ ...



P34 - VALOR POR EXTENSO

JS converter.js

estilos.css

index.html



&gt; OUTLINE

&gt; TIMELINE

JS converter.js &gt; ...

```
209
210     // Atualiza o conteúdo do elemento HTML com id 'resultado' com o valor por extenso,
211     // garantindo que a primeira letra esteja em maiúscula.
212     document.getElementById('resultado').textContent = resultado.charAt(0).toUpperCase() + resultado.slice(1);
213
214 }
215
216 // Adiciona um ouvinte de evento ao documento que escuta pelo evento 'DOMContentLoaded'.
217 document.addEventListener('DOMContentLoaded', function() {
218
219     // Este evento é disparado quando todo o conteúdo HTML foi completamente carregado,
220     // incluindo todos os scripts dependentes, estilos, e imagens.
221
222     // Obtém o elemento com o id 'numeroInput' e adiciona um
223     // ouvinte de evento de 'keypress' a ele.
224     document.getElementById('numeroInput').addEventListener('keypress', function(e) {
225
226         // 'keypress' é um evento que é acionado toda vez que uma tecla é
227         // pressionada no elemento especificado.
228
229         // Verifica se a tecla pressionada foi a tecla 'Enter'.
230         if (e.key === 'Enter') {
231
232             // Se a tecla 'Enter' foi pressionada, executa o seguinte código:
233
234             // 'e.preventDefault()' impede a ação padrão que pertence a essa tecla.
235             // Para a tecla 'Enter' em um formulário, a ação padrão é submeter o formulário.
236             // Impedindo isso, evita-se que a página seja recarregada.
237             e.preventDefault();
238 }
```



EXPLORER

...

## P34 - VALOR POR EXTENSO

JS converter.js

estilos.css

index.html



JS converter.js X

JS converter.js &gt; ⚡ document.addEventListener('DOMContentLoaded') callback &gt; ⚡ addEventListener('keypress') callback

```
239      // Chama a função 'converterParaExtenso', que é definida em outro lugar no código.  
240      // Esta função é responsável por ler o valor do input,  
241      // converter esse valor para extenso,  
242      // e mostrar o resultado na página.  
243      converterParaExtenso();  
244  
245    }  
246  );  
247 };  
248
```

# **UC: Projetos**

## **Assunto:**

### **Small Projects**

### **P35 – Pop-Up**

EXPLORER ...

P35 - POP-UP

- estilos.css
- img.jpg
- index.html**
- scripts.js

index.html X

index.html > ...

```
1  <!DOCTYPE html>
2  <!-- Declaração do tipo de documento, especifica
3      | que este arquivo é HTML5 -->
4
5  <html lang="pt-br">
6  <!-- Abre a tag HTML e define o idioma do conteúdo do
7      | documento como português do Brasil -->
8
9  <head>
10 <!-- Abre a tag head, que contém metadados, links para
11     | scripts e folhas de estilo -->
12
13 <meta charset="UTF-8">
14 <!-- Define a codificação de caracteres para UTF-8, suportando
15     | um vasto conjunto de caracteres para múltiplos idiomas -->
16
17 <meta name="viewport" content="width=device-width, initial-scale=1.0">
18 <!-- Configura a visualização para escalar corretamente em
19     | dispositivos de diferentes tamanhos, ajustando a largura
20     | da página para a largura do dispositivo e iniciando
21     | com o zoom padrão -->
22
23 <title>Exemplos de Pop-ups</title>
24 <!-- Define o título do documento que aparecerá na
25     | aba do navegador -->
26
27 <link rel="stylesheet" href="estilos.css">
28 <!-- Link para a folha de estilos externa que contém os
29     | estilos CSS para estilizar o documento -->
```

Ln 1, Col 1 Spaces: 8 UTF-8 CRLF HTML ⚡ Port : 5500 Reload



EXPLORER

...

P35 - POP-UP

estilos.css

img.jpg

index.html

scripts.js



index.html X

index.html &gt; html &gt; head

```
31   </head>
32   <!-- Fecha a tag head -->
33
34
35 <body>
36   <!-- Abre a tag body, que contém todo o conteúdo principal
37   | que será exibido na página web -->
38
39   <h1>Exemplos de Pop-ups</h1>
40   <!-- Define um cabeçalho de primeiro nível (h1), que
41   | geralmente é usado para o título principal da página -->
42
43   <div class="botoes">
44   <!-- Abre uma divisão (div) com a classe 'botoes', usada
45   | para agrupar elementos visualmente e aplicar
46   | estilos específicos via CSS -->
47
48   <button id="botaoModal">Abrir Modal Pop-up</button>
49   <!-- Define um botão com id 'botaoModal', que ao
50   | ser clicado, pode abrir uma janela modal pop-up -->
51
52   <button id="botaoAlerta">Mostrar Alerta</button>
53   <!-- Define um botão com id 'botaoAlerta', usado
54   | para mostrar uma janela de alerta quando clicado -->
55
56   <button id="botaoTooltip" class="tooltip">Mostrar Tooltip<span class="tooltipTexto">Texto do Tooltip</span></button>
57   <!-- Define um botão com id 'botaoTooltip' e classe 'tooltip',
58   | usado para mostrar um tooltip; um texto explicativo
59   | adicional é fornecido dentro de um span com
60   | classe 'tooltipTexto' -->
```



EXPLORER

...

index.html X



P35 - POP-UP

estilos.css

img.jpg

index.html

scripts.js



index.html &gt; html &gt; body &gt; div.botoes

```
61      <button id="botaoNotificacao">Mostrar Notificação</button>
62      <!-- Define um botão com id 'botaoNotificacao', que quando
63          | clicado, pode exibir uma notificação na página --&gt;
64
65      &lt;button id="botaoLightbox"&gt;Abrir Lightbox&lt;/button&gt;
66      <!-- Define um botão com id 'botaoLightbox', usado para abrir
67          | uma lightbox, geralmente uma janela que exibe imagens
68          | ou vídeos em uma visualização ampliada sobre o
69          | conteúdo da página --&gt;
70
71      &lt;button id="botaoMenuContexto"&gt;Mostrar Menu de Contexto&lt;/button&gt;
72      <!-- Define um botão com id 'botaoMenuContexto', usado para
73          | exibir um menu de contexto personalizado quando clicado --&gt;
74
75      &lt;button id="botaoConfirmacao"&gt;Mostrar Confirmação&lt;/button&gt;
76      <!-- Define um botão com id 'botaoConfirmacao', que ao ser
77          | clicado, pode abrir uma janela pedindo confirmação
78          | do usuário --&gt;
79
80      &lt;button id="botaoEntradaTexto"&gt;Mostrar Entrada de Texto&lt;/button&gt;
81      <!-- Define um botão com id 'botaoEntradaTexto', usado para
82          | abrir uma janela que permite ao usuário inserir texto --&gt;
83
84      &lt;/div&gt;
85      <!-- Fecha a divisão 'botoes' --&gt;
86
87
88      &lt;!-- Modal Pop-up --&gt;
89
90      &lt;!-- Comentário explicativo indicando que o código a</pre>
```

EXPLORER

...

P35 - POP-UP

- estilos.css
- img.jpg
- index.html**
- scripts.js

2

OUTLINE

TIMELINE

0 △ 0 ⌂ 0

index.html X

index.html > html > body

```
91 |         seguir implementa um modal pop-up -->
92 |     <div id="meuModal" class="modal">
93 |         <!-- Abre uma div com id 'meuModal' e classe 'modal'; é o
94 |             contêiner principal do modal, inicialmente oculto, e
95 |             será exibido quando o usuário acionar uma ação específica --&gt;
96 |
97 |         &lt;div class="modal-conteudo"&gt;
98 |             <!-- Abre uma div com classe 'modal-conteudo', que contém os
99 |                 elementos internos do modal, como texto e botões --&gt;
100 |
101 |             &lt;span class="fechar"&gt;&amp;times;&lt;/span&gt;
102 |             <!-- Inclui um elemento span com classe 'fechar', que
103 |                 contém o caractere 'x'. Este span é usado como
104 |                 botão de fechamento do modal --&gt;
105 |
106 |             &lt;p&gt;Este é um modal pop-up!&lt;/p&gt;
107 |             <!-- Parágrafo contendo uma mensagem indicativa de
108 |                 que esta seção é um modal pop-up --&gt;
109 |
110 |         &lt;/div&gt;
111 |         <!-- Fecha a div 'modal-conteudo' --&gt;
112 |
113 |     &lt;/div&gt;
114 |     <!-- Fecha a div 'meuModal' --&gt;
115 |
116 |
117 |     <!-- Tooltip Pop-up --&gt;
118 |     <!-- Tooltip já incluído no botão --&gt;
119 |
120 |     <!-- Notification Pop-up --&gt;</pre>
```

EXPLORER ...

P35 - POP-UP

- estilos.css
- img.jpg
- index.html**
- scripts.js

index.html x

index.html > html > body

```
121     <!-- Comentário indicando que o código a seguir implementa
122         |     um pop-up de notificação -->
123     <div id="notificacao" class="notificacao">
124         |     <!-- Abre uma div com id 'notificacao' e classe 'notificacao'; é
125             |     o contêiner do pop-up de notificação -->
126
127         <span class="botaoFechar" onclick="this.parentElement.style.display='none';">&times;</span>
128         |     <!-- Define um botão de fechar para o pop-up de notificação.
129             |     Quando clicado, o estilo 'display' do elemento
130                 |     pai (a div de notificação) é alterado para 'none',
131                     |     ocultando o pop-up -->
132
133         Esta é uma notificação!
134         |     <!-- Texto exibido dentro do pop-up de notificação -->
135
136     </div>
137     |     <!-- Fecha a div 'notificacao' -->
138
139     <!-- Lightbox Pop-up -->
140     <!-- Comentário indicando que o código a seguir
141         |     implementa um pop-up de lightbox -->
142     <div id="lightbox" class="lightbox">
143         |     <!-- Abre uma div com id 'lightbox' e classe 'lightbox'; é
144             |     o contêiner do lightbox, usado principalmente para
145                 |     exibir conteúdo de mídia como imagens de forma
146                     |     destacada sobre o resto da página -->
147
148         <span class="fechar">&times;</span>
149         |     <!-- Inclui um elemento span com classe 'fechar',
150             |     utilizado como botão de fechamento do lightbox -->
```

Ln 121, Col 4 Spaces: 8 UTF-8 CRLF HTML Ø Port: 5500 Reload

EXPLORER ...

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- scripts.js

index.html

```
index.html > html > body > div#lightbox.lightbox
```

151                   <img class="lightbox-conteudo" id="imagemLightbox">  
152                    <!-- Define um elemento img com classe 'lightbox-conteudo'<br/>153                    | e id 'imagemLightbox', onde a imagem a ser  
154                    | exibida no lightbox será carregada -->  
155  
156                    <div id="legenda"></div>  
157                    <!-- Div vazia com id 'legenda', destinada a conter a<br/>158                    | legenda da imagem exibida no lightbox -->  
159  
160                    </div>  
161                    <!-- Fecha a div 'lightbox' --&gt;<br/>162  
163  
164                      
165                    <!-- Define uma imagem oculta na página, que quando clicada,<br/>166                    | aciona a função 'abrirLightbox(this)', passando a  
167                    | própria imagem como argumento para ser exibida  
168                    | no lightbox -->  
169  
170  
171                    <!-- Context Menu Pop-up --&gt;<br/>172  
173                    <!-- Comentário indicando que o código a seguir implementa um<br/>174                    | menu de contexto personalizado -->  
175                    <div id="menuContexto" class="menu-contexto">  
176                    <!-- Abre uma div com id 'menuContexto' e classe 'menu-contexto'; é<br/>177                    | o contêiner do menu de contexto, geralmente oculto e  
178                    | exibido apenas em resposta a uma interação específica,  
179                    | como um clique direito -->  
180

OUTLINE 1

TIMELINE

✖ 0 △ 0 ⌂ 0

Ln 151, Col 6 Spaces: 8 UTF-8 CRLF HTML ⚙ Port: 5500 Reload

EXPLORER

...

P35 - POP-UP

- estilos.css
- img.jpg
- index.html

JS scripts.js

2

OUTLINE

TIMELINE

1

index.html X

index.html > html > body > div#menuContexto.menu-contexto

```
181     <ul>
182         <!-- Lista não ordenada para agrupar as opções
183             do menu de contexto -->
184
185         <li onclick="acao1()">Ação 1</li>
186         <!-- Item de lista que, ao ser clicado, chama a função 'acao1()', 
187             | executando uma ação específica -->
188
189         <li onclick="acao2()">Ação 2</li>
190         <!-- Item de lista que, ao ser clicado, chama a função 'acao2()', 
191             | executando uma ação específica -->
192
193         <li onclick="acao3()">Ação 3</li>
194         <!-- Item de lista que, ao ser clicado, chama a função 'acao3()', 
195             | executando uma ação específica -->
196
197     </ul>
198     </div>
199     <!-- Fecha a div 'menuContexto' -->
200
201     <!-- Confirmação Pop-up -->
202     <!-- Comentário indicando que o código a seguir implementa um
203         | pop-up de confirmação -->
204     <div id="confirmacao" class="confirmacao">
205     <!-- Abre uma div com id 'confirmacao' e classe 'confirmacao'; é
206         | o contêiner do pop-up de confirmação, geralmente usado
207         | para solicitar a confirmação do usuário antes de
208         | realizar uma ação importante -->
209
210     <div class="confirmacao-conteudo">
```



EXPLORER

...

P35 - POP-UP

estilos.css

img.jpg

index.html

scripts.js



&gt; OUTLINE

&gt; TIMELINE

5 index.html X

index.html &gt; html &gt; body &gt; div#confirmacao.confirmacao &gt; div.confirmacao-conteudo

```
211     <!-- Div que contém o conteúdo real do pop-up de confirmação -->
212
213     <span class="fechar">&times;</span>
214     <!-- Inclui um botão de fechar no formato de 'x' que,
215         quando clicado, oculta o pop-up de confirmação -->
216
217     <p>Você tem certeza?</p>
218     <!-- Parágrafo contendo a mensagem de confirmação -->
219
220     <button id="botaoSim">Sim</button>
221     <!-- Botão que, ao ser clicado, pode representar a aceitação
222         do usuário à confirmação proposta -->
223
224     <button id="botaoNao">Não</button>
225     <!-- Botão que, ao ser clicado, pode representar a
226         negação do usuário à confirmação proposta -->
227
228     </div>
229 </div>
230 <!-- Fecha a div 'confirmacao' -->
231
232
233 <!-- Entrada de Texto Pop-up -->
234 <!-- Comentário indicativo de que o código a seguir
235         trata da implementação de um pop-up de entrada de texto -->
236 <div id="entradaTexto" class="entrada-texto">
237     <!-- Abre uma div com id 'entradaTexto' e classe 'entrada-texto'; é
238         o container principal do pop-up de entrada de texto,
239         geralmente oculto até ser ativado por uma ação do usuário -->
```



EXPLORER

...

P35 - POP-UP

estilos.css

img.jpg

index.html

scripts.js



&gt; OUTLINE

&gt; TIMELINE

5 index.html X

5 index.html &gt; html &gt; body &gt; div#entradaTexto.entrada-texto

```
241     <div class="entrada-texto-conteudo">
242         <!-- Div interna que contém os elementos específicos do
243             | pop-up de entrada de texto --&gt;
244
245         &lt;span class="fechar"&gt;&amp;times;&lt;/span&gt;
246         <!-- Inclui um elemento span com classe 'fechar',
247             | utilizado como botão de fechamento do pop-up,
248             | contendo o caractere 'x' --&gt;
249
250         &lt;p&gt;Digite algo:&lt;/p&gt;
251         <!-- Parágrafo que fornece instrução ao usuário sobre
252             | o que fazer, neste caso, digitar algo no
253             | campo de texto --&gt;
254
255         &lt;input type="text" id="textoEntrada"&gt;
256         <!-- Campo de entrada de texto onde os usuários
257             | podem digitar informações --&gt;
258
259         &lt;button id="botaoEnviar"&gt;Enviar&lt;/button&gt;
260         <!-- Botão com id 'botaoEnviar', utilizado para
261             | submeter o texto digitado pelo usuário --&gt;
262
263     &lt;/div&gt;
264     <!-- Fecha a div 'entrada-texto-conteudo' --&gt;
265
266 &lt;/div&gt;
267     <!-- Fecha a div 'entradaTexto' --&gt;
268
269     &lt;script src="scripts.js"&gt;&lt;/script&gt;
270     <!-- Linha que inclui o arquivo de script externo 'scripts.js'.</pre>
```



EXPLORER

...



P35 - POP-UP

estilos.css

img.jpg

index.html

scripts.js



&gt; OUTLINE

&gt; TIMELINE



0 0 0

Ln 271, Col 3 Spaces: 8 UTF-8 CRLF HTML ⚙ Port : 5500 Reload

EXPLORER ...

P35 - POP-UP

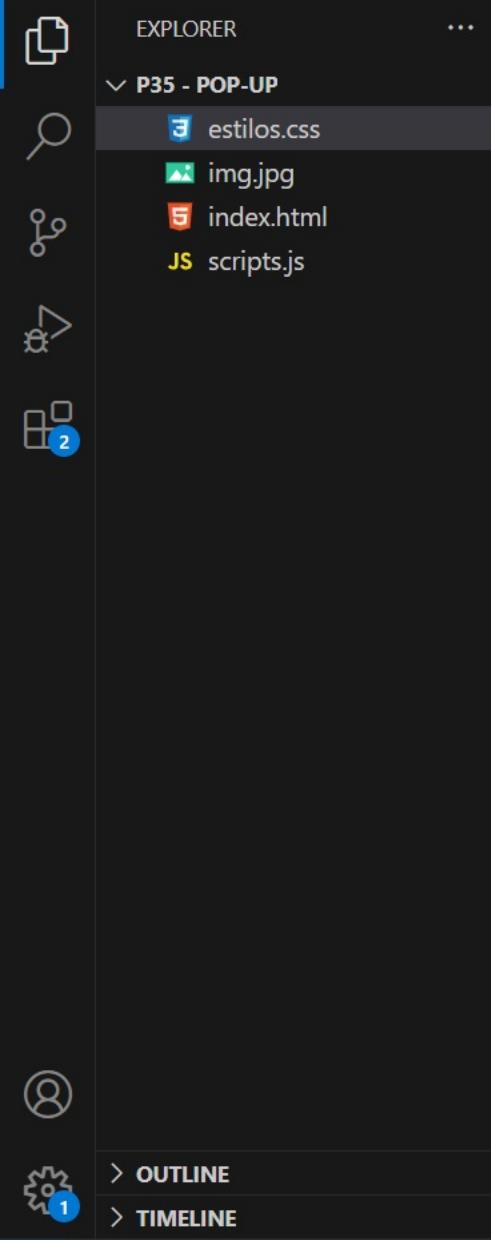
- estilos.css
- img.jpg
- index.html
- scripts.js

estilos.css X

estilos.css > body

```
1 body {  
2     font-family: 'Helvetica Neue', Arial, sans-serif;  
3     /* Define a família da fonte do corpo do documento.  
4         'Helvetica Neue' é a primeira escolha por ser uma  
5             fonte limpa e profissional, seguida por Arial e  
6                 sans-serif como alternativas. */  
7  
8     background-color: #f4f4f9;  
9     /* Define a cor de fundo do corpo do documento como um  
10        cinza muito claro, proporcionando uma  
11            aparência limpa e neutra. */  
12  
13     margin: 0;  
14     /* Remove todas as margens externas do corpo do  
15        documento, garantindo que o conteúdo estenda  
16            até as bordas do navegador. */  
17  
18     padding: 20px;  
19     /* Adiciona um espaçamento interno de 20 pixels em  
20        todas as direções dentro do corpo do documento,  
21            criando espaço entre o conteúdo e as bordas  
22                do navegador. */  
23  
24  
25     color: #333;  
26     /* Define a cor padrão do texto para um cinza escuro,  
27        melhorando a legibilidade em fundos claros. */  
28  
29     font-size: 18px;  
30     /* Aumenta o tamanho padrão da fonte para 18 pixels
```

Ln 1, Col 1 Spaces: 8 UTF-8 CRLF CSS ⚡ Port: 5500 Reload



estilos.css X

estilos.css > body

```
31 |     | para melhorar a legibilidade e acessibilidade. */
32 |
33 }
34
35 h1 {
36
37     text-align: center;
38     /* Centraliza o texto do elemento h1 horizontalmente,
39      | geralmente usado para títulos e cabeçalhos
40      | principais. */
41
42     margin-bottom: 40px;
43     /* Define uma margem inferior de 40 pixels abaixo dos
44      | elementos h1 para separá-los visualmente do
45      | conteúdo que segue. */
46
47     color: #333;
48     /* Define a cor do texto dos elementos h1 para o mesmo
49      | cinza escuro do corpo, mantendo a consistência
50      | da cor do texto em toda a página. */
51
52     font-size: 3em;
53     /* Define o tamanho da fonte dos elementos h1 para 3
54      | vezes o tamanho da fonte base, o que destaca o
55      | título principal e atrai a atenção do usuário. */
56
57 }
58
59
60 .botões {
```

EXPLORER

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- scripts.js

2

OUTLINE

TIMELINE

1

estilos.css X

estilos.css > .botões

```
61 display: flex;
62 /* Define o display como flex, permitindo o uso flexível
63   de alinhamento e distribuição de itens dentro
64   deste contêiner. */
65
66
67 flex-wrap: wrap;
68 /* Permite que os itens dentro do contêiner flexível
69   sejam envolvidos em múltiplas linhas conforme
70   necessário, dependendo do espaço disponível. */
71
72 justify-content: space-around;
73 /* Distribui o espaço restante uniformemente entre os
74   botões, garantindo que haja um espaço consistente
75   em torno de cada botão. */
76
77 gap: 20px;
78 /* Define um espaço de 20 pixels entre cada botão
79   no contêiner flexível. */
80
81 width: 900px;
82 /* Define a largura do contêiner de botões para 900 pixels. */
83
84 height: 100px;
85 /* Define a altura do contêiner de botões para 100 pixels. */
86
87 margin: 0 auto;
88 /* Centraliza o contêiner na página horizontalmente e
89   remove a margem vertical. */
90
```



EXPLORER



### ✓ P35 - POP-UP



3 estilos CSS



```
92     /* Adiciona um espaçamento interno de 20 pixels em todas
93      | as direções dentro do contêiner de botões. */
94 }
95
96
97 .botoes button {
98
99   background: #108478;
100  /* Define a cor de fundo dos botões para um tom
101   | de verde-azulado. */
102
103   color: white;
104  /* Define a cor do texto dos botões para branco,
105   | garantindo uma boa legibilidade contra o
106   | fundo escuro. */
107
108   border: none;
109  /* Remove qualquer borda dos botões para um visual
110   | mais limpo e moderno. */
111
112   padding: 20px 40px;
113  /* Aumenta o espaçamento interno dos botões para 20 pixels
114   | verticalmente e 40 pixels horizontalmente,
115   | melhorando a ergonomia e a visibilidade do texto. */
116
117   text-align: center;
118  /* Centraliza o texto dentro dos botões. */
119
120   text-decoration: none;
```



-  img.jpg
-  index.html
-  scripts.js

JS scripts.js

**IS\_scripts** is

index.html

5 index.html

5 index.html

index.html

mg.jpg

 img.jpg



 img.jpg



img img



⊗ 0 △ 0 Ⓛ 0

Ln 91, Col 11 Spaces: 8 UTF-8 CRLF CSS ⚡ Port: 5500 Reload 🍀

EXPLORER ...

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- scripts.js

estilos.css X

estilos.css > .botões button

```
121 /* Remove qualquer decoração de texto (como sublinhado)
122 | que possa ser herdada. */
123
124 font-size: 32px;
125 /* Aumenta o tamanho da fonte para 32 pixels para fazer os
126 | textos dos botões mais visíveis e acessíveis. */
127
128 margin: 10px;
129 /* Define a margem em torno de cada botão para 10 pixels,
130 | garantindo espaço suficiente entre eles para evitar a
131 | sobreposição e melhorar a estética. */
132
133 width: 900px;
134 /* Define a largura de cada botão para 900 pixels,
135 | garantindo que eles preencham o container. */
136
137 height: 100px;
138 /* Define a altura de cada botão para 100 pixels, oferecendo
139 | uma grande área de toque para interação. */
140
141 cursor: pointer;
142 /* Muda o cursor para um ponteiro quando sobrevoado,
143 | indicando que o botão é clicável. */
144
145 border-radius: 10px;
146 /* Arredonda as bordas dos botões com um raio de 10
147 | pixels para um visual mais suave e moderno. */
148
149 transition: background 0.3s, box-shadow 0.3s;
150 /* Adiciona animações suaves para mudanças de fundo e
```



EXPLORER

...

estilos.css X

▶ □ ⏪ ⏩ ⏴



P35 - POP-UP

estilos.css

img.jpg

index.html

scripts.js



```
estilos.css X
estilos.css > .botoes button
151    |    sombra, melhorando a interatividade visual ao
152    |    interagir com os botões. */
153
154    box-shadow: 0 4px 6px rgba(0,0,0,0.1);
155    /* Aplica uma sombra sutil abaixo dos botões para dar
156    |    uma sensação de elevação e profundidade. */
157
158 }
159
160 .botoes button:hover {
161
162     background: #1cac40;
163     /* Muda a cor de fundo para um verde mais claro quando o
164     |    mouse está sobre o botão, aumentando a resposta
165     |    visual ao hover. */
166
167     box-shadow: 0 6px 8px rgba(0,0,0,0.2);
168     /* Intensifica a sombra sob o botão no hover, aumentando o
169     |    efeito de elevação para uma resposta visual mais
170     |    dinâmica. */
171
172 }
173
174 .botoes button:focus {
175
176     outline: none;
177     /* Remove o contorno padrão que os navegadores
178     |    aplicam ao focar elementos interativos,
179     |    para um controle mais fino sobre a estética. */
180
```



&gt; OUTLINE

&gt; TIMELINE



⊗ 0 ⚡ 0 ⌂ 0

Ln 151, Col 2 Spaces: 8 UTF-8 CRLF CSS ⚡ Port : 5500 Reload ⌂



EXPLORER

...

estilos.css X

&gt; □ ...



P35 - POP-UP

estilos.css

img.jpg

index.html

scripts.js



&gt; OUTLINE

&gt; TIMELINE



⊗ 0 △ 0 ⌂ 0

```
181  box-shadow: 0 0 0 4px rgba(52, 152, 219, 0.5);
182  /* Adiciona um contorno azul claro ao redor do botão
183  | quando focado, proporcionando uma indicação visual
184  | clara do foco enquanto mantém o design limpo. */
185
186 }
187
188 /* Estilos para o modal */
189 .modal {
190
191   display: none;
192   /* O modal está oculto por padrão, só será exibido
193   | mediante uma ação específica (geralmente um
194   | clique de botão). */
195
196   position: fixed;
197   /* A posição fixa é usada para manter o modal no
198   | lugar, independente da rolagem da página. */
199
200   z-index: 10;
201   /* Define a pilha de ordem do modal acima de outros
202   | conteúdos da página (que têm um z-index menor). */
203
204   left: 0;
205   top: 0;
206   /* Posiciona o modal exatamente no topo e à
207   | esquerda da página. */
208
209   width: 100%;
210   height: 100%;
```

EXPLORER ...

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- scripts.js

estilos.css X

estilos.css > .modal

```
211  /* O modal cobre toda a tela ao definir largura e
212  | altura como 100%. */
213
214  overflow: auto;
215  /* Permite a rolagem dentro do modal se o
216  | conteúdo exceder a altura da tela. */
217
218  background-color: □rgba(0,0,0,0.6);
219  /* Fundo semitransparente escurecido para reduzir a
220  | visibilidade do conteúdo abaixo do modal,
221  | ajudando a focar a atenção no próprio modal. */
222
223 }
224
225 .modal-conteudo {
226
227  background-color: ■#fefefe;
228  /* Fundo branco para o conteúdo do modal, garantindo
229  | que o conteúdo seja claramente visível. */
230
231  margin: 5% auto;
232  /* Centraliza o modal na tela verticalmente e
233  | horizontalmente com uma margem superior e
234  | inferior de 5% e margem lateral automática. */
235
236  padding: 50px;
237  /* Espaço interno de 50px em todas as direções dentro do
238  | modal para não aglomerar o conteúdo nos limites do modal. */
239
240  border: none;
```

211, Col 3 Spaces: 8 UTF-8 CRLF CSS ⚙ Port : 5500 Reload

EXPLORER

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- scripts.js

2

1

OUTLINE

TIMELINE

estilos.css X

estilos.css > .modal-conteudo

```
241  /* Remove qualquer borda do modal para um design mais
242  | limpo e moderno. */
243
244  width: 70%; 
245  /* Define a largura do conteúdo do modal como 70% da
246  | largura da tela para não ocupar toda a tela,
247  | mantendo o foco no conteúdo. */
248
249  max-width: 900px;
250  /* Impõe um limite máximo de largura para garantir que o
251  | modal não se torne muito largo em telas grandes. */
252
253  box-shadow: 0 4px 8px rgba(0,0,0,0.1);
254  /* Aplica sombra ao redor do modal para dar uma sensação
255  | de elevação sobre o restante da página. */
256
257  border-radius: 10px;
258  /* Borda arredondada para suavizar as bordas do modal e
259  | proporcionar uma estética mais amigável. */
260
261  text-align: center;
262  /* Centraliza o texto dentro do modal para garantir que o
263  | conteúdo esteja alinhado de maneira uniforme
264  | e centrada. */
265
266  font-size: 1.5em;
267  /* Aumenta o tamanho da fonte dentro do modal para
268  | melhorar a legibilidade e destacar a
269  | importância do conteúdo. */
270
```

EXPLORER ...

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- scripts.js

estilos.css X

estilos.css > .modal-conteudo

```
271 }
272
273 .fechar {
274   color: #aaa;
275   /* Cor cinza clara para o ícone de fechar, fazendo com
276    | que se destaque sobre o fundo branco, mas
277    | não de maneira muito agressiva. */
278
279
280   float: right;
281   /* Posiciona o ícone de fechar no canto superior
282    | direito do modal. */
283
284   font-size: 32px;
285   /* Aumenta o tamanho do ícone de fechar para
286    | torná-lo facilmente clicável. */
287
288   font-weight: bold;
289   /* Torna o ícone de fechar em negrito para
290    | destacar sua importância. */
291
292   cursor: pointer;
293   /* Muda o cursor para um ponteiro quando hover
294    | sobre o ícone de fechar, indicando que
295    | ele é interativo. */
296
297 }
298
299 .fechar:hover,
300 .fechar:focus {
```

2 1

> OUTLINE  
> TIMELINE

✖ 0 △ 0 ⌂ 0

Ln 271, Col 2 Spaces: 8 UTF-8 CRLF CSS ⚡ Port : 5500 Reload

EXPLORER ...

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- scripts.js

estilos.css X

estilos.css > ✎ .fechar:hover

```
301 color: black;
302 /* Muda a cor do ícone de fechar para preto ao
303    passar o mouse ou focar, proporcionando um
304    feedback visual imediato ao usuário. */
305
306 text-decoration: none;
307 /* Remove qualquer decoração de texto para manter o
308    ícone de fechar limpo e distinto. */
309
310 }
311
312 /* Estilos para o tooltip */
313 .tooltip {
314
315 position: relative;
316 /* Define a posição do tooltip como relativa para
317    permitir que o texto do tooltip (tooltipTexto) seja
318    posicionado absolutamente em relação a este contêiner. */
319
320 display: inline-block;
321 /* Muda o display do elemento para inline-block, permitindo
322    que ele tenha uma largura e altura, mas ainda assim
323    flua com outros elementos inline. */
324
325 border-bottom: 1px dotted black;
326 /* Adiciona uma borda pontilhada na parte inferior do
327    tooltip para indicar que há informações adicionais
328    disponíveis ao passar o mouse. */
329
330 }
```

2

OUTLINE

TIMELINE

Ln 301, Col 1 Spaces: 8 UTF-8 CRLF CSS ⚡ Port : 5500 Reload

EXPLORER

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- scripts.js

2

OUTLINE

TIMELINE

1

estilos.css X

estilos.css > .tooltip

```
331    font-size: 1.2em;
332    /* Aumenta o tamanho da fonte para 1.2 vezes o tamanho
333       padrão do texto ao redor, destacando o elemento
334       tooltip. */
335
336 }
337
338 .tooltip .tooltipTexto {
339
340     visibility: hidden;
341     /* Inicialmente oculta o texto do tooltip para que ele
342        não apareça até que o usuário passe o mouse
343        sobre o elemento tooltip. */
344
345     width: 180px;
346     /* Define uma largura fixa para o texto do tooltip para
347        manter a consistência do tamanho, independentemente do
348        conteúdo interno. */
349
350     background-color: black;
351     /* Define a cor de fundo do texto do tooltip para
352        preto, garantindo que o texto branco se destaque. */
353
354     color: #fff;
355     /* Define a cor do texto dentro do tooltip para branco,
356        proporcionando um alto contraste contra o fundo preto. */
357
358     text-align: center;
359     /* Centraliza o texto dentro do tooltip para melhorar a
360        legibilidade e a estética. */
```

EXPLORER ...

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- scripts.js

estilos.css X

estilos.css > .tooltip .tooltipTexto

```
361 border-radius: 6px;
362 /* Aplica bordas arredondadas ao tooltip para um
363 | design mais suave e moderno. */
364
365 padding: 10px 0;
366 /* Adiciona padding vertical dentro do tooltip para dar
367 | espaço ao redor do texto sem alterar a
368 | largura definida. */
369
370
371 position: absolute;
372 /* Posiciona o texto do tooltip de forma absoluta em
373 | relação ao seu contêiner parente (tooltip),
374 | permitindo que ele sobreponha outros conteúdos
375 | conforme necessário. */
376
377 z-index: 1;
378 /* Define o nível de stack do tooltip para garantir que ele
379 | sobreponha outros elementos posicionados de
380 | forma não explícita. */
381
382 bottom: 150%;
383 /* Posiciona o tooltip 150% abaixo do topo do elemento tooltip,
384 | fazendo com que ele apareça abaixo do elemento pai. */
385
386 left: 50%;
387 /* Alinha o centro do tooltip com o centro do elemento pai. */
388
389 margin-left: -90px;
390 /* Desloca o tooltip 90px para a esquerda do centro.
```

Ln 361, Col 2 Spaces: 8 UTF-8 CRLF CSS Ø Port : 5500 Reload

EXPLORER

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- scripts.js

2

OUTLINE

TIMELINE

estilos.css X

estilos.css > .tooltip .tooltipTexto

```
391 |     centralizando-o efetivamente em relação ao elemento pai. */
392 |
393 opacity: 0;
394 /* Define a opacidade inicial do tooltip como 0 para torná-lo
395 | completamente transparente até que seja ativado. */
396 |
397 transition: opacity 0.3s;
398 /* Aplica uma transição suave na propriedade de opacidade
399 | para animar a entrada e saída do tooltip. */
400 |
401 box-shadow: 0 2px 10px rgba(0,0,0,0.1);
402 /* Adiciona uma sombra leve ao redor do tooltip para destacá-lo
403 | visualmente do conteúdo de fundo. */
404 |
405 }
406 |
407 |
408 /* Estilos para exibição do texto do tooltip quando o
409 | mouse está sobre ele */
410 .tooltip:hover .tooltipTexto {
411 |
412 visibility: visible;
413 /* Torna o texto do tooltip visível quando o usuário
414 | passa o mouse sobre o elemento tooltip. */
415 |
416 opacity: 1;
417 /* Altera a opacidade do texto do tooltip para 1, tornando-o
418 | totalmente opaco e claramente visível ao usuário. */
419 |
420 }
```

EXPLORER

...

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- scripts.js

2

OUTLINE

TIMELINE

1

0 0 0 0 0 0

estilos.css X

estilos.css > ...

```
421
422  /* Estilos para a notificação */
423  .notificacao {
424
425      display: none;
426      /* Inicialmente, a notificação é ocultada e não é exibida até
427      | ser ativada por um evento específico no JavaScript. */
428
429      background-color: #555;
430      /* Define a cor de fundo da notificação para um cinza escuro, o
431      | que ajuda o texto branco a se destacar. */
432
433      color: white;
434      /* Define a cor do texto para branco, garantindo que ele
435      | seja legível contra o fundo cinza escuro. */
436
437      text-align: center;
438      /* Centraliza o texto dentro da notificação para melhorar a
439      | legibilidade e a apresentação. */
440
441      border-radius: 5px;
442      /* Aplica bordas arredondadas para suavizar as bordas da
443      | notificação e dar um visual mais amigável e moderno. */
444
445      position: fixed;
446      /* Usa a posição fixa para manter a notificação em uma
447      | posição específica na janela de visualização,
448      | independente da rolagem da página. */
449
450      z-index: 1;
```

EXPLORER

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- scripts.js

2

OUTLINE

TIMELINE

1

estilos.css X

estilos.css > .notificacao

```
451 /* Define a ordem de empilhamento para garantir que a
452     | notificação apareça acima de outros conteúdos que
453     | não tenham um z-index definido. */
454
455 bottom: 30px;
456 /* Posiciona a notificação a 30 pixels da parte inferior
457     | da janela de visualização. */
458
459 right: 30px;
460 /* Posiciona a notificação a 30 pixels da parte direita da
461     | janela de visualização. */
462
463 padding: 20px;
464 /* Adiciona um espaçamento interno de 20 pixels em todas
465     | as direções dentro da notificação para evitar que o
466     | texto fique muito próximo das bordas. */
467
468 box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
469 /* Adiciona uma sombra leve ao redor da notificação para
470     | dar uma sensação de elevação e destacá-la visualmente
471     | do fundo da página. */
472
473 font-size: 1.5em;
474 /* Aumenta o tamanho da fonte para 1.5 vezes o tamanho padrão
475     | do texto, tornando a notificação mais visível e legível. */
476
477 }
478
479 .notificacao.mostrar {
480 }
```



EXPLORER

...

estilos.css X

▶ □ ⏪ ⏩ ⏴



P35 - POP-UP

estilos.css

img.jpg

index.html

scripts.js



&gt; OUTLINE

&gt; TIMELINE



⊗ 0 ▲ 0 ⌂ 0

estilos.css &gt; .notificacao.mostrar

```
481 display: block;
482 /* Altera a propriedade de exibição para 'block' quando a
483 | notificação deve ser mostrada, tornando-a visível
484 | na página. */
485 }
486
487
488 /* Estilos para o lightbox */
489 .lightbox {
490
491 display: none;
492 /* O lightbox é inicialmente oculto e será exibido apenas
493 | quando ativado por uma ação específica, como um clique. */
494
495 position: fixed;
496 /* A posição fixa garante que o lightbox seja posicionado em
497 | relação à janela do navegador, permanecendo visível
498 | mesmo durante a rolagem da página. */
499
500 z-index: 10;
501 /* Define a ordem de empilhamento do lightbox para garantir
502 | que ele fique acima de outros elementos na página. */
503
504 padding-top: 60px;
505 /* Adiciona um espaçamento na parte superior do lightbox para
506 | evitar que o conteúdo fique muito próximo ao topo da página. */
507
508 left: 0;
509 top: 0;
510 /* Posiciona o lightbox no topo esquerdo da janela do
```

EXPLORER ...

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- scripts.js

estilos.css X

estilos.css > .lightbox

```
511 | navegador, cobrindo toda a tela. */
512 width: 100%;
513 height: 100%;
514 /* Estende o lightbox para cobrir toda a largura e
515 | altura da janela do navegador. */
516
517 overflow: auto;
518 /* Permite rolagem dentro do lightbox se o conteúdo
519 | exceder a altura da tela. */
520
521 background-color: #rgba(0,0,0,0.9);
522 /* Usa uma cor de fundo preta com opacidade de 90% para
523 | obscurecer a tela atrás do lightbox, focando a
524 | atenção no conteúdo dentro do lightbox. */
525
526 }
527
528
529 .lightbox-conteudo {
530
531 margin: auto;
532 /* Centraliza o conteúdo do lightbox horizontal e
533 | verticalmente dentro do lightbox. */
534
535 display: block;
536 /* Garante que o conteúdo do lightbox seja tratado como um
537 | bloco, ocupando toda a largura disponível dentro
538 | do seu container. */
539
540 width: 80%;
```

Ln 511, Col 2 Spaces: 8 UTF-8 CRLF CSS ⚡ Port : 5500 Reload



EXPLORER

...

estilos.css X

D □ ...



P35 - POP-UP

estilos.css

img.jpg

index.html

scripts.js



&gt; OUTLINE

&gt; TIMELINE

estilos.css &gt; .lightbox-conteudo

```
541  /* Define a largura do conteúdo do lightbox para 80% da
542  | largura total do lightbox, deixando espaço nas laterais. */
543
544  max-width: 900px;
545  /* Estabelece uma largura máxima para o conteúdo do lightbox
546  | para evitar que ele fique muito largo em telas grandes. */
547
548  box-shadow: 0 4px 8px rgba(0,0,0,0.1);
549  /* Aplica uma sombra ao redor do conteúdo do lightbox para
550  | elevá-lo visualmente e destacá-lo contra o
551  | fundo escurecido. */
552
553  border-radius: 10px;
554  /* Arredonda os cantos do conteúdo do lightbox para suavizar
555  | sua aparência e integrá-lo melhor ao design geral
556  | da página. */
557
558 }
559
560 #legenda {
561
562  text-align: center;
563  /* Centraliza o texto da legenda dentro do elemento. */
564
565  color: #ccc;
566  /* Define a cor do texto para um cinza claro, oferecendo
567  | contraste suficiente contra um fundo potencialmente
568  | escuro, mas de forma sutil. */
569
570  padding: 10px;
```



0 0 0 0 0 0

EXPLORER ...

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- scripts.js

estilos.css X

estilos.css > #legenda

```
571 /* Adiciona espaço interno ao redor do texto da
572 | legenda para separá-lo de outros conteúdos
573 | visuais. */
574
575 font-size: 1.5em;
576 /* Aumenta o tamanho da fonte da legenda para 1.5 vezes o
577 | tamanho normal do texto, tornando-a mais legível e
578 | destacada. */
579
580 }
581
582
583 /* Estilos para o menu de contexto */
584 .menu-contexto {
585
586 display: none;
587 /* O menu de contexto é inicialmente oculto e só é
588 | exibido mediante uma ação específica, como
589 | um clique direito. */
590
591 position: absolute;
592 /* Posicionamento absoluto permite que o menu de
593 | contexto seja posicionado em relação ao seu
594 | contêiner mais próximo com posição não estática. */
595
596 z-index: 1000;
597 /* Um z-index alto assegura que o menu de contexto
598 | apareça acima de outros elementos na página. */
599
600 width: 250px;
```

2

OUTLINE

TIMELINE

Ln 571, Col 3 Spaces: 8 UTF-8 CRLF CSS Ø Port : 5500 Reload



EXPLORER

...

estilos.css X



P35 - POP-UP

estilos.css

img.jpg

index.html

scripts.js



```
601 /* Define uma largura fixa para o menu de contexto,  
602 | proporcionando espaço suficiente para acomodar  
603 | os itens do menu. */  
604  
605 background-color: #fff;  
606 /* Define a cor de fundo do menu de contexto para  
607 | branco, garantindo que os itens sejam  
608 | claramente visíveis. */  
609  
610 box-shadow: 0 4px 8px rgba(0,0,0,0.1);  
611 /* Aplica uma sombra ao redor do menu para dar uma  
612 | sensação de elevação e destacá-lo visualmente do  
613 | restante do conteúdo da página. */  
614  
615 border-radius: 5px;  
616 /* Arredonda as bordas do menu de contexto para um  
617 | visual mais suave e moderno. */  
618  
619 padding: 15px 0;  
620 /* Aplica um padding vertical para garantir que os  
621 | itens do menu não estejam muito próximos das  
622 | bordas superior e inferior. */  
623  
624 }  
625  
626 .menu-contexto ul {  
627  
628 list-style: none;  
629 /* Remove os marcadores de lista, já que não são  
630 | necessários em um menu de contexto. */
```



&gt; OUTLINE

&gt; TIMELINE



0 ▲ 0 ⌂ 0

Ln 601, Col 5 Spaces: 8 UTF-8 CRLF CSS Ø Port : 5500 Reload

EXPLORER ...

