

# **UC: Projetos**

## **Assunto:**

### **Small Projects – 41 a 48**

**UC:** Projetos

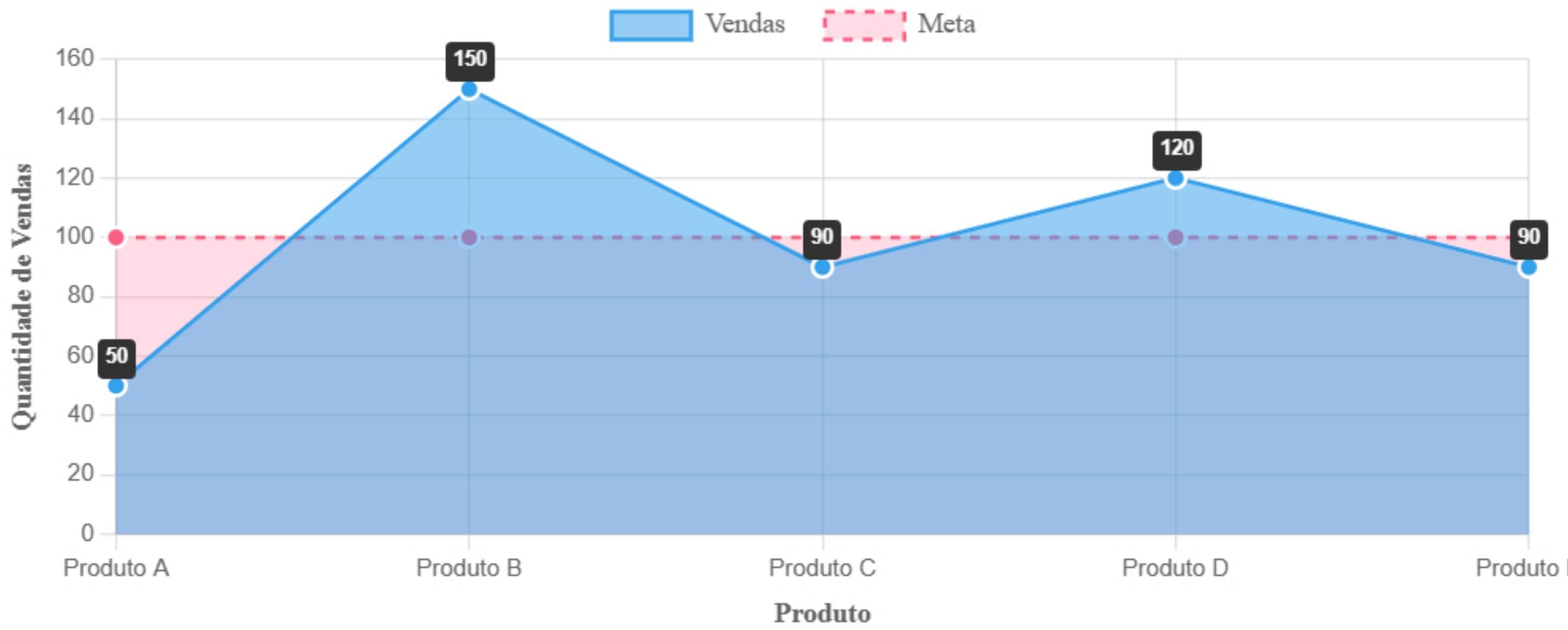
**Assunto:**

Small Projects

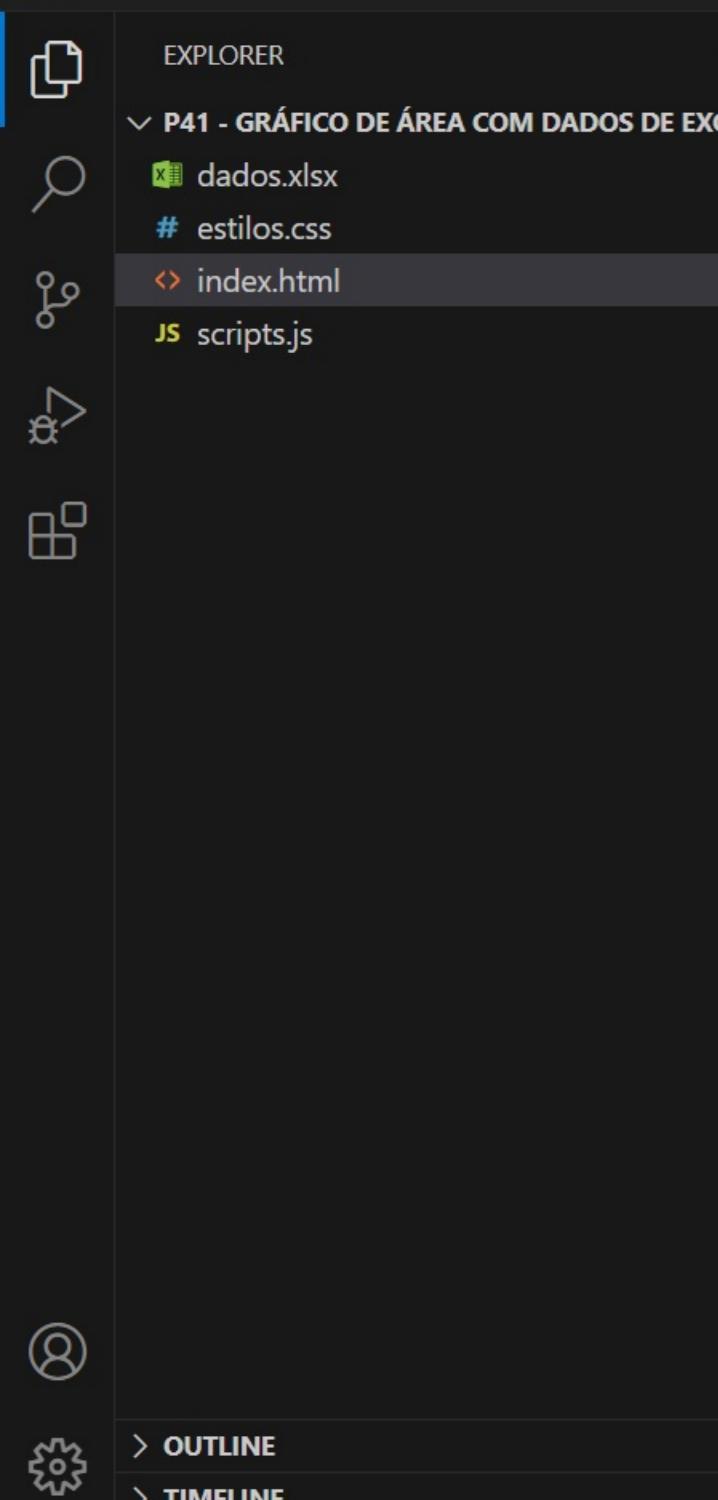
P41 – Gráfico de Área com Dados de Excel

# Gráfico de Áreas com Dados de Excel

Vendas vs Meta por Produto









File Edit Selection View Go Run ... ← → P41 - Gráfico de Área com Dados de Excel

EXPLORER

P41 - GRÁFICO DE ÁREA COM DADOS DE EXCEL

- dados.xlsx
- # estilos.css
- index.html
- scripts.js

index.html

```
91     Os atributos 'width' e 'height' definem as
92         | dimensões do canvas na página. -->
93
94     </div>
95
96     <script src="scripts.js"></script>
97     <!-- A tag &lt;script&gt; com o atributo 'src' apontando para "scripts.js"
98         | inclui um arquivo JavaScript externo que contém o código
99         | necessário para gerar e manipular o gráfico de
100        | áreas no elemento &lt;canvas&gt; acima.
101        Esse script usa a biblioteca Chart.js, já
102        | incluída no cabeçalho, para desenhar e configurar o
103        | gráfico com base nos dados carregados de um
104        | arquivo Excel. --&gt;
105
106    &lt;/body&gt;