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- scripts.js

estilos.css X

estilos.css > .menu-contexto ul

```
631 padding: 0;
632 /* Remove o padding padrão da lista para garantir
   que os itens do menu usem eficientemente o
   espaço horizontal disponível. */
633 margin: 0;
634 /* Remove as margens padrão da lista para evitar
   espaços indesejados em torno dos itens do menu. */
635
636
637 }
638
639
640
641
642
643 .menu-contexto ul li {
644
645   padding: 20px;
646   /* Adiciona um padding substancial em torno de cada
      item da lista para aumentar a área clicável e
      melhorar a usabilidade. */
647
648
649
650   cursor: pointer;
651   /* Muda o cursor para um ponteiro quando sobrevoado,
      indicando que cada item é clicável. */
652
653
654   transition: background-color 0.3s;
655   /* Adiciona uma transição suave para mudanças de cor
      de fundo, melhorando a resposta visual ao
      interagir com os itens do menu. */
656
657
658
659   font-size: 1.5em;
660   /* Aumenta o tamanho da fonte para tornar os textos
```



EXPLORER

...

estilos.css X

...



P35 - POP-UP

estilos.css

img.jpg

index.html

scripts.js



&gt; OUTLINE

&gt; TIMELINE

estilos.css &gt; .menu-contexto ul li

```
661 | dos itens do menu mais legíveis e acessíveis. */
662 |
663 }
664
665 .menu-contexto ul li:hover {
666
667     background-color: #eee;
668     /* Muda a cor de fundo dos itens do menu quando o
669      mouse passa sobre eles, fornecendo um
670      feedback visual imediato. */
671 }
672
673
674 /* Estilos para a janela de confirmação */
675 .confirmacao {
676
677     display: none;
678     /* A janela de confirmação é inicialmente
679      ocultada e só é exibida mediante uma ação
680      específica (como uma solicitação de
681      confirmação). */
682
683     position: fixed;
684     /* A posição fixa garante que a janela de
685      confirmação seja posicionada em relação à
686      janela do navegador, cobrindo todo o conteúdo
687      abaixo dela, independentemente da rolagem da página. */
688
689     z-index: 10;
690     /* Um z-index alto para garantir que a janela de
```

EXPLORER ...

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- scripts.js

estilos.css X

estilos.css > .confirmacao

```
691 |     confirmação apareça acima de outros
692 |     elementos da página. */
693 |
694     left: 0;
695     top: 0;
696     /* Posiciona a janela de confirmação exatamente no
697      topo e à esquerda da página, cobrindo a
698      tela inteira. */
699 |
700     width: 100%;
701     height: 100%;
702     /* Estas propriedades fazem a janela de confirmação
703      cobrir toda a tela, criando uma área de foco
704      para a ação do usuário. */
705 |
706     overflow: auto;
707     /* Permite a rolagem dentro da janela de confirmação
708      se o conteúdo exceder a altura da tela. */
709 |
710     background-color: □rgba(0,0,0,0.6);
711     /* Utiliza uma cor de fundo preta com opacidade de 60%,
712      obscurecendo o conteúdo da página abaixo e
713      focando a atenção do usuário na janela de
714      confirmação. */
715 |
716 }
717 |
718 .confirmacao-conteudo {
719     background-color: ■#fefefe;
```

Ln 691, Col 3 Spaces: 8 UTF-8 CRLF CSS ⚡ Port : 5500 Reload



EXPLORER 111

### ✓ P35 - POP-UP

## 3 estilos.css

 img.jpg

5 index.html

## JS scripts.js

3 estilos.css X

3 estilos.css > .confirmacao-conteudo



EXPLORER

...

estilos.css X

&gt; □ ...



P35 - POP-UP

estilos.css

img.jpg

index.html

scripts.js



&gt; OUTLINE

&gt; TIMELINE



⊗ 0 △ 0 ⌂ 0

estilos.css &gt; .confirmacao-conteudo

```
751 box-shadow: 0 4px 8px rgba(0,0,0,0.1);
752 /* Aplica uma sombra ao redor do conteúdo da janela de
753 | confirmação para dar uma sensação de elevação e
754 | destaca-lo visualmente do fundo. */
755
756 border-radius: 10px;
757 /* Aplica bordas arredondadas ao conteúdo da janela de
758 | confirmação para um design mais suave e moderno. */
759
760 text-align: center;
761 /* Centraliza o texto dentro do conteúdo da janela de
762 | confirmação para melhorar a legibilidade e a
763 | apresentação. */
764
765 font-size: 1.5em;
766 /* Aumenta o tamanho da fonte para 1.5 vezes o tamanho
767 | normal do texto, tornando a janela de confirmação
768 | mais legível e acessível. */
769
770 }
771
772
773 /* Estilos para os botões dentro do conteúdo de confirmação */
774 .confirmacao-conteudo button {
775 background: linear-gradient(135deg, #6EC1E4, #3498db);
776 /* Define um gradiente de fundo começando de um
777 | azul claro (#6EC1E4) a um azul mais escuro (#3498db),
778 | criando um efeito visual atraente. */
779
780 color: white;
```



EXPLORER

...

estilos.css X

...



P35 - POP-UP

estilos.css

img.jpg

index.html

scripts.js



2



&gt; OUTLINE

&gt; TIMELINE



0 0 0 0

estilos.css &gt; .confirmacao-conteudo button

```
781  /* Define a cor do texto dos botões para branco,  
   | garantindo alta legibilidade contra o fundo colorido. */  
782  border: none;  
783  /* Remove a borda do botão para um design mais limpo e integrado. */  
784  
785  padding: 15px 30px;  
786  /* Aumenta o padding do botão para 15 pixels na  
   | vertical e 30 pixels na horizontal, melhorando a  
   | ergonomia e tornando o botão mais fácil de clicar. */  
787  
788  margin: 10px;  
789  /* Adiciona uma margem de 10 pixels ao redor de cada  
   | botão para garantir que eles não fiquem muito  
   | próximos um do outro. */  
790  
791  
792  font-size: 1.2em;  
793  /* Aumenta o tamanho da fonte para 1.2 vezes o tamanho  
   | do texto normal, melhorando a visibilidade do  
   | texto no botão. */  
794  
795  
796  
797  cursor: pointer;  
798  /* Muda o cursor para um ponteiro quando o usuário passa o  
   | mouse sobre o botão, indicando que o elemento é clicável. */  
799  
800  
801  
802  
803  
804  
805  
806  border-radius: 5px;  
807  /* Aplica bordas arredondadas de 5 pixels ao botão para  
   | suavizar o design e torná-lo mais amigável. */  
808  
809  
810  transition: background 0.3s, box-shadow 0.3s;
```



EXPLORER

...

estilos.css X

&gt; □ ...



P35 - POP-UP

estilos.css

img.jpg

index.html

scripts.js



&gt; OUTLINE

&gt; TIMELINE

estilos.css &gt; .confirmacao-conteudo button

```
811  /* Adiciona uma transição suave tanto para o fundo quanto
812   | para a sombra do botão, melhorando a interação
813   | visual quando o estado do botão muda. */
814
815  box-shadow: 2px 2px 10px □rgba(0,0,0,0.1);
816  /* Aplica uma sombra leve ao botão para criar um efeito
817   | de elevação, aumentando o destaque visual do botão. */
818
819 }
820
821 .confirmacao-conteudo button:hover {
822  background: linear-gradient(135deg, □#3498db, □#6EC1E4);
823  /* Inverte o gradiente de fundo do botão ao passar o
824   | mouse, adicionando um feedback dinâmico e visual
825   | para a interação do usuário. */
826
827  box-shadow: 2px 2px 12px □rgba(0,0,0,0.2);
828  /* Intensifica a sombra quando o botão é hoverizado,
829   | aumentando o contraste e destacando ainda mais o botão. */
830
831 }
832
833
834 /* Estilos para a entrada de texto */
835 .entrada-texto {
836
837  display: none;
838  /* A entrada de texto é inicialmente oculta e só aparece
839   | mediante ação específica, similar ao comportamento
840   | de um modal. */
```



EXPLORER

...

estilos.css X



P35 - POP-UP

estilos.css

img.jpg

index.html

scripts.js



&gt; OUTLINE

&gt; TIMELINE



0 ▲ 0 ⌂ 0

```
estilos.css > .entrada-texto
position: fixed;
/* A posição fixa garante que a entrada de texto seja
   posicionada em relação à janela do navegador, e
   não à página, independentemente da rolagem. */

z-index: 10;
/* Um z-index de 10 garante que a entrada de texto apareça
   acima de outros elementos na página, mas abaixo de
   outros possíveis overlays. */

left: 0;
top: 0;
/* Posiciona a entrada de texto no topo esquerdo da
   página, cobrindo toda a tela quando exibida. */

width: 100%;
height: 100%;
/* A entrada de texto cobre toda a largura e altura da
   janela do navegador, criando um fundo completo
   para o usuário focar. */

overflow: auto;
/* Permite a rolagem dentro da entrada de texto se o
   conteúdo exceder a altura da tela, garantindo
   que todo o conteúdo seja acessível. */

background-color: □rgba(0,0,0,0.6);
/* Utiliza um fundo preto semitransparente (60% de opacidade)
   para obscurecer o conteúdo da página abaixo, concentrando a
```



EXPLORER ...

✓ P35 - POP-UP

3 estilos.css

img.jpg

5 index.html

JS scripts.js

3 estilos.css X

```
estilos.css > .entrada-texto
871     |     atenção do usuário na entrada de texto. */
872
873 }
874
875
876 /* Estilos para o conteúdo do popup de entrada de texto */
877 .entrada-texto-conteudo {
878
879     background-color: #fefefe;
880     /* Define a cor de fundo como branco quase puro, criando um
881        | contraste nítido com o texto e outros elementos. */
882
883     margin: 5% auto;
884     /* Ajusta as margens verticalmente para 5% do tamanho da
885        | tela, centralizando horizontalmente o conteúdo dentro
886        | da área do popup. */
887
888     padding: 50px;
889     /* Ajusta um espaçamento interno generoso para evitar que o
890        | conteúdo encoste diretamente nas bordas do popup. */
891
892     border: none;
893     /* Remove qualquer borda predefinida, proporcionando um
894        | design mais limpo e integrado ao restante do layout. */
895
896     width: 70%;
897     /* Define a largura do conteúdo do popup para 70% da largura
898        | total da área visível, focando a atenção do usuário
899        | no centro. */
900
```



EXPLORER

...

estilos.css X



P35 - POP-UP

estilos.css

img.jpg

index.html

scripts.js



```
estilos.css > .entrada-texto-conteudo
901   max-width: 900px;
902   /* Limita a largura máxima do conteúdo a 900px, garantindo
903      que não se torne demasiadamente largo em telas grandes. */
904
905   box-shadow: 0 4px 8px rgba(0,0,0,0.1);
906   /* Adiciona uma sombra suave ao redor do popup para destacá-lo
907      contra o fundo da página, criando uma sensação de profundidade. */
908
909   border-radius: 10px;
910   /* Arredonda as bordas do popup para suavizar sua aparência e
911      torná-lo visualmente mais agradável. */
912
913   text-align: center;
914   /* Centraliza o texto dentro do popup, alinhando qualquer
915      texto ou botão dentro dele ao centro. */
916
917   font-size: 1.5em;
918   /* Aumenta o tamanho da fonte para 1.5 vezes o tamanho
919      normal, melhorando a legibilidade do texto dentro do popup. */
920
921 }
922
923 /* Estilos para o campo de entrada de texto dentro do popup */
924 .entrada-texto-conteudo input[type="text"] {
925
926   width: 80%;
927   /* Define a largura do campo de texto para 80% da largura
928      do container, proporcionando um bom equilíbrio
929      entre a largura e o espaço disponível. */
930 }
```



&gt; OUTLINE

&gt; TIMELINE



0 △ 0 ⌂ 0



EXPLORER

...

estilos.css X

&gt; □ ...

P35 - POP-UP

estilos.css

img.jpg

index.html

scripts.js

```
estilos.css > .entrada-texto-conteudo input[type="text"]
931 padding: 15px;
932 /* Aplica um padding de 15px em todas as direções dentro do
933 campo de entrada, aumentando a área de interação e
934 melhorando a experiência do usuário. */
935
936 margin: 20px 0;
937 /* Define a margem superior e inferior do campo de
938 entrada para 20px, separando-o visualmente de
939 outros elementos dentro do popup. */
940
941 font-size: 1.2em;
942 /* Aumenta o tamanho da fonte dentro do campo de
943 entrada para 1.2 vezes o tamanho normal,
944 facilitando a leitura do texto digitado. */
945
946 border-radius: 5px;
947 /* Arredonda as bordas do campo de entrada, em
948 concordância com o design geral do popup,
949 para uma aparência consistente e suave. */
950
951 border: 1px solid #ccc;
952 /* Adiciona uma borda sutil ao campo de entrada com
953 uma cor cinza claro (#ccc), definindo claramente
954 os limites do campo. */
955
956 }
957
958
959 /* Estilos para o botão dentro do conteúdo
960 de entrada de texto */
```



&gt; OUTLINE

&gt; TIMELINE

EXPLORER

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- scripts.js

2

OUTLINE

TIMELINE

1

estilos.css X

estilos.css > .entrada-texto-conteudo button

```
961 .entrada-texto-conteudo button {  
962     background: linear-gradient(135deg, #6EC1E4, #0d3149);  
963     /* Define um gradiente de fundo que começa com um azul  
964         claro (#6EC1E4) e transita para um azul mais escuro (#0d3149),  
965             criando um efeito visual dinâmico e atraente. */  
966     color: white;  
967     /* Define a cor do texto do botão como branco para  
968         garantir alta legibilidade contra o fundo colorido. */  
969     border: none;  
970     /* Remove a borda do botão para um visual mais limpo e integrado. */  
971     padding: 15px 30px;  
972     /* Define um padding generoso (15px vertical e 30px  
973         horizontal) para melhorar a ergonomia e tornar o  
974             botão mais fácil de clicar. */  
975     margin: 10px;  
976     /* Aplica uma margem de 10px ao redor do botão para evitar  
977         que ele toque diretamente outros elementos ou  
978             as bordas do container. */  
979     font-size: 1.2em;  
980     /* Aumenta o tamanho da fonte para 1.2 vezes o tamanho  
981         normal do texto, facilitando a leitura e  
982             destacando o botão. */  
983     cursor: pointer;
```

EXPLORER ...

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- scripts.js

estilos.css X

estilos.css > .entrada-texto-conteudo button

```
991 /* Muda o cursor para um ponteiro quando o usuário
992 | passa o mouse sobre o botão, indicando interatividade. */
993
994 border-radius: 5px;
995 /* Arredonda as bordas do botão para suavizar sua
996 | aparência e torná-lo esteticamente mais agradável. */
997
998 transition: background 0.3s, box-shadow 0.3s;
999 /* Aplica uma transição suave nas propriedades de fundo e
1000 | sombra do botão, melhorando a experiência visual
1001 | durante as interações do usuário. */
1002
1003 box-shadow: 2px 2px 10px rgba(0,0,0,0.1);
1004 /* Adiciona uma sombra suítil ao redor do botão para criar um
1005 | efeito de elevação e destacar o botão no layout. */
1006
1007 }
1008
1009 .entrada-texto-conteudo button:hover {
1010
1011 background: linear-gradient(135deg, #3498db, #6EC1E4);
1012 /* Inverte o gradiente de fundo do botão ao passar o
1013 | mouse, oferecendo um feedback visual instantâneo e dinâmico. */
1014
1015 box-shadow: 2px 2px 12px rgba(0,0,0,0.2);
1016 /* Intensifica a sombra quando o botão é hoverizado,
1017 | aumentando o efeito de elevação e tornando o botão
1018 | mais proeminente. */
1019
1020 }
```

Ln 991, Col 1 Spaces: 8 UTF-8 CRLF CSS ⚙️ Port : 5500 Reload

EXPLORER ... JS scripts.js X

JS scripts.js > ...

```
1 // Modal Pop-up
2 var modal = document.getElementById("meuModal");
3 // Acessa o elemento HTML que representa o
4 // modal pop-up na página usando seu ID.
5 // Este elemento é o container que será mostrado ou escondido.
6
7 var botaoModal = document.getElementById("botaoModal");
8 // Acessa o elemento HTML que representa o botão que,
9 // quando clicado, irá abrir o modal pop-up.
10
11 var spanFecharModal = document.getElementsByClassName("fechar")[0];
12 // Acessa o primeiro elemento HTML com a classe "fechar"
13 // que geralmente é um botão ou ícone dentro do
14 // modal que permite fechá-lo.
15
16 botaoModal.onclick = function() {
17
18     modal.style.display = "block";
19     // Define o estilo de exibição do modal para 'block',
20     // o que torna o modal visível na tela.
21     // Esse evento é disparado quando o usuário clica
22     // no botão para abrir o modal.
23
24 }
25
26
27 // Fechamento do Modal Pop-up
28 spanFecharModal.onclick = function() {
29
30     modal.style.display = "none";

```



EXPLORER

...

JS scripts.js X

&gt; □ ...



P35 - POP-UP

estilos.css

img.jpg

index.html

JS scripts.js



&gt; OUTLINE

&gt; TIMELINE



JS scripts.js &gt; onclick

```
31 // Define o estilo de exibição do modal para 'none', o
32     // que torna o modal invisível na tela.
33 // Essa função é chamada quando o usuário clica no
34     // elemento span que geralmente contém um ícone de fechar (X).
35
36 }
37
38 // Clique fora do Modal para Fechar
39 window.onclick = function(event) {
40
41     // Verifica se o local do clique foi no modal (na
42         // área fora do conteúdo do modal, mas dentro
43             // do elemento do modal).
44     if (event.target == modal) {
45
46         // Se verdadeiro, altera a propriedade de exibição
47             // para 'none', fechando o modal.
48         // Isso permite que o usuário clique fora do
49             // conteúdo do modal para fechá-lo.
50         modal.style.display = "none";
51
52     }
53 }
54
55
56 // Alert Pop-up
57 var botaoAlerta = document.getElementById("botaoAlerta");
58 // Acessa o elemento HTML que representa o botão que,
59     // quando clicado, irá disparar um alerta pop-up.
```

EXPLORER

...

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- JS scripts.js

2

OUTLINE

TIMELINE

0 0 0 0 0 0

JS scripts.js X

JS scripts.js > ...

```
61 botaoAlerta.onclick = function() {
62
63     alert("Esta é uma mensagem de alerta!");
64     // Define o que acontece quando o botão de alerta é clicado.
65     // Mostra uma caixa de alerta com a mensagem "Esta é
66         // uma mensagem de alerta!".
67     // A função `alert` é uma função nativa do navegador que
68         // exibe uma caixa de diálogo com uma mensagem
69             // especificada e espera que o usuário pressione 'OK'.
70
71 }
72
73 // Definição de variável para o botão que ativa a notificação.
74 var botaoNotificacao = document.getElementById("botaoNotificacao");
75 // Recupera o elemento do DOM que representa o botão de
76     // notificação usando seu ID.
77
78 botaoNotificacao.onclick = function() {
79     // Adiciona um manipulador de eventos de clique ao
80         // botão de notificação.
81
82     var notificacao = document.getElementById("notificacao");
83     // Recupera o elemento do DOM que representa a
84         // notificação pop-up usando seu ID.
85
86     notificacao.className = "notificacao mostrar";
87     // Define a classe do elemento de notificação para
88         // incluir 'mostrar', que geralmente é usado para
89             // tornar o elemento visível modificando seu
90                 // estilo, como 'display: block'.
```

EXPLORER

...

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- JS scripts.js

2

OUTLINE

TIMELINE

1

JS scripts.js X

JS scripts.js > onclick

```
91    setTimeout(function() {
92        // Inicia um temporizador que executa uma função após um
93        // intervalo especificado. Aqui, o intervalo é
94        // de 3000 milissegundos, ou 3 segundos.
95
96
97        notificacao.className = notificacao.className.replace("mostrar", "");
98        // Após 3 segundos, remove a classe 'mostrar' do
99        // elemento de notificação.
100       // Isso geralmente reverte a visibilidade do elemento,
101       // fazendo com que ele desapareça ou se torne 'display: none'.
102    }, 3000);
103
104 }
105
106
107 // Recuperação de elementos do DOM necessários para o
108 // funcionamento do lightbox.
109
110 // Acessa o botão que, quando clicado, irá abrir o lightbox.
111 var botaoLightbox = document.getElementById("botaoLightbox");
112
113 // Acessa o elemento principal do lightbox que inclui o
114 // fundo escurecido e a área onde a imagem será mostrada.
115 var lightbox = document.getElementById("lightbox");
116
117 // Acessa o elemento de imagem dentro do lightbox, onde a
118 // imagem a ser ampliada será exibida.
119 var imagemLightbox = document.getElementById("imagemLightbox");
120
```



EXPLORER ...

✓ P35 - POP-UP

## 3 estilos.css

 img.jpg

## 5 index.html

## JS scripts.js

JS scripts.js X

JS scripts.js > ..

```
121 // Acessa o segundo elemento com a classe 'fechar' que
122     // geralmente está associado ao botão ou ícone
123     // para fechar o lightbox.
124 var spanFecharLightbox = document.getElementsByClassName("fechar")[1];
125
126 // Define o manipulador de eventos para o clique no
127     // botão que abre o lightbox.
128 botaoLightbox.onclick = function() {
129     // A função é ativada quando o botão 'botaoLightbox' é clicado.
130
131     // Acessa o elemento de imagem que supostamente deve
132         // ser mostrado no lightbox.
133 var imagem = document.getElementById("imagem");
134
135     // Inicialmente, garante que o elemento de imagem
136         // esteja visível para manipulação.
137 imagem.style.display = "block";
138
139     // Programaticamente dispara um evento de clique na imagem.
140     // Isso pode ser usado para disparar outro handler que, por
141         // exemplo, poderia ser usado para exibir a
142             // imagem em um lightbox.
143 imagem.click();
144
145     // Immediatamente oculta a imagem novamente. Isso pode
146         // ser parte de um comportamento de pré-carregamento ou
147             // uma maneira de garantir que a imagem só seja
148                 // visível dentro do lightbox.
149 imagem.style.display = "none";
150
```



EXPLORER

...

JS scripts.js X

▷ ⏪ ⏴ ⏵



P35 - POP-UP

estilos.css

img.jpg

index.html

JS scripts.js



&gt; OUTLINE

&gt; TIMELINE

JS scripts.js &gt; ⚒ onclick

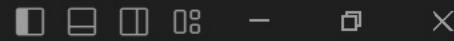
```
151 }  
152  
153  
154 // Define a ação quando o ícone de fechar (usualmente  
155 // um 'X') dentro do lightbox é clicado.  
156 spanFecharLightbox.onclick = function() {  
157  
158 // Configura o display do lightbox para 'none',  
159 // efetivamente ocultando o lightbox.  
160 lightbox.style.display = "none";  
161 }  
162  
163  
164 // Adiciona um ouvinte de evento ao objeto window para  
165 // capturar cliques em qualquer lugar da janela.  
166 window.onclick = function(event) {  
167  
168 // Verifica se o clique ocorreu diretamente no  
169 // fundo do lightbox (não nos conteúdos).  
170 if (event.target == lightbox) {  
171  
172 // Se o fundo do lightbox foi clicado, oculta o lightbox.  
173 lightbox.style.display = "none";  
174 }  
175 }  
176 }  
177  
178 // Define a função para abrir o lightbox, que é chamada  
179 // quando uma imagem que deve abrir o lightbox é clicada.  
180 function abrirLightbox(elemento) {
```



EXPLORER

...

JS scripts.js X



▷ ⏪ ⏹ ⏷



P35 - POP-UP

estilos.css

img.jpg

index.html

JS scripts.js



JS scripts.js &gt; ⓘ abrirLightbox

```
181 // Ajusta o display do lightbox para 'block', tornando-o visível.  
182 lightbox.style.display = "block";  
183  
184 // Define o atributo 'src' da imagem dentro do  
185 // lightbox para corresponder ao 'src' do elemento clicado.  
186 imagemLightbox.src = elemento.src;  
187  
188 // Define o conteúdo HTML do elemento de legenda dentro do  
189 // lightbox para corresponder ao texto  
190 // alternativo ('alt') do elemento clicado.  
191 // Isso fornece contexto adicional ou descrição da imagem mostrada.  
192 document.getElementById("legenda").innerHTML = elemento.alt;  
193  
194 }  
195  
196  
197 // Context Menu Pop-up  
198 // Acessa o botão que, quando clicado, irá abrir o menu de contexto.  
199 var botaoMenuContexto = document.getElementById("botaoMenuContexto");  
200  
201 // Acessa o elemento do DOM que representa o menu de contexto.  
202 var menuContexto = document.getElementById("menuContexto");  
203  
204 // Define o manipulador de eventos para o clique no  
205 // botão que abre o menu de contexto.  
206 botaoMenuContexto.onclick = function(event) {  
207 // A função é ativada quando o botão 'botaoMenuContexto' é clicado.  
208  
209 event.preventDefault();  
210 // Previne a ação padrão do evento de clique. Para um
```

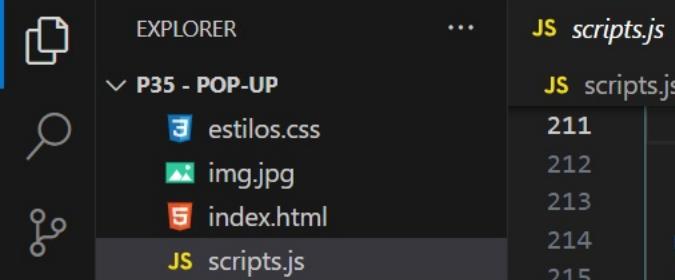


&gt; OUTLINE

&gt; TIMELINE



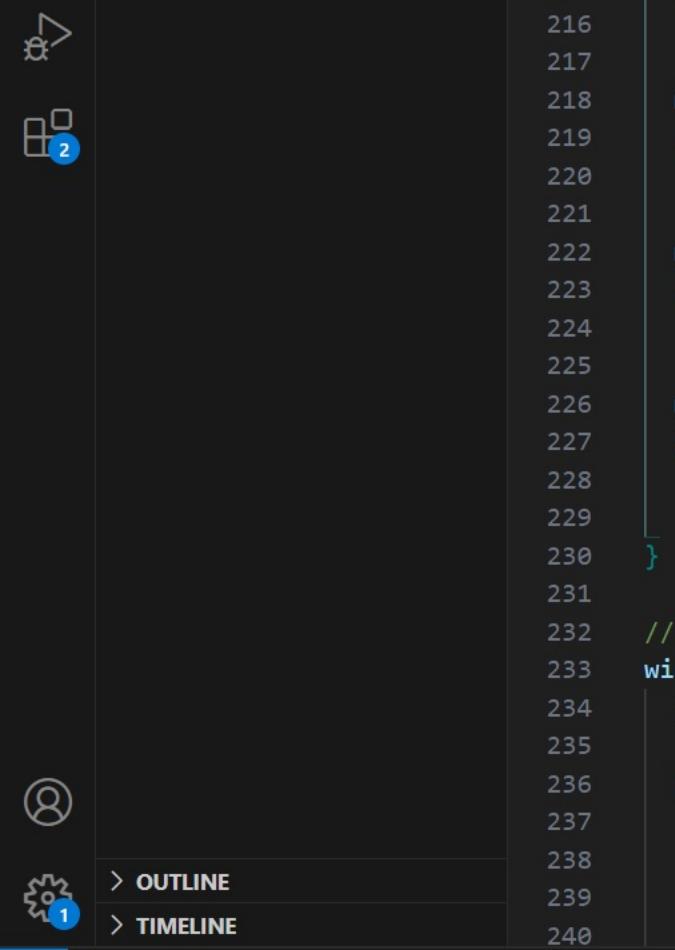
⊗ 0 △ 0 ⌂ 0



JS scripts.js X

JS scripts.js > onclick

```
211     // botão, isso normalmente previne a submissão de
212     // um formulário ou a navegação para um link.
213
214     menuContexto.style.top = "50%";
215     // Define a posição superior do menu de contexto
216     // para 50% da altura da janela, centralizando-o verticalmente.
217
218     menuContexto.style.left = "50%";
219     // Define a posição esquerda do menu de contexto para 50% da
220     // largura da janela, centralizando-o horizontalmente.
221
222     menuContexto.style.transform = "translate(-50%, -50%)";
223     // Aplica uma transformação CSS para ajustar a posição do
224     // menu de contexto, movendo-o para o centro exato da janela.
225
226     menuContexto.style.display = "block";
227     // Torna o menu de contexto visível ao definir seu
228     // display para 'block'.
229
230 }
231
232 // Define um ouvinte de evento para cliques na janela inteira.
233 window.onclick = function(event) {
234     // A função é ativada para qualquer clique na janela.
235
236     if (event.button !== 2 && event.target !== botaoMenuContexto) {
237         // Verifica se o botão do mouse não é o botão
238         // direito (botão 2) e se o alvo do clique não é
239         // o próprio botão do menu de contexto.
```





EXPLORER

...

JS scripts.js X

...

P35 - POP-UP

estilos.css

img.jpg

index.html

JS scripts.js



JS scripts.js &gt; onclick

```
241     menuContexto.style.display = "none";
242     // Se as condições forem verdadeiras, oculta o menu
243     // de contexto definindo seu display para 'none'.
244
245 }
246
247
248 // Funções para as ações do menu de contexto
249 function acao1() {
250
251     // Exibe um alerta com a mensagem "Ação 1 selecionada"
252     alert("Ação 1 selecionada");
253
254     // Oculta o menu de contexto definindo seu display para 'none'
255     menuContexto.style.display = "none";
256
257 }
258
259 function acao2() {
260
261     // Exibe um alerta com a mensagem "Ação 2 selecionada"
262     alert("Ação 2 selecionada");
263
264     // Oculta o menu de contexto definindo seu display para 'none'
265     menuContexto.style.display = "none";
266
267 }
268
269 function acao3() {
270 }
```



0 ▲ 0 ⌂ 0

Ln 241, Col 12 Spaces: 8 UTF-8 CRLF {} JavaScript Ø Port : 5500 Reload

EXPLORER

...

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- JS scripts.js

2 OUTLINE

1 TIMELINE

JS scripts.js X

JS scripts.js > acao3

```
271 // Exibe um alerta com a mensagem "Ação 3 selecionada"
272 alert("Ação 3 selecionada");
273
274 // Oculta o menu de contexto definindo seu display para 'none'
275 menuContexto.style.display = "none";
276 }
277
278
279
280 // Confirmacao Pop-up
281 // Acessa o botão que, quando clicado, irá abrir o pop-up de confirmação.
282 var botaoConfirmacao = document.getElementById("botaoConfirmacao");
283
284 // Acessa o elemento do DOM que representa o pop-up de confirmação.
285 var confirmacao = document.getElementById("confirmacao");
286
287 // Acessa o terceiro elemento com a classe 'fechar' que é
288 // usado para fechar o pop-up de confirmação.
289 var spanFecharConfirmacao = document.getElementsByClassName("fechar")[2];
290
291 // Acessa o botão dentro do pop-up de confirmação que
292 // corresponde à ação "Sim".
293 var botaoSim = document.getElementById("botaoSim");
294
295 // Acessa o botão dentro do pop-up de confirmação que
296 // corresponde à ação "Não".
297 var botaoNao = document.getElementById("botaoNao");
298
299 // Define o manipulador de eventos para o clique no
300 // botão que abre o pop-up de confirmação.
```

EXPLORER

... JS scripts.js X

P35 - POP-UP

- estilos.css
- img.jpg
- index.html
- JS scripts.js

2

OUTLINE

TIMELINE

0 0 △ 0 ⌂ 0

```
JS scripts.js > ...
301 botaoConfirmacao.onclick = function() {
302     // A função é ativada quando o botão 'botaoConfirmacao' é clicado.
303
304     // Torna o pop-up de confirmação visível ao definir
305     // seu display para 'block'.
306     confirmacao.style.display = "block";
307
308 }
309
310
311 // Define o manipulador de eventos para o clique no
312 // ícone de fechar dentro do pop-up de confirmação.
313 spanFecharConfirmacao.onclick = function() {
314     // A função é ativada quando o ícone de fechar é clicado.
315
316     confirmacao.style.display = "none";
317     // Oculta o pop-up de confirmação definindo seu
318     // display para 'none'.
319
320 }
321
322 // Define o manipulador de eventos para o clique no
323 // botão "Sim" dentro do pop-up de confirmação.
324 botaoSim.onclick = function() {
325     // A função é ativada quando o botão "Sim" é clicado.
326
327     confirmacao.style.display = "none";
328     // Oculta o pop-up de confirmação definindo seu
329     // display para 'none'.
330 }
```



EXPLORER

...

JS scripts.js X

D □ ...



P35 - POP-UP

estilos.css

img.jpg

index.html

JS scripts.js



&gt; OUTLINE

&gt; TIMELINE

JS scripts.js &gt; ⚒ onclick

```
331     alert("Você clicou em Sim!");
332     // Exibe uma mensagem de alerta informando que o
333     // usuário clicou em "Sim".
334
335 }
336
337 // Define o manipulador de eventos para o clique no
338 // botão "Não" dentro do pop-up de confirmação.
339 botaoNao.onclick = function() {
340     // A função é ativada quando o botão "Não" é clicado.
341
342     confirmacao.style.display = "none";
343     // Oculta o pop-up de confirmação definindo
344     // seu display para 'none'.
345
346 }
347
348 // Define um ouvinte de evento para cliques
349 // na janela inteira.
350 window.onclick = function(event) {
351     // A função é ativada para qualquer clique na janela.
352
353     if (event.target == confirmacao) {
354         // Verifica se o clique ocorreu diretamente no
355         // fundo do pop-up de confirmação (não nos conteúdos).
356
357         confirmacao.style.display = "none";
358         // Se o fundo do pop-up de confirmação foi
359         // clicado, oculta o pop-up de confirmação
360         // definindo seu display para 'none'.
```



EXPLORER

...

JS scripts.js X



P35 - POP-UP

estilos.css

img.jpg

index.html

JS scripts.js

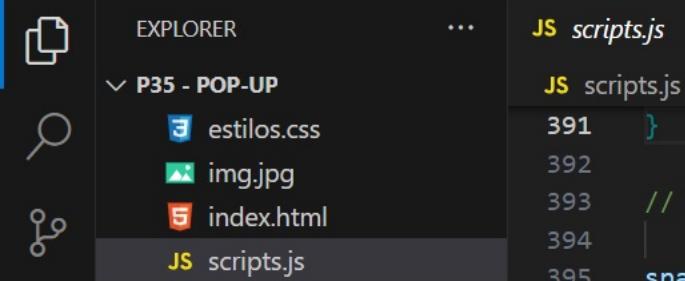


&gt; OUTLINE

&gt; TIMELINE

JS scripts.js &gt; onclick

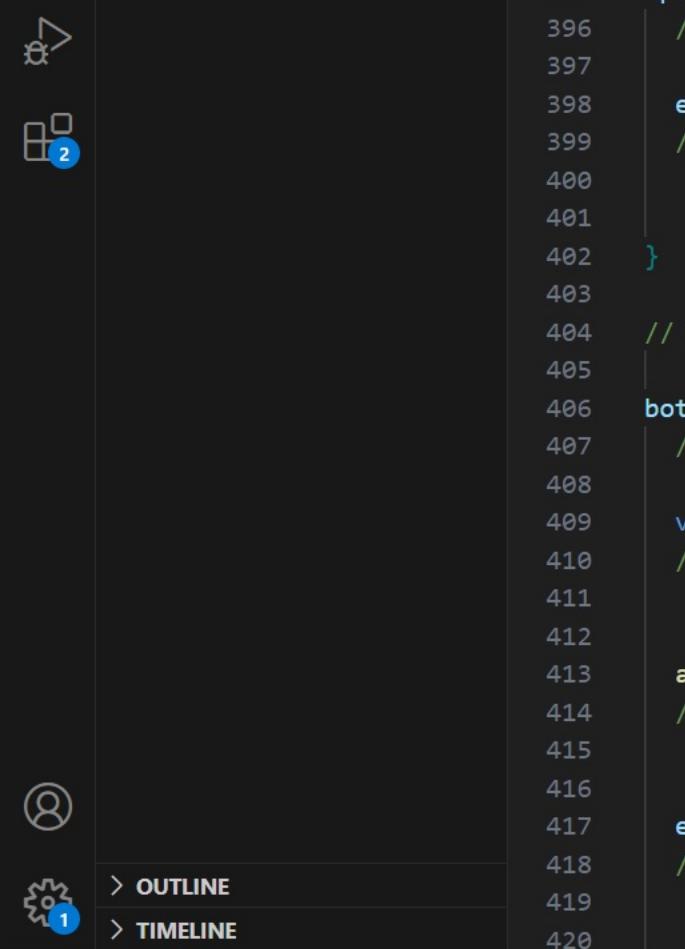
```
361
362    }
363 }
364
365 // Entrada de Texto Pop-up
366 // Acessa o botão que, quando clicado, irá abrir o
367 // pop-up de entrada de texto.
368 var botaoEntradaTexto = document.getElementById("botaoEntradaTexto");
369
370 // Acessa o elemento do DOM que representa o pop-up
371 // de entrada de texto.
372 var entradaTexto = document.getElementById("entradaTexto");
373
374 // Acessa o quarto elemento com a classe 'fechar',
375 // que é usado para fechar o pop-up de entrada de texto.
376 var spanFecharEntradaTexto = document.getElementsByClassName("fechar")[3];
377
378 // Acessa o botão dentro do pop-up de entrada de texto que
379 // será usado para enviar o texto digitado.
380 var botaoEnviar = document.getElementById("botaoEnviar");
381
382 // Define o manipulador de eventos para o clique no
383 // botão que abre o pop-up de entrada de texto.
384 botaoEntradaTexto.onclick = function() {
385     // A função é ativada quando o botão 'botaoEntradaTexto' é clicado.
386
387     entradaTexto.style.display = "block";
388     // Torna o pop-up de entrada de texto visível ao
389     // definir seu display para 'block'.
```



JS scripts.js X

JS scripts.js > onclick

```
391 }
392
393 // Define o manipulador de eventos para o clique no
394 |    // ícone de fechar dentro do pop-up de entrada de texto.
395 spanFecharEntradaTexto.onclick = function() {
396     // A função é ativada quando o ícone de fechar é clicado.
397
398     entradaTexto.style.display = "none";
399     // Oculta o pop-up de entrada de texto definindo
400     |    // seu display para 'none'.
401
402 }
403
404 // Define o manipulador de eventos para o clique no
405 |    // botão 'Enviar' dentro do pop-up de entrada de texto.
406 botaoEnviar.onclick = function() {
407     // A função é ativada quando o botão 'botaoEnviar' é clicado.
408
409     var texto = document.getElementById("textoEntrada").value;
410     // Recupera o valor digitado pelo usuário no
411     |    // campo de entrada de texto.
412
413     alert("Você digitou: " + texto);
414     // Exibe uma mensagem de alerta mostrando o
415     |    // texto digitado pelo usuário.
416
417     entradaTexto.style.display = "none";
418     // Oculta o pop-up de entrada de texto definindo seu
419     |    // display para 'none'.
420 }
```





EXPLORER ...

✓ P35 - POP-UP

3 estilos.css

img.jpg

## 5 index.html

JS scripts.js X

JS scripts.js >  onclick

```
421 }
422
423 // Define um ouvinte de evento para cliques
424 // na janela inteira.
425 window.onclick = function(event) {
426     // A função é ativada para qualquer clique na janela.
427
428     if (event.target == entradaTexto) {
429         // Verifica se o clique ocorreu diretamente no
430         // fundo do pop-up de entrada de
431         // texto (não nos conteúdos).
432
433         entradaTexto.style.display = "none";
434         // Se o fundo do pop-up de entrada de texto foi
435         // clicado, oculta o pop-up definindo seu
436         // display para 'none'.
437
438     }
439 }
```

# **UC: Projetos**

## **Assunto:**

### **Small Projects**

#### **P36 – Scroller de Texto**



EXPLORER

...

index.html X

&gt; □ ...



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



&gt; OUTLINE

&gt; TIMELINE

index.html &gt; ...

```
1  <!DOCTYPE html>
2  <!-- Essa linha informa ao navegador que este é um
3      documento HTML5. O DOCTYPE é necessário para ativar o
4      modo de padrões de conformidade, assegurando que o
5      conteúdo seja interpretado de forma correta. -->
6
7  <html lang="pt-br">
8  <!-- Abre a tag HTML e define o idioma do documento como
9      Português do Brasil. Isso ajuda os navegadores e as
10     ferramentas de SEO a entenderem o idioma principal da
11     página, o que é importante para acessibilidade e
12     indexação correta pela busca. -->
13
14 <head>
15 <!-- Abre a tag HEAD, que contém metadados (dados sobre dados) que
16     geralmente não são exibidos diretamente na página web.
17     Eses dados incluem scripts, instruções de estilo,
18     metainformações, etc. -->
19
20 <meta charset="UTF-8">
21 <!-- A tag META com o atributo charset define o conjunto de
22     caracteres para o documento HTML. UTF-8 é um padrão de
23     codificação de caracteres que inclui quase todos os
24     caracteres de todos os idiomas do mundo, proporcionando uma
25     boa interoperabilidade internacional. -->
26
27 <title>Exemplos de Efeitos</title>
28 <!-- A tag TITLE define o título do documento, que é mostrado na
29     aba do navegador. É uma informação crucial para SEO e
30     usabilidade, ajudando os usuários a identificar o
```



EXPLORER

...

index.html X

&gt; □ ...



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



&gt; OUTLINE

&gt; TIMELINE

index.html &gt; html &gt; head

```
31     conteúdo da aba. -->
32
33     <link rel="stylesheet" href="estilos.css">
34     <!-- Esta linha liga um arquivo CSS externo, chamado estilos.css,
35         ao documento HTML. Este arquivo contém os estilos
36         que serão aplicados ao documento HTML. O atributo 'rel'
37         especifica o relacionamento entre o documento HTML e
38         o arquivo vinculado. Neste caso, é uma folha
39         de estilo. --&gt;
40
41     &lt;/head&gt;
42     <!-- Fecha a tag HEAD. Todo o conteúdo dentro de &lt;head&gt; é processado
43         pelo navegador, mas não é exibido diretamente
44         na página web. --&gt;
45
46     &lt;body&gt;
47     <!-- A tag BODY contém todo o conteúdo visível da página web,
48         como texto, imagens, links, etc. É aqui que você
49         coloca tudo o que os usuários irão interagir
50         diretamente. --&gt;
51
52     &lt;div class="container"&gt;
53     <!-- Divisão com a classe 'container', geralmente usada para
54         centralizar o conteúdo da página e dar um controle
55         mais fácil sobre o layout através do CSS. --&gt;
56
57     &lt;h1&gt;Exemplos de Efeitos Visuais&lt;/h1&gt;
58     <!-- A tag H1 define um cabeçalho, que é geralmente o
59         título mais importante na página. Os motores de
60         busca utilizam esses cabeçalhos para entender</pre>
```



EXPLORER

...

index.html X

&gt; □ ...



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



&gt; OUTLINE

&gt; TIMELINE

61

melhor o conteúdo principal da página, impactando SEO. --&gt;

62

```
<div class="botoes">
    <!-- Divisão com a classe 'botoes', contendo um conjunto de
        botões. Esta classe pode ser usada para aplicar
        estilos específicos aos botões dentro dela
        através do CSS. --&gt;</pre>
```

63

```
<button onclick="abrirPagina('fade.html')>Efeito de Fade</button>
    <!-- Botão que, quando clicado, aciona a função 'abrirPagina'
        com 'fade.html' como argumento, redirecionando o
        navegador para a página 'fade.html'. O texto no
        botão é 'Efeito de Fade'. --&gt;</pre>
```

64

```
<button onclick="abrirPagina('bounce.html')>Efeito de Bounce</button>
    <!-- Botão que inicia a função 'abrirPagina' redirecionando
        para 'bounce.html'. O texto no botão
        é 'Efeito de Bounce'. --&gt;</pre>
```

65

```
<button onclick="abrirPagina('zoom.html')>Efeito de Zoom</button>
    <!-- Botão que, ao ser clicado, executa a função 'abrirPagina'
        e carrega 'zoom.html'. O texto no
        botão é 'Efeito de Zoom'. --&gt;</pre>
```

66

```
<button onclick="abrirPagina('rotate.html')>Efeito de Rotação</button>
    <!-- Botão que chama 'abrirPagina' com 'rotate.html' como
        parâmetro, carregando essa página. O texto no
        botão é 'Efeito de Rotação'. --&gt;</pre>
```

67

```
<button onclick="abrirPagina('image.html')>Imagen e Texto</button>
```

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90



EXPLORER

...

index.html X



&gt; □ ...



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

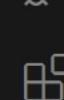
rotate.html

script.js

zoom.html

index.html > html > body > div.container > div.botoes

```
91      <!-- Botão que quando clicado, aciona a função 'abrirPagina'  
92          direcionando para 'image.html'. O texto no botão  
93          explica o conteúdo a ser visto: 'Imagen e Texto'. -->  
94  
95      <button onclick="abrirPagina('led.html')>Efeito de LED</button>  
96      <!-- Botão que dispara a função 'abrirPagina' redirecionando  
97          para 'led.html'. O texto no botão é 'Efeito de LED'. -->  
98  
99      </div>  
100     </div>  
101     <script>  
102         /* Início da tag SCRIPT, que é usada para embutir ou referenciar  
103             código JavaScript executável diretamente no HTML.  
104             Todo código dentro desta tag é tratado como  
105             JavaScript pelo navegador. */  
106  
107     function abrirPagina(pagina) {  
108         /* Declaração da função 'abrirPagina'. A função recebe um  
109             parâmetro 'pagina', que se espera ser uma string  
110             representando o caminho para outra página HTML.  
111             As funções são blocos de código que realizam uma  
112             tarefa específica e podem ser chamadas sempre  
113             que necessário. */  
114  
115         window.location.href = pagina;  
116         /* A linha de código dentro da função modifica a  
117             propriedade 'href' do objeto 'location', que é  
118             parte do objeto 'window'. 'window' é um objeto  
119             global que representa a janela do navegador.  
120             A propriedade 'location.href' pode ser usada para
```



&gt; OUTLINE

&gt; TIMELINE



0 ▲ 0 ⌂ 0





EXPLORER

...

index.html X

&gt; □ ...



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

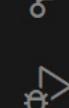
rotate.html

script.js

zoom.html

index.html &gt; html &gt; body &gt; script &gt; abrirPagina

```
121      obter ou definir a URL atual da janela. Aqui,  
122      ela é definida para o valor do parâmetro 'pagina',  
123      causando um redirecionamento da janela do navegador  
124      para a nova URL especificada. Isso muda a página  
125      atual para a página especificada pelo  
126      argumento 'pagina'. */  
127  }  
128  /* Encerramento da função 'abrirPagina'. */  
129  
130 </script>  
131 <!-- Fim da tag SCRIPT. -->  
132  
133 </body>  
134 </html>  
135 <!-- Fim da tag BODY e do documento HTML. -->
```



&gt; OUTLINE

&gt; TIMELINE



0 0 0 0

Ln 121, Col 8 Spaces: 8 UTF-8 CRLF HTML Ø Port : 5500 Reload



EXPLORER

...

fade.html X



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



fade.html &gt; ...

```
1  <!DOCTYPE html>
2  <!-- Essa declaração indica ao navegador que a
3      | versão do documento é HTML5. É o padrão
4      | atual para HTML e assegura que recursos modernos
5      | sejam corretamente interpretados pelo navegador. -->
6
7  <html lang="pt-br">
8  <!-- Abre o elemento <html>, que serve como o contêiner
9      | raiz de todo o documento HTML. O atributo 'lang="pt-br"' 
10     | especifica que o idioma do conteúdo da página é
11     | português brasileiro, que ajuda na acessibilidade e
12     | na indexação correta pelo Google e outros
13     | motores de busca. -->
14
15 <head>
16 <!-- A tag <head> encapsula informações sobre o documento
17     | que não são diretamente exibidas na página da web,
18     | incluindo links para scripts e folhas de estilo,
19     | metadados e outras instruções para o navegador. -->
20
21 <meta charset="UTF-8">
22 <!-- Esta linha define a codificação de caracteres para UTF-8,
23     | que inclui suporte para praticamente todos os
24     | caracteres de qualquer idioma. Isso garante que
25     | qualquer texto no documento seja exibido corretamente. -->
26
27 <title>Efeito de Fade</title>
28 <!-- A tag <title> define o título da página, que é
29     | mostrado na aba do navegador. Este título é
30     | também usado pelos motores de busca e pode
```





EXPLORER

...

fade.html X

&gt; □ ...



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



&gt; OUTLINE

&gt; TIMELINE



0 ▲ 0 ⌂ 0

61

```
62     <script src="script.js"></script>
63     <!-- A tag &lt;script&gt; com o atributo 'src' que aponta
64     para 'script.js', onde estão
65     contidos os scripts JavaScript que gerenciam o
66     comportamento dinâmico do scroller, como iniciar e
67     manter o efeito de fade ou responder a
68     interações do usuário. --&gt;
69
70 &lt;/body&gt;
71 &lt!-- Encerramento da tag &lt;array&gt; que delimita todo o
72     conteúdo visível da página. --&gt;
73
74 &lt;/html&gt;
75 &lt!-- Fim do documento HTML. Toda a estrutura básica
76     necessária para a página está completa. --&gt;</pre>
```



EXPLORER

...



P36 - SCROLLER DE TEXTO

bounce.html



carro.jpg



estilos.css



fade.html



fundo\_led\_2.jpg



fundo\_led\_3.jpg



fundo\_led.jpg



image.html



index.html



led.html



rotate.html



script.js



zoom.html

bounce.html X

bounce.html &gt; ...

```
1  <!DOCTYPE html>
2  <!-- A linha DOCTYPE declara o tipo do documento
3          como HTML5, que é a versão mais recente do HTML.
4          Isso ajuda o navegador a renderizar a página de
5          acordo com as regras do HTML5. -->
6
7  <html lang="pt-br">
8  <!-- A tag <html> inicia o documento HTML, com o atributo 'lang'
9          especificando o idioma do conteúdo da página como
10         português do Brasil. Isso é útil para acessibilidade
11         e SEO, indicando aos navegadores e ferramentas de
12         pesquisa o idioma principal da página. -->
13
14 <head>
15   <!-- A tag <head> contém metadados sobre o documento HTML,
16       que não são exibidos diretamente na página web,
17       mas são importantes para o navegador. -->
18
19 <meta charset="UTF-8">
20   <!-- Define o conjunto de caracteres usado no documento
21       como UTF-8, que suporta quase todos os caracteres
22       de todos os idiomas e é a codificação mais
23       comum para web. Isso garante que o texto seja
24       exibido corretamente. -->
25
26 <title>Efeito de Bounce</title>
27   <!-- A tag <title> define o título da página, que é
28       exibido na aba do navegador. Este título
29       também é usado nos resultados de busca e é
30       importante para SEO. -->
```



EXPLORER

...

bounce.html X



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



31

```
<link rel="stylesheet" href="estilos.css">
<!-- A tag &lt;link&gt; é usada aqui para conectar uma folha
de estilos externa, 'estilos.css', ao documento.
Isso permite que você aplique estilos CSS externos
ao conteúdo da página. --&gt;</pre>
```

32

```
</head>
<!-- Fim da tag &lt;head&gt;. --&gt;</pre>
```

33

```
<body>
<!-- A tag &lt;body&gt; contém todo o conteúdo principal da
página, como texto, imagens, vídeos, jogos,
links de navegação, etc., que são exibidos
no navegador. --&gt;</pre>
```

34

```
<div class="scroller-bounce">
<!-- A tag &lt;div&gt; cria um contêiner para o texto com
efeito de bounce. A classe 'scroller-bounce' é
usada para aplicar estilos CSS e alguns
comportamentos de JavaScript específicos. --&gt;</pre>
```

35

```
<div class="conteudo-bounce">Este é um scroller de texto com efeito de bounce! Divirta-se!</div>
```

36

```
<!-- Outra <div> contém o texto que será animado com o
efeito de bounce. A classe 'conteudo-bounce'
especificamente define o conteúdo que terá o
comportamento animado aplicado via CSS ou JavaScript. -->
```

37

```
</div>
```

38

```
Ln 31, Col 1 Spaces: 8 UTF-8 CRLF HTML ⚡ Port: 5500 Reload
```



EXPLORER

...

bounce.html X

&gt; □ ...



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html

bounce.html > html > body > script

```
61     <script src="script.js"></script>
62     <!-- A tag <script> com o atributo 'src' inclui um arquivo
63             JavaScript externo, 'script.js', que contém
64             código JavaScript para interagir ou modificar
65             dinamicamente o comportamento dos elementos na
66             página, incluindo possivelmente o scroller bounce. -->
67
68 </body>
69 <!-- Fim da tag <body>. -->
70
71 </html>
72 <!-- Fim do documento HTML. -->
```



&gt; OUTLINE

&gt; TIMELINE



⊗ 0 △ 0 ⌂ 0

Ln 61, Col 5 Spaces: 8 UTF-8 CRLF HTML ⚡ Port: 5500 Reload





EXPLORER

...

## P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html

## zoom.html X

## zoom.html &gt; ...

```
1  <!DOCTYPE html>
2  <!-- A declaração DOCTYPE é essencial para informar ao
3      navegador que este documento segue o padrão HTML5,
4      garantindo que as funcionalidades HTML5 sejam
5      interpretadas e executadas corretamente. -->
6
7  <html lang="pt-br">
8  <!-- A tag <html> serve como a raiz do documento HTML e o
9      atributo 'lang="pt-br"' especifica que o idioma
10     principal do conteúdo é o português do Brasil. Isso é
11     importante para acessibilidade e para otimização de
12     mecanismos de busca (SEO), ajudando na correta
13     indexação do conteúdo. -->
14
15 <head>
16  <!-- A tag <head> contém informações meta sobre o
17      documento, incluindo links para folhas de
18      estilo, scripts e outras dependências que não
19      são diretamente visíveis na página. -->
20
21 <meta charset="UTF-8">
22 <!-- A meta tag com charset="UTF-8" especifica a codificação
23      de caracteres usada no documento, suportando
24      quase todos os caracteres de todos os idiomas.
25      Isso garante a exibição correta de qualquer
26      texto no navegador. -->
27
28 <title>Efeito de Zoom</title>
29 <!-- A tag <title> define o título da página, que
30      aparece na aba do navegador. Este título é
```



&gt; OUTLINE



&gt; TIMELINE



⊗ 0 △ 0 ⌂ 0



EXPLORER

...

zoom.html X



## P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



zoom.html &gt; html &gt; head

```
31      também utilizado pelos motores de busca e
32      afeta a SEO, sendo um fator importante na
33      forma como os usuários percebem o conteúdo
34      da página. -->
35
36      <link rel="stylesheet" href="estilos.css">
37      <!-- O elemento <link> com o atributo rel="stylesheet"
38          refere-se a uma folha de estilos externa
39          chamada 'estilos.css'. Este arquivo CSS
40          contém as regras de estilo que definem a
41          aparência de elementos na página, incluindo o
42          layout, cores e animações. -->
43
44  </head>
45  <!-- Finaliza a seção <head>, que prepara o documento
46      com todas as informações necessárias antes de
47      carregar o conteúdo visível. -->
48
49  <body>
50      <!-- A tag <body> engloba todo o conteúdo principal
51          que será exibido aos usuários no navegador. -->
52
53  <div class="scroller-zoom">
54      <!-- Uma <div> que atua como o container principal para o
55          scroller com efeito de zoom. A classe 'scroller-zoom' é
56          utilizada para aplicar estilos CSS e possíveis
57          scripts que manipulem seu comportamento
58          visual e de interação. -->
59
60  <div class="conteudo-zoom">Este é um scroller de texto com efeito de zoom! Aproveite o visual!</div>
```



EXPLORER

...

## P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html

zoom.html X

zoom.html &gt; html &gt; body &gt; div.scroller-zoom

```
61      <!-- Dentro do container principal, há uma segunda <div>
62          que contém diretamente o texto que será animado
63          com o efeito de zoom. A classe 'conteudo-zoom'
64          define as propriedades CSS que controlam a
65          animação, como tamanho de fonte,
66          transições e transformações. -->
67
68      </div>
69
70      <script src="script.js"></script>
71      <!-- A tag <script> com o atributo src="script.js" vincula um
72          arquivo JavaScript externo ao documento. Este
73          script pode incluir funções responsáveis por
74          iniciar, pausar ou modificar dinamicamente a
75          animação de zoom, além de tratar interações do
76          usuário com o scroller. -->
77
78  </body>
79  <!-- Encerra a tag <body>, finalizando a área de conteúdo
80          que os usuários interagem diretamente. -->
81
82  </html>
83  <!-- Fecha a tag <html>, marcando o fim do documento HTML. -->
```



&gt; OUTLINE

&gt; TIMELINE



0 ▲ 0 ⌂ 0

Ln 61, Col 7 Spaces: 4 UTF-8 CRLF HTML Ø Port : 5500 Reload





EXPLORER

...

P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html

rotate.html X

rotate.html &gt; ...

```
1  <!DOCTYPE html>
2  <!-- Esta linha declara o tipo de documento como HTML5,
3  |     informando ao navegador que o documento deve ser
4  |     processado de acordo com os padrões HTML5. -->
5
6  <html lang="pt-br">
7  <!-- Abre a tag <html>, que encapsula todo o conteúdo HTML do
8  |     documento. O atributo 'lang="pt-br"' define o idioma
9  |     principal do conteúdo como português do Brasil, o
10 |     que é útil para acessibilidade e SEO. -->
11
12 <head>
13  <!-- A tag <head> contém metadados e links para recursos
14  |     externos, como folhas de estilo e scripts, que não
15  |     são exibidos diretamente na página, mas são
16  |     essenciais para a configuração e funcionamento
17  |     do documento. -->
18
19 <meta charset="UTF-8">
20 <!-- Define a codificação de caracteres do documento
21  |     como UTF-8, garantindo suporte a uma ampla
22  |     variedade de caracteres e símbolos de diferentes
23  |     idiomas. Isso previne problemas de exibição de texto. -->
24
25 <title>Efeito de Rotação</title>
26 <!-- Define o título da página, que será exibido na aba do
27  |     navegador. Este título também é utilizado por
28  |     motores de busca e aparece nos resultados de
29  |     busca, sendo importante para SEO e usabilidade. -->
30
```



EXPLORER

...

rotate.html X



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



rotate.html &gt; html &gt; head

```
31     <link rel="stylesheet" href="estilos.css">
32     <!-- Conecta o documento HTML a uma folha de estilos externa
33             chamada 'estilos.css'. Esta folha de estilo contém as
34             definições de CSS que controlam a aparência e o
35             layout do conteúdo HTML. -->
36
37 </head>
38 <!-- Fecha a tag <head>, concludendo a seção de metadados e
39             links para recursos externos. -->
40
41 <body>
42     <!-- A tag <body> contém todo o conteúdo visível da página
43             web, incluindo textos, imagens, vídeos e
44             outros elementos. -->
45
46     <div class="scroller-rotate">
47         <!-- Cria uma divisão (div) que serve como container
48             para o scroller com efeito de rotação.
49             A classe 'scroller-rotate' aplica estilos
50             específicos que definem o comportamento e a
51             aparência do scroller. -->
52
53         <div class="conteudo-rotate">Este é um scroller de rotação!</div>
54         <!-- Outra <div> dentro do scroller que contém o texto a
55             ser animado com o efeito de rotação. A classe 'conteudo-rotate'
56             aplica estilos específicos de animação e visual ao conteúdo. -->
57
58     </div>
59
60     <script src="script.js"></script>
```

File Edit Selection View Go Run ... ← → 🔎 P36 - Scroller de Texto

EXPLORER ...

P36 - SCROLLER DE TEXTO

- bounce.html
- carro.jpg
- estilos.css
- fade.html
- fundo\_led\_2.jpg
- fundo\_led\_3.jpg
- fundo\_led.jpg
- image.html
- index.html
- led.html
- rotate.html
- script.js
- zoom.html

rotate.html X

rotate.html > html > body

```
61      <!-- Inclui um arquivo JavaScript externo chamado 'script.js'.  
62          Este script pode adicionar funcionalidades dinâmicas,  
63          como animações e interações, ao conteúdo do scroller. -->  
64      </body>  
65      <!-- Fecha a tag <body>, encerrando a seção do documento que  
66          contém o conteúdo visível para o usuário. -->  
67  
68      </html>  
69      <!-- Fecha a tag <html>, concluindo o documento HTML. Esta  
70          estrutura completa garante que o conteúdo seja  
71          exibido e funcione corretamente nos navegadores. -->
```

Ln 61, Col 12 Spaces: 4 UTF-8 CRLF HTML ⚡ Port: 5500 Reload



EXPLORER

...

image.html X



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



&gt; OUTLINE

&gt; TIMELINE

image.html &gt; ...

```
1  <!DOCTYPE html>
2  <!-- A declaração DOCTYPE define o tipo de documento
3  como HTML5. Essa linha é essencial para
4  garantir que o navegador interprete e renderize o
5  conteúdo da página de acordo com os padrões
6  modernos do HTML5. -->
7
8  <html lang="pt-br">
9  <!-- A tag <html> abre o documento HTML e o atributo 'lang="pt-br"'
10 <!-- especifica que o idioma do documento é português
11 do Brasil. Isso ajuda na acessibilidade, SEO e
12 garante que funcionalidades de idioma específicas,
13 como a formatação de datas e números, sejam
14 corretamente aplicadas pelo navegador. -->
15
16 <head>
17  <!-- A tag <head> contém metadados e links para scripts ou
18  folhas de estilo. Essa seção não é visível
19  diretamente na interface do usuário. -->
20
21  <meta charset="UTF-8">
22  <!-- A meta tag com o atributo charset="UTF-8" define a
23  codificação de caracteres universal para o
24  documento, suportando quase todos os caracteres e
25  símbolos de todos os idiomas, garantindo que
26  qualquer texto seja exibido corretamente. -->
27
28  <title>Imagen e Texto</title>
29  <!-- A tag <title> define o título da página, que é
30  exibido na aba do navegador. Esse título é
```



EXPLORER

...

image.html X

&gt; □ ...



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html

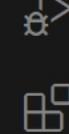


image.html &gt; html &gt; head

```
31     |     importante para a experiência do usuário e SEO,
32     |     pois fornece um contexto sobre o conteúdo da página. -->
33
34     <link rel="stylesheet" href="estilos.css">
35     |     A tag <link> é usada para incluir uma folha de estilos
36     |     externa, especificada pelo atributo href.
37     |     Neste caso, 'estilos.css' contém as regras CSS
38     |     que definem a aparência da página. -->
39
40 </head>
41     |     Fim da tag <head>. -->
42
43 <body>
44     |     A tag <body> contém todo o conteúdo principal da
45     |     página que é visível para os usuários. -->
46
47 <div class="scroller-image">
48     |     Uma <div> com a classe 'scroller-image'. Essa classe
49     |     é usada no CSS para aplicar estilos e
50     |     comportamentos específicos ao scroller de imagem,
51     |     como definições de tamanho, overflow, etc. -->
52
53 <div class="conteudo-image">
54     |     Outra <div> interna que contém efetivamente o
55     |     conteúdo do scroller. A classe 'conteudo-image' é
56     |     usada para especificar estilos e animações
57     |     para o conteúdo que contém imagem e texto. -->
58
59     
60     |     A tag <img> é usada para incluir uma imagem no
```

EXPLORER

P36 - SCROLLER DE TEXTO

- bounce.html
- carro.jpg
- estilos.css
- fade.html
- fundo\_led\_2.jpg
- fundo\_led\_3.jpg
- fundo\_led.jpg
- image.html**
- index.html
- led.html
- rotate.html
- script.js
- zoom.html

> OUTLINE

> TIMELINE

image.html X

image.html > html > body > div.scroller-image > div.conteudo-image

```
61 | documento. O atributo 'src' especifica o
62 | caminho do arquivo de imagem, e 'alt'
63 | fornece uma descrição alternativa da imagem,
64 | essencial para acessibilidade e quando a
65 | imagem não pode ser carregada. -->
66 |
67 | <span>Este é um scroller de texto com imagem! Fantástico!</span>
68 | <!-- Um <span> contém texto adicional, proporcionando
69 |           informações ou complementando a imagem. -->
70 |
71 |         </div>
72 |     </div>
73 |
74 |     <script src="script.js"></script>
75 |     <!-- A tag <script> com o atributo 'src' inclui um arquivo
76 |           JavaScript externo. 'script.js' é utilizado para
77 |           adicionar funcionalidades dinâmicas ao scroller,
78 |           como iniciar a animação, responder a eventos do
79 |           usuário, etc. -->
80 |
81 |   </body>
82 |   <!-- Fim da tag <body>. -->
83 |
84 | </html>
85 | <!-- Fim do documento HTML. Toda a estrutura do
86 |       documento está completa. -->
```



EXPLORER

...

led.html X

&gt; □ ...



P36 - SCROLLE...



bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



&gt; OUTLINE

&gt; TIMELINE

led.html &gt; ...

```
1  <!DOCTYPE html>
2  <!-- Essa declaração define o tipo de documento
3  |       como HTML5, o mais recente padrão de HTML.
4  |       Ela informa aos navegadores modernos como
5  |       interpretar o restante do conteúdo do
6  |       documento corretamente. -->
7
8  <html lang="pt-br">
9  <!-- Abre a tag <html>, definindo o escopo do documento HTML.
10 |      O atributo 'lang="pt-br"' especifica que o idioma
11 |      principal do conteúdo é o português do Brasil, o
12 |      que ajuda mecanismos de busca e ferramentas de
13 |      acessibilidade a processar a página adequadamente. -->
14
15 <head>
16  <!-- A tag <head> contém informações sobre o documento que
17  |      não são exibidas diretamente na página web, como
18  |      metadados, links para folhas de estilo e scripts. -->
19
20 <meta charset="UTF-8">
21  <!-- Esta meta tag define a codificação de caracteres do
22  |      documento para UTF-8, que suporta todos os
23  |      caracteres de qualquer idioma, assegurando
24  |      que não haja problemas de exibição de texto. -->
25
26 <title>Efeito de LED</title>
27  <!-- O <title> define o título da página, que é mostrado
28  |      na aba do navegador e utilizado pelos motores
29  |      de busca para informar sobre o conteúdo da
30  |      página. É importante para a usabilidade e SEO. -->
```



EXPLORER

...

led.html X

D I ...



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



led.html &gt; html &gt; head

```
31 |           <link rel="stylesheet" href="estilos.css">
32 |           <!-- A tag &lt;link&gt; aqui é usada para conectar o documento HTML a
33 |           |           uma folha de estilos externa chamada 'estilos.css'.
34 |           |           Esta folha de estilo contém as regras CSS que
35 |           |           controlam a aparência visual do documento. --&gt;
36 |
37 |
38 |       &lt;/head&gt;
39 |       <!-- Fecha a tag &lt;head&gt;. --&gt;
40 |
41 |   &lt;body&gt;
42 |       <!-- A tag &lt;body&gt; envolve todo o conteúdo principal que é
43 |           |           visível para os usuários na página web. --&gt;
44 |
45 |       &lt;div class="scroller-led"&gt;
46 |           <!-- A &lt;div&gt; atua como um contêiner para o conteúdo do
47 |               |           efeito de LED. A classe 'scroller-led' é
48 |               |           usada para aplicar estilos específicos relacionados
49 |               |           ao efeito de scroller que simula um display de LED. --&gt;
50 |
51 |           &lt;div class="conteudo-led"&gt;APROVEITE A PROMOÇÃO!&lt;/div&gt;
52 |           <!-- Outra &lt;div&gt; contém o texto que será exibido com o
53 |               |           efeito de LED. A classe 'conteudo-led' pode ser
54 |               |           usada para aplicar estilos específicos de animação
55 |               |           ou visual, como cores brilhantes, piscar, ou
56 |               |           deslizar o texto. --&gt;
57 |
58 |       &lt;/div&gt;
59 |
60 |   &lt;script src="script.js"&gt;&lt;/script&gt;</pre>
```

The screenshot shows a dark-themed code editor interface. At the top, there's a navigation bar with icons for File, Edit, Selection, View, Go, Run, and a search bar containing the text "P36 - Scroller de Texto". On the far right, there are window control buttons. Below the navigation bar is a toolbar with various icons: a clipboard, magnifying glass, gear, and others.

The main area has three panes:

- EXPLORER** pane on the left, showing a tree view of files under "P36 - SCROLLER DE TEXTO". The files listed are: bounce.html, carro.jpg, estilos.css, fade.html, fundo\_led\_2.jpg, fundo\_led\_3.jpg, fundo\_led.jpg, image.html, index.html, led.html (which is selected and highlighted in grey), rotate.html, script.js, and zoom.html.
- OUTLINE** pane at the bottom left, showing sections: OUTLINE and TIMELINE.
- Code Editor** pane on the right, displaying the content of the "led.html" file. The code is as follows:

```
61 <!-- A tag <script> inclui um arquivo JavaScript externo, 'script.js',  
62 | que contém código para controlar ou adicionar  
63 | comportamentos dinâmicos ao efeito de LED, como  
64 | controlar a animação, iniciar automaticamente, etc. -->  
65 </body>  
66 <!-- Fim da tag <body>. -->  
67 </html>  
68 <!-- Encerra o documento HTML. Esta estrutura assegura que o  
69 | conteúdo seja exibido corretamente nos navegadores e  
70 | esteja bem organizado. -->  
71 <!-- A tag <script> inclui um arquivo JavaScript externo, 'script.js',  
72 | que contém código para controlar ou adicionar  
| comportamentos dinâmicos ao efeito de LED, como  
| controlar a animação, iniciar automaticamente, etc. -->
```

At the bottom, there are status indicators: Line 61, Column 6, Spaces: 4, UTF-8, CRLF, HTML, Port: 5500, Reload, and a refresh icon.



EXPLORER ...

P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html

estilos.css X

estilos.css &gt; button

```
1 body {  
2     background-color: #fff;  
3     /* Fundo branco */  
4  
5     color: #222;  
6     /* Define a cor de texto geral para um cinza  
7      escuro, aumentando o contraste com o  
8      fundo branco */  
9  
10    font-family: Arial, sans-serif;  
11    /* Define a família de fontes para Arial, que é  
12       sans-serif, dando um look limpo e  
13       moderno ao texto */  
14  
15    display: flex;  
16    /* Define o modelo de layout para Flexbox, o que  
17       facilita a organização dos itens internos de  
18       maneira eficiente e flexível */  
19  
20    justify-content: center;  
21    /* Alinha os itens filho horizontalmente ao centro,  
22       funcionando aqui porque o 'display' foi  
23       definido como 'flex' */  
24  
25    align-items: center;  
26    /* Alinha os itens filho verticalmente ao centro,  
27       isso também depende do 'display: flex'  
28       definido anteriormente */  
29  
30}
```



EXPLORER

✓ P36 - SCROLLER DE TEXTO

5 bounce.html

carro.jpg

3 estilos.css

 fade.html

## fundo\_led\_2.jpg

 fundo\_led\_3.jpg

 fundo\_led.jpg

5 image.html

5 index.htm

5 led.html

rotate.html

## JS script.js

 zoom.html

3 estilos.css X

3 estilos.css > 



EXPLORER

...

## P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



estilos.css X

estilos.css &gt; h1

```
61      /* Define a cor do texto do cabeçalho para um cinza escuro,  
62      | | | proporcionando um bom contraste com um  
63      | | | fundo claro */  
64      }  
65  
66      .botões {  
67  
68          display: flex;  
69          /* Usa Flexbox para organizar os botões dentro deste  
70          | | | contêiner, proporcionando um layout  
71          | | | mais adaptável */  
72  
73          flex-direction: column;  
74          /* Define a orientação dos itens flexíveis como  
75          | | | vertical, empilhando os botões  
76          | | | verticalmente */  
77  
78          align-items: center;  
79          /* Centraliza os botões na direção transversal (horizontal  
80          | | | neste caso), dentro de seu contêiner */  
81  
82          gap: 10px;  
83          /* Adiciona um espaço de 15px entre cada botão  
84          | | | dentro do contêiner */  
85  
86          }  
87  
88      button {  
89
```



EXPLORER

...

## P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html

estilos.css X

estilos.css &gt; button

```
91     display: block;  
92     /* Faz com que cada botão se comporte como um bloco,  
93      |   ocupando uma linha inteira, mas personalizado  
94      |   para ajustar largura */  
95  
96     width: 750px;  
97     /* Define a largura específica de cada botão para 750px,  
98      |   garantindo que todos os botões sejam  
99      |   largos e visíveis */  
100  
101    height: 80px;  
102    /* Define a altura de cada botão para 130px, tornando-os  
103       |   mais fáceis de clicar especialmente em  
104       |   dispositivos touch */  
105  
106    padding: 15px 30px;  
107    /* Adiciona um preenchimento dentro de cada botão para  
108       |   tornar o texto mais distante das bordas e  
109       |   mais confortável de visualizar */  
110  
111    background-color: #444;  
112    /* Define a cor de fundo dos botões para um cinza  
113       |   escuro (#444), o que destaca os botões  
114       |   no layout geral */  
115  
116    color: #fff;  
117    /* Define a cor do texto dentro dos botões para  
118       |   branco, criando um alto contraste com o  
119       |   fundo escuro */  
120
```



&gt; OUTLINE

&gt; TIMELINE



⊗ 0 ▲ 0 ⌂ 0

Ln 91, Col 9 Spaces: 4 UTF-8 CRLF CSS Ø Port : 5500 Reload





EXPLORER

...

estilos.css X

▶ □ ⏪ ⏩ ⏴



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



&gt; OUTLINE

&gt; TIMELINE

estilos.css &gt; button

```
121 border: none;
122 /* Remove qualquer borda padrão que poderia estar
123 |     presente, criando um visual mais limpo */
124
125 cursor: pointer;
126 /* Muda o cursor para indicar que o elemento é
127 |     clicável, melhorando a usabilidade */
128
129 font-size: 24px;
130 /* Define um tamanho de fonte maior para o texto
131 |     dentro dos botões, tornando-os mais
132 |     fáceis de ler */
133
134 text-align: center;
135 /* Centraliza o texto dentro dos botões, garantindo
136 |     que o texto esteja bem alinhado horizontalmente */
137
138 line-height: 20px;
139 /* Define a altura da linha do texto dentro dos botões, o
140 |     que pode afetar a maneira como o texto é
141 |     visualmente posicionado verticalmente */
142
143 }
144
145
146 button:hover {
147
148 background-color: #e0f44b;
149 /* Muda a cor de fundo do botão para um amarelo vivo quando o
150 |     mouse está sobre ele, aumentando a
```



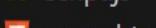
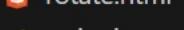
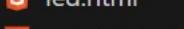
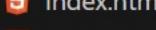
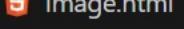
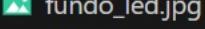
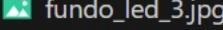
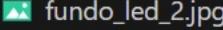
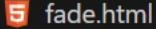
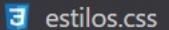
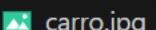
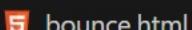


EXPLORER

3 estilos.css X



## ✓ P36 - SCROLLER DE TEXTO



```
estilos.css > .scroller-fade

181         posicionar absolutamente elementos internos
182         como 'conteudo-fade' */
183
184     background-color: #000;
185     /* Define a cor de fundo do scroller para preto,
186      proporcionando um fundo distinto para o
187      texto claro */
188
189     box-shadow: 0 2px 5px rgba(0,0,0,0.2);
190     /* Aplica uma sombra sutil ao scroller, adicionando
191      profundidade e ajudando o scroller a se
192      destacar do resto do conteúdo */
193
194     display: flex;
195     /* Utiliza Flexbox para alinhar facilmente itens
196      dentro do scroller, neste caso,
197      centralizando verticalmente */
198
199     align-items: center;
200     /* Alinha itens verticalmente ao centro dentro do
201      scroller, centralizando o conteúdo
202      verticalmente */
203
204 }
205
206 .conteudo-fade {
207
208     position: absolute;
209     /* Posiciona absolutamente o conteúdo dentro do
210      scroller, permitindo que ele deslize
```



EXPLORER

3

3 estilos.css X

3 estilos.css >  .conteudo-fac

```
211         independentemente do scroller */
212
213     white-space: nowrap;
214     /* Impede a quebra de linha do texto dentro do
215      scroller, essencial para o efeito de
216      deslizamento contínuo */
217
218     font-size: 100px;
219     /* Define um tamanho de fonte muito grande para o
220      texto, garantindo que ele seja legível e
221      tenha impacto visual */
222
223     color: ■#fff;
224     /* Define a cor do texto para branco, criando um
225      alto contraste com o fundo escuro para
226      legibilidade */
227
228     padding: 10px 20px;
229     /* Adiciona um espaçamento interno ao redor do
230      texto, aumentando a legibilidade e a
231      estética geral */
232
233     animation: deslizar-fade 15s linear infinite;
234     /* Aplica uma animação chamada 'deslizar-fade', que
235      desliza o texto horizontalmente ao longo
236      de 15 segundos de forma linear e infinita */
237
238 }
239
240
```

Ln 211, Col 4 Spaces: 4 UTF-8 CRLF CSS Ø Port : 5500 Reload ⚙



EXPLORER

...

## P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



estilos.css X

estilos.css &gt; ⚙️ @keyframes deslizar-fade

```
241  @keyframes deslizar-fade {  
242      0% {  
243          transform: translateX(100%);  
244          /* No início da animação, posiciona o  
245             conteúdo 100% para a direita do ponto de  
246             partida, tornando-o inicialmente invisível à  
247             direita da tela. */  
248          opacity: 0;  
249          /* Configura a opacidade para 0, tornando o conteúdo  
250             completamente transparente no início da animação. */  
251      }  
252      10% {  
253          opacity: 1;  
254          /* Aumenta a opacidade para 1 em 10% do tempo de  
255             animação, tornando o conteúdo totalmente visível. */  
256      }  
257      90% {  
258          opacity: 1;  
259          /* Mantém a opacidade em 1 até 90% do tempo de  
260             animação, garantindo que o conteúdo permaneça  
261             visível durante a maior parte da animação. */  
262      }  
263  }  
264  }  
265  }  
266  }  
267  }  
268  }  
269  }
```

EXPLORER

P36 - SCROLLER DE TEXTO

- bounce.html
- carro.jpg
- estilos.css
- fade.html
- fundo\_led\_2.jpg
- fundo\_led\_3.jpg
- fundo\_led.jpg
- image.html
- index.html
- led.html
- rotate.html
- script.js
- zoom.html

estilos.css

```
estilos.css > @keyframes deslizar-fade
271
272 }
273
274 100% {
275
276     transform: translateX(-100%);
277     /* No final da animação, desloca o conteúdo para 100% à
278      | esquerda, fazendo-o sair completamente da tela. */
279
280     opacity: 0;
281     /* Reduz a opacidade para 0 novamente, tornando o
282      | conteúdo completamente transparente no
283      | final da animação. */
284 }
285
286 }
287
288
289 /* Scroller com Efeito de Bounce */
290 .scroller-bounce {
291
292     width: 100%;
293     /* Define a largura do contêiner do scroller
294      | para 100% da largura do elemento pai,
295      | garantindo que ele ocupe toda a
296      | largura disponível. */
297
298     overflow: hidden;
299     /* Configura o scroller para esconder qualquer
300      | conteúdo que ultrapasse o seu tamanho
```

OUTLINE

TIMELINE

Ln 271, Col 1 Spaces: 4 UTF-8 CRLF CSS ⚡ Port : 5500 Reload



EXPLORER

...

estilos.css X

&gt; □ ...



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



&gt; OUTLINE

&gt; TIMELINE

estilos.css &gt; .scroller-bounce

```
301      |     especificado, essencial para manter o
302      |     conteúdo controlado dentro do scroller. */
303
304      height: 400px;
305      /* Define a altura do scroller para 400px, proporcionando
306      |     espaço suficiente para o efeito de quique ser
307      |     observado sem cortes. */
308
309      position: relative;
310      /* Define a posição como relativa, o que permite que
311      |     qualquer posicionamento absoluto de elementos
312      |     filhos seja relativo a este contêiner. */
313
314      background-color: #000;
315      /* Configura a cor de fundo do scroller para preto,
316      |     proporcionando um forte contraste com
317      |     elementos de conteúdo claro. */
318
319      box-shadow: 0 2px 5px rgba(0,0,0,0.2);
320      /* Aplica uma sombra suítil ao scroller para adicionar
321      |     profundidade e destaque visual. */
322
323      display: flex;
324      /* Utiliza Flexbox para facilitar o alinhamento dos
325      |     itens dentro do scroller. */
326
327      align-items: center;
328      /* Alinha os itens no centro verticalmente dentro do
329      |     scroller, garantindo que o conteúdo esteja
330      |     centrado independentemente do tamanho. */
```



EXPLORER

...

estilos.css X



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



P36 - SCROLLER DE TEXTO

estilos.css &gt; .scroller-bounce

```
331 }  
332 }  
333  
334  
335 .conteudo-bounce {  
336     position: absolute;  
337     /* Posiciona o elemento de forma absoluta dentro de seu  
338      |   contêiner mais próximo com posição não estática,  
339      |   permitindo que ele se mova livremente dentro  
340      |   do scroller-bounce. */  
341  
342     white-space: nowrap;  
343     /* Impede que o texto dentro deste elemento quebre em  
344      |   várias linhas, garantindo que o texto permaneça em  
345      |   uma única linha horizontal. */  
346  
347     font-size: 100px;  
348     /* Define um tamanho de fonte grande, tornando o texto  
349      |   claramente visível e impactante. */  
350  
351     color: #fff;  
352     /* Define a cor do texto para branco, criando um alto  
353      |   contraste com um fundo escuro para  
354      |   facilitar a leitura. */  
355  
356     padding: 10px 20px;  
357     /* Adiciona espaço dentro do elemento ao redor do texto,  
358      |   melhorando a legibilidade e a estética  
359      |   geral do design. */  
360 }
```



EXPLORER

...

estilos.css X

&gt; □ ...



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



&gt; OUTLINE

&gt; TIMELINE

estilos.css &gt; .conteudo-bounce

```
361
362     animation: deslizar-bounce 10s ease-in-out infinite;
363     /* Aplica uma animação chamada 'deslizar-bounce'.
364      A animação ocorre durante 10 segundos com um
365      efeito 'ease-in-out' (aceleração no início e
366      no fim) e se repete infinitamente. */
367
368 }
369
370 @keyframes deslizar-bounce {
371     0% {
372         transform: translateX(100%);
373         /* No início da animação, move o elemento 100% para a
374          direita de sua posição original, fazendo com
375          que comece fora da tela à direita. */
376
377     }
378
379 }
380
381     25% {
382
383         transform: translateX(75%) translateY(-50px);
384         /* Em 25% da animação, move o elemento para 75% da
385          direita para a esquerda e simultaneamente
386          move 50px para cima, criando um efeito de "salto". */
387
388     }
389
390     50% {
```



EXPLORER

...

estilos.css X

D □ ...



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html

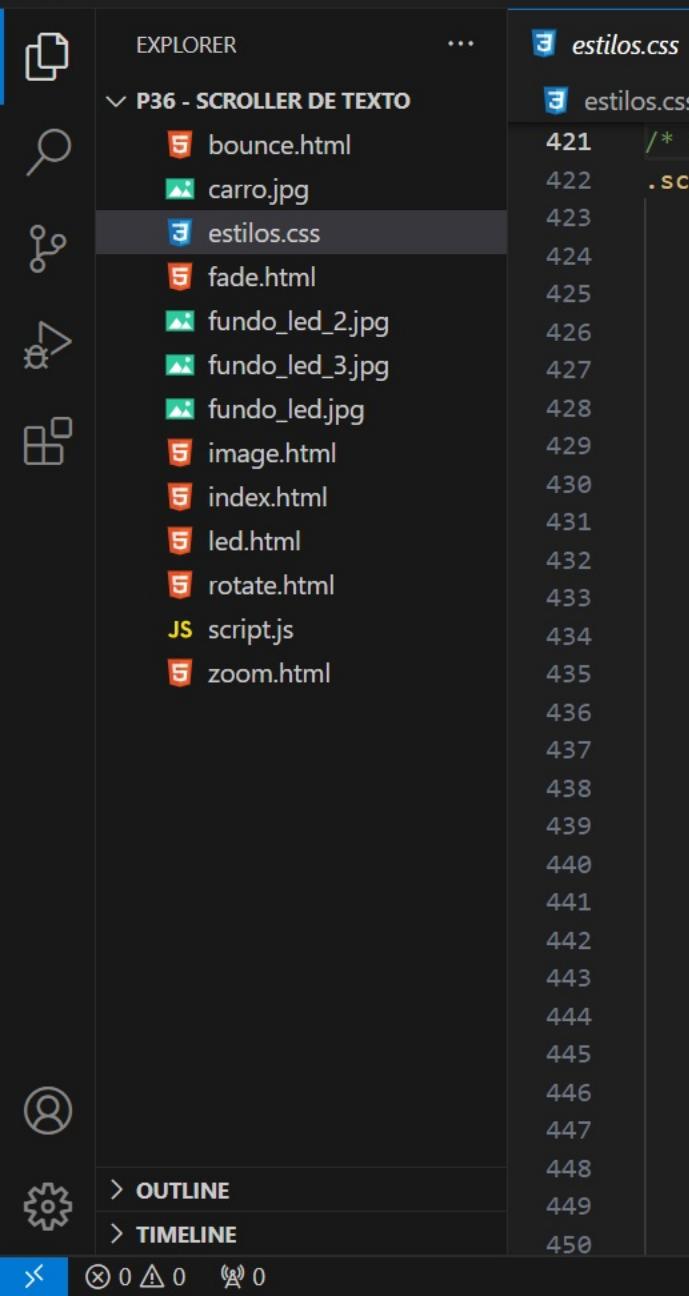


&gt; OUTLINE

&gt; TIMELINE

estilos.css &gt; @keyframes deslizar-bounce

```
391 transform: translateX(50%) translateY(0);
392 /* Em 50% do tempo, o elemento é trazido para o
393 centro horizontalmente (50% para a esquerda
394 do início) e retorna ao alinhamento
395 vertical inicial. */
396 }
397
398 75% {
399
400     transform: translateX(25%) translateY(-50px);
401     /* Em 75% da animação, o elemento é ainda mais
402     deslocado para a esquerda (25% para a direita
403     do ponto de partida) e novamente move 50px para
404     cima, reforçando o efeito de "salto". */
405
406 }
407
408 100% {
409
410     transform: translateX(-100%);
411     /* No final da animação, o elemento é completamente
412     movido para a esquerda, saindo da tela,
413     passando 100% de sua largura original
414     para a esquerda. */
415
416 }
417
418 }
419 }
420 }
```



## estilos.css X

## estilos.css &gt; ...

```
421  /* Scroller com Efeito de Zoom */
422  .scroller-zoom {
423
424      width: 100%;
425      /* Ajusta a largura do scroller para ocupar 100%
426      |   |   da largura do contêiner pai. */
427
428      overflow: hidden;
429      /* Oculta qualquer conteúdo que ultrapasse o limite do
430      |   |   scroller, essencial para manter um design limpo. */
431
432      height: 800px;
433      /* Define uma altura fixa para o scroller, o suficiente
434      |   |   para mostrar claramente o conteúdo e permitir um
435      |   |   bom espaço para o efeito de zoom. */
436
437      position: relative;
438      /* Configura o scroller para ter posição relativa, que é
439      |   |   a base para o posicionamento absoluto dos
440      |   |   elementos internos. */
441
442      background-color: □#000;
443      /* Aplica uma cor de fundo preta ao scroller, garantindo um
444      |   |   alto contraste com o conteúdo de cor clara. */
445
446      box-shadow: 0 2px 5px □rgba(0,0,0,0.2);
447      /* Adiciona uma sombra leve para estilizar e elevar
448      |   |   visualmente o scroller acima de outros conteúdos. */
449
450      display: flex;
```



EXPLORER

...

estilos.css X



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



&gt; OUTLINE

&gt; TIMELINE

estilos.css &gt; .scroller-zoom

```
451     /* Utiliza Flexbox para gerenciar o layout dos
452      |   elementos internos, permitindo alinhamentos
453      |   mais simples e designs responsivos. */
454
455     align-items: center;
456     /* Alinha o conteúdo dentro do scroller verticalmente ao
457      |   centro, melhorando a estética geral. */
458
459 }
460
461 .conteudo-zoom {
462
463     position: absolute;
464     /* Posiciona o conteúdo de forma absoluta dentro do
465      |   scroller, permitindo que ele se mova livre
466      |   de restrições do contêiner pai. */
467
468     white-space: nowrap;
469     /* Evita quebra de linha do conteúdo, essencial para
470      |   manter o texto contínuo e legível durante o
471      |   efeito de zoom. */
472
473     font-size: 50px;
474     /* Define o tamanho da fonte do conteúdo, garantindo que
475      |   seja grande o suficiente para leitura em zoom. */
476
477     color: #fff;
478     /* Define a cor do texto para branco, garantindo
479      |   excelente legibilidade contra o fundo escuro. */
480
```



EXPLORER

...

## P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html

estilos.css X

```
estilos.css > .conteudo-zoom
481     padding: 10px 20px;
482     /* Adiciona preenchimento ao redor do conteúdo para
483      |   espaço extra, evitando que o texto toque
484      |   diretamente nas bordas do scroller. */
485
486     animation: deslizar-zoom 15s linear infinite;
487     /* Define uma animação de 15 segundos que se repete
488      |   infinitamente, com um movimento suave (linear)
489      |   para simular o deslizamento e zoom. */
490
491 }
492
493 @keyframes deslizar-zoom {
494
495     0% {
496
497         transform: translateX(100%) scale(1);
498         /* Inicia com o conteúdo 100% à direita da
499         |   sua posição normal, em escala normal (1). */
500
501     }
502
503     50% {
504
505         transform: translateX(0%) scale(2);
506         /* Atinge o centro do scroller e aumenta o zoom para o
507         |   dobro do tamanho original (escala 2), criando um
508         |   destaque visual no meio da animação. */
509
510 }
```



&gt; OUTLINE



&gt; TIMELINE



⊗ 0 △ 0 ⌂ 0

Ln 481, Col 4 Spaces: 4 UTF-8 CRLF CSS Ø Port : 5500 Reload



EXPLORER

...

## P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html

estilos.css X

estilos.css &gt; ⚒ @keyframes deslizar-zoom

```
511    100% {  
512        transform: translateX(-100%) scale(1);  
513        /* Move o conteúdo para 100% à esquerda da sua  
514         posição normal e retorna à escala normal,  
515         fazendo com que saia completamente da  
516         tela à esquerda. */  
517    }  
518}  
519  
520    /* Scroller com Efeito de LED */  
521    .scroller-led {  
522        width: 100%;  
523        /* Estabelece a largura do scroller para preencher 100%  
524         da largura do container pai, o que maximiza a  
525         área de exibição do scroller. */  
526        overflow: hidden;  
527        /* Oculta qualquer conteúdo que exceda as dimensões do  
528         scroller, essencial para manter o controle  
529         sobre a aparência visual, especialmente  
530         em animações. */  
531        height: 300px;  
532        /* Define a altura do scroller para 300px, o suficiente  
533         para acomodar o texto e proporcionar uma  
534         boa visibilidade. */  
535    }
```



&gt; OUTLINE

&gt; TIMELINE

0 ▲ 0 ⌂ 0

Ln 511, Col 1 Spaces: 4 UTF-8 CRLF CSS ⌂ Port : 5500 Reload



EXPLORER

...

## P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



estilos.css X

estilos.css &gt; .scroller-led

```
541      position: relative;
542      /* Configura a posição como relativa, o que é importante
543         para posicionar corretamente elementos
544         absolutamente posicionados dentro
545         deste container. */
546
547      background: url('fundo_led.jpg') repeat;
548      /* Define o fundo do scroller como uma imagem de LED,
549         repetindo-a para cobrir toda a área do
550         elemento. Isso ajuda a criar um efeito
551         visual de painel de LED. */
552
553      display: flex;
554      /* Usa o Flexbox para gerenciar o layout dos
555         elementos dentro do scroller, possibilitando um
556         alinhamento mais fácil e flexível do conteúdo. */
557
558      align-items: center;
559      /* Alinha o conteúdo verticalmente ao centro do scroller,
560         garantindo que qualquer texto ou elemento
561         esteja centralizado na altura. */
562
563      justify-content: center;
564      /* Alinha o conteúdo horizontalmente ao centro, o que é
565         crucial para que o texto apareça no meio
566         do scroller. */
567
568
569    }
570
```



EXPLORER

...

estilos.css X

&gt; □ ...



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



&gt; OUTLINE

&gt; TIMELINE

estilos.css &gt; .conteudo-led

```
571 .conteudo-led {  
572     font-size: 100px;  
573     /* Define um tamanho de fonte grande, 100px, tornando o  
574      texto grande o suficiente para ser visto  
575      claramente e impactante visualmente. */  
576     color: #ffffff;  
577     /* Configura a cor do texto para branco, oferecendo um  
578      alto contraste com a imagem de fundo e outros  
579      elementos visuais, melhorando a legibilidade. */  
580     text-shadow:  
581         0 0 5px #ff0000;  
582     /* Adiciona uma sombra de texto vermelha, que simula o  
583      brilho de um display de LED, ampliando o  
584      efeito visual de tecnologia de luz LED. */  
585     animation: deslizar-led 20s linear infinite, piscar 1s step-start infinite;  
586     /* Combina duas animações: uma que desliza o texto  
587      através do scroller em 20 segundos de forma  
588      contínua e linear, e outra que faz o texto  
589      piscar a cada segundo. Isso cria um dinamismo  
590      visual característico de displays de LED. */  
591 }  
592  
593 @keyframes deslizar-led {  
594     0% { transform: translateX(0); }  
595     100% { transform: translateX(-100%); }  
596 }
```



EXPLORER

...

estilos.css X

D I ...



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



&gt; OUTLINE

&gt; TIMELINE

estilos.css &gt; ⚙️ @keyframes deslizar-led

```
601    0% {  
602        transform: translateX(100%);  
603        /* No início da animação, o elemento começa 100% para a  
604         direita do seu ponto de partida visível,  
605         essencialmente fora da tela à direita. */  
606    }  
607  
608    100% {  
609        transform: translateX(-100%);  
610        /* No final da animação, o elemento se move 100% para a  
611         esquerda do seu ponto de partida original, movendo-se  
612         completamente fora da tela à esquerda. */  
613    }  
614  
615    }  
616  
617    }  
618  
619    /* Esta animação cria um efeito de deslizamento contínuo de um  
620     lado ao outro, simulando um display LED em movimento. */  
621  
622  
623    @keyframes piscar {  
624  
625        0%, 49% {  
626            opacity: 1;  
627            /* De 0% a 49% da duração da animação, o elemento  
628             permanece totalmente visível (opacidade 1),  
629             criando a impressão de que está constantemente ligado. */  
630    }
```

EXPLORER ...

P36 - SCROLLER DE TEXTO

- bounce.html
- carro.jpg
- estilos.css
- fade.html
- fundo\_led\_2.jpg
- fundo\_led\_3.jpg
- fundo\_led.jpg
- image.html
- index.html
- led.html
- rotate.html
- script.js
- zoom.html

estilos.css X

estilos.css > ⚒ @keyframes piscar

```
631
632    }
633
634    50%, 100% {
635        opacity: 0;
636        /* De 50% a 100% da duração da animação, o elemento se
637         torna completamente invisível (opacidade 0),
638         criando um efeito de "piscar". */
639    }
640
641    }
642 }
643 /* Esta animação de piscar faz com que o texto ou elemento
644      alternem entre visível e invisível, simbolizando o
645      intermitente piscar de um display de LED, frequentemente
646      usado para chamar a atenção ou indicar algo importante. */
647
648
649 /* Scroller com Efeito de Rotação */
650 .scroller-rotate {
651
652     width: 100%;
653     /* Define a largura do contêiner do scroller para
654      ocupar 100% da largura disponível no
655      contêiner pai, assegurando que ele se
656      estenda por toda a largura da tela ou do
657      contêiner pai. */
658
659     overflow: hidden;
660     /* Oculta qualquer conteúdo que ultrapasse os limites
```

Ln 631, Col 1 Spaces: 4 UTF-8 CRLF CSS ⚒ Port : 5500 Reload 🔍

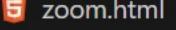
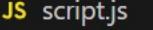
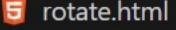
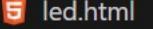
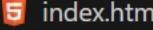
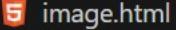
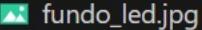
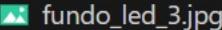
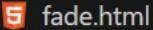
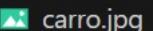
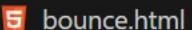


EXPLORER

3 estilos.css X



✓ P36 - SCROLLER DE TEXTO



estilos.css > .scroller-rotate

```
661         deste contêiner, crucial para manter o scroller  
662         limpo e evitar a visualização de elementos que  
663         se movem para fora da área definida. */  
664  
665     height: 800px;  
666     /* Estabelece uma altura fixa de 800px para o scroller,  
667         proporcionando espaço suficiente verticalmente para  
668         visualizar o conteúdo enquanto ele roda e move-se  
669         horizontalmente. */  
670  
671     position: relative;  
672     /* Define a posição do scroller como relativa, permitindo  
673         posicionar absolutamente elementos internos em  
674         relação a este contêiner. */  
675  
676     background-color: □#000;  
677     /* Configura a cor de fundo do scroller para preto,  
678         oferecendo um forte contraste com o conteúdo de  
679         cor clara (texto branco), realçando a  
680         visibilidade do efeito de rotação. */  
681  
682     display: flex;  
683     /* Utiliza Flexbox para alinhar itens dentro do scroller,  
684         facilitando o alinhamento vertical e horizontal  
685         do conteúdo. */  
686  
687     align-items: center;  
688     /* Centraliza o conteúdo verticalmente dentro do  
689         scroller, garantindo que o conteúdo esteja  
690         alinhado ao centro da altura do scroller. */
```



EXPLORER

...

estilos.css X

&gt; □ ...



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



&gt; OUTLINE

&gt; TIMELINE

estilos.css &gt; .scroller-rotate

```
691      justify-content: flex-start;
692      /* Alinha o conteúdo ao início do contêiner do
693      scroller (lado direito), assegurando que a
694      animação comece fora da visualização à direita. */
695
696 }
697
698
699 .conteudo-rotate {
700
701     position: absolute;
702     /* Posiciona o conteúdo de forma absoluta dentro do
703     scroller, permitindo que ele se mova
704     livremente conforme as especificações da
705     animação, sem ser restrito pelo layout
706     do contêiner. */
707
708     white-space: nowrap;
709     /* Evita que o texto dentro do conteúdo quebre em
710     várias linhas, mantendo tudo em uma única
711     linha contínua; crucial para manter a
712     coesão visual durante a rotação e o
713     movimento horizontal. */
714
715     font-size: 50px;
716     /* Define o tamanho da fonte do conteúdo para 50px,
717     tornando-o suficientemente grande para
718     ser lido claramente à medida que se
719     move e roda. */
720
```



EXPLORER

...

estilos.css X

...



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



estilos.css &gt; .conteudo-rotate

```
721     color: #fff;
722     /* Define a cor do texto para branco, criando um
723        contraste alto com o fundo preto para
724        melhorar a legibilidade. */
725
726     padding: 10px 20px;
727     /* Adiciona preenchimento ao redor do texto para
728        evitar que ele toque diretamente as bordas
729        do scroller, melhorando a estética e a
730        legibilidade. */
731
732     left: 100%;
733     /* Posiciona inicialmente o conteúdo fora da tela à
734        direita (100% para a direita da borda
735        esquerda do container), garantindo que
736        ele comece fora de vista e entre na
737        tela suavemente. */
738
739     animation: deslizar-rotate 20s linear infinite;
740     /* Aplica a animação chamada 'deslizar-rotate' com
741        duração de 20 segundos, ocorrendo de
742        maneira linear e repetindo infinitamente,
743        para um efeito contínuo de rotação e
744        deslocamento horizontal. */
745
746 }
747
748 @keyframes deslizar-rotate {
749
750     0% {
```

EXPLORER ...

P36 - SCROLLER DE TEXTO

- bounce.html
- carro.jpg
- estilos.css
- fade.html
- fundo\_led\_2.jpg
- fundo\_led\_3.jpg
- fundo\_led.jpg
- image.html
- index.html
- led.html
- rotate.html
- script.js
- zoom.html

estilos.css

@keyframes deslizar-rotate

```
751 transform: translateX(0) rotate(0deg);
752 /* Inicia com o conteúdo alinhado à borda
753 direita, dentro do campo de visão,
754 pronto para entrar. */
755 }
756
757 50% {
758     transform: translateX(-150%) rotate(720deg);
759     /* Movimenta o conteúdo para a esquerda (150% da
760      largura do próprio conteúdo) e realiza duas
761      voltas completas, posicionando o texto mais
762      ao centro durante este ponto da animação. */
763
764 }
765
766 100% {
767     transform: translateX(-450%) rotate(1440deg);
768     /* Continua o movimento para a esquerda (500% da
769      largura do conteúdo) e completa quatro voltas
770      totais, garantindo que o conteúdo saia
771      completamente da tela à esquerda ao
772      final da animação. */
773
774 }
775
776 }
777
778 }
779 }
780 }
```

OUTLINE

TIMELINE

Ln 751, Col 1 Spaces: 4 UTF-8 CRLF CSS ⚡ Port : 5500 Reload

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P36 - Scroller de Texto
- Sidebar:** EXPLORER, P36 - SCROLLER DE TEXTO (containing files: bounce.html, carro.jpg, estilos.css, fade.html, fundo\_led\_2.jpg, fundo\_led\_3.jpg, fundo\_led.jpg, image.html, index.html, led.html, rotate.html, script.js, zoom.html), OUTLINE, and TIMELINE.
- Central Area:** A code editor window titled "estilos.css". The file content is as follows:

```
781 /* Scroller com Imagem e Texto */
782 .scroller-image {
783     width: 100%; /* Define a largura do scroller para 100% da largura do container pai, o que permite que ocupe toda a largura disponível. */
784     overflow: hidden; /* Mantém qualquer conteúdo que ultrapasse os limites do scroller oculto, essencial para uma aparência limpa e controlada. */
785     height: 300px; /* Define a altura do scroller para 300px, o suficiente para acomodar imagens e texto de forma adequada. */
786     position: relative; /* Configura o scroller para ter posição relativa, o que serve de referência para os elementos posicionados de forma absoluta dentro dele. */
787     background-color: #000; /* Define a cor de fundo do scroller para preto, criando um fundo sólido que aumenta o contraste para o conteúdo claro. */
788     box-shadow: 0 2px 5px rgba(0,0,0,0.2); /* Adiciona uma sombra sutil ao redor do scroller */
789 }
```

The code is written in Portuguese and describes the styling for a scrollable container that holds both images and text.



EXPLORER

...

estilos.css X

&gt; □ ...



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



&gt; OUTLINE

&gt; TIMELINE

estilos.css &gt; .scroller-image

```
811     para dar profundidade e destacá-lo
812     visualmente do resto do conteúdo. */
813
814     display: flex;
815     /* Usa Flexbox para gerenciar o layout dos
816     elementos dentro do scroller, simplificando o
817     alinhamento vertical. */
818
819     align-items: center;
820     /* Alinha os conteúdos internos verticalmente ao
821     centro, assegurando que tanto texto
822     quanto imagem estejam alinhados. */
823
824 }
825
826 .conteudo-image {
827
828     position: absolute;
829     /* Posiciona o conteúdo de forma absoluta dentro do
830     scroller, permitindo movimentação livre
831     dentro do espaço definido. */
832
833     white-space: nowrap;
834     /* Impede que o conteúdo (texto ou imagem) quebre em
835     várias linhas, essencial para a consistência
836     do deslizamento horizontal. */
837
838     font-size: 50px;
839     /* Define um tamanho de fonte grande para o texto,
840     garantindo visibilidade e impacto visual. */
```



EXPLORER

P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html

estilos.css

estilos.css &gt; .conteudo-image

```
841      color: #fff;
842      /* Configura a cor do texto para branco, promovendo alto
843         contraste com o fundo escuro. */
844
845      padding: 10px 20px;
846      /* Aplica preenchimento em torno do conteúdo,
847         aumentando a área de toque e
848         melhorando a legibilidade. */
849
850      display: flex;
851      /* Utiliza Flexbox dentro do conteúdo para alinhar
852         elementos como texto e imagens lado a lado. */
853
854      align-items: center;
855      /* Garante que tanto o texto quanto as imagens
856         estejam alinhados verticalmente ao
857         centro dentro do próprio conteúdo. */
858
859
860      animation: deslizar-image 15s linear infinite;
861      /* Aplica uma animação que move o conteúdo
862         horizontalmente de um lado ao outro do
863         scroller de forma contínua e suave. */
864
865 }
866
867 .conteudo-image img {
868
869     height: 100px; /* Define a altura da imagem para 100px,
870                     garantindo que ela se ajuste bem dentro do
```



EXPLORER

...

estilos.css X

&gt; □ ...



P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

script.js

zoom.html



&gt; OUTLINE

&gt; TIMELINE

estilos.css &gt; .conteudo-image img

```
871     scroller sem sobrepor o texto. */
872
873     margin-right: 20px;
874     /* Adiciona uma margem à direita da imagem,
875        criando espaço entre a imagem e o texto
876        seguinte, evitando que os elementos
877        fiquem visualmente comprimidos. */
878 }
879
880 @keyframes deslizar-image {
881
882     0% {
883
884         transform: translateX(calc(0% + 100vw));
885         /* Inicia a animação movendo o conteúdo de modo
886            que comece completamente fora da tela à
887            direita. O valor '100vw' representa 100% da
888            largura da viewport, garantindo que o
889            elemento inicie fora da área visível. */
890
891     }
892
893
894     100% {
895
896         transform: translateX(calc(-50% - 50vw));
897         /* Finaliza a animação empurrando o conteúdo para
898            além da largura da viewport para a esquerda.
899            Essa transformação garante que ele saia
900            completamente da área de visualização. */
```

The image shows a screenshot of a code editor application window. The top bar includes standard menu items: File, Edit, Selection, View, Go, Run, ..., and a search bar labeled "P36 - Scroller de Texto". The right side of the top bar features window control icons: minimize, maximize, close, and others.

The left sidebar contains several icons: a clipboard, a magnifying glass, a gear, a person, and a timeline. Below these are sections for "EXPLORER" and "P36 - SCROLLER DE TEXTO". The "P36 - SCROLLER DE TEXTO" section lists files: bounce.html, carro.jpg, estilos.css (which is selected), fade.html, fundo\_led\_2.jpg, fundo\_led\_3.jpg, fundo\_led.jpg, image.html, index.html, led.html, rotate.html, script.js, and zoom.html.

The main editor area displays the "estilos.css" file. The title bar of the editor says "estilos.css". The code shown is:

```
estilos.css > @keyframes deslizar-image
901 { }
902   }
903 }
```

The status bar at the bottom provides information about the current file: "Ln 901, Col 2", "Spaces: 4", "UTF-8", "CRLF", "CSS", and "Port : 5500 Reload".



EXPLORER ...

✓ P36 - SCROLLE... ⌂ ⌃ ⌄ ⌅

5 bounce.html

3 estilos.css

5 fade.html

 fundo\_led\_2.jpg

 fundo\_led\_3.jpg

 fundo\_led.jpg

5 image.htm

5 index.htm

5 led.html

5 rotate.htm

JS script.js

5 zoom.htm

JS script.js X

JS script is >



EXPLORER

...

JS script.js X

&gt; □ ...

## P36 - SCROLLER DE TEXTO

bounce.html

carro.jpg

estilos.css

fade.html

fundo\_led\_2.jpg

fundo\_led\_3.jpg

fundo\_led.jpg

image.html

index.html

led.html

rotate.html

JS script.js

zoom.html

JS script.js &gt; ⚡ document.addEventListener('DOMContentLoaded') callback &gt; ⚡ scrollers.forEach() callback &gt; ⚡ conteudoTexto.addEventListener('mouseout') callback

31 | };

32 | };

33 | };

34 | };



&gt; OUTLINE

&gt; TIMELINE



0 0 0

Ln 31, Col 4 Spaces: 8 UTF-8 CRLF {} JavaScript ⚡ Port : 5500 Reload

# **UC: Projetos**

## **Assunto:**

### **Small Projects**

### **P37 – Temporizador**

EXPLORER ...

P37 - TEMPORIZADOR

index.html

```
index.html > ...
1  <!DOCTYPE html>
2  <!-- Especifica que este documento usa HTML5, o que
3      ajuda o navegador a renderizar a página corretamente. -->
4
5  <html lang="pt-BR">
6  <!-- Abre a tag <html> que envolve todo o conteúdo da página.
7      O atributo 'lang' define o idioma principal do
8      documento como português do Brasil. -->
9
10 <head>
11   <!-- A tag <head> contém metadados (informações sobre os dados)
12       que não são exibidos diretamente na página web, mas
13       são usados pelo navegador. -->
14
15   <meta charset="UTF-8">
16   <!-- Define o conjunto de caracteres utilizado no documento
17       como UTF-8, que inclui quase todos os caracteres e
18       símbolos usados nas línguas escritas humanas.
19       Isso garante que qualquer texto no documento seja
20       exibido corretamente. -->
21
22   <meta name="viewport" content="width=device-width, initial-scale=1.0">
23   <!-- A tag <meta> com o atributo 'viewport' informa ao
24       navegador como controlar as dimensões e a escala da página.
25       - width=device-width: ajusta a largura da página
26           para a largura do dispositivo.
27       - initial-scale=1.0: define o nível de zoom inicial
28           quando a página é carregada pela primeira vez. -->
29
30   <title>Temporizador</title>
```

Ln 1, Col 1 Spaces: 7 UTF-8 CRLF HTML ⌂ Go Live Reload

EXPLORER ...

P37 - TEMPORIZADOR

- index.html
- script.js
- styles.css

index.html > html > head

```
31  <!-- A tag <title> define o título da página, que é mostrado
32  | na aba do navegador. Neste caso, 'Temporizador' é
33  | o título da página. -->
34
35  <link rel="stylesheet" href="styles.css">
36  <!-- A tag <link> é usada para vincular a folha de estilos
37  | externa, 'styles.css', ao documento HTML.
38  | Isso permite que o documento use as regras de estilo
39  | definidas no arquivo CSS especificado. -->
40
41 </head>
42 <!-- Fecha a tag <head>, indicando o fim do
43 | cabeçalho do documento. -->
44
45
46 <body>
47  <!-- A tag <body> contém todo o conteúdo da página
48  | que é visível para o usuário. -->
49
50 <div class="temporizador-container">
51  <!-- Div principal que contém todos os elementos do
52  | temporizador, usando uma classe para aplicar
53  | estilos específicos. -->
54
55 <h1>Temporizador</h1>
56  <!-- Título da página exibido como um cabeçalho, indicando
57  | que a funcionalidade principal é um temporizador. -->
58
59 <div id="temporizador">
60  <!-- Contêiner específico para mostrar o tempo do
```

Ln 31, Col 8 Spaces: 7 UTF-8 CRLF HTML ⚙ Go Live Reload



EXPLORER



#### ✓ P37 - TEMPORIZADOR



5 index.html >



EXPLORER

...

index.html X



P37 - TEMPORIZADOR

index.html

script.js

styles.css



&gt; OUTLINE

&gt; TIMELINE



⊗ 0 △ 0 ⌂ 0

index.html &gt; html &gt; body &gt; div.temporizador-container &gt; div.definir-tempo

```
91 <div class="definir-tempo">
92     <!-- Div que permite ao usuário definir o tempo
93         | inicial do temporizador. --&gt;
94
95     &lt;label for="entrada-horas"&gt;Horas:&lt;/label&gt;
96     <!-- Rótulo associado ao campo de entrada para horas. --&gt;
97
98     &lt;input type="number" id="entrada-horas" min="0" value="0"&gt;
99     <!-- Campo de entrada para definir as horas, do tipo
100        | numérico, com valor mínimo de 0 e inicialmente
101        | definido como 0. --&gt;
102
103    &lt;label for="entrada-minutos"&gt;Minutos:&lt;/label&gt;
104    <!-- Rótulo associado ao campo de entrada para minutos. --&gt;
105
106    &lt;input type="number" id="entrada-minutos" min="0" max="59" value="0"&gt;
107    <!-- Campo de entrada para definir os minutos, do tipo
108        | numérico, com valor mínimo de 0, máximo de 59 e
109        | inicialmente definido como 0. --&gt;
110
111    &lt;label for="entrada-segundos"&gt;Segundos:&lt;/label&gt;
112    <!-- Rótulo associado ao campo de entrada para segundos. --&gt;
113
114    &lt;input type="number" id="entrada-segundos" min="0" max="59" value="0"&gt;
115    <!-- Campo de entrada para definir os segundos, do tipo numérico,
116        | com valor mínimo de 0, máximo de 59 e inicialmente
117        | definido como 0. --&gt;
118
119    &lt;/div&gt;
120 &lt;/div&gt;</pre>
```



EXPLORER

...

index.html X

▷ □ ...



P37 - TEMPORIZADOR

index.html

script.js

styles.css



index.html &gt; html &gt; body

```
121      <script src="script.js"></script>
122      <!-- Vincula um arquivo JavaScript externo chamado 'script.js',
123          | que contém a lógica para operar o temporizador. --&gt;
124
125      &lt;/body&gt;
126      &lt;/html&gt;
127
128      <!-- Fecha as tags &lt;body&gt; e &lt;html&gt;, encerrando o documento HTML. --&gt;</pre>
```



0 0 0 0

Ln 121, Col 1 Spaces: 7 UTF-8 CRLF HTML ⚡ Go Live Reload



EXPLORER

...

## P37 - TEMPORIZADOR

index.html

script.js

styles.css



&gt; OUTLINE

&gt; TIMELINE



⑧ 0 △ 0 ⌂ 0

Ln 25, Col 1 Spaces: 8 UTF-8 CRLF CSS ⚙ Go Live Reload

styles.css X

styles.css &gt; body

```
1  body {  
2  
3      font-family: 'Arial', sans-serif;  
4      /* Define a família de fontes para o corpo do documento.  
5         'Arial' é a fonte principal, e 'sans-serif' é a  
6         fonte de reserva.  
7         Isso assegura que, se 'Arial' não estiver disponível,  
8         qualquer fonte sem serifa será usada. */  
9  
10     background-color: #2C3E50;  
11     /* Define a cor de fundo do corpo da página como um azul  
12        escuro (#2C3E50). Essa cor proporciona um contraste  
13        agradável com o texto e outros elementos claros,  
14        facilitando a leitura e melhorando a estética da página. */  
15  
16     color: #ECF0F1;  
17     /* Define a cor do texto para um cinza claro (#ECF0F1).  
18        Isso garante que o texto se destaque bem contra o  
19        fundo escuro, melhorando a legibilidade. */  
20  
21     display: flex;  
22     /* Configura o modelo de caixa do corpo da página para  
23        Flexbox, o que facilita o alinhamento e a distribuição  
24        dos elementos filhos dentro do corpo. */  
25  
26     justify-content: center;  
27     /* Alinha horizontalmente todos os elementos filhos (diretamente  
28        dentro do <body>) ao centro do corpo da página.  
29        Isso é útil para centrar todo o conteúdo principal  
30        na visualização horizontal. */
```



EXPLORER

...

styles.css X

&gt; □ ...



P37 - TEMPORIZADOR

index.html

script.js

styles.css



&gt; OUTLINE

&gt; TIMELINE

styles.css &gt; .temporizador-container

```
61     para um tom de azul-escuro (#34495E), proporcionando
62     um contraste agradável com textos e botões claros,
63     além de oferecer um aspecto visual sofisticado. */
64
65     padding: 40px;
66     /* Adiciona um preenchimento interno de 40 pixels em todas as
67      direções dentro do contêiner, criando espaço
68      entre o conteúdo e as bordas do contêiner, o que
69      melhora a estética e a legibilidade. */
70
71     border-radius: 10px;
72     /* Arredonda os cantos do contêiner com um raio de 10 pixels,
73      suavizando a aparência geral e oferecendo um
74      design mais moderno e menos rígido. */
75
76     box-shadow: 0 0 20px rgba(0, 0, 0, 0.3);
77     /* Aplica uma sombra ao redor do contêiner. A sombra é
78      projetada para ser sutil (com uma opacidade
79      de 30% - '0.3') e espalha-se por 20 pixels,
80      criando um efeito de elevação visual que destaca o
81      contêiner sobre o restante da interface. */
82
83 }
84
85 .temporizador-container h1 {
86
87     margin-bottom: 20px;
88     /* Define uma margem inferior de 20 pixels para o
89      cabeçalho dentro do contêiner, separando-o
90      visualmente dos elementos que estão abaixo dele.
```



EXPLORER



#### ✓ P37 - TEMPORIZADOR



5 index.html

5 index.html

JS script.js

3 styles.css

3 styles.css X



EXPLORER

...

styles.css X

D I ...



P37 - TEMPORIZADOR

index.html

script.js

styles.css



&gt; OUTLINE

&gt; TIMELINE

styles.css &gt; .botões

```
121     display: flex;
122     /* Estabelece o modelo de layout flexível para o
123      | contêiner, facilitando o alinhamento e ajuste
124      | dos botões internos. */
125
126     justify-content: center;
127     /* Centraliza os botões horizontalmente dentro do contêiner,
128      | garantindo uma apresentação visual equilibrada. */
129
130     gap: 10px;
131     /* Define um espaço de 10 pixels entre cada botão, evitando
132      | que eles toquem uns aos outros e melhorando a
133      | estética geral. */
134
135     margin-bottom: 20px;
136     /* Aplica uma margem inferior de 20 pixels após o contêiner
137      | dos botões, criando espaço entre este contêiner e
138      | quaisquer elementos abaixo dele. */
139
140 }
141
142 button {
143
144     background: #1ABC9C;
145     /* Define a cor de fundo dos botões para um tom atraente
146     | de verde-água (#1ABC9C), tornando-os visualmente
147     | distintos e agradáveis. */
148
149     border: none;
150     /* Remove a borda padrão dos botões para um design
```





EXPLORER

...

## P37 - TEMPORIZADOR

index.html

script.js

styles.css



styles.css X

styles.css &gt; button

```
151     |     mais limpo e moderno. /*  
152  
153     padding: 10px 20px;  
154     /* Adiciona preenchimento interno de 10 pixels na vertical  
155     |     e 20 pixels na horizontal, aumentando a área de  
156     |     clique e melhorando o conforto visual. */  
157  
158     border-radius: 5px;  
159     /* Arredonda as bordas dos botões com um raio de 5 pixels,  
160     |     suavizando o design geral e proporcionando uma  
161     |     estética mais amigável. */  
162  
163     color: #2C3E50;  
164     /* Define a cor do texto dentro dos botões como um azul  
165     |     escuro (#2C3E50), o que oferece um contraste  
166     |     eficaz com o fundo claro. */  
167  
168     font-size: 1em;  
169     /* Mantém o tamanho da fonte dos botões alinhado com o  
170     |     tamanho padrão da fonte do documento, garantindo  
171     |     consistência e legibilidade. */  
172  
173     cursor: pointer;  
174     /* Muda o cursor para uma mão apontando quando passa sobre o  
175     |     botão, indicando que o elemento é clicável. */  
176  
177     transition: background 0.3s;  
178     /* Adiciona uma transição suave de 0,3 segundos para a cor  
179     |     de fundo, melhorando a experiência do usuário ao  
180     |     interagir com o botão. */
```

EXPLORER ...

P37 - TEMPORIZADOR

- index.html
- script.js
- styles.css

styles.css X

styles.css > button

```
181
182 }
183
184 button:hover {
185
186     background: #095344;
187     /* Altera a cor de fundo do botão para um tom de verde
188      mais escuro (#095344) quando o mouse está sobre
189      ele, proporcionando feedback visual imediato. */
190
191     /* Altera a cor da fonte do botão para um branco. */
192     color: #FFFFFF;
193
194 }
195
196 button:disabled {
197
198     background: #95A5A6;
199     /* Define a cor de fundo para botões desabilitados como
200      cinza (#95A5A6), comunicando visualmente que o
201      botão está inativo. */
202
203     cursor: not-allowed;
204     /* Muda o cursor para um ícone de operação não permitida,
205      reforçando que o botão não pode ser clicado no
206      estado desabilitado. */
207
208 }
209
210 .definir-tempo {
```

Ln 181, Col 1 Spaces: 8 UTF-8 CRLF CSS Go Live Reload



EXPLORER

•

## 3 styles.css



5 index.html



5 index.htm



JS script.js



> OUTLINE

> TIMELINE



EXPLORER

...

## P37 - TEMPORIZADOR

index.html

script.js

styles.css



styles.css X

```
  styles.css > .definir-tempo input
241      width: 50px;
242      /* Define a largura dos campos de input para 50 pixels,
243         o que é suficiente para acomodar dois dígitos,
244         mantendo o design compacto e uniforme. */
245
246      padding: 5px;
247      /* Aplica um preenchimento interno de 5 pixels em todos os
248         lados dentro dos campos de input, aumentando a área
249         de clique e melhorando a experiência tátil. */
250
251      border-radius: 5px;
252      /* Arredonda as bordas dos campos de input com um raio de 5
253         pixels, suavizando a aparência geral dos campos e
254         tornando o design mais amigável e moderno. */
255
256      border: 1px solid #ccc;
257      /* Adiciona uma borda sólida de 1 pixel com a cor cinza
258         clara (#ccc) ao redor dos campos de input. Isso
259         define claramente os limites dos campos, melhorando a
260         definição visual em fundos escuros ou claros. */
261
262      text-align: center;
263      /* Alinha o texto digitado nos campos de input ao centro,
264         garantindo que os números estejam visualmente centrados
265         dentro do campo, o que ajuda na uniformidade e
266         na estética. */
267
268 }
```



EXPLORER

...

P37 - TEMPOR... ✚ ⟳ ⟳ ⟳

index.html

JS script.js

styles.css



&gt; OUTLINE

&gt; TIMELINE

JS script.js X

JS script.js &gt; ...

```
1 let horas = 0, minutos = 0, segundos = 0;
2 /* Declara três variáveis, 'horas', 'minutos' e
3     'segundos', inicializando todas com 0.
4     Essas variáveis armazenam os valores atuais de
5         horas, minutos e segundos do temporizador. */
6
7 let intervalo;
8 /* Declara a variável 'intervalo' sem inicializar um
9     valor. Esta variável será usada para
10    armazenar a referência ao setInterval que controla a
11    contagem do tempo, permitindo que ela seja parada ou
12    reiniciada conforme necessário. */
13
14 let pausado = false;
15 /* Declara e inicializa a variável 'pausado' com o
16     valor booleano false. Esta variável é usada para controlar
17     se o temporizador está pausado. Se true, a contagem do
18     tempo está pausada; se false, o temporizador está ativo. */
19
20 const atualizarDisplay = () => {
21     /* Define uma função chamada 'atualizarDisplay' que
22         atualiza o conteúdo textual dos elementos HTML
23         que mostram o tempo do temporizador, formatando as
24         horas, minutos e segundos para sempre mostrarem
25         dois dígitos. */
26
27     document.getElementById('horas').textContent = String(horas).padStart(2, '0');
28     /* Acessa o elemento com ID 'horas' e atualiza seu
29         conteúdo textual.
30         Converte o valor da variável 'horas' para uma string e
```



EXPLORER

...

JS script.js X

&gt; □ ...

P37 - TEMPORIZADOR

index.html

JS script.js

styles.css

JS script.js &gt; [?] atualizarDisplay

```
31     usa 'padStart(2, '0')' para garantir que o valor
32     sempre tenha pelo menos dois dígitos, preenchendo
33     com zeros à esquerda se necessário. */
34
35     document.getElementById('minutos').textContent = String(minutos).padStart(2, '0');
36     /* Funciona de maneira semelhante à linha anterior, mas
37      para o elemento com ID 'minutos' e a
38      variável 'minutos'. */
39
40     document.getElementById('segundos').textContent = String(segundos).padStart(2, '0');
41     /* Funciona de maneira semelhante às linhas anteriores,
42      mas para o elemento com ID 'segundos' e a
43      variável 'segundos'. */
44
45 }
46
47
48 const iniciarTemporizador = () => {
49     /* Define a função 'iniciarTemporizador' que é responsável
50      por iniciar a contagem do temporizador. */
51
52     if (intervalo) clearInterval(intervalo);
53     /* Verifica se já existe um intervalo ativo armazenado
54      na variável 'intervalo'.
55      Se existir, utiliza 'clearInterval' para parar o temporizador
56      anterior, evitando que múltiplos
57      temporizadores funcionem ao mesmo tempo. */
58
59     horas = parseInt(document.getElementById('entrada-horas').value);
60     minutos = parseInt(document.getElementById('entrada-minutos').value);
```



&gt; OUTLINE

&gt; TIMELINE



0 ⚠ 0 ⚡ 0

Ln 31, Col 5 Spaces: 8 UTF-8 CRLF {} JavaScript ⚡ Go Live Reload



EXPLORER ...

#### ✓ P37 - TEMPORIZADOR

5 index.html

JS script.js

JS script.js >

JS script.js > [?] iniciarTemporizado

```
61     segundos = parseInt(document.getElementById('entrada-segundos').value);
62     /* Recupera os valores dos campos de entrada (inputs) para
63      | horas, minutos e segundos no formulário.
64      | Utiliza 'document.getElementById' para acessar cada
65      | campo pelo seu ID e 'parseInt' para converter
66      | o valor de string para um número inteiro.
67      | Esses valores são então atribuídos às variáveis
68      | 'horas', 'minutos' e 'segundos'. */
69
70     if (horas === 0 && minutos === 0 && segundos === 0) {
71         alert('Defina um tempo válido para o temporizador.');
72         /* Verifica se todos os campos de tempo estão
73          | definidos como zero.
74          | Se verdadeiro, exibe um alerta ao usuário para
75          | definir um tempo válido, indicando que o
76          | temporizador não pode ser iniciado com
77          | todos os campos zerados. */
78         return;
79         /* Encerra a execução da função se o tempo não
80          | for válido, evitando que o temporizador seja iniciado. */
81     }
82
83     atualizarDisplay();
84     /* Chama a função 'atualizarDisplay' para atualizar a
85      | interface do usuário com os valores de tempo
86      | recém-definidos nos campos de entrada. */
87
88     pausado = false;
89     /* Define a variável 'pausado' como false, indicando
```



EXPLORER

...

JS script.js X

...



P37 - TEMPORIZADOR

index.html

JS script.js

styles.css



&gt; OUTLINE

&gt; TIMELINE

JS script.js &gt; [?] iniciarTemporizador

```
91     que o temporizador não está pausado e está pronto
92     para começar ou continuar a contagem. */
93
94     const intervalo = setInterval(() => {
95         /* Cria um intervalo que executa a função anônima a
96         |   cada 1000 milissegundos (1 segundo).
97         A variável 'intervalo' armazena a referência
98         |   desse intervalo para permitir que ele seja
99         |   interrompido com clearInterval quando necessário. */
100
101    if (!pausado) {
102        /* Verifica se o temporizador não está pausado antes
103        |   de proceder com a contagem regressiva. */
104
105    if (segundos === 0) {
106        /* Verifica se os segundos chegaram a zero para então
107        |   processar minutos e horas. */
108
109    if (minutos === 0) {
110        /* Verifica se os minutos também estão a zero, indicando
111        |   que é necessário decrementar as horas. */
112
113    if (horas === 0) {
114        /* Se as horas também estão a zero, o temporizador
115        |   alcançou o final da contagem.
116        |   É necessário então parar o intervalo e
117        |   notificar o usuário. */
118
119        clearInterval(intervalo);
120        /* Para o intervalo de tempo, interrompendo a
```

EXPLORER ... JS script.js X

P37 - TEMPORIZADOR

- index.html
- script.js
- styles.css

JS script.js > [i] iniciarTemporizador > setInterval() callback

```
121     | contagem regressiva. */  
122     |  
123         | alert('O tempo acabou!');  
124         | /* Exibe um alerta indicando que o tempo do  
125             | temporizador expirou. */  
126         |  
127         | reiniciarBotoes();  
128         | /* Chama a função 'reiniciarBotoes' para restabelecer os  
129             | estados iniciais dos botões. */  
130         |  
131         | return;  
132         | /* Sai da função para evitar mais execuções após o  
133             | término do temporizador. */  
134     } else {  
135         |  
136         |/* Se ainda houver horas restantes, decrementa uma hora e  
137             | ajusta minutos e segundos para 59. */  
138         | horas--;  
139         | minutos = 59;  
140         | segundos = 59;  
141         |  
142     }  
143     |  
144 } else {  
145     |/* Se ainda houver minutos, apenas decrementa um  
146         | minuto e ajusta os segundos para 59. */  
147     | minutos--;  
148     | segundos = 59;
```

Ln 121, Col 7 Spaces: 8 UTF-8 CRLF {} JavaScript ⚡ Go Live Reload

EXPLORER ... JS script.js X

JS script.js > [e] iniciarTemporizador > ⚡ setInterval() callback

```
151
152 }
153
154 } else {
155
156     /* Se os segundos ainda não estão a zero,
157      | simplesmente decrementa um segundo. */
158     segundos--;
159
160 }
161
162 atualizarDisplay();
163 /* Atualiza o display do temporizador para
164  | mostrar o tempo atualizado. */
165
166 }
167
168 }, 1000);
169 /* O intervalo executa a função a cada 1 segundo (1000 milissegundos). */
170
171 document.getElementById('iniciar').disabled = true;
172 /* Desabilita o botão de iniciar, pois o temporizador já
173  | está em execução e não deve ser reiniciado. */
174
175 document.getElementById('pausar').disabled = false;
176 /* Habilita o botão de pausar, permitindo ao usuário
177  | pausar o temporizador se necessário. */
178
179 document.getElementById('continuar').disabled = true;
180 /* Mantém o botão de continuar desabilitado, já que só
```



EXPLORER

...

JS script.js

X

P37 - TEMPORIZADOR

index.html

JS script.js

styles.css

JS script.js &gt; [?] iniciarTemporizador

```
181     deve ser habilitado após o temporizador
182     ser pausado. */
183
184     document.getElementById('resetar').disabled = false;
185     /* Habilita o botão de resetar, permitindo ao usuário
186         resetar o temporizador a qualquer momento
187         após o início. */
188 }
189
190
191
192
193 const reiniciarBotoes = () => {
194     /* Define a função 'reiniciarBotoes' que é responsável
195         por configurar o estado inicial dos botões
196         quando o temporizador é resetado ou quando a
197         página é carregada. */
198
199     document.getElementById('iniciar').disabled = false;
200     /* Habilita o botão 'Iniciar'. Isso permite ao
201         usuário iniciar o temporizador.
202         É importante que este botão esteja habilitado
203             inicialmente para permitir que o temporizador
204             seja iniciado. */
205
206     document.getElementById('pausar').disabled = true;
207     /* Desabilita o botão 'Pausar' porque não faz sentido
208         pausar um temporizador que ainda não foi iniciado. */
209
210     document.getElementById('continuar').disabled = true;
```



&gt; OUTLINE

&gt; TIMELINE



0 0 0 0

Ln 181, Col 4 Spaces: 8 UTF-8 CRLF {} JavaScript Go Live Reload



EXPLORER

...

JS script.js

X



P37 - TEMPORIZADOR

index.html

JS script.js

styles.css



&gt; OUTLINE

&gt; TIMELINE



⊗ 0 △ 0 ⌂ 0

...

Ln 211, Col 1 Spaces: 8 UTF-8 CRLF {} JavaScript ⌂ Go Live Reload

```
JS script.js > [?] reiniciarBotoes
211     /* Desabilita o botão 'Continuar' inicialmente, pois ele
212      | só deve ser habilitado após o temporizador ser pausado. */
213
214     document.getElementById('resetar').disabled = true;
215     /* Desabilita o botão 'Resetar' inicialmente, pois não
216      | há necessidade de resetar um temporizador que
217      | não começou. */
218
219 }
220
221 const pausarTemporizador = () => {
222     /* Define a função 'pausarTemporizador' que é
223      | responsável por pausar a contagem do
224      | temporizador. */
225
226     pausado = true;
227     /* Atribui o valor true à variável 'pausado'. Isso é
228      | usado para interromper a contagem no intervalo
229      | definido na função 'iniciarTemporizador'.
230      Quando 'pausado' é true, o código dentro do intervalo
231      | que decrementa o tempo não será executado. */
232
233     document.getElementById('pausar').disabled = true;
234     /* Busca o botão 'Pausar' pelo seu ID e desabilita-o. Isso
235      | impede que o usuário clique novamente em 'Pausar'
236      | enquanto o temporizador já está pausado, o que
237      | evita comportamentos redundantes ou confusos
238      | na interface. */
239
240     document.getElementById('continuar').disabled = false;
```

EXPLORER ... JS script.js X

P37 - TEMPORIZADOR

- index.html
- script.js
- styles.css

JS script.js > [?] pausarTemporizador

```
241     /* Busca o botão 'Continuar' pelo seu ID e habilita-o.  
242     Isso permite que o usuário possa continuar a contagem  
243     do temporizador após ele ter sido pausado. */  
244  
245 }  
246  
247 const continuarTemporizador = () => {  
248     /* Define a função 'continuarTemporizador' que é responsável  
249     por continuar a contagem do temporizador após  
250     ter sido pausado. */  
251  
252     pausado = false;  
253     /* Atribui o valor false à variável 'pausado'. Isso permite  
254     que o intervalo na função 'iniciarTemporizador'  
255     continue a contagem do tempo. */  
256  
257     document.getElementById('pausar').disabled = false;  
258     /* Busca o botão 'Pausar' pelo seu ID e habilita-o novamente.  
259     Isso permite que o usuário possa pausar o  
260     temporizador novamente se necessário. */  
261  
262     document.getElementById('continuar').disabled = true;  
263     /* Busca o botão 'Continuar' pelo seu ID e desabilita-o.  
264     Isso impede que o usuário clique em 'Continuar'  
265     enquanto o temporizador já está contando, mantendo a  
266     consistência e evitando comportamentos redundantes  
267     na interface. */  
268  
269 }  
270
```

< ⊞ 0 ⚠ 0 ⚡ 0 🔍 0 > OUTLINE > TIMELINE

Ln 241, Col 6 Spaces: 8 UTF-8 CRLF {} JavaScript ⚡ Go Live Reload

EXPLORER ... JS script.js X

JS script.js > ...

```
271 const resetarTemporizador = () => {
272     /* Define a função 'resetarTemporizador' responsável
273      por reiniciar completamente o temporizador,
274      zerando todas as variáveis de tempo e parando
275      qualquer contagem ativa. */
276
277     clearInterval(intervalo);
278     /* Utiliza a função 'clearInterval' para parar o
279      intervalo de tempo que está controlando
280      o temporizador.
281      'intervalo' é a referência ao setInterval
282      iniciado pela função 'iniciarTemporizador'.
283      Isso efetivamente para o temporizador, impedindo
284      que ele continue a contar. */
285
286     horas = 0;
287     /* Reinicia a variável 'horas' para 0, zerando a
288      contagem de horas no temporizador. */
289
290     minutos = 0;
291     /* Reinicia a variável 'minutos' para 0, zerando a
292      contagem de minutos no temporizador. */
293
294     segundos = 0;
295     /* Reinicia a variável 'segundos' para 0, zerando a
296      contagem de segundos no temporizador. */
297
298     pausado = false;
299     /* Define a variável 'pausado' como false. Isso garante
300      que o temporizador não esteja em estado de pausa
```

Ln 271, Col 1 Spaces: 8 UTF-8 CRLF {} JavaScript ⌂ Go Live Reload

EXPLORER ... JS script.js X

JS script.js > [?] resetarTemporizador

```
301     |     após ser resetado, permitindo que ele seja
302     |     reiniciado sem interferência. */
303
304     |     atualizarDisplay();
305     /* Chama a função 'atualizarDisplay' para atualizar os
306         |     elementos de display do temporizador na interface do usuário.
307         |     Como as variáveis de tempo foram zeradas, o
308         |     display mostrará '00:00:00'. */
309
310     |     reiniciarBotoes();
311     /* Chama a função 'reiniciarBotoes' para restabelecer os
312         |     estados iniciais dos botões de controle.
313         |     Isso geralmente envolve desabilitar os botões que
314         |     não devem ser usáveis quando o temporizador está parado
315         |     e garantindo que o botão 'Iniciar' esteja habilitado. */
316
317 }
318
319
320
321 document.getElementById('iniciar').addEventListener('click', iniciarTemporizador);
322 /* Adiciona um ouvinte de evento ao botão 'Iniciar'. Quando
323     |     clicado, a função 'iniciarTemporizador' será chamada,
324     |     iniciando a contagem do temporizador. */
325
326 document.getElementById('pausar').addEventListener('click', pausarTemporizador);
327 /* Adiciona um ouvinte de evento ao botão 'Pausar'. Quando
328     |     clicado, a função 'pausarTemporizador' será chamada,
329     |     pausando a contagem do temporizador. */
330
```

Ln 301, Col 1 Spaces: 8 UTF-8 CRLF {} JavaScript ⚙ Go Live Reload



EXPLORER

...

JS script.js X

▷ ⌂ ...



P37 - TEMPORIZADOR

index.html

JS script.js

styles.css



&gt; OUTLINE

&gt; TIMELINE



⊗ 0 ▲ 0 ⌂ 0

# **UC: Projetos**

## **Assunto:**

### **Small Projects**

#### **P38 – Projeto com Tooltip**



EXPLORER

...

5 index.html X

&gt; □ ...



P38 - PROJETO COM TOOLTIP

&gt; imagemProduto

&gt; imagens

5 index.html

JS script.js

3 styles.css

x Vendedor.xlsx



5 index.html &gt; ...

```
1  <!DOCTYPE html>
2  <!-- A declaração DOCTYPE é uma instrução para o navegador
3  sobre qual versão do HTML a página está escrita. Neste
4  caso, indica HTML5, o padrão atual da web, assegurando que os
5  navegadores renderizem a página usando as regras de
6  sintaxe do HTML5. -->
7
8  <html lang="pt-br">
9  <!-- A tag <html> marca o início do documento HTML. O atributo 'lang'
10 | define o idioma principal do documento como português do
11 | Brasil (pt-br), o que ajuda os motores de busca e as tecnologias
12 | assistivas a entenderem o conteúdo da página. -->
13
14 <head>
15 <!-- A tag <head> contém metadados, links para folhas de estilo, scripts e
16 | outras informações que não são diretamente visíveis na área de
17 | conteúdo do navegador. -->
18
19 <meta charset="UTF-8">
20 <!-- A tag <meta> com o atributo 'charset' definido como "UTF-8"
21 | especifica a codificação de caracteres para o documento. UTF-8 é
22 | uma codificação de caracteres universal que inclui praticamente
23 | todos os caracteres e símbolos do mundo, assegurando que qualquer
24 | texto seja exibido corretamente. -->
25
26 <meta name="viewport" content="width=device-width, initial-scale=1.0">
27 <!-- Outra tag <meta> que configura a viewport para controlar como a
28 | página é dimensionada e visualizada em diferentes dispositivos.
29 | O atributo 'content' com 'width=device-width' instrui o navegador a
30 | definir a largura da viewport de acordo com a largura do dispositivo.
```



&gt; OUTLINE

&gt; TIMELINE



⊗ 0 △ 0 ⌂ 0

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF HTML Ø Port: 5500 Reload



EXPLORER

## P38 - PROJETO COM TOOLTIP

&gt; imagemProduto

&gt; imagens

5 index.html

JS script.js

3 styles.css

Vendedor.xlsx

## index.html

## index.html &gt; html &gt; head

```
31          o que é crucial para design responsivo. 'initial-scale=1.0'  
32          estabelece o nível de zoom inicial quando a página é carregada,  
33          promovendo uma melhor acessibilidade e visualização inicial. -->  
34  
35      <title>Projeto Vendedores</title>  
36      <!-- A tag <title> define o título da página, que é exibido na aba do navegador.  
37          Este título é também utilizado pelos motores de busca e pode  
38          influenciar o SEO (Search Engine Optimization) da página.  
39          "Projeto Vendedores" é um título que dá uma ideia clara do  
40          conteúdo da página. -->  
41  
42      <link rel="stylesheet" href="styles.css">  
43      <!-- A tag <link> é usada aqui para vincular uma folha de estilo  
44          externa, localizada no arquivo 'styles.css'. Esse arquivo  
45          contém todas as regras de estilo CSS que definem a aparência  
46          da página. Isso ajuda a separar o conteúdo (HTML) da  
47          apresentação (CSS), facilitando a manutenção e atualizações do design. -->  
48  
49      <script src="https://cdnjs.cloudflare.com/ajax/libs/xlsx/0.18.5/xlsx.full.min.js"></script>  
50      <!-- A tag <script> com um atributo 'src' que aponta para um URL  
51          CDN (Content Delivery Network) que hospeda a biblioteca 'xlsx.full.min.js'.  
52          Esta biblioteca JavaScript é utilizada para ler e manipular  
53          arquivos do Excel diretamente no navegador, permitindo que a  
54          página processe e exiba dados de arquivos Excel sem a necessidade  
55          de um backend para parseamento dos dados. -->  
56  
57  </head>  
58  
59  <body>  
60  <!-- A tag <body> contém todo o conteúdo visível da página web. Aqui, são
```



&gt; OUTLINE

&gt; TIMELINE



⊗ 0 △ 0 ⌂ 0



EXPLORER

...

index.html X

&gt; □ ...



P38 - PROJETO COM TOOLTIP

&gt; imagemProduto

&gt; imagens

index.html

script.js

styles.css

Vendedor.xlsx



&gt; OUTLINE

&gt; TIMELINE

index.html &gt; html &gt; body

```
61      definidos todos os elementos que os usuários podem interagir e
62      ver no navegador. -->
63
64      <h1>Dados dos Vendedores</h1>
65      <!-- A tag <h1> define um cabeçalho de nível um, o mais importante dentro
66          de uma página. "Dados dos Vendedores" serve como o título principal
67          da página, indicando o tema ou o conteúdo principal para os usuários e
68          também é importante para SEO. -->
69
70      <div class="filtros">
71      <!-- Um <div> com a classe "filtros" agrupa visualmente e semanticamente os
72          campos de entrada utilizados para filtrar os dados da tabela.
73          Isso melhora a organização do layout e ajuda na aplicação
74          de estilos CSS específicos. -->
75
76      <input type="text" id="filtro-vendedor" placeholder="Filtrar por Vendedor">
77      <!-- Campo de entrada para filtrar os dados da tabela por vendedor.
78          O atributo 'placeholder' fornece uma dica ao usuário sobre o
79          que inserir no campo. -->
80
81      <input type="text" id="filtro-produto" placeholder="Filtrar por Produto">
82      <!-- Similar ao campo anterior, mas destinado ao filtro por produto. -->
83
84      <input type="text" id="filtro-preco" placeholder="Filtrar por Preço">
85      <!-- Campo de entrada para filtrar os dados da tabela por preço. Isso
86          permite aos usuários restringir a visualização dos dados
87          conforme especificado. -->
88
89      </div>
```

EXPLORER

P38 - PROJETO COM TOOLTIP

- > imagemProduto
- > imagens
- 5 index.html
- JS script.js
- 3 styles.css
- Vendedor.xlsx

index.html X

index.html > html > body > table#tabela-vendedores

```
91     <table id="tabela-vendedores">
92         <!-- A tag <table> é usada para definir uma tabela HTML, que será
93             preenchida dinamicamente com os dados dos vendedores.
94             O atributo 'id' permite referenciar e manipular a
95             tabela via JavaScript. -->
96
97     <thead>
98         <!-- A seção <thead> define o cabeçalho da tabela. -->
99
100        <tr>
101            <!-- A tag <tr> define uma linha na tabela. -->
102
103            <th>Imagen</th>
104                <!-- Cada <th> define uma célula de cabeçalho, que descreve os
105                    dados das colunas. 'Imagen' indica que nesta coluna serão
106                    exibidas imagens. -->
107
108            <th>Vendedor</th>
109                <!-- Descreve a coluna que mostrará o nome do vendedor. -->
110
111            <th>Produto</th>
112                <!-- Descreve a coluna para os produtos vendidos. -->
113
114            <th>Total</th>
115                <!-- Descreve a coluna que mostrará o total de vendas ou preço
116                    total dos produtos vendidos. -->
117
118        </tr>
119    </thead>
```

Ln 91, Col 7 Spaces: 4 UTF-8 CRLF HTML ⚡ Port: 5500 Reload

EXPLORER

P38 - PROJETO COM TOOLTIP

- > imagemProduto
- > imagens
- 5 index.html
- JS script.js
- 3 styles.css
- Vendedor.xlsx

index.html X

index.html > html > body > table#tabela-vendedores > tbody

```
121     <tbody>
122
123         <!-- A seção <tbody> contém o corpo principal da tabela, onde os
124             |   |   |   dados serão inseridos dinamicamente via JavaScript. -->
125             |   |   |       <!-- Dados serão carregados aqui -->
126
127     </tbody>
128
129 </table>
130
131
132     <div id="tooltip" class="tooltip">
133         <!-- Uma <div> com a classe 'tooltip' e o id 'tooltip' é usada para mostrar
134             |   |   |   informações adicionais quando o usuário passa o mouse sobre
135             |   |   |       elementos da tabela. Estilo e comportamento são definidos
136             |   |   |       no CSS e JavaScript, respectivamente. -->
137
138     </div>
139
140     <script src="script.js"></script>
141         <!-- A tag <script> com o atributo 'src' que aponta para o arquivo 'script.js'
142             |   |   |   é usada para incluir um arquivo JavaScript externo que contém a
143             |   |   |       lógica para carregar os dados na tabela, filtrar os dados, e
144             |   |   |       gerenciar outras interações do usuário na página. -->
145
146 </body>
147 </html>
```

OUTLINE

TIMELINE

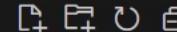
x 0 ▲ 0 ⌂ 0

Ln 121, Col 11 Spaces: 4 UTF-8 CRLF HTML ⚡ Port : 5500 Reload



EXPLORER

P38 - PROJETO COM TOOLTIP



&gt; imagemProduto

&gt; imagens

5 index.html

JS script.js

styles.css

Vendedor.xlsx



&gt; OUTLINE

&gt; TIMELINE

styles.css X

styles.css &gt; body

```
1  body {  
2  
3      font-family: Arial, sans-serif;  
4      /* Define a família de fontes para o corpo do documento como 'Arial',  
5         com um fallback para qualquer fonte 'sans-serif' se 'Arial' não  
6         estiver disponível. 'Arial' é uma escolha comum por sua legibilidade e  
7         aparência limpa, adequada para textos em interfaces de usuário. */  
8  }  
9  
10 h1 {  
11  
12     text-align: center;  
13     /* Alinha o texto do cabeçalho principal (h1) ao centro do contêiner. Isso é  
14        útil para títulos para garantir que eles sejam um ponto  
15        focal claro na página. */  
16  }  
17  
18 table {  
19  
20  
21     width: 100%;  
22     border-collapse: collapse;  
23     /* Estabelece que as tabelas ocupem 100% da largura do contêiner pai,  
24        proporcionando uma disposição mais fluida e integrada.  
25  
26     'border-collapse: collapse;' remove as bordas duplas entre células  
27     adjacentes, tornando a tabela mais coesa e esteticamente agradável. */  
28  }  
29  
30 }
```

EXPLORER

P38 - PROJETO COM TOOLTIP

- > imagemProduto
- > imagens
- index.html
- script.js
- styles.css
- Vendedor.xlsx

styles.css

```
31 th, td {  
32     border: 1px solid #ddd;  
33     padding: 8px;  
34     text-align: center;  
35     /* Define um estilo comum para as células de cabeçalho (th)  
36     e as células de dados (td):  
37     - 'border: 1px solid #ddd;' adiciona uma borda sólida com uma  
38         cor cinza claro (#ddd) para delimitar claramente as células.  
39     - 'padding: 8px;' adiciona um espaçamento interno de 8 pixels em  
40         todas as direções dentro de cada célula para evitar que o  
41         conteúdo fique muito compacto.  
42     - 'text-align: center;' centraliza o conteúdo textual dentro das  
43         células, melhorando a legibilidade e a apresentação  
44         uniforme dos dados. */  
45 }  
46  
47 th {  
48     background-color: #f2f2f2;  
49     /* Aplica uma cor de fundo cinza claro (#f2f2f2) para as células  
50         de cabeçalho. Isso ajuda a diferenciar visualmente as  
51         células de cabeçalho das células de dados, tornando a  
52         tabela mais fácil de navegar visualmente. */  
53 }  
54  
55 }  
56  
57 }  
58  
59 tr:hover {  
60 }
```

OUTLINE

TIMELINE

Ln 31, Col 5 Spaces: 4 UTF-8 CRLF CSS Ø Port: 5500 Reload

EXPLORER

P38 - PROJETO COM TOOLTIP

- > imagemProduto
- > imagens
- 5 index.html
- JS script.js
- styles.css
- Vendedor.xlsx

styles.css X

styles.css > tr:hover

```
background-color: #f5f5f5;
/* Define uma cor de fundo cinza muito claro (#f5f5f5) para uma
   linha da tabela quando o mouse passa sobre ela ('hover').
   Esta interatividade visual ajuda os usuários a focar em uma
   linha por vez, facilitando a leitura de dados em tabelas densas. */

.filtros {
    margin-bottom: 20px;
    /* Adiciona uma margem inferior de 20px ao contêiner `.filtros`.
       Isso cria um espaço vertical entre os filtros e os elementos
       subsequentes na página, evitando que os componentes fiquem
       visualmente amontoados e melhorando a clareza e a separação dos
       elementos de interface do usuário. */
}

.filtros input {
    padding: 10px;
    /* Adiciona um preenchimento de 10px ao redor do conteúdo interno dos
       elementos de entrada dentro do `.filtros`. Isso aumenta a área
       clicável e melhora a legibilidade do texto inserido. */

    margin-right: 10px;
    /* Aplica uma margem direita de 10px a cada campo de entrada, criando espaço
       entre eles quando dispostos horizontalmente. Isso previne que os
       elementos de entrada se toquem, mantendo uma distância visual
```

0 ▲ 0 ⌂ 0

Ln 61, Col 5 Spaces: 4 UTF-8 CRLF CSS Ø Port: 5500 Reload

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P38 - Projeto com Tooltip
- Left Sidebar (Explorer):**
  - P38 - PROJETO COM TOOLTIP
    - imagemProduto
    - imagens
    - index.html
    - script.js
    - styles.css (selected)
    - Vendedor.xlsx
  - OUTLINE
  - TIMELINE
- Central Area:** styles.css (Line 91 to Line 120)

```
91     |     agradável e funcional. */
92
93     border: 1px solid #ddd;
94     /* Adiciona uma borda sólida de 1px com a cor cinza claro (#ddd) ao redor
95     dos campos de entrada, definindo claramente os limites do campo e
96     melhorando o contraste com o fundo. */
97
98     border-radius: 5px;
99     /* Arredonda os cantos dos campos de entrada com um raio de 5px, proporcionando
100    uma estética moderna e suave que é frequentemente associada a designs
101    amigáveis e acessíveis. */
102
103 }
104
105 .tooltip {
106
107     position: absolute;
108     /* Define a posição do elemento `.tooltip` como absoluta, o que permite que
109     ele seja posicionado em relação ao seu contêiner posicionado mais
110     próximo. Isso é crucial para o posicionamento dinâmico do tooltip
111     com base nas interações do usuário. */
112
113     display: none;
114     /* Inicialmente define o tooltip como oculto (`display: none`), fazendo
115     com que ele não afete o layout da página até ser necessário.
116     Isso evita distrações visuais até que o tooltip seja ativado. */
117
118     background-color: white;
119     /* Define a cor de fundo do tooltip como branca, garantindo que o conteúdo
120     dentro do tooltip seja fácil de ler contra o fundo claro. */
```
- Bottom Status Bar:** Line 91, Col 4, Spaces: 4, UTF-8, CRLF, CSS, Port: 5500, Reload

EXPLORER

P38 - PROJETO COM TOOLTIP

- > imagemProduto
- > imagens
- 5 index.html
- JS script.js
- styles.css
- Vendedor.xlsx

styles.css X

styles.css > .tooltip

```
121 border: 1px solid #ddd;
122 /* Adiciona uma borda de 1px de espessura com cor cinza claro (#ddd), que
123     delimita o tooltip do restante do conteúdo da página e reforça a
124     percepção visual do mesmo como um elemento flutuante e distinto. */
125
126 padding: 10px;
127 /* Aplica um preenchimento interno de 10px para todas as direções
128     dentro do tooltip, criando espaço entre o conteúdo e as
129     bordas do tooltip. Isso melhora a legibilidade e a
130     apresentação estética do conteúdo. */
131
132 box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
133 /* Adiciona uma sombra ao redor do tooltip, aumentando a percepção de
134     profundidade e destacando o tooltip sobre outros elementos
135     da página. A sombra é sutil, usando preto com 10% de
136     opacidade, o que é suficiente para criar um contraste
137     visual sem ser dominante. */
138
139 z-index: 1000;
140 /* Define um alto valor para `z-index` garantindo que o tooltip fique
141     acima de outros elementos na página. Isso é importante para
142     assegurar que o tooltip não seja bloqueado ou escondido
143     por outros elementos visuais. */
144
145 width: 420px;
146 /* Define a largura do tooltip para 420px, garantindo que haja espaço
147     suficiente para exibir o conteúdo desejado de forma clara e
148     legível. Essa largura fixa ajuda na consistência do layout dos
149     tooltips independentemente do conteúdo exibido. */
150
```

Ln 121, Col 1 Spaces: 4 UTF-8 CRLF CSS ⚡ Port : 5500 Reload

EXPLORER

P38 - PROJETO COM TOOLTIP

- > imagemProduto
- > imagens
- index.html
- script.js
- styles.css
- Vendedor.xlsx

styles.css

styles.css > .tooltip

```
151
152 }
153
154
155 .tooltip img {
156     display: block;
157     /* Define que as imagens dentro do tooltip serão tratadas como blocos.
158      | Isso significa que ocuparão uma linha inteira por si só, o que é
159      | útil para centralizá-las horizontalmente. */
160
161     margin: 0 auto;
162     /* Aplica margem automática em ambos os lados da imagem. Isso centraliza a
163      | | imagem horizontalmente dentro do seu contêiner, criando um layout
164      | | visualmente equilibrado e atraente. */
165
166 }
167
168 .tooltip .produto-imagem {
169     width: 200px;
170     /* Define a largura da imagem do produto dentro do tooltip como 200px,
171      | garantindo que a imagem seja grande o suficiente para ser
172      | claramente visível, mas não tão grande a ponto de dominar o
173      | espaço do tooltip. */
174
175     height: 200px;
176     /* Define a altura da imagem do produto para corresponder à sua largura,
177      | mantendo a imagem proporcional e bem dimensionada. Isso ajuda a
178      | manter a qualidade visual e a consistência das imagens apresentadas. */
179
180
```

OUTLINE

TIMELINE

0 ▲ 0 ⌂ 0

Ln 151, Col 1 Spaces: 4 UTF-8 CRLF CSS Ø Port: 5500 Reload

EXPLORER

P38 - PROJETO COM TOOLTIP

- > imagemProduto
- > imagens
- 5 index.html
- JS script.js
- styles.css
- Vendedor.xlsx

styles.css

styles.css > .tooltip .produto-imagem

```
181
182 }
183
184 .tooltip .produto-nome {
185     text-align: center;
186     /* Centraliza o texto do nome do produto dentro do tooltip, facilitando a
187      | leitura e mantendo a consistência com o alinhamento centralizado
188      | da imagem acima. */
189
190
191     font-size: 18px;
192     /* Define o tamanho da fonte para o nome do produto como 18px, o que é
193      | adequado para destacar o nome sem sobrecarregar
194      | visualmente o usuário. */
195
196     margin-top: 10px;
197     /* Aplica uma margem superior de 10px acima do nome do produto,
198      | criando um espaço visual entre a imagem do produto e o
199      | texto, o que ajuda a separar claramente os elementos
200      | dentro do tooltip. */
201
202 }
203
204 .tooltip .total {
205     text-align: center;
206     /* Centraliza o texto que apresenta o total das vendas ou preço,
207      | alinhando-o visualmente com o nome do produto e a imagem. */
208
209     font-size: 24px;
```

Ln 181, Col 1 Spaces: 4 UTF-8 CRLF CSS ⚡ Port : 5500 Reload

EXPLORER

P38 - PROJETO COM TOOLTIP

- > imagemProduto
- > imagens
- 5 index.html
- JS script.js
- styles.css
- Vendedor.xlsx

styles.css

.tooltip .total

```
211  /* Define o tamanho da fonte para o total das vendas como 24px,  
212  | | | tornando-o proeminente e fácil de ler, o que é importante  
213  | | | para destacar valores numéricos significativos. */  
214  
215  margin-top: 10px;  
216  /* Adiciona uma margem superior de 10px para separar visualmente o  
217  | | | valor total do nome do produto acima, ajudando a organizar e  
218  | | | diferenciar claramente as informações dentro do tooltip. */  
219  
220  font-weight: bold;  
221  /* Aplica um peso de fonte negrito para o total das vendas,  
222  | | | enfatizando ainda mais essa informação crucial,  
223  | | | garantindo que se destaque entre outros textos  
224  | | | dentro do tooltip. */  
225  
226 }
```

OUTLINE

TIMELINE

x 0 △ 0 ⌂ 0

Ln 211, Col 13 Spaces: 4 UTF-8 CRLF CSS Ø Port : 5500 Reload

The image shows a code editor interface with a tooltip displayed over a variable in a script file. The tooltip provides detailed information about the variable's purpose and usage.

**File Edit Selection View Go Run ...**

P38 - Projeto com Tooltip

EXPLORER

P38 - PROJETO COM TOOLTIP

- imagemProduto
- imagens
- index.html
- JS script.js**
- styles.css
- Vendedor.xlsx

JS script.js

```
document.addEventListener('DOMContentLoaded', function () {  
    /* Adiciona um ouvinte de evento ao objeto document para o  
    evento 'DOMContentLoaded', que é disparado quando o HTML  
    inicial do documento foi completamente carregado e analisado,  
    sem esperar pelo CSS, imagens e subframes para terminar de carregar.  
    Isso garante que o código JavaScript não será executado até que o  
    HTML esteja totalmente disponível para manipulação, evitando erros  
    causados por tentar acessar elementos que ainda não foram carregados. */  
  
    const tabela = document.getElementById('tabela-vendedores').getElementsByTagName('tbody')[0];  
    /* Declara uma constante 'tabela' e atribui a ela o primeiro elemento 'tbody'  
     * dentro do elemento com ID 'tabela-vendedores'.  
     * - 'document.getElementById('tabela-vendedores')' seleciona o elemento  
     *   tabela pelo seu ID.  
     * - '.getElementsByTagName('tbody')[0]' obtém o primeiro (e, neste caso único)  
     *   elemento 'tbody' dentro da tabela.  
     * Esta abordagem é utilizada para direcionar diretamente o corpo da tabela  
     * onde os dados dos vendedores serão inseridos dinamicamente. */  
  
    const tooltip = document.getElementById('tooltip');  
    /* Declara uma constante 'tooltip' e atribui a ela o elemento  
     * com ID 'tooltip'. Este elemento será usado para mostrar  
     * informações adicionais sobre itens específicos na tabela  
     * quando o usuário interagir com eles, como passar o mouse  
     * sobre uma linha. Este tooltip é inicialmente oculto no CSS e  
     * será posicionado e mostrado dinamicamente via JavaScript. */  
  
    let dadosTabela = [];  
    /* Declara uma variável 'dadosTabela' e inicializa-a como um array vazio.  
     * Esta variável será usada para armazenar os dados extraídos do arquivo
```

In 1 Col 1 Spaces: 8 UFT-8 CRLF {} JavaScript ⚡ Port: 5500 Reload ⚡



EXPLORER

## P38 - PROJETO COM TOOLTIP

&gt; imagemProduto

&gt; imagens

5 index.html

JS script.js

styles.css

Vendedor.xlsx

JS script.js X

JS script.js &gt; document.addEventListener('DOMContentLoaded') callback

```
31     Excel. A utilização de um array permite que os dados sejam facilmente
32     manipulados e atualizados dinamicamente na tabela HTML conforme necessário. */
33
34     // Função para carregar o arquivo Excel
35     function carregarExcel() {
36         /* Define a função 'carregarExcel', que é responsável por carregar e
37         processar o arquivo Excel. Esta função não recebe argumentos e
38         encapsula todo o processo de acesso ao arquivo, leitura e extração
39         de dados, seguido pela atualização da tabela na interface do usuário. */
40
41     fetch('Vendedor.xlsx')
42     /* Utiliza a função global 'fetch' para fazer uma requisição HTTP GET
43     ao arquivo 'Vendedor.xlsx'. A função 'fetch' retorna uma promessa
44     que resolve em um objeto de resposta do qual podemos extrair o corpo
45     da resposta em vários formatos, neste caso, como um ArrayBuffer. */
46
47     .then(response => response.arrayBuffer())
48     /* O primeiro '.then' captura a resposta da requisição e chama
49     'response.arrayBuffer()', que lê o corpo da resposta e o
50     retorna como um ArrayBuffer. Este formato é necessário para o
51     processamento subsequente pelo XLSX.js. */
52
53     .then(data => {
54         /* O segundo '.then' recebe o ArrayBuffer e procede com o
55         processamento do arquivo Excel. */
56
57         const workbook = XLSX.read(data, { type: 'array' });
58         /* Utiliza a função 'read' do XLSX.js para converter o ArrayBuffer
59         em um objeto 'workbook' que representa o arquivo Excel.
60         A opção '{ type: 'array' }' informa ao XLSX.js que os
```



&gt; OUTLINE

&gt; TIMELINE



0 0 △ 0 ⌂ 0

EXPLORER

P38 - PROJETO COM TOOLTIP

- > imagemProduto
- > imagens
- 5 index.html
- JS script.js**
- 3 styles.css
- Vendedor.xlsx

JS script.js

```
JS script.js > document.addEventListener('DOMContentLoaded') callback > carregarExcel > then() callback
dados estão em um ArrayBuffer. */

const sheetName = workbook.SheetNames[0];
/* Extrai o nome da primeira planilha do arquivo Excel.
'workbook.SheetNames' é um array que contém os nomes de
todas as planilhas no arquivo; '[0]' acessa o nome da
primeira planilha. */

const sheet = workbook.Sheets[sheetName];
/* Acessa a primeira planilha usando o nome obtido anteriormente.
'workbook.Sheets' é um objeto onde cada chave é o nome
de uma planilha e o valor correspondente é a planilha em si. */

const json = XLSX.utils.sheet_to_json(sheet);
/* Converte a planilha em um array de objetos JSON, onde cada objeto
representa uma linha da planilha e cada chave do objeto
representa uma coluna. Isso facilita a manipulação e exibição
dos dados na tabela HTML. */

dadosTabela = json;
/* Atribui os dados convertidos para a variável 'dadosTabela', que
armazena os dados que serão usados para preencher a tabela
na interface do usuário. */

preencherTabela(dadosTabela);
/* Chama a função 'preencherTabela' passando os dados da tabela
como argumento. Esta função é responsável por inserir os
dados na tabela HTML na página. */

}
```

Ln 61, Col 7 Spaces: 8 UTF-8 CRLF {} JavaScript Ø Port: 5500 Reload



EXPLORER

## P38 - PROJETO COM TOOLTIP

&gt; imagemProduto

&gt; imagens

5 index.html

JS script.js

3 styles.css

Vendedor.xlsx

JS script.js X

JS script.js &gt; document.addEventListener('DOMContentLoaded') callback &gt; preencherTabela &gt; dados.forEach() callback

```
91
92     .catch(error => console.error('Erro ao carregar o arquivo Excel:', error));
93     /* O método '.catch' é usado para capturar e tratar qualquer erro
94      que possa ocorrer durante o processo de requisição ou
95      processamento do arquivo Excel. Os erros são exibidos no
96      console, ajudando na diagnóstico e correção de problemas. */
97
98 }
99
100 // Função para preencher a tabela com os dados do Excel
101 function preencherTabela(dados) {
102     /* Define a função 'preencherTabela', que recebe um array 'dados'
103      como argumento. Esta função é responsável por limpar e
104      então preencher a tabela HTML com os novos dados
105      extraídos de um arquivo Excel. */
106
107     tabela.innerHTML = '';
108     /* Limpa o conteúdo atual da tabela, definindo 'innerHTML' para
109      uma string vazia. Isso é necessário para remover quaisquer
110      linhas de dados anteriores antes de adicionar novas, garantindo
111      que a tabela mostre apenas os dados mais recentes. */
112
113     dados.forEach(linha => {
114         /* Itera sobre cada objeto 'linha' no array 'dados'. Cada 'linha'
115             representa uma linha de dados do Excel. */
116
117         const novaLinha = tabela.insertRow();
118         /* Cria uma nova linha na tabela e a armazena na variável 'novaLinha'.
119             'insertRow()' é um método que adiciona uma nova linha
120             ao final da tabela. */
```



&gt; OUTLINE

&gt; TIMELINE



0 ▲ 0 ⌂ 0



EXPLORER

## P38 - PROJETO COM TOOLTIP

- > imagemProduto
- > imagens
- 5 index.html
- JS script.js**
- styles.css
- Vendedor.xlsx

JS script.js X

JS script.js &gt; document.addEventListener('DOMContentLoaded') callback &gt; preencherTabela &gt; dados.forEach() callback

```
121 const celulaImagen = novaLinha.insertCell();
122 /* Insere uma nova célula na 'novaLinha' criada e armazena esta
123    célula na variável 'celulaImagen'. Esta célula será usada
124    para exibir a imagem do vendedor. */
125
126 const imagem = document.createElement('img');
127 /* Cria um novo elemento 'img', que será usado para exibir a
128    imagem do vendedor dentro da célula. */
129
130
131 imagem.src = `imagens/${linha.Vendedor}.jpg`;
132 /* Define o atributo 'src' do elemento 'img', construindo o
133    caminho do arquivo baseado no nome do vendedor. Assumimos
134    que as imagens estão nomeadas de acordo com os vendedores e
135    armazenadas na pasta 'imagens/'. */
136
137 imagem.alt = linha.Vendedor;
138 /* Define o atributo 'alt' para a imagem, fornecendo uma descrição
139    alternativa que é o nome do vendedor. Isso é útil para
140    acessibilidade e caso a imagem não possa ser carregada. */
141
142 imagem.width = 50;
143 imagem.height = 50;
144 /* Define a largura e a altura da imagem para 50 pixels, garantindo
145    que todas as imagens tenham um tamanho uniforme e não
146    distorçam a apresentação da tabela. */
147
148 celulaImagen.appendChild(imagem);
149 /* Adiciona o elemento 'img' à 'celulaImagen', fazendo com que a
150    imagem seja exibida dentro dessa célula específica na tabela. */
```



&gt; OUTLINE

&gt; TIMELINE



0 ▲ 0 ⌂ 0

0 ▲ 0 ⌂ 0

Ln 121, Col 1 Spaces: 8 UTF-8 CRLF {} JavaScript Ø Port: 5500 Reload

EXPLORER

P38 - PROJETO COM TOOLTIP

- > imagemProduto
- > imagens
- 5 index.html
- JS script.js**
- 3 styles.css
- Vendedor.xlsx

JS script.js

JS script.js > document.addEventListener('DOMContentLoaded') callback > preencherTabela > dados.forEach() callback

```
151 const celulaVendedor = novaLinha.insertCell();
152 /* Insere outra célula na linha para o nome do vendedor e armazena
153 | essa célula na variável 'celulaVendedor'. */
154
155 celulaVendedor.textContent = linha.Vendedor;
156 /* Define o conteúdo de texto da 'celulaVendedor' como o nome do
157 | vendedor, exibindo-o na tabela. */
158
159
160 const celulaProduto = novaLinha.insertCell();
161 /* Insere uma célula para o produto vendido e armazena esta
162 | célula na variável 'celulaProduto'. */
163
164 celulaProduto.textContent = linha.Produto;
165 /* Define o conteúdo de texto da 'celulaProduto' como o nome
166 | do produto, exibindo-o na tabela. */
167
168 const totalFormatado = new Intl.NumberFormat('pt-BR', { style: 'currency', currency: 'BRL' }).format(linha.Total);
169 /* Formata o valor total das vendas para o formato de moeda
170 | brasileiro (Real), usando a classe 'Intl.NumberFormat'. Isso
171 | transforma números em uma string formatada que é mais legível e
172 | adequada para exibição monetária. */
173
174 const celulaTotal = novaLinha.insertCell();
175 /* Insere uma célula para o total formatado e armazena essa
176 | célula na variável 'celulaTotal'. */
177
178 celulaTotal.textContent = totalFormatado;
179 /* Define o conteúdo de texto da 'celulaTotal' como o total das
```

Ln 151, Col 1 Spaces: 8 UTF-8 CRLF {} JavaScript Ø Port : 5500 Reload

EXPLORER

P38 - PROJETO COM TOOLTIP

- > imagemProduto
- > imagens
- 5 index.html
- JS script.js**
- 3 styles.css
- Vendedor.xlsx

JS script.js

JS script.js > document.addEventListener('DOMContentLoaded') callback > preencherTabela > dados.forEach() callback

```
180 vendas já formatado, exibindo-o na tabela. */

181
182
183 novaLinha.addEventListener('mouseover', () => {
184     /* Adiciona um ouvinte de evento 'mouseover' à 'novaLinha'
185     criada na tabela. O evento 'mouseover' é disparado sempre
186     que o cursor do mouse entra no espaço ocupado pela 'novaLinha'
187     na tabela. A função de callback (função anônima) é definida
188     para ser executada quando este evento ocorre. */
189
190     tooltip.innerHTML =
191         
193             <div class="produto-nome">${linha.Produto}</div>
194             <div class="total">${totalFormatado}</div>
195         ;
196     /* Define o conteúdo HTML interno do 'tooltip' quando o evento 'mouseover' ocorre.
197     - A primeira linha dentro do `tooltip` é um elemento 'img' que
198         busca uma imagem com base no nome do produto,
199         as imagens são nomeadas de acordo com os produtos e
200         armazenadas na pasta 'imagemProduto/'. O atributo 'class' é
201         definido como 'produto-imagem', que pode ser usado para
202             estilização específica no CSS.
203     - A segunda linha é um 'div' que exibe o nome do produto. Esta
204         informação ajuda o usuário a identificar rapidamente o
205             produto sem ter que ler a tabela.
206     - A terceira linha é outro 'div' que mostra o 'totalFormatado',
207         apresentando o valor total das vendas ou o preço do
208             produto em formato monetário. */
```

Ln 180, Col 1 Spaces: 8 UTF-8 CRLF {} JavaScript ⚡ Port : 5500 Reload 🔍



EXPLORER

## P38 - PROJETO COM TOOLTIP

&gt; imagemProduto

&gt; imagens

5 index.html

JS script.js

3 styles.css

Vendedor.xlsx



&gt; OUTLINE

&gt; TIMELINE



JS script.js X

```
JS script.js > document.addEventListener('DOMContentLoaded') callback > preencherTabela > dados.forEach() callback > novaLinha.addEventListener('DOMContentLoad...')

209     tooltip.style.display = 'block';
210     /* Altera a propriedade de estilo 'display' do 'tooltip' para 'block', o
211      que torna o 'tooltip' visível na página. Por padrão, o
212      'tooltip' pode ter 'display: none' aplicado no CSS para
213      mantê-lo oculto quando não está em uso. Definir para 'block'
214      assegura que ele se torne visível e sobreponha outros elementos
215      conforme necessário, fornecendo informações contextuais
216      adicionais ao usuário. */
217
218 });
219
220
221 novaLinha.addEventListener('mousemove', (event) => {
222     /* Adiciona um ouvinte de evento 'mousemove' à 'novaLinha'. Este
223      evento é disparado sempre que o mouse se move sobre o
224      elemento 'novaLinha'. A função de callback recebe o
225      objeto 'event', que contém informações sobre o evento de
226      mouse, incluindo a posição do cursor. */
227
228     tooltip.style.left = `${event.pageX + 10}px`;
229     /* Define a propriedade CSS 'left' do 'tooltip', posicionando-o
230      horizontalmente. 'event.pageX' fornece a posição horizontal
231      do cursor na página. Adicionando 10 pixels à posição, o
232      'tooltip' é posicionado um pouco à direita do cursor, evitando
233      que o cursor cubra o conteúdo do 'tooltip' e melhorando a
234      visibilidade e a acessibilidade. */
235
236     tooltip.style.top = `${event.pageY + 10}px`;
237     /* Define a propriedade CSS 'top' do 'tooltip', posicionando-o
238      verticalmente. Similarmente, 'event.pageY' fornece a
```

File Edit Selection View Go Run ... ← → P38 - Projeto com Tooltip

EXPLORER ... JS script.js X

P38 - PROJETO COM TOOLTIP

- > imagemProduto
- > imagens
- 5 index.html
- JS script.js
- styles.css
- Vendedor.xlsx

```
ment.addEventListener('DOMContentLoaded') callback > preencherTabela > dados.forEach() callback > novaLinha.addEventListener('mousemove') callback
239     posição vertical do cursor na página. Adicionando 10
240 pixels, o 'tooltip' é posicionado um pouco abaixo do
241 cursor, seguindo a mesma lógica de não obstruir o
242 conteúdo com o cursor. */
243 );
244
245 novaLinha.addEventListener('mouseout', () => {
246     /* Adiciona um ouvinte de evento 'mouseout' à 'novaLinha'.
247     Este evento é disparado quando o cursor do mouse
248     deixa o espaço ocupado pelo elemento 'novaLinha'. Não
249     recebe nenhum argumento adicional porque a ação necessária
250     não depende da posição ou do movimento do cursor. */
251
252     tooltip.style.display = 'none';
253     /* Define a propriedade de estilo 'display' do 'tooltip' para 'none',
254     tornando o 'tooltip' invisível. Este comportamento é
255     crucial para garantir que o 'tooltip' só apareça quando o
256     usuário está interagindo diretamente com a linha correspondente,
257     evitando confusão visual e mantendo a interface limpa e
258     organizada quando o 'tooltip' não é necessário. */
259
260 );
261
262 });
263 }
264
265 // Função para filtrar a tabela
266 function filtrarTabela() {
267     /* Define a função 'filtrarTabela', que não recebe argumentos. Esta
```

Ln 239, Col 4 Spaces: 8 UTF-8 CRLF {} JavaScript Ø Port: 5500 Reload



EXPLORER



✓ P38 - PROJETO COM TOOLTIP



```
> imagemProduto 269
> imagens 270
> index.html 271
JS script.js 272
styles.css 273
Vendedor.xlsx 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

    função é responsável por aplicar filtros aos dados da tabela
    com base nos valores inseridos nos campos de filtro da página. */

const filtroVendedor = document.getElementById('filtro-vendedor').value.toLowerCase();
/* Obtém o valor do campo de filtro de vendedor, converte para
   minúsculas e armazena na constante 'filtroVendedor'. A conversão
   para minúsculas garante que a comparação de strings seja insensível a
   maiúsculas/minúsculas, aumentando a robustez do filtro. */

const filtroProduto = document.getElementById('filtro-produto').value.toLowerCase();
/* Similar ao 'filtroVendedor', obtém o valor do campo de filtro de
   produto, converte para minúsculas e armazena na
   constante 'filtroProduto'. */

const filtroPreco = document.getElementById('filtro-preco').value;
/* Obtém o valor do campo de filtro de preço e armazena na
   constante 'filtroPreco'. Note que não convertemos o
   valor para minúsculas, pois se trata de um valor numérico
   que será formatado e comparado como string. */

const dadosFiltrados = dadosTabela.filter(linha => {
    /* Utiliza o método 'filter' para criar um novo array 'dadosFiltrados'.
       Este método percorre cada 'linha' do array 'dadosTabela' e
       aplica uma função de teste que retorna 'true' para as linhas
       que correspondem aos critérios de filtragem. */

    const vendedorMatch = linha.Vendedor.toLowerCase().includes(filtroVendedor);
    /* Verifica se o nome do vendedor na linha atual contém o texto
       do filtro de vendedor. Retorna 'true' se contém, significando
       que a linha deve ser incluída nos dados filtrados. */
```

EXPLORER

P38 - PROJETO COM TOOLTIP

- > imagemProduto
- > imagens
- 5 index.html
- JS script.js**
- 3 styles.css
- Vendedor.xlsx

JS script.js

```
JS script.js > document.addEventListener('DOMContentLoaded') callback > filtrarTabela > dadosFiltrados > dadosTabela.filter() callback
```

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

```
const produtoMatch = linha.Produto.toLowerCase().includes(filtroProduto);
/* Verifica se o nome do produto na linha atual contém o texto
   do filtro de produto. */

const precoMatch = filtroPreco ? new Intl.NumberFormat('pt-BR', { style: 'currency', currency: 'BRL' }).format(linha.Total).includes(filtroPreco) : true;
/* Avalia se o valor total formatado como moeda contém o texto
   do filtro de preço. Se 'filtroPreco' estiver vazio, 'precoMatch'
   automaticamente retorna 'true', assumindo que o filtro por
   preço não deve ser aplicado. */

return vendedorMatch && produtoMatch && precoMatch;
/* Retorna 'true' se todas as condições de filtro forem satisfeitas
   para a linha, indicando que ela deve ser mantida nos dados filtrados. */

};

preencherTabela(dadosFiltrados);
/* Chama a função 'preencherTabela' com o array 'dadosFiltrados'
   como argumento, atualizando a tabela na interface do usuário
   para mostrar apenas as linhas que correspondem aos critérios de filtro. */

}

document.getElementById('filtro-vendedor').addEventListener('input', filtrarTabela);
/* Adiciona um ouvinte de evento ao campo de entrada do filtro de
   vendedor. O evento 'input' é acionado sempre que o usuário
   digita ou altera o conteúdo no campo de entrada.
```



# **UC: Projetos**

## **Assunto:**

### Small Projects

### P39 – Jogo Tetris

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P40 - Jogo Tetris
- Icons:** Explorer, Search, Open, Save, Help, Outline, Timeline.
- Left Sidebar (Explorer):** P40 - JOGO TETRIS folder containing index.html, script.js, and styles.css.
- Active File:** index.html
- Code Content:**

```
1  <!DOCTYPE html>
2  <!-- Declaração do tipo de documento, que é HTML5 --&gt;
3
4  &lt;html lang="pt-br"&gt;
5  <!-- Define o idioma da página como português do Brasil --&gt;
6
7  &lt;head&gt;
8      <!-- Cabeçalho do documento HTML --&gt;
9
10 &lt;meta charset="UTF-8"&gt;
11  <!-- Define a codificação de caracteres como UTF-8, que
12      suporta a maioria dos caracteres de todas as línguas escritas --&gt;
13
14 &lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt;
15  <!-- Define como a página deve ser exibida em diferentes
16      dispositivos, garantindo que ela seja responsiva (se
17      adapte a diferentes tamanhos de tela) --&gt;
18
19 &lt;title&gt;Tetris&lt;/title&gt;
20  <!-- Título da página, que será exibido na aba
21      do navegador --&gt;
22
23 &lt;link rel="stylesheet" href="styles.css"&gt;
24  <!-- Inclui um arquivo CSS externo para estilizar a página --&gt;
25
26 &lt;/head&gt;
27
28 &lt;body&gt;
29  <!-- Corpo do documento HTML, onde o conteúdo
30      visível para o usuário é colocado --&gt;</pre>- Bottom Status Bar: Q, Ln 1, Col 1, Spaces: 4, UTF-8, CRLF, HTML, Go Live, Bell icon.

```

The screenshot shows a code editor interface with the following details:

- File Bar:** File Edit Selection View Go Run ...
- Search Bar:** P40 - Jogo Tetris
- Icons:** Explorer, Find, Open, Save, Help, Outline (with a blue circle containing a white number 1), Timeline.
- Explorer:** P40 - JOGO TETRIS contains index.html, script.js, and styles.css.
- Code Editor:** The file index.html is open, showing the following structure:

```
<h1>Tetris</h1>
<!-- Cabeçalho de nível 1 com o título do jogo --&gt;

&lt;div id="Pontuacao"&gt;
    <!-- Div que exibe a pontuação do jogador --&gt;

    Pontuação:

    &lt;span id="PontuacaoValor"&gt;0&lt;/span&gt;
    <!-- O valor inicial da pontuação é 0 e está dentro de um elemento &lt;span&gt;
        com o id "PontuacaoValor". Este valor será atualizado dinamicamente
        pelo JavaScript conforme o jogador ganha pontos no jogo --&gt;

&lt;/div&gt;

&lt;div id="jogo"&gt;
    <!-- Div que contém o canvas do jogo e as peças disponíveis --&gt;

    &lt;canvas id="tetris" width="300" height="600"&gt;&lt;/canvas&gt;
    <!-- Elemento canvas onde o jogo Tetris será desenhado.
        A largura é de 300 pixels e a altura é de 600 pixels.
        O canvas é uma área de desenho onde o JavaScript pode renderizar gráficos
        e animações, como o jogo Tetris. --&gt;

    &lt;div id="pecasDisponiveis"&gt;
        <!-- Div que contém as peças disponíveis para o
            jogador selecionar --&gt;

        &lt;h2&gt;Peças Disponíveis&lt;/h2&gt;</pre>
```
- Status Bar:** Ln 31, Col 1 Spaces: 4 UTF-8 CRLF HTML Go Live

The screenshot shows a code editor interface with the following details:

- File Bar:** File Edit Selection View Go Run ...
- Search Bar:** P40 - Jogo Tetris
- Icons:** Explorer, Find, Open, Save, Help, Outline, Timeline.
- Outline View:** Shows files: index.html, script.js, styles.css.
- Editor Area:** Displays the content of index.html.

```
<!-- Cabeçalho de nível 2 para a seção de peças disponíveis -->
<div id="pecas"></div>
<!-- Div que irá conter todas as peças disponíveis
     que o jogador pode escolher.
     Esta área será preenchida dinamicamente pelo
     JavaScript com as diferentes
     peças de Tetris que o jogador pode clicar para selecionar. -->
</div>
</div>

<script src="script.js"></script>
<!-- Inclui um arquivo JavaScript externo que contém a
     lógica do jogo Tetris.
     Este arquivo irá controlar a renderização do jogo no
     canvas, a atualização da pontuação
     e a lógica para selecionar e desenhar as peças disponíveis. -->
</body>
</html>
```

A screenshot of a code editor interface, likely Visual Studio Code, displaying CSS code for a Tetris game. The editor has a dark theme with light-colored syntax highlighting.

The top bar includes standard file navigation like File, Edit, Selection, View, Go, Run, and a search bar labeled "P40 - Jogo Tetris".

The left sidebar contains icons for Explorer, Search, Problems, and Outline/Timeline, with "OUTLINE" currently selected.

The main workspace shows two tabs: "index.html" and "# styles.css". The "# styles.css" tab is active, showing the following CSS code:

```
/* Estilos para o corpo (body) da página */
body {
    /* Define que o layout será flexível usando Flexbox */
    display: flex;

    /* Define que os elementos filhos serão dispostos em
       uma coluna (verticalmente) */
    flex-direction: column;

    /* Centraliza os elementos filhos horizontalmente */
    align-items: center;

    /* Define a fonte padrão para o texto no corpo
       como Arial, ou sans-serif se Arial não estiver disponível */
    font-family: Arial, sans-serif;
}

/* Estilos para o cabeçalho de nível 1 (h1) */
h1 {
    /* Define uma margem superior de 20 pixels */
    margin-top: 10px;
}

/* Estilos para o elemento com id 'Pontuacao' */
#Pontuacao {
```

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P40 - Jogo Tetris
- Explorer:** Shows a project structure with files: index.html, script.js, and styles.css.
- Code Editor:** The # styles.css file is open, containing the following CSS code:

```
/* Define uma margem inferior de 20 pixels */
margin-bottom: 10px;

/* Define o tamanho da fonte como 30 pixels */
font-size: 30px;

}

/* Estilos para o elemento com id 'jogo' */
#jogo {
    /* Define que o layout será flexível usando Flexbox */
    display: flex;
}

/* Estilos para o elemento <canvas> */
canvas {
    /* Define uma borda ao redor do canvas com 2
       pixels de largura e cor preta */
    border: 2px solid black;

}

/* Estilos para o elemento com id 'pecasDisponiveis' */
#pecasDisponiveis {
```

The code editor has a dark theme with syntax highlighting for CSS. The file # styles.css is currently selected in the sidebar.

The screenshot shows a code editor interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P40 - Jogo Tetris
- Icons:** Explorer, Find, Open, Save, Help, Outline, Timeline.
- Project Explorer:** P40 - JOGO TETRIS (index.html, script.js, # styles.css)
- Current File:** # styles.css
- Code Content:**

```
/* Define que o layout será flexível usando Flexbox */
display: flex;

/* Define que os elementos filhos serão dispostos em
   uma coluna (verticalmente) */
flex-direction: column;

/* Centraliza os elementos filhos horizontalmente */
align-items: center;

/* Define uma margem à esquerda de 20 pixels para criar
   espaço entre o canvas e as peças disponíveis */
margin-left: 20px;

}

/* Estilos para o elemento com id 'pecas' */
#pecas {

    /* Define que o layout será flexível usando Flexbox */
    display: flex;

    /* Define que os elementos filhos serão
       dispostos em uma coluna (verticalmente) */
    flex-direction: column;

    /* Define um espaço (gap) de 10 pixels entre
       os elementos filhos */
    gap: 10px;
```

- Status Bar:** Ln 61, Col 5, Spaces: 4, UTF-8, CRLF, CSS, Go Live, Bell icon.

A screenshot of a code editor interface, likely Visual Studio Code, displaying CSS code for a Tetris game. The editor has a dark theme with light-colored syntax highlighting.

The top bar includes standard file navigation: File, Edit, Selection, View, Go, Run, and a search bar labeled "P40 - Jogo Tetris".

The left sidebar contains icons for Explorer, Search, Problems, Editor Groups, Preview, and a Help section. The "OUTLINE" and "TIMELINE" sections are visible at the bottom of the sidebar.

The main workspace shows two tabs: "index.html" and "# styles.css". The "# styles.css" tab is active, showing the following CSS code:

```
/* Estilos para os elementos com a classe 'peca' */
.peca {
    /* Define que o layout será em grade (grid) */
    display: grid;
    /* Define um espaço (gap) de 2 pixels entre
       os itens da grade */
    gap: 2px;
}

/* Estilos específicos para a peça com o formato "I" */
.peca[data-shape="I"] {
    /* Define uma grade (grid) com 4 colunas, cada uma
       com 30 pixels de largura */
    grid-template-columns: repeat(4, 30px);
}

/* Estilos específicos para a peça com o formato "O" */
.peca[data-shape="O"] {
    /* Define uma grade (grid) com 2 colunas, cada uma
       com 30 pixels de largura */
    grid-template-columns: repeat(2, 30px);
}
```

The code editor status bar at the bottom shows: "Ln 91, Col 1" and "Spaces: 4".

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P40 - Jogo Tetris
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Split, ...
- Left Sidebar (Explorer):** EXPLORER, P40 - JOGO TETRIS (index.html, script.js, # styles.css), OUTLINE (1), TIMELINE.
- Central Area:** Editor showing CSS code for a Tetris game.

```
# styles.css > .peca[data-shape="O"]  
121 }  
122  
123 /* Estilos para as peças com os  
124 | formatos "T", "L", "J", "S" e "Z" */  
125 .peca[data-shape="T"],  
126 .peca[data-shape="L"],  
127 .peca[data-shape="J"],  
128 .peca[data-shape="S"],  
129 .peca[data-shape="Z"] {  
130  
131 /* Define uma grade (grid) com 3 colunas, cada  
132 | uma com 30 pixels de largura */  
133 grid-template-columns: repeat(3, 30px);  
134  
135 }  
136  
137  
138 /* Estilos para cada bloco individual dentro de uma peça */  
139 .peca div {  
140  
141 /* Define a largura de cada bloco como 30 pixels */  
142 width: 30px;  
143  
144 /* Define a altura de cada bloco como 30 pixels */  
145 height: 30px;  
146  
147 /* Define uma borda de 1 pixel de largura e  
148 | cor preta ao redor de cada bloco */  
149 border: 1px solid black;  
150 }
```
- Bottom Status Bar:** ⌂ 0 △ 0 ⌂ 0, Ln 121, Col 2, Spaces: 4, UTF-8, CRLF, CSS, Go Live, Bell icon.

A screenshot of a code editor interface, likely Microsoft Visual Studio Code, displaying a script for a Tetris game. The title bar reads "P40 - Jogo Tetris". The left sidebar shows icons for Explorer, Search, Problems, and Outline/Timeline, with "OUTLINE" having a blue notification badge. The Explorer panel lists files: "index.html", "# styles.css", "script.js" (selected), and "# styles.css". The main editor area contains the following JavaScript code:

```
1 const canvas = document.getElementById('tetris');
2 // Obtém o elemento canvas com o id 'tetris' do documento HTML.
3 // O canvas é a área de desenho onde o jogo será renderizado.
4
5 const contexto = canvas.getContext('2d');
6 // Obtém o contexto de renderização 2D do canvas.
7 // O contexto 2D é usado para desenhar formas, texto,
8 // imagens e outros objetos gráficos.
9
10 const tamanhoBloco = 30;
11 // Define o tamanho de cada bloco do jogo como 30 pixels.
12 // Cada peça do Tetris é composta por blocos de 30x30 pixels.
13
14 const linhas = 20;
15 // Define o número de linhas do tabuleiro de jogo.
16 // O tabuleiro terá 20 linhas verticais.
17
18 const colunas = 10;
19 // Define o número de colunas do tabuleiro de jogo.
20 // O tabuleiro terá 10 colunas horizontais.
21
22 const tabuleiro = [];
23 // Inicializa um array vazio que representará o tabuleiro do jogo.
24 // Cada célula do tabuleiro armazenará informações sobre se
25 // um bloco está presente nessa posição.
26
27 let pecaAtual;
28 // Declara uma variável para armazenar a peça atual que está caindo.
29 // A peça será um array bidimensional representando sua forma.
30
```

A screenshot of a code editor interface, likely Microsoft Visual Studio Code, displaying a JavaScript file for a Tetris game. The title bar reads "P40 - Jogo Tetris".

The left sidebar shows the project structure under "P40 - JOGO TETRIS": "index.html", "script.js" (selected), and "# styles.css".

The main editor area contains the following code:

```
31 let intervalo;
32 // Declara uma variável para armazenar o
33 // identificador do intervalo do jogo.
34 // Esta variável será usada para controlar o loop do jogo.
35
36 let pontuacao = 0;
37 // Inicializa a pontuação do jogador como 0.
38 // A pontuação aumentará conforme o jogador completa linhas no jogo.
39
40 let pecaX, pecaY;
41 // Declara variáveis para armazenar a posição x e y da peça atual.
42 // Essas variáveis serão usadas para mover e desenhar a peça no tabuleiro.
43
44 let gameOver = false;
45 // Inicializa a variável gameOver como falsa.
46 // Esta variável será usada para determinar se o jogo terminou.
47
48 const pecas = [
49
50   { shape: [[1, 1, 1, 1]], name: "I" }, // I
51   // Peça em formato de linha. Representada por uma matriz 1x4.
52   // O nome "I" refere-se ao seu formato semelhante a uma linha reta.
53
54   { shape: [[1, 1], [1, 1]], name: "O" }, // O
55   // Peça em formato de quadrado. Representada por uma matriz 2x2.
56   // O nome "O" refere-se ao seu formato semelhante a um quadrado.
57
58   { shape: [[0, 1, 0], [1, 1, 1]], name: "T" }, // T
59   // Peça em formato de "T". Representada por uma matriz 2x3.
60   // O nome "T" refere-se ao seu formato que lembra a letra T.
```

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P40 - Jogo Tetris
- Toolbar:** Standard window control icons.
- Left Sidebar:** Explorer, Search, Find, Preview, Outline, Timeline.
- Central Area:** Script.js file open, showing code for a Tetris game.

```
index.html styles.css script.js (active)
```

**script.js Content:**

```
61 { shape: [[1, 1, 0], [0, 1, 1]], name: "S" }, // S
62 // Peça em formato de "S". Representada por uma matriz 2x3.
63 // O nome "S" refere-se ao seu formato que lembra a letra S.

64 { shape: [[0, 1, 1], [1, 1, 0]], name: "Z" }, // Z
65 // Peça em formato de "Z". Representada por uma matriz 2x3.
66 // O nome "Z" refere-se ao seu formato que lembra a letra Z.

67 { shape: [[1, 1, 1], [1, 0, 0]], name: "L" }, // L
68 // Peça em formato de "L". Representada por uma matriz 2x3.
69 // O nome "L" refere-se ao seu formato que lembra a letra L.

70 { shape: [[1, 1, 1], [0, 0, 1]], name: "J" } // J
71 // Peça em formato de "J". Representada por uma matriz 2x3.
72 // O nome "J" refere-se ao seu formato que lembra a letra J.

73
74 ];
75
76
77
78
79
80 // Função para inicializar a pontuação a partir do LocalStorage
81 function inicializarPontuacao() {
82
83     // Tenta obter o valor da pontuação salvo no LocalStorage do
84     // navegador, usando a chave 'pontuacao'
85     const pontuacaoSalva = localStorage.getItem('pontuacao');
86
87     // Verifica se há um valor salvo no LocalStorage
88     if (pontuacaoSalva !== null) {
89
90         // Se houver um valor salvo, converte essa string para
```

**Bottom Status Bar:** < > 0 △ 0 ⌂ 0 Q Ln 61, Col 1 Spaces: 4 UTF-8 CRLF {} JavaScript ⌂ Go Live ⌂



A screenshot of a code editor interface, likely Visual Studio Code, displaying a JavaScript file for a Tetris game. The top bar shows the title "P40 - Jogo Tetris". The left sidebar contains icons for Explorer, Search, Problems, and Outline/Timeline, with "OUTLINE" and "TIMELINE" currently selected. The main area shows the file structure and content:

- Files: index.html, styles.css, script.js
- Script.js content (lines 121-150):

```
// Inicializa o tabuleiro
function inicializarTabuleiro() {
    // Loop que percorre cada linha do tabuleiro
    // Começa com 'linha' igual a 0 e incrementa até chegar ao
    // número total de linhas definido pela variável 'linhas'
    for (let linha = 0; linha < linhas; linha++) {
        // Cria uma nova linha no array tabuleiro
        // Cada índice do array 'tabuleiro' será um array
        // representando uma linha do tabuleiro
        tabuleiro[linha] = [];

        // Loop que percorre cada coluna da linha atual do tabuleiro
        // Começa com 'coluna' igual a 0 e incrementa até chegar ao
        // número total de colunas definido pela variável 'colunas'
        for (let coluna = 0; coluna < colunas; coluna++) {
            // Inicializa cada célula da linha atual com 0
            // Cada posição no array bidimensional 'tabuleiro' será inicializada com o valor 0
            // O valor 0 indica que a célula está vazia (sem bloco), pois
            // nenhuma peça ocupa essa posição
            tabuleiro[linha][coluna] = 0;
        }
    }
}

// Desenha um bloco
function desenharBloco(x, y, cor) {
```

The "script.js" file is the active tab, indicated by a yellow background. The code initializes a 2D array "tabuleiro" to represent the game board, with rows and columns both starting at 0.

The screenshot shows a code editor interface with the following details:

- File Bar:** File Edit Selection View Go Run ...
- Search Bar:** P40 - Jogo Tetris
- Icons:** Explorer, Search, Find, Open, Save, Help, Outline, Timeline.
- Project Explorer:** P40 - JOGO TETRIS (index.html, script.js, # styles.css)
- Code Editor:** Script.js (selected tab)
- Code Content:** (Lines 151-180) A snippet of JavaScript code for drawing blocks and the board in a canvas. It defines fill and stroke styles, calculates pixel positions, and loops through the board grid.
- Status Bar:** OUTLINE (1), TIMELINE, Line 151, Col 1, Spaces: 4, UTF-8, CRLF, {} JavaScript, Go Live, Bell icon.

```
// Define a cor de preenchimento para o bloco
contexto.fillStyle = cor;

// Desenha um retângulo preenchido no canvas
// x * tamanhoBloco e y * tamanhoBloco calculam a
// posição do bloco em pixels
// tamanhoBloco define a largura e altura do bloco
contexto.fillRect(x * tamanhoBloco, y * tamanhoBloco, tamanhoBloco, tamanhoBloco);

// Define a cor da borda do bloco como preto
contexto.strokeStyle = 'black';

// Desenha a borda do retângulo no canvas
// x * tamanhoBloco e y * tamanhoBloco calculam a
// posição do bloco em pixels
// tamanhoBloco define a largura e altura do bloco
contexto.strokeRect(x * tamanhoBloco, y * tamanhoBloco, tamanhoBloco, tamanhoBloco);

}

// Desenha o tabuleiro
function desenharTabuleiro() {

    // Loop que percorre cada linha do tabuleiro
    for (let linha = 0; linha < linhas; linha++) {

        // Loop que percorre cada coluna da linha atual do tabuleiro
        for (let coluna = 0; coluna < colunas; coluna++) {
```

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P40 - Jogo Tetris
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Split, ...
- Left Sidebar (EXPLORER):** P40 - JOGO TETRIS (index.html, script.js, styles.css)
- Current File:** script.js
- Code Content:** A portion of a JavaScript script for a Tetris game. The code includes functions for checking the board, generating new pieces, and drawing pieces. The code is in Portuguese.

```
// Verifica se há um bloco na célula atual do tabuleiro
if (tabuleiro[linha][coluna]) {
    // Se houver um bloco, chama a função desenharBloco
    // Passa a coluna como x, a linha como y e a cor armazenada na célula do tabuleiro
    desenharBloco(coluna, linha, tabuleiro[linha][coluna]);
}

// Gera uma nova peça aleatória
function gerarNovaPeca() {
    // Gera um índice aleatório entre 0 e o comprimento
    // do array de peças
    const indice = Math.floor(Math.random() * pecas.length);

    // Retorna a forma da peça correspondente ao
    // índice gerado aleatoriamente
    return pecas[indice].shape;
}

// Desenha uma peça
function desenharPeca() {
    // Loop que percorre cada linha da peça atual
```

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P40 - Jogo Tetris
- Left Sidebar (Icons):** EXPLORER, P40 - JOGO TETRIS (index.html, script.js, # styles.css), Find, Open, Find in Files, Outline, Timeline.
- Left Sidebar (Text):** > OUTLINE, > TIMELINE (with a blue notification badge).
- Central Area:** Script.js file open, showing code for drawing a piece and moving it down.
- Bottom Status Bar:** Line 211, Column 1, Spaces: 4, UTF-8, CRLF, JavaScript, Go Live, and a bell icon.

```
index.html    # styles.css  JS script.js  X
JS script.js > desenharPeca
211   for (let linha = 0; linha < pecaAtual.length; linha++) {
212     // Loop que percorre cada coluna da linha atual da peça
213     for (let coluna = 0; coluna < pecaAtual[linha].length; coluna++) {
214       // Verifica se há um bloco na célula atual da peça
215       if (pecaAtual[linha][coluna]) {
216         // Se houver um bloco, chama a função desenharBloco
217         // A posição do bloco é deslocada por pecaX e pecaY para a posição correta no tabuleiro
218         // A cor do bloco é definida como 'blue'
219         desenharBloco(coluna + pecaX, linha + pecaY, 'blue');
220       }
221     }
222   }
223 }
224 }
225 }
226 }
227 }
228
229 // Move a peça para baixo
230 function moverPecaBaixo() {
231
232   // Verifica se a peça colide com outra peça ou com o
233   // fundo do tabuleiro ao tentar se mover para baixo
234   if (verificarColisao(0, 1)) {
235
236     // Se houver uma colisão, fixa a peça na
237     // posição atual no tabuleiro
238     fixarPeca();
239
240   // Gera uma nova peça aleatória para começar a cair
```

A screenshot of a code editor interface, likely Microsoft Visual Studio Code, displaying a script for a Tetris game. The title bar reads "P40 - Jogo Tetris".

The left sidebar shows the project structure under "P40 - JOGO TETRIS":

- index.html
- script.js (selected)
- # styles.css

The main editor area contains the following JavaScript code:

```
241 pecaAtual = gerarNovaPeca();
242
243 // Calcula a posição inicial da nova peça em x (horizontal)
244 // Centraliza a peça no meio do tabuleiro
245 pecaX = Math.floor(colunas / 2) - Math.floor(pecaAtual[0].length / 2);
246
247 // Define a posição inicial da nova peça em
248 // y (vertical) no topo do tabuleiro
249 pecaY = 0;
250
251 // Verifica se a nova peça colide com peças
252 // existentes na posição inicial, indicando fim de jogo
253 if (verificarFimDeJogo()) {
254
255     // Se houver colisão, define gameOver como verdadeiro
256     gameOver = true;
257
258     // Exibe uma mensagem de alerta
259     // informando que o jogo terminou
260     alert('Fim de Jogo!');
261 }
262
263 } else {
264
265     // Se não houver colisão, move a peça para
266     // baixo aumentando a posição y em 1
267     pecaY++;
268
269 }
270 }
```

The status bar at the bottom shows: ⌂ 0 ▲ 0 ⌂ 0 Ln 241, Col 8 Spaces: 4 UTF-8 CRLF {} JavaScript ⌂ Go Live ⌂

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P40 - Jogo Tetris
- Left Sidebar (Icons):** EXPLORER, P40 - JOGO TETRIS (with index.html, script.js, # styles.css), magnifying glass, gear, user profile, outline, timeline.
- Left Sidebar (Text):** > OUTLINE, > TIMELINE (with a blue notification badge).
- Central Area:** Script.js tab selected. The code is a JavaScript function to check if a game is over.

```
index.html styles.css script.js (selected)
JS script.js > moverPecaBaixo
271 }
272
273 // Verifica fim de jogo
274 function verificarFimDeJogo() {
275
276     // Loop que percorre cada coluna da primeira linha do tabuleiro
277     for (let coluna = 0; coluna < colunas; coluna++) {
278
279         // Verifica se há um bloco na célula da
280         // primeira linha e coluna atual
281         // tabuleiro[0][coluna] acessa a célula na
282         // primeira linha (índice 0) e na coluna atual
283         // Se esta célula estiver ocupada (não for 0),
284         // significa que o topo do tabuleiro está cheio
285         if (tabuleiro[0][coluna]) {
286
287             // Se encontrar qualquer célula ocupada na
288             // primeira linha, retorna verdadeiro (true)
289             // Isso indica que o jogo terminou porque não
290             // há espaço para novas peças
291             return true;
292
293         }
294     }
295
296     // Se o loop terminar sem encontrar células ocupadas na
297     // primeira linha, retorna falso (false)
298     // Isso indica que o jogo ainda pode continuar
299     return false;
300 }
```



File Edit Selection View Go Run ·

← →

Q P40 - Jogo Tetris



EXPLORER



JS script.js > verificarFimDeJog

```
301 }
302
303
304 // Limpa linhas completas
305 function limparLinhasCompletas() {
306
307     // Inicializa um contador para o número de linhas eliminadas
308     let linhasEliminadas = 0;
309
310     // Loop que percorre cada linha do tabuleiro,
311     // começando da linha inferior
312     for (let linha = linhas - 1; linha >= 0; linha--) {
313
314         // Verifica se todos os blocos da linha atual
315         // estão preenchidos (não são zero)
316         if (tabuleiro[linha].every(bloco => bloco)) {
317
318             // Remove a linha preenchida do tabuleiro
319             tabuleiro.splice(linha, 1);
320
321             // Insere uma nova linha vazia no topo do tabuleiro
322             // Array(colunas).fill(0) cria um novo array
323             // com 'colunas' elementos, todos preenchidos com 0
324             tabuleiro.unshift(Array(colunas).fill(0));
325
326             // Incrementa o contador de linhas eliminadas
327             linhasEliminadas++;
328
329             // Incrementa 'linha' para verificar
330             // novamente a linha removida
```



✓ P40 - JOGO TETRIS

<> index.htm

JS script.js



> OUTLINE

> TIMELINE

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P40 - Jogo Tetris
- Toolbar:** Minimize, Maximize, Close
- Left Sidebar:** EXPLORER, P40 - JOGO TETRIS (index.html, script.js, # styles.css), SEARCH, PROBLEMS, HIGHLIGHTS, OUTLINE (1), TIMELINE.
- Central Area:** Script Editor showing the script.js file content.

```
// Isso é necessário porque ao remover uma
// linha, as linhas acima se deslocam para baixo
linha++;
}

}

// Atualiza a pontuação do jogador com o
// número de linhas eliminadas
pontuacao += linhasEliminadas;

// Atualiza o elemento HTML com id 'pontuacaoValor' para
// mostrar a pontuação atualizada
document.getElementById('pontuacaoValor').textContent = pontuacao;

// Salva a pontuação atualizada no LocalStorage
salvarPontuacao();

}

}

// Desenha o jogo
function desenhar() {

    // Limpa o canvas, removendo qualquer desenho anterior
    // clearRect(x, y, largura, altura) limpa um retângulo
    // começando em (0, 0) até (canvas.width, canvas.height)
    contexto.clearRect(0, 0, canvas.width, canvas.height);

    // Desenha o tabuleiro no canvas
```

A screenshot of a code editor interface, likely Visual Studio Code, displaying a JavaScript file for a Tetris game. The editor has a dark theme with syntax highlighting for code. The top bar shows the title "P40 - Jogo Tetris".

The left sidebar includes icons for Explorer, Search, Problems, and Outline/Timeline, with "OUTLINE" currently selected.

The Explorer sidebar shows files: "index.html", "# styles.css", "script.js" (selected), and "P40 - JOGO TETRIS" which contains "index.html", "script.js", and "# styles.css".

The main code editor area displays the following JavaScript code:

```
index.html    # styles.css    JS script.js    X
JS script.js > desenhar
361     desenharTabuleiro();
362
363     // Desenha a peça atual no canvas
364     desenharPeca();
365
366 }
367
368 // Loop do jogo
369 function loopJogo() {
370
371     // Move a peça atual para baixo
372     moverPecaBaixo();
373
374     // Desenha o estado atual do jogo (tabuleiro e peça) no canvas
375     desenhar();
376
377     // Verifica se o jogo não acabou
378     if (!gameOver) {
379
380         // Se o jogo não acabou, continua o loop do jogo
381         // setTimeout chama a função loopJogo novamente
382         // após 500 milissegundos (0,5 segundos)
383         intervalo = setTimeout(loopJogo, 500);
384
385     }
386
387 }
388
389 // Inicializa a área de peças disponíveis
390 function inicializarPecasDisponiveis() {
```

The screenshot shows a code editor interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P40 - Jogo Tetris
- Icons:** Explorer, Search, Problems, Editor, Outline, Timeline.
- Editor Area:** The main area displays a script named `script.js`. The code is written in JavaScript and handles the initialization of tetromino pieces.
- Code Preview:** A preview window shows the execution of the code, specifically the part where it initializes pieces.
- Bottom Status Bar:** Shows file count (1), spaces (Spaces: 4), CRLF, and a Go Live button.

```
// Obtém o elemento HTML que irá conter as peças
// disponíveis usando o id 'pecas'
const pecasDiv = document.getElementById('pecas');

// Percorre o array de peças, onde cada peça é
// um objeto contendo sua forma e nome
pecas.forEach(peca, indice) => {

    // Cria um novo elemento <div> para representar
    // cada peça disponível
    const pecaDiv = document.createElement('div');

    // Adiciona a classe 'peca' ao novo elemento <div>
    // Isso permite estilizar todas as peças disponíveis de
    // forma consistente usando CSS
    pecaDiv.classList.add('peca');

    // Adiciona um atributo de dados 'indice' ao
    // <div> para armazenar o índice da peça
    // Isso pode ser útil para identificar qual peça foi clicada
    pecaDiv.dataset.indice = indice;

    // Adiciona um atributo de dados 'shape' ao
    // <div> para armazenar o nome da peça
    // Isso permite identificar o formato da peça
    pecaDiv.dataset.shape = pecas[indice].name;

    // Percorre cada linha da forma da peça
    peca.shape.forEach(linha => {
```

The screenshot shows a code editor interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P40 - Jogo Tetris
- Icons:** Explorer, Search, Problems, Outline, Timeline.
- Project Explorer:** P40 - JOGO TETRIS (index.html, script.js, styles.css).
- Code Editor:** Script.js file open, showing code for initializing Tetris pieces.
- Code Preview:** Shows the call stack for the current function: script.js > inicializarPecasDisponiveis > pecas.forEach() callback > peca.shape.forEach() callback.
- Code Content:** (Lines 421-450) A snippet of JavaScript code for initializing Tetris pieces, defining colors based on block value (1 for blue, 0 for white), and adding click listeners to each piece's div.
- Status Bar:** Line 421, Col 1, Spaces: 4, UTF-8, CRLF, {} JavaScript, Go Live.

```
// Percorre cada bloco (célula) na linha da peça
linha.forEach(bloco => {

    // Cria um novo elemento <div> para
    // representar cada bloco da peça
    const blocoDiv = document.createElement('div');

    // Define a cor de fundo do bloco
    // Se o valor do bloco for 1, a cor
    // será 'blue'; se for 0, a cor será 'white'
    blocoDiv.style.backgroundColor = bloco ? 'blue' : 'white';

    // Adiciona o bloco <div> ao <div> da peça
    pecaDiv.appendChild(blocoDiv);

});

// Adiciona o <div> da peça ao elemento que
// contém todas as peças disponíveis
pecasDiv.appendChild(pecaDiv);

// Adiciona um ouvinte de evento de clique ao <div> da peça
// Quando a peça for clicada, a peça atual no
// jogo será atualizada para a peça clicada
pecaDiv.addEventListener('click', () => {

    // Define a peça atual no jogo para a peça que foi clicada
    pecaAtual = pecas[indice].shape;
```

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P40 - Jogo Tetris
- Icons:** Explorer, Search, Open, Save, Help, Outline, Timeline.
- Left Sidebar (EXPLORER):** P40 - JOGO TETRIS (index.html, script.js, styles.css)
- Current File:** script.js
- Code Preview:** Shows the call stack: script.js > inicializarPecasDisponiveis > pecas.forEach() callback > pecaDiv.addEventListener('click') callback
- Code Content:** The script.js file contains the following code (lines 451 to 480):

```
// Calcula a posição inicial da peça atual em x (horizontal)
// Math.floor(colunas / 2) centraliza a peça na metade do tabuleiro
// - Math.floor(pecaAtual[0].length / 2) ajusta para o centro da peça
pecaX = Math.floor(colunas / 2) - Math.floor(pecaAtual[0].length / 2);

// Define a posição inicial da peça atual em y (vertical) no topo do tabuleiro
pecaY = 0;

};

}

// Verifica colisão da peça com o tabuleiro
function verificarColisao(deslocX, deslocY) {

    // Loop que percorre cada linha da peça atual
    for (let linha = 0; linha < pecaAtual.length; linha++) {

        // Loop que percorre cada coluna da linha atual da peça
        for (let coluna = 0; coluna < pecaAtual[linha].length; coluna++) {

            // Verifica se há um bloco na célula atual da peça
            // Se o valor for 1, significa que essa célula da
            // peça está ocupada por um bloco
            if (pecaAtual[linha][coluna]) {

                // Calcula a nova posição x da célula da peça após o deslocamento
                const novoX = coluna + pecaX + deslocX;
```



The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P40 - Jogo Tetris
- Icons:** Explorer, Search, Open, Find, Help, Outline, Timeline.
- Left Sidebar (EXPLORER):** P40 - JOGO TETRIS (index.html, script.js, # styles.css)
- Active Tab:** script.js
- Code Content:** A snippet of JavaScript code for fixing a piece in a Tetris game tablero. The code iterates through each cell of the current piece and sets its color to 'blue' if it contains a block. It then calls a function to clear completed rows.

```
for (let linha = 0; linha < pecaAtual.length; linha++) {
    for (let coluna = 0; coluna < pecaAtual[linha].length; coluna++) {
        if (pecaAtual[linha][coluna]) {
            tabuleiro[linha + pecaY][coluna + pecaX] = 'blue';
        }
    }
}

// Após fixar a peça no tabuleiro, chama a
// função limparLinhasCompletas
// Isso verifica e limpa qualquer linha
// completa formada pela peça fixada
limparLinhasCompletas();
```
- Bottom Status Bar:** 0 0 0 0, Ln 511, Col 1, Spaces: 4, UTF-8, CRLF, {}, JavaScript, Go Live

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P40 - Jogo Tetris
- Left Sidebar (Icons):** EXPLORER, P40 - JOGO TETRIS (index.html, script.js, # styles.css), Find, Open, Find in Files, Outline, Timeline.
- Left Sidebar (Text):** > OUTLINE, > TIMELINE (with a blue notification badge).
- Central Area:** Script.js tab selected. The code is a JavaScript function for rotating a tetromino piece.

```
// Rotaciona a peça atual
function rotacionarPeca() {
    // Cria um novo array para armazenar a peça rotacionada
    const novaPeca = [];

    // Loop que percorre cada coluna da peça atual
    for (let coluna = 0; coluna < pecaAtual[0].length; coluna++) {
        // Cria uma nova linha para a peça rotacionada
        const novaLinha = [];

        // Loop que percorre cada linha da peça atual, de baixo para cima
        for (let linha = pecaAtual.length - 1; linha >= 0; linha--) {
            // Adiciona o valor da célula atual da peça
            // original à nova linha da peça rotacionada
            novaLinha.push(pecaAtual[linha][coluna]);
        }
        // Adiciona a nova linha à peça rotacionada
        novaPeca.push(novaLinha);
    }
}

// Verifica se a nova posição da peça rotacionada
// colide com algo no tabuleiro
// deslocX e deslocY são ambos 0, então a
// verificação é feita para a posição atual
```

Bottom status bar: < x 0 △ 0 ⌂ 0 Ln 541, Col 2 Spaces: 4 UTF-8 CRLF {} JavaScript Go Live

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P40 - Jogo Tetris
- Icons:** Explorer, Search, Problems, Outline, Timeline.
- Left Sidebar (EXPLORER):** P40 - JOGO TETRIS (index.html, script.js, # styles.css)
- Active Tab:** script.js
- Code Content:** A portion of a JavaScript file for a Tetris game. The code handles keydown events to move the current piece left or right. It includes collision detection and piece rotation logic.

```
index.html    # styles.css  JS script.js  X
JS script.js > rotacionarPeca
571 if (!verificarColisao(0, 0, novaPeca)) {
572     // Se não houver colisão, atualiza a
573     // peça atual para a peça rotacionada
574     pecaAtual = novaPeca;
575
576 }
577
578 }
579
580
581
582 // Adiciona um ouvinte de evento para pressionamentos de tecla
583 document.addEventListener('keydown', function(event) {
584
585     // Verifica se a tecla pressionada é a seta para a esquerda
586     if (event.key === 'ArrowLeft' && !verificarColisao(-1, 0)) {
587
588         // Se a tecla for a seta para a esquerda e não
589         // houver colisão ao mover a peça para a esquerda
590         // Move a peça uma posição para a esquerda, decrementando pecaX
591         pecaX--;
592
593     } else if (event.key === 'ArrowRight' && !verificarColisao(1, 0)) {
594
595         // Verifica se a tecla pressionada é a seta para a direita
596         // Se a tecla for a seta para a direita e não
597         // houver colisão ao mover a peça para a direita
598         // Move a peça uma posição para a direita, incrementando pecaX
599         pecaX++;
600 }
```

Bottom status bar: ⌂ 0 △ 0 ⌂ 0 Ln 571, Col 1 Spaces: 4 UTF-8 CRLF {} JavaScript Go Live

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P40 - Jogo Tetris
- Toolbar:** Standard window control icons.
- Left Sidebar (EXPLORER):**
  - P40 - JOGO TETRIS
    - index.html
    - script.js**
    - # styles.css
- Central Area:** Script.js file content.
- Bottom Status Bar:** Includes icons for file operations, a gear icon with a '1' (indicating one untracked file), and status indicators for rows, columns, and spaces.

```
< > index.html # styles.css JS script.js X
JS script.js > document.addEventListener('keydown') callback
601 } else if (event.key === 'ArrowDown') {
602     // Verifica se a tecla pressionada é a seta para baixo
603     // Se a tecla for a seta para baixo, move a peça
604     // para baixo chamando a função moverPecaBaixo
605     moverPecaBaixo();
606
607 } else if (event.key === 'ArrowUp') {
608     // Verifica se a tecla pressionada é a seta para cima
609     // Se a tecla for a seta para cima, rotaciona a
610     // peça chamando a função rotacionarPeca
611     rotacionarPeca();
612
613 }
614
615
616     // Após mover ou rotacionar a peça, redesenha o
617     // jogo para atualizar a posição da peça no canvas
618     desenhar();
619
620 }
621 );
622
623 // Inicialização do jogo
624
625 // Chama a função para inicializar o tabuleiro do jogo
626 // Isso cria um tabuleiro vazio onde as peças serão posicionadas
627 inicializarTabuleiro();
628
629 // Chama a função para inicializar a pontuação
630 // a partir do LocalStorage
```

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P40 - Jogo Tetris
- Icons:** Explorer, Search, Find, Split, Outline, Timeline, Go Live.
- Left Sidebar:** EXPLORER, P40 - JOGO TETRIS (index.html, script.js, # styles.css).
- Active Tab:** script.js
- Code Content:** (Lines 631 to 654)

```
631 // Isso carrega a pontuação anterior salva ou
632 // define como 0 se não houver pontuação salva
633 inicializarPontuacao();
634
635 // Gera uma nova peça aleatória para começar o jogo
636 pecaAtual = gerarNovaPeca();
637
638 // Calcula a posição inicial da peça atual em x (horizontal)
639 // Math.floor(colunas / 2) centraliza a peça na metade do tabuleiro
640 // - Math.floor(pecaAtual[0].length / 2) ajusta para o centro da peça
641 pecaX = Math.floor(colunas / 2) - Math.floor(pecaAtual[0].length / 2);
642
643 // Define a posição inicial da peça atual
644 // em y (vertical) no topo do tabuleiro
645 pecaY = 0;
646
647 // Chama a função para inicializar a área de peças disponíveis
648 // Isso desenha as peças disponíveis que o
649 // jogador pode escolher à direita do tabuleiro
650 inicializarPecasDisponiveis();
651
652 // Inicia o loop do jogo, que controla o movimento
653 // das peças e a atualização do tabuleiro
654 loopJogo();
```
- Bottom Status Bar:** < 0 △ 0 ⌂ 0 🔍 Ln 631, Col 1 Spaces: 4 UTF-8 CRLF {} JavaScript Go Live

**UC: Projetos  
Assunto:  
Small Projects  
P40 – Jogo Space Invaders**

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Explorer Bar (Left):** Shows project files under "P41 - JOGO SPACE INVADERS": chefe.png, index.html (selected), inimigo.png, nave.png, space-invaders.js, styles.css, and tiro.mp3.
- Code Editor (Main Area):** Displays the content of "index.html".
- Code Editor Content:**

```
1 <!DOCTYPE html>
2 <!-- Especifica que este documento é um HTML5. -->
3
4 <html lang="pt-br">
5 <!-- Inicia o documento HTML e define o idioma
6 como português do Brasil. -->
7
8 <head>
9 <!-- Abre a seção de cabeçalho da página, onde são
10 colocadas informações sobre o documento,
11 como metadados, título e links para CSS. -->
12
13 <meta charset="UTF-8">
14 <!-- Define a codificação de caracteres do
15 documento para UTF-8, que inclui a maioria
16 dos caracteres de idiomas humanos. -->
17
18 <meta name="viewport" content="width=device-width, initial-scale=1.0">
19 <!-- Assegura que a página é responsiva. A largura
20 da viewport é definida como a largura do
21 dispositivo, e a escala inicial é 1.0,
22 significando que a página será exibida em
23 uma escala de 100%. -->
24
25 <title>Space Invaders</title>
26 <!-- Define o título da página, que aparecerá na
27 aba do navegador. -->
28
29 <link rel="stylesheet" href="styles.css">
30 <!-- Inclui um arquivo CSS externo chamado 'styles.css', que
```
- Bottom Status Bar:** Shows file information: Ln 1, Col 1, Spaces: 8, UTF-8, CRLF, HTML, Go Live, and a bell icon.

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Explorer:** Shows files in the project: chefe.png, index.html, inimigo.png, nave.png, space-invaders.js, styles.css, and tiro.mp3.
- Code Editor:** The current file is index.html.
  - Line 31:** </head> | contém os estilos para formatar a página. -->
  - Line 33:** </head>
  - Line 34:** <!-- Fecha a seção de cabeçalho da página. -->
  - Line 36:** <body>
  - Line 37:** <!-- Inicia o corpo do documento HTML, onde todo o conteúdo visível da página é inserido. -->
  - Line 40:** <div id="containerJogo">
  - Line 41:** <!-- Cria uma divisão (div) que serve como um container para todos os elementos do jogo, facilitando a organização e estilização. -->
  - Line 45:** <h1 id="tituloJogo">Space Invaders</h1>
  - Line 46:** <!-- Insere um cabeçalho de primeiro nível (h1) que contém o título do jogo. -->
  - Line 49:** <p id="Pontuacao">Pontuação: 0</p>
  - Line 50:** <!-- Cria um parágrafo para mostrar a pontuação do jogador. Inicialmente, está definido como 0. -->
  - Line 53:** <p id="vidas">Vidas: 3</p>
  - Line 54:** <!-- Cria um parágrafo para mostrar as vidas restantes do jogador, começando com 3. -->
  - Line 57:** <audio id="somTiro" src="tiro.mp3"></audio>
  - Line 58:** <!-- Inclui um elemento de áudio para reproduzir um som de tiro. O arquivo 'tiro.mp3' é especificado como a fonte do som. -->
- Bottom Status Bar:** Shows file information (Ln 31, Col 1), spaces used (Spaces: 8), encoding (UTF-8), line endings (CRLF), file type (HTML), and Go Live button.



EXPLORER

...

## P41 - JOGO SPACE INVADERS

chefe.png

index.html

inimigo.png

nave.png

space-invaders.js

# styles.css

tiro.mp3



&gt; OUTLINE

&gt; TIMELINE

## index.html X

&lt; index.html &gt; html &gt; body &gt; div#containerJogo

```
61      <canvas id="telaJogo" width="800" height="600"></canvas>
62      <!-- Cria um elemento canvas, que é usado para desenhar
63          gráficos via scripting (geralmente JavaScript). Aqui,
64          ele é usado como a área de jogo com largura de 800
65          pixels e altura de 600 pixels. --&gt;
66
67      &lt;/div&gt;
68      <!-- Fecha a div que contém os elementos do jogo. --&gt;
69
70
71      &lt;div id="modal" class="modal"&gt;
72      <!-- Cria uma nova divisão com identificador 'modal' e a
73          classe 'modal'. Esta div será usada como uma janela
74          modal para mensagens de jogo, como game over ou pausa. --&gt;
75
76      &lt;div class="modal-content"&gt;
77      <!-- Define o conteúdo da janela modal, utilizando uma div
78          com a classe 'modal-content' para facilitar a estilização. --&gt;
79
80      &lt;p id="mensagemModal"&gt;&lt;/p&gt;
81      <!-- Cria um parágrafo dentro da modal para exibir
82          mensagens para o jogador. --&gt;
83
84      &lt;button id="botaoContinuar" class="botao"&gt;Continuar&lt;/button&gt;
85      <!-- Adiciona um botão para permitir ao jogador
86          continuar o jogo. Este botão só aparece se a
87          modal permitir continuação. --&gt;
88
89      &lt;button id="botaoTerminar" class="botao"&gt;Terminar&lt;/button&gt;
90      <!-- Adiciona um botão que permite ao jogador</pre>
```



A screenshot of a code editor interface, likely Visual Studio Code, displaying CSS code for a game. The top bar shows the title "P41 - Jogo Space Invaders". The left sidebar has icons for Explorer, Search, Problems, and Outline/Timeline. The Explorer panel shows files like "index.html", "styles.css", and image files "chefe.png", "inimigo.png", "nave.png". The "styles.css" file is selected and shown in the main editor area.

```
index.html # styles.css
# styles.css > body
body {
    margin: 0;
    /* Remove a margem padrão do corpo do documento,
       garantindo que não haverá espaço extra ao
       redor dos elementos dentro dele. */
    padding: 0;
    /* Remove o preenchimento padrão do corpo do documento,
       para que o conteúdo esteja diretamente alinhado
       com as bordas da janela do navegador. */
    display: flex;
    /* Define o modelo de layout para Flexbox, o que permite um
       alinhamento avançado de elementos filhos dentro
       do corpo. */
    justify-content: center;
    /* Centraliza os elementos filhos horizontalmente
       dentro do corpo quando usando Flexbox. */
    align-items: center;
    /* Centraliza os elementos filhos verticalmente dentro do
       corpo quando usando Flexbox. */
    height: 100vh;
    /* Define a altura do corpo para ser igual a 100% da altura
       da viewport (área visível da janela do navegador). */
    background-color: #000;
```

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Explorer:** Shows files and folders for "P41 - JOGO SPACE INVADERS". Files listed include: chefe.png, index.html, inimigo.png, nave.png, space-invaders.js, styles.css, and tiro.mp3.
- Code Editor:** The current file is "styles.css". The code is as follows:

```
/* Define a cor de fundo do corpo para preto. */
body {
    font-family: Arial, sans-serif;
    /* Define a família de fontes para Arial, que é uma fonte
       sem serifa, com uma alternativa genérica 'sans-serif' se
       o Arial não estiver disponível. */

}

#containerJogo {
    text-align: center;
    /* Alinha o texto e, consequentemente, alguns tipos de
       elementos filhos (como botões e cabeçalhos) ao
       centro horizontalmente. */

    position: relative;
    /* Define a posição do contêiner do jogo como relativa,
       permitindo que os elementos filhos posicionados
       absolutamente se posicionem em relação a este contêiner. */

}

#tituloJogo {
    font-size: 24px;
    /* Define o tamanho da fonte do título do jogo para 36 pixels. */

    color: #fff;
    /* Define a cor do texto para branco. */
}
```

Bottom status bar: Line 31, Col 1 | Spaces: 8 | UTF-8 | CRLF | CSS | Go Live

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Explorer:** Shows files in the project: chefe.png, index.html, inimigo.png, nave.png, space-invaders.js, styles.css, and tiro.mp3.
- Editor:** The # styles.css file is open, containing the following CSS code:

```
61 # tituloJogo
62 margin-top: 6em;
63 /* Adiciona uma margem acima do título do jogo para separá-lo
64 | visualmente dos elementos que estão acima dele. */
65
66 margin-bottom: 10px;
67 /* Adiciona uma margem abaixo do título do jogo para separá-lo
68 | visualmente dos elementos que estão abaixo dele. */
69
70 }
71
72
73
74 # pontuacao {
75
76 font-size: 16px;
77 /* Define o tamanho da fonte para 24 pixels,
78 | tornando o texto facilmente legível. */
79
80 color: ■#fff;
81 /* Define a cor do texto para branco, garantindo
82 | contraste com o fundo escuro. */
83
84 margin-bottom: 10px;
85 /* Adiciona uma margem abaixo da pontuação para
86 | separá-la visualmente dos elementos abaixo. */
87
88 }
89
90 # vidas {
```

The code editor has a light theme with syntax highlighting for CSS. The file is saved at Ln 61, Col 1 with Spaces: 8, UTF-8 encoding, CRLF line endings, and CSS selected as the language. A status bar at the bottom shows file counts (0), line numbers (0), and other metrics.

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Explorer:** Shows files in the project: chefe.png, index.html, inimigo.png, nave.png, space-invaders.js, styles.css, and tiro.mp3.
- Editor Area:** Displays the contents of the styles.css file.

```
index.html # styles.css <# styles.css > #vidas
91
92     font-size: 16px;
93     /* Define o tamanho da fonte para 24 pixels,
94      | similar ao elemento de pontuação,
95      | mantendo consistência no design. */
96
97     color: #fff;
98     /* Define a cor do texto para branco, mantendo a
99      | legibilidade contra o fundo escuro. */
100
101 }
102
103 canvas {
104
105     border: 2px solid #fff;
106     /* Adiciona uma borda branca de 2 pixels ao redor do
107      | elemento canvas, destacando-o no fundo preto e
108      | delineando a área de jogo. */
109
110 }
111
112 .modal {
113
114     display: none;
115     /* Inicialmente oculta a janela modal, que só
116      | será exibida mediante ação específica no jogo. */
117
118     position: fixed;
119     /* Fixa a posição da janela modal em relação à viewport,
120      | fazendo com que ela não se move mesmo ao rolar a página. */
```

The code defines styles for a modal window and a canvas element, including font size, color, and border properties. It also includes comments explaining the purpose of each style rule.

A screenshot of a code editor interface, likely Visual Studio Code, displaying CSS code for a modal window. The editor has a dark theme with light-colored syntax highlighting.

The top navigation bar includes: File, Edit, Selection, View, Go, Run, ..., < >, and a search bar containing "P41 - Jogo Space Invaders". The top right corner features window control buttons (minimize, maximize, close) and a tab bar with index.html, # styles.css (which is the active tab), and X.

The left sidebar contains icons for Explorer, Search, Find, Open, and Outline/Timeline, with "OUTLINE" and "TIMELINE" currently selected.

The Explorer panel shows a project structure under "P41 - JOGO SPACE INVADERS": chefe.png, index.html, inimigo.png, nave.png, space-invaders.js, # styles.css (selected), and tiro.mp3.

The main code editor area displays the following CSS code:

```
121 # styles.css > .modal
122     z-index: 1;
123     /* Define a pilha de ordem de camadas da janela modal. Um valor
124      maior traz a modal para frente de outros elementos
125      com z-index menor. */
126
127     left: 0;
128     /* Alinha o lado esquerdo da modal com o lado
129      esquerdo da viewport. */
130
131     top: 0;
132     /* Alinha o topo da modal com o topo da viewport. */
133
134     width: 40%;
135     /* Define a largura da modal para cobrir 100% da largura da viewport. */
136
137     height: 40%;
138     /* Define a altura da modal para cobrir 100% da altura da viewport. */
139
140     overflow: auto;
141     /* Permite a rolagem dentro da modal se o conteúdo
142      exceder a altura ou largura da mesma. */
143
144     background-color: □rgba(0,0,0,0.4);
145     /* Define a cor de fundo da modal para preto com uma
146      opacidade de 40%, criando um efeito de sobreposição
147      que ainda permite ver o fundo. */
148
149     justify-content: center;
150     /* Centraliza horizontalmente o conteúdo da modal usando Flexbox. */
```

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Icons:** Explorer, Search, Find, Open, Save, Outline, Timeline.
- Left Sidebar (EXPLORER):** Shows project files: chefe.png, index.html, inimigo.png, nave.png, space-invaders.js, styles.css, tiro.mp3.
- Active File:** styles.css
- Code Content:** CSS rules for a modal window, starting with .modal and .modal-content.

```
index.html # styles.css <# styles.css > .modal
151
152     align-items: center;
153     /* Centraliza verticalmente o conteúdo da modal usando Flexbox. */
154 }
155
156
157
158 .modal-content {
159
160     background-color: #fff;
161     /* Define a cor de fundo do conteúdo da modal para branco,
162      | garantindo que o texto e os elementos dentro dela
163      | sejam facilmente visíveis. */
164
165     margin: auto;
166     /* Centraliza automaticamente o conteúdo da modal no
167      | contêiner pai, que é útil para manter o conteúdo
168      | alinhado e centrado horizontalmente. */
169
170     padding: 20px;
171     /* Adiciona preenchimento de 20 pixels em todas as
172      | direções dentro do contêiner do conteúdo da modal,
173      | aumentando o espaço entre os bordos internos da
174      | modal e seu conteúdo. */
175
176     border: 1px solid #888;
177     /* Adiciona uma borda sólida com 1 pixel de espessura e
178      | cor cinza (#888) ao redor do conteúdo da modal,
179      | ajudando a definir visualmente a área do
180      | conteúdo da modal. */
```

A screenshot of a code editor interface, likely Visual Studio Code, displaying a CSS file for a modal button. The editor has a dark theme with syntax highlighting. The left sidebar shows project files including images and a JavaScript file. The top bar includes standard file operations like File, Edit, Selection, View, Go, Run, and a search bar.

The current file is `# styles.css`, which contains the following CSS for a button class:

```
181     width: 60%;  
182     /* Define a largura do conteúdo da modal como 80% da  
183      | largura de seu contêiner pai, o que ajuda a  
184      | manter a modal responsivamente adequada a  
185      | diferentes tamanhos de tela. */  
186  
187     text-align: center;  
188     /* Alinha o texto e outros elementos inline ao  
189      | centro do contêiner do conteúdo da modal. */  
190  
191 }  
192  
193 .botao {  
194     margin: 10px;  
195     /* Adiciona uma margem de 10 pixels em todas as direções ao  
196      | redor de cada botão, separando-os visualmente de  
197      | outros elementos. */  
198  
199     padding: 10px 20px;  
200     /* Adiciona um preenchimento dentro do botão com 10 pixels na  
201      | vertical e 20 pixels na horizontal, tornando o botão  
202      | maior e mais fácil de clicar. */  
203  
204     font-size: 16px;  
205     /* Define o tamanho da fonte dentro do botão para 16 pixels, o  
206      | que é adequado para a legibilidade e destaque do  
207      | texto dentro do botão. */  
208  
209 }  
210 }
```

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Icons:** Explorer, Search, Open, Recent, Outline, Timeline.
- Left Sidebar (EXPLORER):** P41 - JOGO SPACE INVADERS (chefe.png, index.html, inimigo.png, nave.png, space-invaders.js, styles.css, tiro.mp3)
- Active Tab:** JS space-invaders.js
- Code Content:** A script for a Space Invaders game in JavaScript. The code uses document.getElementById to find elements like the canvas ('telaJogo'), audio ('somTiro'), modal ('modal'), and message container ('mensagemModal'). It also finds buttons for continuing ('botaoContinuar') and quitting ('botaoTerminar'). The code is annotated with comments in Portuguese explaining its purpose.
- Bottom Status Bar:** 0 0 0 0, Ln 1, Col 1, Spaces: 6, UTF-8, CRLF, {}, JavaScript, Go Live, Bell icon.

```
// Obtém o elemento canvas do documento HTML usando
// seu ID 'telaJogo' e armazena na variável 'telaJogo'.
const telaJogo = document.getElementById('telaJogo');

// Cria um contexto de desenho 2D para a variável 'telaJogo',
// que será usado para desenhar gráficos no canvas.
const contexto = telaJogo.getContext('2d');

// Obtém o elemento de áudio do documento HTML usando
// seu ID 'somTiro' e armazena na variável 'somTiro'.
const somTiro = document.getElementById('somTiro');

// Obtém o contêiner da modal do documento HTML usando
// seu ID 'modal' e armazena na variável 'modal'.
const modal = document.getElementById('modal');

// Obtém o elemento dentro da modal que será usado para
// exibir mensagens, usando seu ID 'mensagemModal' e armazena
// na variável 'mensagemModal'.
const mensagemModal = document.getElementById('mensagemModal');

// Obtém o botão dentro da modal que permite ao usuário
// continuar o jogo, usando seu ID 'botaoContinuar' e
// armazena na variável 'botaoContinuar'.
const botaoContinuar = document.getElementById('botaoContinuar');

// Obtém o botão dentro da modal que permite ao usuário
// terminar o jogo, usando seu ID 'botaoTerminar' e
// armazena na variável 'botaoTerminar'.
const botaoTerminar = document.getElementById('botaoTerminar');
```

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Icons:** Explorer, Search, Find, Refresh, Outline, Timeline, Go Live.
- Left Sidebar (EXPLORER):** P41 - JOGO SPACE INVADERS (chefe.png, index.html, inimigo.png, nave.png, space-invaders.js, styles.css, tiro.mp3)
- Active Tab:** space-invaders.js
- Code Content:** A snippet of JavaScript defining a class 'Objeto' for game objects.

```
31 // Define a classe 'Objeto', que é uma estrutura
32 // base para outros elementos gráficos do jogo.
33 class Objeto {
34
35     // O construtor é um método especial para criar e
36     // inicializar um objeto criado a partir de uma classe.
37     constructor(x, y, largura, altura, cor) {
38
39         // Inicializa a propriedade 'x' do objeto, que
40         // representa a posição horizontal no canvas.
41         this.x = x;
42
43         // Inicializa a propriedade 'y' do objeto, que
44         // representa a posição vertical no canvas.
45         this.y = y;
46
47         // Inicializa a propriedade 'largura' do objeto, que
48         // representa a largura do objeto no canvas.
49         this.largura = largura;
50
51         // Inicializa a propriedade 'altura' do objeto, que
52         // representa a altura do objeto no canvas.
53         this.altura = altura;
54
55         // Inicializa a propriedade 'cor' do objeto, que
56         // define a cor de preenchimento do objeto.
57         this.cor = cor;
58
59     }
60 }
```

The code defines a class 'Objeto' with a constructor that initializes properties x, y, largura, altura, and cor. The code is annotated with comments explaining each step.

File Edit Selection View Go Run ... ← → 🔍 P41 - Jogo Space Invaders

EXPLORER ...

✓ P41 - JOGO SPACE INVADERS

- chefe.png
- index.html
- inimigo.png
- nave.png
- JS space-invaders.js
- # styles.css
- tiro.mp3

JS space-invaders.js > Objeto

```
61 // Define o método 'desenhar' para a classe 'Objeto'.
62 desenhar() {
63
64     // Define a cor de preenchimento para o desenho no
65     // contexto do canvas, usando a propriedade 'cor' do objeto.
66     contexto.fillStyle = this.cor;
67
68     // Desenha um retângulo no canvas nas coordenadas
69     // especificadas (x, y) com as dimensões especificadas (largura, altura).
70     // O método 'fillRect' é usado para desenhar um retângulo preenchido.
71     contexto.fillRect(this.x, this.y, this.largura, this.altura);
72
73 }
74
75 }
76
77
78 // Define a classe 'Jogador' como uma subclasse de 'Objeto',
79 // o que significa que 'Jogador' herda características e
80 // comportamentos da classe 'Objeto'.
81 class Jogador extends Objeto {
82
83     // O construtor é uma função especial que é chamada para
84     // criar um novo objeto da classe. Aqui, ele configura um novo jogador.
85     constructor(x, y, largura, altura, imagem) {
86
87         // 'super' é usado para chamar o construtor da classe
88         // pai, 'Objeto'. Isso é necessário para garantir que o
89         // jogador tenha todas as propriedades de um 'Objeto'.
90         super(x, y, largura, altura);
```

OUTLINE

TIMELINE

0 △ 0 ⌂ 0

Ln 61, Col 3 Spaces: 6 UTF-8 CRLF {} JavaScript Go Live

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Explorer:** Shows files in the project: chefe.png, index.html, inimigo.png, nave.png, space-invaders.js, styles.css, and tiro.mp3.
- Code Editor:** The current file is space-invaders.js. The code defines a constructor for a player object and its drawing method.

```
// 'this.imagem' armazena uma referência para o
// objeto 'Image' que representa a imagem visual do
// jogador. Isso será usado ao desenhar o jogador no canvas.
this.imagem = imagem;

// 'this.velocidade' é usada para controlar a rapidez com
// que o jogador se move no eixo x. Inicializado como 0,
// significa que o jogador está parado.
this.velocidade = 0;

// 'this.vidas' armazena o número de vidas que o jogador tem.
// Ela tenta recuperar esse número do localStorage; se
// não houver nada salvo, o padrão será 3.
// 'parseInt' é usado para garantir que o valor
// recuperado é um número inteiro.
// O segundo argumento, '10', especifica a base decimal para a conversão.
this.vidas = parseInt(localStorage.getItem('vidas'), 10) || 3;

}

// 'desenhar' é um método para exibir o jogador no canvas.
desenhar() {

    // 'contexto.drawImage' é um método do contexto 2D do canvas que desenha uma imagem.
    // 'this.imagem' é a imagem do jogador a ser desenhada.
    // 'this.x' e 'this.y' são as coordenadas no canvas onde a imagem será posicionada.
    // 'this.largura' e 'this.altura' especificam o tamanho da imagem ao ser desenhada.
    contexto.drawImage(this.imagem, this.x, this.y, this.largura, this.altura);
}
```
- Bottom Bar:** Shows status icons (File, Edit, Selection, View, Go, Run, ...), a search icon, and file statistics (Ln 91, Col 3, Spaces: 6, UTF-8, CRLF, JavaScript, Go Live).



The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Explorer:** Shows a project structure under P41 - JOGO SPACE INVADERS:
  - chefe.png
  - index.html
  - inimigo.png
  - nave.png
  - space-invaders.js
  - styles.css
  - tiro.mp3
- Code Editor:** The active file is space-invaders.js. The code implements methods for a player character (Jogador) to move left or right and lose lives.

```
// Define a 'velocidade' do jogador baseado na
// direção passada ao método.
// Se a direção for 'esquerda', 'velocidade' é
// definida como -5, fazendo com que 'x' diminua.
// Se não for 'esquerda' (ou seja, 'direita'),
// 'velocidade' é 5, fazendo com que 'x' aumente.
this.velocidade = direcao === 'esquerda' ? -velocidade : velocidade;

}

// Método 'parar' é usado para parar o movimento do
// jogador ao zerar a 'velocidade'.
parar() {

    // Atribui 0 à 'velocidade' do jogador, efetivamente
    // parando seu movimento lateral.
    this.velocidade = 0;

}

// Método 'perderVida' na classe Jogador. Este método é
// chamado sempre que o jogador é atingido por um
// inimigo ou por um projétil.
perderVida() {

    // Decrementa o número de vidas do jogador em 1.
    this.vidas--;

}

// Atualiza o texto do elemento HTML que mostra
// as vidas restantes do jogador.
```
- Bottom Status Bar:** Shows file information: Ln 151, Col 7, Spaces: 6, UTF-8, CRLF, and a tab for JavaScript.
- Bottom Icons:** File, Save, Undo, Redo, Minimize, Maximize, Close, and a Go Live icon.

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Explorer:** Shows files and folders for "P41 - JOGO SPACE INVADERS".
- Code Editor:** The active tab is "space-invaders.js".
  - Content: JavaScript code for a game.
  - Line 181: `// 'document.getElementById('vidas')' obtém o`  
`// elemento HTML com ID 'vidas'.`  - Line 182: `// '.innerText' é uma propriedade que permite`  
`// modificar o texto dentro de um elemento HTML.`  - Line 183: `// 'Vidas: \${this.vidas}' é uma template string que`  
`// incorpora o valor atual de 'this.vidas'.`  - Line 184: `document.getElementById('vidas').innerText = `Vidas: \${this.vidas}`;
  - Line 185: `// Verifica se o jogador está sem vidas.
  - Line 186: `if (this.vidas === 0) {`
  - Line 187: ` // Se o jogador não tem mais vidas, redefine o`  
` // número de vidas no localStorage para 3.
  - Line 188: ` // 'localStorage.setItem('vidas', 3)' armazena a`  
` // chave 'vidas' com o valor '3' no localStorage do navegador.
  - Line 189: ` // Isso é útil para reiniciar o estado do jogo`  
` // com 3 vidas na próxima vez que o jogo for carregado.
  - Line 190: ` localStorage.setItem('vidas', 3);`
  - Line 191: ` // Exibe uma modal informando ao jogador que`  
` // ele perdeu e que o jogo terminou.
  - Line 192: ` // 'exibirModal' é uma função que cria e mostra`  
` // uma janela modal com uma mensagem.
  - Line 193: ` // 'Você perdeu! Game Over' é a mensagem exibida.
  - Line 194: ` // O segundo argumento, 'false', indica que o botão`  
` // para continuar não deve ser exibido na modal.
  - Line 195: ` exibirModal('Você perdeu! Game Over', false);`
  - Line 196: `}`
- Bottom Bar:** Icons for Outline, Timeline, and various status indicators.

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Icons:** Explorer, Search, Open, Find, Outline, Timeline.
- Left Sidebar (EXPLORER):** P41 - JOGO SPACE INVADERS (chefe.png, index.html, inimigo.png, nave.png, space-invaders.js, styles.css, tiro.mp3)
- Active Tab:** space-invaders.js
- Code Content:** The code defines a 'Bala' class extending 'Objeto'. It includes a constructor that calls the parent's constructor with initial position, size, and color, and sets velocity. An 'atualizar()' method is also defined.

```
index.html styles.css space-invaders.js
JS space-invaders.js > Jogador
211
212 }
213
214 // Define a classe 'Bala', que herda características da classe 'Objeto'.
215 // Essa classe é usada para criar e gerenciar as balas disparadas no jogo.
216 class Bala extends Objeto {
217
218     // O construtor é chamado quando um novo objeto 'Bala' é criado.
219     constructor(x, y, largura, altura, cor, velocidade) {
220
221         // Chama o construtor da classe pai (Objeto) com
222         // as propriedades necessárias.
223         // Essas propriedades definem a posição inicial,
224         // tamanho e cor da bala.
225         super(x, y, largura, altura, cor);
226
227         // 'this.velocidade' armazena a velocidade da bala. Esta
228         // propriedade controla a rapidez com que a
229         // bala se move no canvas.
230         // O valor de 'velocidade' pode ser positivo ou negativo,
231         // dependendo da direção que a bala deve se
232         // mover (para cima ou para baixo).
233         this.velocidade = velocidade;
234
235     }
236
237     // Método 'atualizar' é usado para atualizar o estado da bala.
238     // É chamado a cada frame do jogo para mover a bala na tela.
239     atualizar() {
240 }
```

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Split, ...
- Left Sidebar (EXPLORER):**
  - P41 - JOGO SPACE INVADERS
    - chefe.png
    - index.html
    - inimigo.png
    - nave.png
    - space-invaders.js**
    - styles.css
    - tiro.mp3
- Central Area:** A code editor window with tabs for index.html, styles.css, and space-invaders.js. The space-invaders.js tab is active.
- Code Content (space-invaders.js):**

```
// Incrementa 'this.y' pela 'this.velocidade'. Isso
// move a bala verticalmente no canvas.
// Se 'velocidade' for positiva, a bala move-se para
// baixo; se negativa, para cima.
this.y += this.velocidade;

}

}

// Define a classe 'Inimigo' que herda da classe 'Objeto'.
// Esta classe é utilizada para criar e gerenciar os
// inimigos no jogo, definindo suas
// características e comportamentos.
class Inimigo extends Objeto {

    // O construtor é chamado para criar uma nova instância de 'Inimigo'.
    constructor(x, y, largura, altura, imagem) {

        // Chama o construtor da classe base 'Objeto' para
        // inicializar as propriedades comuns como
        // posição e tamanho.
        super(x, y, largura, altura);

        // 'this.imagem' armazena uma referência ao objeto
        // Image que representa visualmente o inimigo.
        // Esta imagem é usada quando o inimigo é desenhado no canvas.
        this.imagem = imagem;
    }
}
```
- Bottom Status Bar:** Line 241, Col 5, Spaces: 6, UTF-8, CRLF, {}, JavaScript, Go Live, ...

File Edit Selection View Go Run ... ← → 🔍 P41 - Jogo Space Invaders

EXPLORER

✓ P41 - JOGO SPACE INVADERS

- chefe.png
- index.html
- inimigo.png
- nave.png
- JS space-invaders.js
- # styles.css
- tiro.mp3

JS space-invaders.js # styles.css JS space-invaders.js > Inimigo > constructor

```
271 // 'this.chanceTiro' define a probabilidade de o
272 |    // inimigo disparar um tiro em cada atualização de frame.
273 // Um valor muito pequeno como 0.0003 significa que o
274 |    // inimigo tem uma chance baixa, mas constante, de atirar.
275 this.chanceTiro = 0.0003;
276
277 // 'this.direcao' inicializa a direção do movimento do inimigo.
278 // O valor 1 indica que o inimigo se moverá para a
279 |    // direita. Um valor -1 indicaria movimento para a esquerda.
280 // Esta propriedade é usada para controlar a
281 |    // direção horizontal do inimigo no jogo.
282 this.direcao = 1;
283 }
284
285 // Método 'desenhar' para exibir o inimigo no canvas.
286 desenhar() {
287
288     // 'contexto.drawImage' é uma função do contexto 2D
289     // do canvas usada para desenhar uma imagem.
290     // 'this.imagem' é a imagem que será desenhada no canvas.
291     // 'this.x' e 'this.y' são as coordenadas no canvas onde a
292     // imagem do inimigo será posicionada.
293     // 'this.largura' e 'this.altura' definem o tamanho da imagem quando desenhada.
294     contexto.drawImage(this.imagem, this.x, this.y, this.largura, this.altura);
295
296 }
297
298 // Método 'atualizar' que é chamado a cada frame para
299 |    // atualizar a posição e verificar se o inimigo deve atirar.
300 atualizar() {
```

Ln 271, Col 3 Spaces: 6 UTF-8 CRLF {} JavaScript ⚡ Go Live

File Edit Selection View Go Run ... ⏪ ⏩ 🔍 P41 - Jogo Space Invaders

EXPLORER ...

✓ P41 - JOGO SPACE INVADERS

- chefe.png
- index.html
- inimigo.png
- nave.png
- JS space-invaders.js
- # styles.css
- tiro.mp3

JS space-invaders.js > Inimigo > atualizar

```
301 // Incrementa a posição horizontal 'x' do inimigo pela sua direção.  
302 // Se 'this.direcao' for 1, o inimigo se move para a  
303 // direita; se for -1, para a esquerda.  
304 this.x += this.direcao;  
305  
306 // Utiliza 'Math.random()' para gerar um número aleatório entre 0 e 1.  
307 // Se esse número for menor que 'this.chanceTiro', o inimigo atira.  
308 // 'this.chanceTiro' é a probabilidade de o inimigo  
309 // atirar a cada atualização de frame,  
310 // estabelecida como 0.0003.  
311 if (Math.random() < this.chanceTiro) {  
312  
313     // Chama o método 'atirar' se a condição for verdadeira.  
314     this.atirar();  
315  
316 }  
317  
318 }  
319  
320 // Método 'atirar' responsável por criar uma  
321 // nova bala e adicioná-la ao jogo.  
322 atirar() {  
323  
324     // Cria uma nova instância de 'Bala'.  
325     // A posição 'x' da bala é calculada para ser  
326     // disparada do meio do inimigo, ajustando-se  
327     // com 'this.largura / 2 - 2'.  
328     // A posição 'y' é ajustada para começar logo após o  
329     // final do inimigo, usando 'this.y + this.altura'.  
330     // '4' e '10' são a largura e altura da bala, respectivamente.
```

1 > OUTLINE > TIMELINE

✖ 0 △ 0 ⌂ 0 🔍 0 🔍 Ln 301, Col 1 Spaces: 6 UTF-8 CRLF {} JavaScript ⚡ Go Live



The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Explorer:** Shows files in the project: chefe.png, index.html, inimigo.png, nave.png, space-invaders.js, styles.css, and tiro.mp3.
- Code Editor:** The active file is 'space-invaders.js'. The code defines a class 'GradeInimigos' for managing enemy formations. It includes methods for setting up the grid structure and defining enemy classes like 'GradeInimigos' and 'Inimigo'.
- Bottom Status Bar:** Shows file information: Ln 361, Col 4, Spaces: 6, UTF-8, CRLF, and a note about JavaScript.
- Bottom Icons:** Includes icons for selection, outline, timeline, and go live.

```
// 'none' faz com que o botão não seja exibido e
// também não ocupe espaço no layout.
botaoContinuar.style.display = 'none';

}

// Configura a modal para ser exibida.
// 'flex' é um valor de display que ativa o modelo de
// layout flexível para o container, permitindo
// que itens dentro dele sejam alinhados e distribuídos.
modal.style.display = 'flex';

}

// Define a classe 'GradeInimigos', que é responsável
// por criar e gerenciar uma formação de inimigos.
class GradeInimigos {

    // O construtor é chamado para criar uma nova instância de 'GradeInimigos'.
    constructor(linhas, colunas, espacamentoInimigo, larguraInimigo, alturaInimigo) {

        // 'this.linhas' define o número de linhas de inimigos na grade.
        this.linhas = linhas;

        // 'this.colunas' define o número de colunas de inimigos na grade.
        this.colunas = colunas;

        // 'this.espacamentoInimigo' estabelece a distância entre
        // cada inimigo na grade, tanto vertical
        // quanto horizontalmente.
```

File Edit Selection View Go Run ... ← → 🔍 P41 - Jogo Space Invaders

EXPLORER

✓ P41 - JOGO SPACE INVADERS

- chefe.png
- index.html
- inimigo.png
- nave.png
- JS space-invaders.js
- # styles.css
- tiro.mp3

JS space-invaders.js > Gradelnimigos > constructor

```
391     this.espacamentoInimigo = espacamentoInimigo;
392
393     // 'this.larguraInimigo' define a largura de
394     // cada inimigo na grade.
395     this.larguraInimigo = larguraInimigo;
396
397     // 'this.alturaInimigo' define a altura de cada inimigo na grade.
398     this.alturaInimigo = alturaInimigo;
399
400     // 'this.inimigos' é um array para armazenar todos os
401     // inimigos criados. Inicialmente, está vazio.
402     this.inimigos = [];
403
404     // 'this.criarInimigos()' é um método chamado para
405     // popular a grade com inimigos usando os
406     // parâmetros fornecidos.
407     this.criarInimigos();
408
409     // 'this.velocidade' define a velocidade com que a
410     // grade inteira de inimigos se move. Inicialmente é 1,
411     // o que significa um pixel por frame.
412     this.velocidade = 1;
413
414     // 'this.direcao' estabelece a direção inicial do
415     // movimento da grade de inimigos. Um valor
416     // de 1 indica para a direita; -1 para a esquerda.
417     this.direcao = 1;
418
419     // 'this.pontuacao' inicializa a pontuação gerada pela
420     // destruição dos inimigos. Começa em 0 e é
```

Ln 391, Col 1 Spaces: 6 UTF-8 CRLF {} JavaScript ⚡ Go Live

File Edit Selection View Go Run ... ⏪ ⏩ 🔎 P41 - Jogo Space Invaders

EXPLORER

P41 - JOGO SPACE INVADERS

- chefe.png
- index.html
- inimigo.png
- nave.png
- space-invaders.js
- styles.css
- tiro.mp3

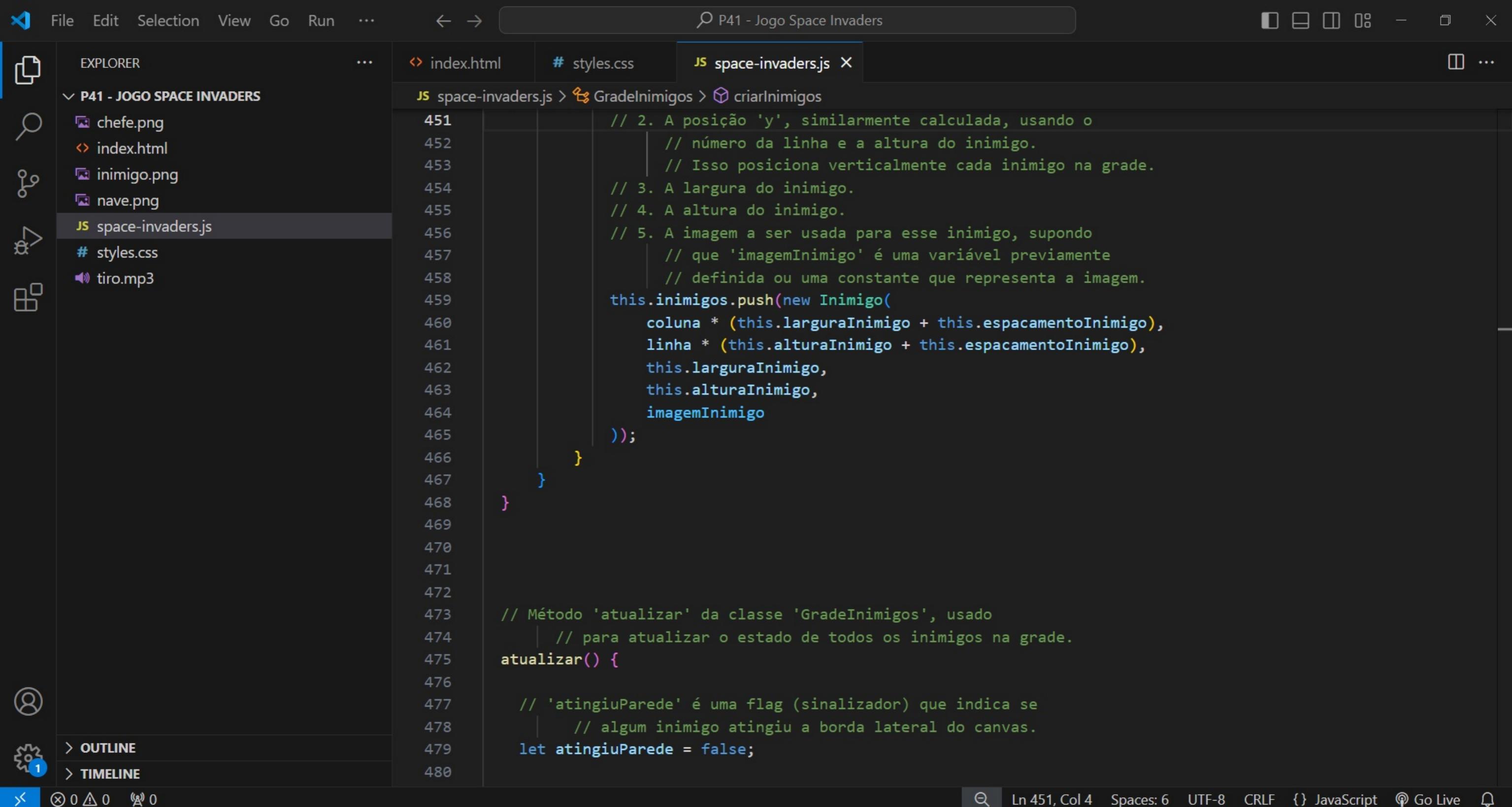
space-invaders.js

```
index.html styles.css space-invaders.js
JS space-invaders.js > Gradelnimigos > constructor
421     // incrementada conforme os inimigos são derrotados.
422     this.pontuacao = 0;
423
424 }
425
426
427 // Método 'criarInimigos' define como os inimigos são
428 // inicialmente criados e posicionados em uma grade.
429 criarInimigos() {
430
431     // Limpa o array 'this.inimigos' para garantir que ele
432     // esteja vazio antes de adicionar novos inimigos.
433     // Isso é útil quando precisamos recriar a grade de
434     // inimigos sem acumular entradas antigas.
435     this.inimigos = [];
436
437     // Laço 'for' que itera sobre cada linha da grade de inimigos.
438     for (let linha = 0; linha < this.linhas; linha++) {
439
440         // Laço 'for' aninhado que itera sobre cada coluna na linha atual.
441         for (let coluna = 0; coluna < this.colunas; coluna++) {
442
443             // Cria uma nova instância da classe 'Inimigo'
444             // para cada posição na grade.
445             // Os parâmetros para o construtor de 'Inimigo' incluem:
446             // 1. A posição 'x', calculada como o número da
447             // coluna multiplicado pela soma da largura do
448             // inimigo e o espaçamento entre inimigos.
449             // Isso assegura que cada inimigo é posicionado
450             // horizontalmente com o espaçamento correto.
```

OUTLINE

TIMELINE

0 ▲ 0 ⌂ 0 🔍 Ln 421, Col 1 Spaces: 6 UTF-8 CRLF {} JavaScript Go Live



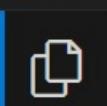
The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Explorer:** Shows files and folders for "P41 - JOGO SPACE INVADERS".
  - Image files: chefe.png, index.html, inimigo.png, nave.png.
  - JavaScript file: space-invaders.js (selected).
  - CSS file: styles.css.
  - Sound file: tiro.mp3.
- Code Editor:** Displays the "space-invaders.js" file content.

```
// 'this.inimigos.forEach' é um método que itera sobre
// cada elemento no array 'this.inimigos'.
// Para cada 'inimigo', executa a função passada como argumento.
this.inimigos.forEach(inimigo => {
    // Chama o método 'atualizar' de cada inimigo, que
    // atualiza a posição do inimigo
    // baseado em sua direção e velocidade.
    inimigo.atualizar();

    // Verifica se o inimigo atingiu a borda esquerda
    // do canvas (inimigo.x <= 0) ou
    // a borda direita do canvas (inimigo.x + inimigo.largura >= telaJogo.width).
    // 'telaJogo.width' é a largura total do canvas
    // onde os inimigos estão sendo desenhados.
    if (inimigo.x <= 0 || inimigo.x + inimigo.largura >= telaJogo.width) {
        // Se qualquer inimigo atingir uma das bordas, a
        // flag 'atingiuParede' é setada como true.
        atingiuParede = true;
    }
});

// Este bloco de código é executado se algum inimigo na
// grade atingiu a parede lateral do canvas, como
// detectado pela flag 'atingiuParede'.
if (atingiuParede) {
```
- Bottom Bar:** Icons for Selection, View, Go, Run, Outline, Timeline, and a status bar showing Ln 481, Col 5, Spaces: 6, UTF-8, CRLF, {} JavaScript, Go Live, and a refresh icon.



EXPLORER



#### ✓ P41 - JOGO SPACE INVADER:



index.html # styles.css space-invaders.js

```
JS space-invaders.js > Gradelnimigos > atualizar
511      // Inverte a direção horizontal de movimento da
512      // grade de inimigos.
513      // Multiplica a direção atual por -1. Se a direção
514      // era 1 (movendo-se para a direita), agora
515      // será -1 (movendo-se para a esquerda), e vice-versa.
516      this.direcao *= -1;
517
518      // 'offsetY' é uma variável que calcula o quanto
519      // todos os inimigos na grade devem se mover
520      // verticalmente para baixo.
521      // É definido como a altura de um inimigo mais o
522      // espaçamento entre os inimigos, garantindo que
523      // eles desçam um espaço claro sem sobreposição.
524      const offsetY = this.alturaInimigo + this.espacamentoInimigo;
525
526      // Itera sobre cada inimigo na grade novamente.
527      this.inimigos.forEach(inimigo => {
528
529          // Atualiza a direção de cada inimigo
530          // individualmente para a nova direção.
531          // Isso assegura que todos os inimigos começarão a
532          // se mover na nova direção horizontal depois
533          // de atingir a parede.
534          inimigo.direcao = this.direcao;
535
536          // Move cada inimigo para baixo adicionando
537          // 'offsetY' à posição 'y' de cada inimigo.
538          // Isso faz com que a grade inteira de inimigos desça
539          // cada vez que qualquer inimigo atinge uma
540          // das bordas laterais do canvas.
```

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Explorer:** Shows files and folders for "P41 - JOGO SPACE INVADERS".
  - Image files: chefe.png, inimigo.png, nave.png.
  - HTML file: index.html.
  - JavaScript file: space-invaders.js (selected).
  - CSS file: styles.css.
  - Sound file: tiro.mp3.
- Code Editor:** The "space-invaders.js" tab is active.

```
index.html    # styles.css    JS space-invaders.js X
JS space-invaders.js > Gradelnimigos > atualizar > inimigos.forEach() callback
541         inimigo.y += offsetY;
542
543     });
544 }
545
546
547 // Itera sobre cada inimigo na lista 'this.inimigos'.
548 this.inimigos.forEach(inimigo => {
549
550     // Primeira condição: verifica se há uma
551     // colisão entre o inimigo e o jogador.
552     // A condição verifica quatro aspectos para determinar
553     // se os retângulos que representam o inimigo e
554     // o jogador se sobrepõem.
555     if (
556
557         // Verifica se o lado direito do inimigo está à
558         // direita do lado esquerdo do jogador.
559         inimigo.x < jogador.x + jogador.largura &&
560
561         // Verifica se o lado esquerdo do inimigo está à
562         // esquerda do lado direito do jogador.
563         inimigo.x + inimigo.largura > jogador.x &&
564
565         // Verifica se a parte inferior do inimigo
566         // está abaixo da parte superior do jogador.
567         inimigo.y + inimigo.altura > jogador.y &&
568
569         // Verifica se a parte superior do inimigo está
570         // acima da parte inferior do jogador.
```
- Bottom Status Bar:** Shows file count (1), outline, timeline, and various system icons.

File Edit Selection View Go Run ... ⏪ ⏩ 🔎 P41 - Jogo Space Invaders

EXPLORER ...

✓ P41 - JOGO SPACE INVADERS

- chefe.png
- index.html
- inimigo.png
- nave.png
- space-invaders.js
- # styles.css
- tiro.mp3

JS space-invaders.js # styles.css index.html

```
JS space-invaders.js > Gradelnimigos > atualizar > inimigos.forEach() callback
    inimigo.y < jogador.y + jogador.altura
)
    // Se todas as condições acima forem verdadeiras,
    // uma colisão foi detectada.
    // Chama o método 'perderVida' do jogador,
    // indicando que o jogador foi atingido.
    jogador.perderVida();

}
// Segunda condição: verifica se algum inimigo
// atingiu a parte inferior do canvas.
// Isso é geralmente usado para indicar que o jogo
// está perdido, ou para fazer o jogador perder uma vida,
// dependendo das regras específicas do jogo.
if (inimigo.y + inimigo.altura >= telaJogo.height) {
    // Se a parte inferior do inimigo atingir
    // ou passar da altura do canvas,
    // chama 'perderVida' no jogador.
    jogador.perderVida();
}
// Itera sobre cada bala na lista 'balasJogador',
// usando um loop forEach que também fornece
// o 'índice' da bala atual.
```

Ln 571, Col 1 Spaces: 6 UTF-8 CRLF {} JavaScript 🔍 Go Live

EXPLORER

...

✓ P41 - JOGO SPACE INVADERS

- chefe.png
- index.html
- inimigo.png
- nave.png
- JS space-invaders.js
- # styles.css
- tiro.mp3

...

OUTLINE

TIMELINE

index.html # styles.css JS space-invaders.js X

JS space-invaders.js > Gradelnimigos > atualizar > inimigos.forEach() callback

```
601     balasJogador.forEach((bala, indice) => {
602
603         // Condição para detectar colisão entre uma bala e um inimigo.
604         // A condição verifica se os retângulos que
605             // representam a bala e o inimigo se sobrepõem.
606         if (
607
608             // Verifica se o lado direito da bala está à
609                 // direita do lado esquerdo do inimigo.
610             bala.x < inimigo.x + inimigo.largura &&
611
612             // Verifica se o lado esquerdo da bala está à
613                 // esquerda do lado direito do inimigo.
614             bala.x + bala.largura > inimigo.x &&
615
616             // Verifica se a parte inferior da bala está
617                 // abaixo da parte superior do inimigo.
618             bala.y < inimigo.y + inimigo.altura &&
619
620             // Verifica se a parte superior da bala está
621                 // acima da parte inferior do inimigo.
622             bala.y + bala.altura > inimigo.y
623
624         ) {
625
626             // Se todas as condições acima forem verdadeiras,
627                 // uma colisão foi detectada.
628             // Remove a bala do array 'balasJogador'
629                 // usando 'splice', que elimina o
630                     // elemento no 'indice' especificado.
```

File Edit Selection View Go Run ... ← → 🔍 P41 - Jogo Space Invaders

EXPLORER

✓ P41 - JOGO SPACE INVADERS

- chefe.png
- index.html
- inimigo.png
- nave.png
- space-invaders.js
- # styles.css
- tiro.mp3

space-invaders.js

# styles.css

space-invaders.js > Gradelnimigos > atualizar > inimigos.forEach() callback > balasJogador.forEach() callback

```
631     balasJogador.splice(indice, 1);

632         // Remove o inimigo do array 'this.inimigos'.
633         // 'this.inimigos.indexOf(inimigo)' encontra o
634             // índice do inimigo que foi atingido para
635                 // que possa ser removido.
636             this.inimigos.splice(this.inimigos.indexOf(inimigo), 1);

637             // Chama o método 'aumentarPontuacao' para
638                 // incrementar a pontuação do jogador
639                     // após acertar um inimigo.
640                     this.aumentarPontuacao();

641             }

642         });

643     }

644     });

645 });

646 }

647 }

648     // Itera sobre cada bala na lista 'balasInimigas',
649         // usando um loop forEach que também fornece o 'indice' da bala atual.
650     balasInimigas.forEach((bala, indice) => {

651         // Condição para detectar colisão entre uma
652             // bala inimiga e o jogador.
653             // A condição verifica se os retângulos que
654                 // representam a bala e o jogador se sobrepõem.
655                 if (
656
657                     // Verifica se o lado direito da bala está à
658                         // direita do lado esquerdo do jogador.
659                         bala.x < jogador.x + jogador.largura &&
```

Ln 631, Col 1 Spaces: 6 UTF-8 CRLF {} JavaScript ⚡ Go Live

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Toolbar:** Back, Forward, Search icon, Minimize, Maximize, Close.
- Left Sidebar (EXPLORER):** Shows project files:
  - P41 - JOGO SPACE INVADERS
    - chefe.png
    - index.html
    - inimigo.png
    - nave.png
  - space-invaders.js
  - # styles.css
  - tiro.mp3
- Code Editor:** The active tab is space-invaders.js. The code implements collision detection between player bullets and enemy ships. It checks if the bullet's left side is to the right of the player and if its bottom is above the player's top. If so, it removes the bullet from the array and calls the player's `perderVida()` method.

```
// Verifica se o lado esquerdo da bala está à
// esquerda do lado direito do jogador.
bala.x + bala.largura > jogador.x &&

// Verifica se a parte inferior da bala está
// abaixo da parte superior do jogador.
bala.y < jogador.y + jogador.altura &&

// Verifica se a parte superior da bala está
// acima da parte inferior do jogador.
bala.y + bala.altura > jogador.y

) {

    // Se todas as condições acima forem verdadeiras,
    // uma colisão foi detectada.
    // Remove a bala do array 'balasInimigas' usando
    // 'splice', que elimina o elemento no 'indice' especificado.
    balasInimigas.splice(indice, 1);

    // Chama o método 'perderVida' do jogador,
    // indicando que o jogador foi atingido pela bala.
    jogador.perderVida();

}

};

});
```
- Bottom Status Bar:** Shows file information: Ln 661, Col 1, Spaces: 6, UTF-8, CRLF, {}, JavaScript, Go Live, and a notification icon.

File Edit Selection View Go Run ... ← → 🔍 P41 - Jogo Space Invaders

EXPLORER ...

✓ P41 - JOGO SPACE INVADERS

- chefe.png
- index.html
- inimigo.png
- nave.png
- JS space-invaders.js
- # styles.css
- tiro.mp3

JS space-invaders.js > Gradelnimigos > atualizar

```
691 // Verifica se a lista de inimigos está vazia e se o
692 // jogador ainda tem vidas restantes.
693 if (this.inimigos.length === 0 && jogador.vidas > 0) {
694
695 // Se não há mais inimigos na lista (todos foram
696 // destruídos) e o jogador ainda possui pelo menos uma vida,
697 // um novo 'chefe' é criado para introduzir um novo desafio ao jogo.
698
699
700 // Cria uma nova instância do 'Chefe'. Os parâmetros
701 // para o construtor incluem:
702 // 1. Posição 'x' inicial do chefe, que é calculada
703 // para ser o centro horizontal do canvas menos 200 pixels.
704 // Isso é feito para centralizar o chefe que tem
705 // uma largura de 400 pixels.
706 // 2. Posição 'y' inicial do chefe, que é definida
707 // como 0, posicionando-o no topo do canvas.
708 // 3. Largura do chefe, definida como 400 pixels, fazendo
709 // com que o chefe seja visualmente maior que os inimigos comuns.
710 // 4. Altura do chefe, definida como 200 pixels,
711 // aumentando sua presença visual.
712 // 5. 'imagemChefe' é a imagem gráfica usada para
713 // representar o chefe visualmente no jogo.
714 chefe = new Chefe(telaJogo.width / 2 - 200, 0, 400, 200, imagemChefe);
715
716 }
717
718 }
719
720 // Método 'desenhar' da classe 'GradeInimigos'.
```

OUTLINE

TIMELINE

0 △ 0 ⌂ 0

Ln 691, Col 1 Spaces: 6 UTF-8 CRLF {} JavaScript Go Live

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Explorer:** Shows files in the project: chefe.png, index.html, inimigo.png, nave.png, space-invaders.js, styles.css, and tiro.mp3.
- Code Editor:** The active file is space-invaders.js. The code is written in Portuguese and handles rendering enemies and updating player score.
- Bottom Status Bar:** Includes icons for file operations, a gear icon with a '1' (likely notifications), and status indicators for rows, columns, and spaces.

```
index.html styles.css space-invaders.js
JS space-invaders.js > Gradelnimigos
721 desenhar() {
722
723     // Itera sobre cada 'inimigo' no array 'this.inimigos'.
724     // Para cada inimigo, chama o método 'desenhar()' que
725         // é definido na classe 'Inimigo'.
726     // Esse método é responsável por renderizar
727         // visualmente o inimigo no canvas.
728     this.inimigos.forEach(inimigo => inimigo.desenhar());
729
730 }
731
732
733 // Método 'aumentarPontuacao' é usado para incrementar a
734 // pontuação do jogador e atualizar a exibição da pontuação.
735 aumentarPontuacao() {
736
737     // Incrementa a pontuação por 1. 'this.pontuacao' mantém a
738     // pontuação total acumulada pelo jogador ao destruir inimigos.
739     this.pontuacao++;
740
741     // Formata a pontuação para o formato numérico local
742     // brasileiro usando 'toLocaleString'.
743     // Isso adiciona separadores de milhar conforme
744     // apropriado, facilitando a leitura dos números.
745     const pontuacaoFormatada = this.pontuacao.toLocaleString('pt-BR');
746
747     // Atualiza o elemento HTML que mostra a pontuação. O elemento é
748     // identificado por seu ID 'pontuacao'.
749     // '.innerText' define o texto dentro desse elemento
750     // para mostrar a pontuação formatada.
```

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Explorer:** Shows files in the project: P41 - JOGO SPACE INVADERS (chefe.png, index.html, inimigo.png, nave.png, space-invaders.js, styles.css, tiro.mp3).
- Code Editor:** The active file is space-invaders.js.
  - Line 751: `document.getElementById('Pontuacao').innerText = 'Pontuação: \${PontuacaoFormatada}';`  
A tooltip for 'PontuacaoFormatada' is shown: JS space-invaders.js > Gradelnimigos > aumentarPontuacao.
  - Line 753: `// Salva a pontuação atual no 'localStorage'. Isso permite`  
`| // que a pontuação seja persistida entre sessões de jogo.`
  - Line 755: `// A pontuação pode ser recuperada quando o jogo é recarregado,`  
`| // permitindo que o jogador continue de onde parou.`
  - Line 757: `localStorage.setItem('Pontuacao', this.Pontuacao);`
  - Line 759: `}`
  - Line 761: `}`
  - Line 763: `
  - Line 764: `// Define a classe 'Chefe', que herda da classe 'Objeto'.`  
`// 'Chefe' representa um inimigo mais forte e desafiador no jogo.`
  - Line 766: `class Chefe extends Objeto {`
  - Line 768: `// Construtor de 'Chefe', chamado ao criar uma nova`  
`| // instância desta classe.`
  - Line 770: `constructor(x, y, largura, altura, imagem) {`
  - Line 772: `// Chama o construtor da classe base 'Objeto' com`  
`| // parâmetros de posição e tamanho.`
  - Line 774: `// Esses parâmetros definem onde e quanto grande o`  
`| // chefe aparecerá no canvas.`
  - Line 776: `super(x, y, largura, altura);`
  - Line 778: `// 'this.imagem' armazena a imagem gráfica que`  
`| // representa o chefe visualmente.`
  - Line 779: `// Esta imagem é usada para desenhar o`
  - Line 780: `//`
- Bottom Bar:** Icons for File, Edit, Selection, View, Go, Run, ...; status bar showing Ln 751, Col 1, Spaces: 6, UTF-8, CRLF, {} JavaScript, Go Live.

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Print, Help, ...
- Left Sidebar (EXPLORER):** Shows project files:
  - P41 - JOGO SPACE INVADERS
    - chefe.png
    - index.html
    - inimigo.png
    - nave.png
    - space-invaders.js
  - # styles.css
  - tiro.mp3
- Code Editor:** The active file is space-invaders.js. The code defines a 'Chefe' class with a constructor and a 'desenhar' method.

```
// 'this.imagem' define a imagem que o chefe vai desenhar.
this.imagem = imagem;

// 'this.vidas' define a quantidade de vidas
// ou saúde que o chefe tem.
// 100 vidas significam que o chefe pode ser
// atingido muitas vezes antes de ser derrotado.
this.vidas = 100;

// 'this.chanceTiro' define a probabilidade de o
// chefe atirar em cada ciclo de atualização do jogo.
// 0.02 indica uma chance de 2% de atirar a cada
// ciclo, o que é relativamente alto comparado a outros inimigos.
this.chanceTiro = 0.02; // Aumentado para atirar mais frequentemente

// 'this.direcao' inicializa a direção horizontal de movimento do chefe.
// Um valor de 1 significa que o chefe começa se movendo para a direita.
this.direcao = 1; // Direção inicial

// 'this.velocidade' define quanto rápido o chefe se move pelo canvas.
// Um valor de 2 significa que o chefe se move 2 pixels
// por ciclo de atualização, o que é relativamente rápido.
this.velocidade = 2; // Velocidade de movimento do chefe

}

// Método 'desenhar' da classe 'Chefe'.
desenhar() {
```
- Bottom Status Bar:** Shows file information: Ln 781, Col 3, Spaces: 6, UTF-8, CRLF, {}, JavaScript, Go Live.

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Icons:** Explorer, Search, Open, Outline, Timeline, and a gear icon with a '1'.
- Left Sidebar (EXPLORER):** Shows project files: chefe.png, index.html, inimigo.png, nave.png, space-invaders.js, styles.css, and tiro.mp3.
- Active Tab:** space-invaders.js
- Code Content:** The code is written in Portuguese and describes a class for a boss enemy in a Space Invaders game. It includes methods for drawing the enemy on the canvas and updating its position based on direction and speed. It also handles boundary checking to prevent the enemy from going off-screen.

```
// Utiliza o contexto do canvas para desenhar a imagem do chefe.  
// 'this.imagem' é a imagem gráfica do chefe.  
// 'this.x' e 'this.y' são as coordenadas no  
// canvas onde a imagem será posicionada.  
// 'this.largura' e 'this.altura' são as dimensões da imagem.  
// Esses valores determinam onde e quanto grande o chefe aparece no jogo.  
contexto.drawImage(this.imagem, this.x, this.y, this.largura, this.altura);  
  
}  
  
// Método 'atualizar' é chamado em cada ciclo de  
// atualização do jogo para alterar o estado do chefe.  
atualizar() {  
  
    // Atualiza a posição horizontal do chefe,  
    // incrementando 'this.x' pela direção multiplicada pela velocidade.  
    // 'this.direcao' pode ser 1 (movendo para a direita) ou -1 (movendo para a esquerda).  
    // 'this.velocidade' é a taxa de deslocamento do  
    // chefe em pixels por ciclo.  
    this.x += this.direcao * this.velocidade;  
  
    // Verifica se o chefe atinge as bordas laterais do canvas.  
    // 'this.x <= 0' verifica se o chefe atinge ou  
    // ultrapassa a borda esquerda do canvas.  
    // 'this.x + this.largura >= telaJogo.width' verifica  
    // se o chefe atinge ou ultrapassa a borda direita do canvas.  
    // 'telaJogo.width' é a largura total do canvas.  
    if (this.x <= 0 || this.x + this.largura >= telaJogo.width) {  
  
        // Se qualquer uma das condições for verdadeira,
```

File Edit Selection View Go Run ... ⏪ ⏩ 🔎 P41 - Jogo Space Invaders

EXPLORER ...

✓ P41 - JOGO SPACE INVADERS

- chefe.png
- index.html
- inimigo.png
- nave.png
- space-invaders.js**
- # styles.css
- tiro.mp3

JS space-invaders.js X

JS space-invaders.js > 📁 Chefe > 🔄 atualizar

```
841     | // inverte a direção do movimento do chefe.
842     | this.direcao *= -1; // Inverte a direção ao bater na borda
843
844 }
845
846 // Gera um número aleatório e verifica se é menor
847 // que 'this.chanceTiro'.
848 // 'this.chanceTiro' é a probabilidade do chefe atirar em
849 // um dado ciclo, definido como 0.02 ou 2%.
850 if (Math.random() < this.chanceTiro) {
851
852     // Se a condição for verdadeira, chama o método 'atirar'.
853     this.atirar();
854
855 }
856
857 }
858
859
860 // Método 'atirar' da classe 'Chefe', usado para que o
861 // chefe dispare projéteis contra o jogador.
862 atirar() {
863
864     // Adiciona um novo objeto 'Bala' ao array 'balasInimigas'.
865     // Esta bala é criada com as seguintes especificações:
866     // 1. Posição x: Calculada para ser disparada do centro
867     // do chefe, ajustado ligeiramente para alinhar
868     // visualmente com sua posição.
869     // 'this.x + this.largura / 2 - 2' calcula o meio do
870     // chefe e ajusta a posição inicial da bala para
```

1 OUTLINE

1 TIMELINE

✖ 0 ▲ 0 ⌂ 0 🔍 0 ⏹ Ln 841, Col 1 Spaces: 6 CRLF {} JavaScript ⏹ Go Live ⏹

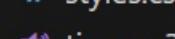
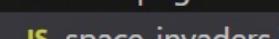
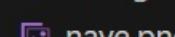
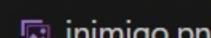
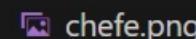




EXPLORER

...

## P41 - JOGO SPACE INVADERS



index.html

# styles.css

JS space-invaders.js X

...

JS space-invaders.js &gt; ⚡ Chefe &gt; 📁 levarDano

```
901     gradeInimigos.pontuacao += 100;  
902  
903     // Formata a pontuação para o formato numérico  
904     // local brasileiro usando 'toLocaleString'.  
905     // Isso adiciona separadores de milhar conforme  
906     // apropriado, facilitando a leitura dos números.  
907     const pontuacaoFormatada = gradeInimigos.pontuacao.toLocaleString('pt-BR');  
908  
909     // Atualiza o elemento HTML que mostra a pontuação.  
910     // O elemento é identificado por seu ID 'pontuacao'.  
911     // '.innerText' define o texto dentro desse elemento  
912     // para mostrar a pontuação formatada.  
913     document.getElementById('pontuacao').innerText = `Pontuação: ${pontuacaoFormatada}`;  
914  
915     // Salva a pontuação atual no 'localStorage'. Isso  
916     // permite que a pontuação seja persistida  
917     // entre sessões de jogo.  
918     // A pontuação pode ser recuperada quando o jogo é  
919     // recarregado, permitindo que o jogador continue de onde parou.  
920     localStorage.setItem('pontuacao', gradeInimigos.pontuacao);  
921  
922     // Salva a quantidade de vidas do jogador no 'localStorage'.  
923     // Isso é feito para persistir o estado do jogo e permitir  
924     // que o jogador continue com o mesmo número de vidas  
925     // se o jogo for recarregado.  
926     localStorage.setItem('vidas', jogador.vidas);  
927  
928     // Recarrega a página, reiniciando o jogo do zero.  
929     // 'document.location.reload()' é um método que  
930     // recarrega a página atual, reiniciando todo o estado do jogo
```



&gt; OUTLINE

&gt; TIMELINE



⊗ 0 △ 0 ⌂ 0



Ln 901, Col 1

Spaces: 6

UTF-8 CRLF



{} JavaScript

⊕ Go Live



File Edit Selection View Go Run ... ← → 🔍 P41 - Jogo Space Invaders

EXPLORER ...

✓ P41 - JOGO SPACE INVADERS

- chefe.png
- index.html
- inimigo.png
- nave.png
- JS space-invaders.js
- # styles.css
- tiro.mp3

JS space-invaders.js # styles.css JS space-invaders.js > 📁 Chefe > 📁 levarDano

```
931 // e a execução do script desde o início.  
932 document.location.reload();  
933  
934 }  
935 }  
936 }  
937 }  
938  
939  
940 // Inicia o processo de carregar imagens para serem usadas no jogo.  
941 // Cada variável do tipo Image será usada para representar  
942 // graficamente os diferentes personagens e objetos no jogo.  
943  
944 // Cria um novo objeto Image para representar o jogador.  
945 const imagemJogador = new Image();  
946  
947 // Define o caminho do arquivo da imagem que  
948 // representa a nave do jogador.  
949 imagemJogador.src = 'nave.png';  
950  
951 // Cria um novo objeto Image para representar  
952 // os inimigos comuns.  
953 const imagemInimigo = new Image();  
954  
955 // Define o caminho do arquivo da imagem que  
956 // representa um inimigo comum.  
957 imagemInimigo.src = 'inimigo.png';  
958  
959 // Cria um novo objeto Image para representar o chefe.  
960 const imagemChefe = new Image();
```

Ln 931, Col 8 Spaces: 6 UTF-8 CRLF {} JavaScript ⚡ Go Live

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Explorer:** Shows files in the project: chefe.png, index.html, inimigo.png, nave.png, space-invaders.js (selected), styles.css, and tiro.mp3.
- Code Editor:** The current file is space-invaders.js, containing the following code:

```
961 // Define o caminho do arquivo da imagem que
962 // será usada para o chefe no jogo.
963 imagemChefe.src = 'chefe.png';
964
965 // Cria um novo objeto 'Jogador', passando parâmetros
966 // que definem sua posição inicial e a imagem a ser usada.
967 // 'telaJogo.width / 2 - 15' posiciona o jogador
968 // horizontalmente no centro do canvas, ajustando
969 // para que o meio da nave fique centrado.
970 // 'telaJogo.height - 30' posiciona o jogador na
971 // parte inferior do canvas.
972 // '30' e '10' são a largura e a altura da nave, respectivamente.
973 const jogador = new Jogador(telaJogo.width / 2 - 15, telaJogo.height - 30, 30, 10, imagemJogador);
974
975 // Inicializa um array vazio para armazenar as
976 // balas disparadas pelo jogador.
977 const balasJogador = [];
978
979 // Cria uma nova 'GradeInimigos', que gerencia
980 // uma coleção de inimigos comuns.
981 // '10', '12' representam o número de linhas e
982 // colunas de inimigos.
983 // '10', '30', '20' são o espaçamento entre os
984 // inimigos, largura e altura de cada inimigo.
985 const gradeInimigos = new GradeInimigos(10, 12, 10, 30, 20);
986
987 // Inicializa um array vazio para armazenar as
988 // balas disparadas pelos inimigos.
989 const balasInimigas = [];
```

The code is written in Portuguese and defines a player object and an enemy grid for a Space Invaders game.

File Edit Selection View Go Run ... ← → 🔍 P41 - Jogo Space Invaders

EXPLORER ...

P41 - JOGO SPACE INVADERS

- chefe.png
- index.html
- inimigo.png
- nave.png
- space-invaders.js
- # styles.css
- tiro.mp3

JS space-invaders.js X

```
991 // Inicializa uma variável 'chefe' como null, que será
992     // usada para armazenar uma instância do chefe
993     // quando ele for criado no jogo.
994
995 let chefe = null;
996
997
998 // Define a função 'loopJogo', que é o ciclo
999 // de atualização e renderização principal do jogo.
1000 function loopJogo() {
1001
1002     // Limpa toda a tela a cada ciclo, preparando o
1003         // canvas para um novo frame de desenho.
1004     // Isso evita sobreposições gráficas dos frames anteriores.
1005     contexto.clearRect(0, 0, telaJogo.width, telaJogo.height);
1006
1007     // Chama o método 'atualizar' do jogador para
1008         // processar movimentos e outras lógicas de atualização.
1009     jogador.atualizar();
1010
1011     // Chama o método 'desenhar' do jogador para
1012         // renderizá-lo na nova posição no canvas.
1013     jogador.desenhar();
1014
1015     // Itera sobre cada 'bala' no array 'balasJogador'
1016         // para atualizar e desenhar cada uma.
1017     balasJogador.forEach((bala, indice) => {
1018
1019         // Atualiza a posição da bala com base em sua velocidade.
1020         bala.atualizar();
```

< > 🔍 Ln 991, Col 1 Spaces: 6 UTF-8 CRLF {} JavaScript ⚡ Go Live

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Icons:** Explorer, Search, Find, Refresh, Outline, Timeline.
- Left Sidebar (EXPLORER):** P41 - JOGO SPACE INVADERS (chefe.png, index.html, inimigo.png, nave.png, space-invaders.js, styles.css, tiro.mp3)
- Active Tab:** space-invaders.js
- Code Content:** The code is part of the `loopJogo` function, specifically the `balasJogador.forEach()` callback.

```
1021 // Verifica se a bala saiu da tela pelo topo (coordenada y negativa).
1022 if (bala.y + bala.altura < 0) {
1023     // Remove a bala do array 'balasJogador' se ela saiu da tela.
1024     // Isso libera recursos e evita processamento desnecessário.
1025     balasJogador.splice(indice, 1);
1026
1027 } else {
1028     // Se a bala ainda estiver visível na tela,
1029     // chama seu método 'desenhar' para renderizá-la.
1030     bala.desenhar();
1031
1032 }
1033
1034
1035
1036 });
1037
1038 // Itera sobre cada 'bala' no array 'balasInimigas',
1039 // atualizando seu estado e renderizando-as.
1040 balasInimigas.forEach((bala, indice) => {
1041
1042     // Atualiza a posição da bala com base em sua velocidade.
1043     bala.atualizar();
1044
1045     // Verifica se a bala ultrapassou o limite inferior do canvas.
1046     if (bala.y > telaJogo.height) {
1047
1048         // Se a bala passou do limite inferior do
1049         // canvas, ela é removida do array.
1050         // Isso evita processamento desnecessário e
```
- Bottom Status Bar:** ⌘ 0 ⌘ 0 ⌘ 0, Ln 1021, Col 1, Spaces: 6, UTF-8, CRLF, {}, JavaScript, Go Live, ⌛

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Toolbar:** Minimize, Maximize, Close
- Left Sidebar (EXPLORER):**
  - P41 - JOGO SPACE INVADERS
    - chefe.png
    - index.html
    - inimigo.png
    - nave.png
    - space-invaders.js**
    - # styles.css
    - tiro.mp3
- Code Editor:** The active tab is "space-invaders.js".

```
index.html    # styles.css    JS space-invaders.js X
JS space-invaders.js > loopJogo > balasInimigas.forEach() callback
1051          // possíveis problemas de desempenho.
1052          balasInimigas.splice(indice, 1);
1053
1054      } else {
1055
1056          // Se a bala ainda estiver dentro dos limites
1057          // do canvas, ela é desenhada na tela.
1058          bala.desenhar();
1059
1060      }
1061  );
1062
1063 // Verifica se ainda existem inimigos vivos na grade de inimigos.
1064 if (gradeInimigos.inimigos.length > 0) {
1065
1066     // Se houver inimigos, atualiza a posição e
1067     // o estado de cada inimigo na grade.
1068     gradeInimigos.atualizar();
1069
1070     // Desenha os inimigos atualizados na tela.
1071     gradeInimigos.desenhar();
1072
1073 }
1074
1075
1076 // Verifica se a variável 'chefe' é verdadeira,
1077 // indicando que um chefe foi criado e está ativo no jogo.
1078 if (chefe) {
1079
1080     // Chama o método 'atualizar' do chefe para
```
- Bottom Status Bar:** x 0 △ 0 ⌂ 0 🔍 Ln 1051, Col 3 Spaces: 6 UTF-8 CRLF {} JavaScript Go Live

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Print, Help, ...
- Left Sidebar (EXPLORER):** Shows project files:
  - P41 - JOGO SPACE INVADERS
    - chefe.png
    - index.html
    - inimigo.png
    - nave.png
    - space-invaders.js
    - styles.css
    - tiro.mp3
- Current File:** space-invaders.js (selected tab)
- Code Content:** The code implements a game loop logic. It processes the movement and actions of the boss enemy (chefe), calling its update and draw methods. It then iterates over each bullet (bala) in the player's bullet array (balasJogador) to check if it has collided with the boss. The collision detection is based on the bullet's right side being within the boss's width range and the bullet's left side being within the boss's width range. Finally, it checks if the bullet's bottom is below the top of the boss.

```
// processar seu movimento e ações, como disparar balas.
chefe.atualizar();

// Chama o método 'desenhar' do chefe para
// renderizá-lo na tela na posição atualizada.
chefe.desenhar();

}

// Itera sobre cada 'bala' no array 'balasJogador' para
// atualizar seu estado e verificar colisões com o chefe.
balasJogador.forEach((bala, indiceBala) => {

    // Verifica se 'chefe' existe e se a 'bala' atual está colidindo com ele.
    // A colisão é verificada por sobreposição de coordenadas.
    if (chefe &&

        // Verifica se o lado direito da bala está à
        // direita do lado esquerdo do chefe.
        bala.x < chefe.x + chefe.largura &&

        // Verifica se o lado esquerdo da bala está à
        // esquerda do lado direito do chefe.
        bala.x + bala.largura > chefe.x &&

        // Verifica se a parte inferior da bala está
        // abaixo da parte superior do chefe.
        bala.y < chefe.y + chefe.altura &&

    ) {
        // Ação de colisão (por exemplo, apagar a bala)
    }
})
```
- Bottom Status Bar:** Shows file information: Ln 1081, Col 7, Spaces: 6, UTF-8, CRLF, {}, JavaScript, Go Live, and a bell icon.

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Explorer:** Shows a project structure under P41 - JOGO SPACE INVADERS:
  - chefe.png
  - index.html
  - inimigo.png
  - nave.png
  - space-invaders.js (selected)
  - styles.css
  - tiro.mp3
- Code Editor:** The active file is space-invaders.js, containing the following code:

```
1111 // Verifica se a parte superior da bala está
1112 // acima da parte inferior do chefe.
1113 bala.y + bala.altura > chefe.y) {
1114
1115     // Se todas as condições de colisão forem
1116     // verdadeiras, o chefe recebe dano.
1117     chefe.levarDano();
1118
1119     // A bala que colidiu é removida do array 'balasJogador'
1120     // para não ser mais processada ou desenhada.
1121     balasJogador.splice(indiceBala, 1);
1122 }
1123
1124
1125
1126     // Itera sobre cada inimigo na lista de
1127     // inimigos mantida por 'gradeInimigos'.
1128     gradeInimigos.inimigos.forEach((inimigo, indiceInimigo) => {
1129
1130         // Verifica se uma bala específica está colidindo com o inimigo atual.
1131         // A colisão é determinada pela sobreposição das
1132         // áreas retangulares de ambos.
1133         if (
1134
1135             // Verifica se o lado direito da bala está à
1136             // direita do lado esquerdo do inimigo.
1137             bala.x < inimigo.x + inimigo.largura &&
1138
1139             // Verifica se o lado esquerdo da bala está à
1140             // esquerda do lado direito do inimigo.
```
- Bottom Bar:** Icons for File, Edit, Selection, View, Go, Run, ...; status bar showing Line 1111, Col 5, Spaces: 6, UTF-8, CRLF; and tabs for {} JavaScript and Go Live.

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Split, ...
- Left Sidebar (EXPLORER):**
  - P41 - JOGO SPACE INVADERS
    - chefe.png
    - index.html
    - inimigo.png
    - nave.png
    - space-invaders.js
  - # styles.css
  - tiro.mp3
- Central Area:** A code editor window titled "space-invaders.js". The code is a JavaScript function for collision detection between a bullet and an enemy. It uses loops to iterate through arrays of bullets and enemies, and conditional statements to check for collisions based on position coordinates.
- Bottom Status Bar:** Includes icons for file operations, a search bar, line number (Ln 1141, Col 1), character count (Spaces: 6), encoding (UTF-8), line endings (CRLF), language (JavaScript), and "Go Live".

```
index.html styles.css space-invaders.js
JS space-invaders.js > JS loopJogo > JS balasJogador.forEach() callback > JS gradeInimigos.inimigos.forEach() callback
bala.x + bala.largura > inimigo.x &&
// Verifica se a parte inferior da bala está
// abaixo da parte superior do inimigo.
bala.y < inimigo.y + inimigo.altura &&
// Verifica se a parte superior da bala está
// acima da parte inferior do inimigo.
bala.y + bala.altura > inimigo.y
)
// Se a bala e o inimigo estão colidindo,
// realiza as seguintes ações:
// Remove o inimigo do array de inimigos, utilizando o
// 'splice' para remover o item no índice detectado.
gradeInimigos.inimigos.splice(indiceInimigo, 1);
// Remove a bala do array de balas do jogador,
// também utilizando 'splice'.
// Isso evita que a mesma bala seja processada mais de
// uma vez ou cause múltiplas colisões.
balasJogador.splice(indiceBala, 1);
// Chama o método 'aumentarPontuacao' da 'gradeInimigos'
// para incrementar a pontuação do jogador.
// Isso recompensa o jogador por acertar um inimigo.
gradeInimigos.aumentarPontuacao();
```

File Edit Selection View Go Run ... ← → 🔍 P41 - Jogo Space Invaders

EXPLORER

✓ P41 - JOGO SPACE INVADERS

- chefe.png
- index.html
- inimigo.png
- nave.png
- space-invaders.js
- # styles.css
- tiro.mp3

JS space-invaders.js X

JS space-invaders.js > loopJogo > balasJogador.forEach() callback > gradelnimigos.inimigos.forEach() callback

```
1171      }
1172      });
1173  });
1174  });
1175
1176 // Itera sobre cada bala no array 'balasInimigas'
1177 // para verificar colisões com o jogador.
1178 balasInimigas.forEach((bala, indice) => {
1179
1180     // Verifica se a bala atual está colidindo com o jogador.
1181     // A colisão é determinada pela sobreposição das
1182     // áreas retangulares de ambos.
1183     if (
1184
1185         // Verifica se o lado direito da bala está à direita
1186         // do lado esquerdo do jogador.
1187         bala.x < jogador.x + jogador.largura &&
1188
1189         // Verifica se o lado esquerdo da bala está à
1190         // esquerda do lado direito do jogador.
1191         bala.x + bala.largura > jogador.x &&
1192
1193         // Verifica se a parte inferior da bala está
1194         // abaixo da parte superior do jogador.
1195         bala.y < jogador.y + jogador.altura &&
1196
1197         // Verifica se a parte superior da bala está
1198         // acima da parte inferior do jogador.
1199         bala.y + bala.altura > jogador.y
1200
```

1 OUTLINE

1 TIMELINE

x ⌂ 0 ⌂ 0 ⌂ 0 ⌂ 0 🔍 Ln 1171, Col 2 Spaces: 6 UTF-8 CRLF {} JavaScript ⌂ Go Live ⌂

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Explorer:** Shows a project folder "P41 - JOGO SPACE INVADERS" containing files: chefe.png, index.html, inimigo.png, nave.png, space-invaders.js (selected), styles.css, and tiro.mp3.
- Code Editor:** The file "space-invaders.js" is open. The code is written in JavaScript and handles collision detection between player bullets and enemy ships.
- Code Content:**

```
1201     ) {
1202
1203         // Se a bala e o jogador estão colidindo,
1204         // realiza as seguintes ações:
1205
1206         // Remove a bala do array 'balasInimigas', utilizando
1207         // 'splice' para remover o item no índice detectado.
1208         balasInimigas.splice(indice, 1);
1209
1210         // Chama o método 'perderVida' do jogador,
1211         // indicando que ele foi atingido por uma bala.
1212         jogador.perderVida();
1213
1214     }
1215 );
1216
1217
1218 // Verifica se o jogador ainda tem vidas restantes.
1219 if (jogador.vidas > 0) {
1220
1221     // Se o jogador ainda possui vidas, o ciclo de jogo continua.
1222     // 'requestAnimationFrame' é usado para agendar a
1223     // próxima chamada do 'loopJogo', mantendo o
1224     // jogo em execução.
1225     requestAnimationFrame(loopJogo);
1226
1227 }
1228
1229
1230
```

- Bottom Bar:** Includes icons for file operations, a gear icon with a '1' (likely notifications), and tabs for OUTLINE and TIMELINE.
- Status Bar:** Shows the current line (Ln 1201, Col 1), spaces (Spaces: 6), encoding (UTF-8), line endings (CRLF), file type (JavaScript), and Go Live.

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Icons:** Explorer, Search, Open, Find, Outline, Timeline.
- Left Sidebar (EXPLORER):** Shows project files:
  - P41 - JOGO SPACE INVADERS
    - chefe.png
    - index.html
    - inimigo.png
    - nave.png
    - space-invaders.js
  - # styles.css
  - tiro.mp3
- Active Editor:** space-invaders.js (Line 1231)
- Code Content:** (Partial view)

```
1231 }
1232
1233 // Adiciona um ouvinte de eventos ao botão 'botaoTerminar' que
1234 // executa uma função quando o botão é clicado.
1235 botaoTerminar.addEventListener('click', () => {
1236
1237     // Salva o valor 3 no localStorage com a chave 'vidas',
1238     // indicando que o jogador começará com 3 vidas na
1239     // próxima vez que o jogo for carregado.
1240     localStorage.setItem('vidas', 3); // Salva a quantidade de vidas no localStorage como 3
1241
1242     // Recarrega a página atual, efetivamente
1243     // reiniciando o jogo do zero.
1244     // 'document.location.reload()' recarrega o
1245     // documento atual, reiniciando todo o estado
1246     // do jogo e a execução do script desde o início.
1247     document.location.reload(); // Reinicia o jogo do zero
1248
1249 });
1250
1251
1252 // Adiciona um ouvinte de eventos ao objeto 'window'
1253 // para escutar eventos de pressionamento de tecla.
1254 window.addEventListener('keydown', (e) => {
1255
1256     // Verifica qual tecla foi pressionada e executa
1257     // ações baseadas na tecla específica.
1258     if (e.key === 'ArrowLeft') {
1259
1260         // Se a tecla pressionada for a seta para a esquerda,
```
- Status Bar:** Line 1231, Col 1, Spaces: 6, UTF-8, CRLF, {} JavaScript, Go Live

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Split, ...
- Left Sidebar (EXPLORER):** Shows project files:
  - P41 - JOGO SPACE INVADERS
    - chefo.png
    - index.html
    - inimigo.png
    - nave.png
  - space-invaders.js
  - # styles.css
  - tiro.mp3
- Code Editor:** The current file is space-invaders.js. The code handles keydown events to move the player ship and fire bullets.

```
index.html    # styles.css  JS space-invaders.js X
JS space-invaders.js > window.addEventListener('keydown') callback
1261          | // chama o método 'mover' do jogador com o argumento 'esquerda'.
1262          | // Isso faz com que o jogador comece a se
1263          |   // mover para a esquerda.
1264          | jogador.mover('esquerda');
1265
1266      } else if (e.key === 'ArrowRight') {
1267
1268          // Se a tecla pressionada for a seta para a direita,
1269          | // chama o método 'mover' do jogador com o argumento 'direita'.
1270          | // Isso faz com que o jogador comece a se mover para a direita.
1271          | jogador.mover('direita');
1272
1273      } else if (e.key === ' ' && balasJogador.length < 3) {
1274
1275          // Se a tecla pressionada for a barra de espaço e
1276          | // o número de balas do jogador em jogo for
1277          |   // menor que 3, permite disparar uma nova bala.
1278          |   // Adiciona uma nova bala ao array 'balasJogador' com
1279          |     // as seguintes propriedades:
1280          |     // - 'x': posição horizontal inicial da bala, ajustada
1281          |       // para sair do centro da nave do jogador.
1282          |     // - 'y': posição vertical inicial da bala, saindo da
1283          |       // frente da nave do jogador.
1284          |     // - 'largura': largura da bala, definida como 4 pixels.
1285          |     // - 'altura': altura da bala, definida como 10 pixels.
1286          |     // - 'cor': cor da bala, definida como 'red' para diferenciar visualmente.
1287          |     // - 'velocidade': velocidade da bala, definida como -4,
1288          |       // indicando que a bala se moverá para cima.
1289          |   balasJogador.push(new Bala(jogador.x + jogador.largura / 2 - 2, jogador.y, 4, 10, 'red', -4));
1290
```

- Bottom Status Bar:** Shows file information: index.html, Line 1261, Column 7, Spaces: 6, UTF-8, CRLF, and a tab for JavaScript.
- Bottom Icons:** Save, Undo, Redo, Find, Go Live, etc.

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Split, ...
- Left Sidebar (EXPLORER):**
  - P41 - JOGO SPACE INVADERS
    - chefe.png
    - index.html
    - inimigo.png
    - nave.png
    - space-invaders.js**
    - styles.css
    - tiro.mp3
- Code Editor:** The active tab is "space-invaders.js".

```
// Adiciona um ouvinte de eventos ao objeto 'window'  
// para escutar eventos de liberação de tecla.  
window.addEventListener('keydown', (e) => {  
    // Verifica se a tecla liberada é a seta para a  
    // esquerda ou a seta para a direita.  
    if (e.key === 'ArrowLeft' || e.key === 'ArrowRight') {  
        // Chama o método 'parar' do jogador, fazendo  
        // com que ele pare de se mover.  
        jogador.parar();  
    }  
});  
  
// Adiciona um ouvinte de eventos ao objeto 'document'  
// para escutar o evento 'DOMContentLoaded'.  
document.addEventListener('DOMContentLoaded', () => {  
    // Inicializa o jogo.  
    iniciarJogo();  
});
```
- Bottom Status Bar:** Line 1291, Col 3, Spaces: 6, UTF-8, CRLF, {} JavaScript, Go Live, ...

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Explorer:** Shows files in the project: chefe.png, index.html, inimigo.png, nave.png, space-invaders.js (selected), styles.css, and tiro.mp3.
- Code Editor:** The current file is space-invaders.js, line 1321 to 1350. The code handles loading saved game data from localStorage and updating UI elements.
- Bottom Status Bar:** Includes icons for selection, search, and navigation, followed by Ln 1321, Col 1, Spaces: 6, UTF-8, CRLF, {}, JavaScript, Go Live, and a refresh icon.

```
1321 document.addEventListener('DOMContentLoaded', () => {
1322
1323     // Obtém a pontuação salva do localStorage.
1324     const pontuacaoSalva = localStorage.getItem('pontuacao');
1325
1326     // Verifica se há uma pontuação salva.
1327     if (pontuacaoSalva !== null) {
1328
1329         // Atualiza o elemento HTML que exibe a
1330         // pontuação com o valor salvo.
1331         document.getElementById('pontuacao').innerText = `Pontuação: ${pontuacaoSalva}`;
1332
1333         // Atualiza a pontuação na 'gradeInimigos' com a pontuação salva.
1334         gradeInimigos.pontuacao = parseInt(pontuacaoSalva, 10);
1335
1336     }
1337
1338     // Obtém as vidas salvas do localStorage, ou define 3
1339     // como padrão se não houver vidas salvas.
1340     const vidasSalvas = parseInt(localStorage.getItem('vidas'), 10) || 3;
1341
1342     // Atualiza o número de vidas do jogador com o valor salvo.
1343     jogador.vidas = vidasSalvas;
1344
1345     // Atualiza o elemento HTML que exibe as vidas com o valor salvo.
1346     document.getElementById('vidas').innerText = `Vidas: ${jogador.vidas}`;
1347
1348 });
1349
1350
```

The screenshot shows a code editor interface with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Jogo Space Invaders
- Explorer:** Shows a folder named "P41 - JOGO SPACE INVADERS" containing files: chefe.png, index.html, inimigo.png, nave.png, space-invaders.js, styles.css, and tiro.mp3.
- Editor Area:** The "space-invaders.js" file is open, showing the following code:

```
1351 // Inicia o ciclo principal do
1352 |    // jogo chamando 'loopJogo'.
1353 loopJogo();
1354
1355
```
- Bottom Status Bar:** Shows file statistics: 0 0 0, Ln 1351, Col 2, Spaces: 6, UTF-8, CRLF, {}, JavaScript, Go Live, and a bell icon.