107    &lt;/html&gt;
108    <!-- As tags &lt;/body&gt; e &lt;/html&gt; marcam o fim do corpo e do
109        | documento HTML, respectivamente. Todos os scripts e
110        | conteúdo que devem ser renderizados devem estar dentro
111        | do &lt;body&gt;, enquanto o &lt;html&gt; encerra todos os
112        | elementos da página. --&gt;</pre>

OUTLINE



TIMELINE



x 0 △ 0 ⌂ 0



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


```



EXPLORER

✓ P41 - GRÁFICO DE ÁREA COM... [+] [+] ⌂

 dados.xlsx

# estilos.css

↔ index.html

JS scripts.js

```
index.html # estilos.css X
# estilos.css > body
1 body {
2
3     font-family: 'Roboto', sans-serif;
4     /* Define a família de fontes para o corpo do documento.
5      'Roboto' é uma fonte sans-serif limpa e moderna.
6      Caso não esteja disponível, o navegador usará
7      qualquer fonte sans-serif padrão. */
8
9     background-color: #f4f4f4;
10    /* Define a cor de fundo do corpo do documento como um cinza
11       claro (#f4f4f4), que é suave para os olhos e oferece um
12       contraste moderado com texto escuro. */
13
14    color: #333;
15    /* Define a cor padrão do texto em todo o corpo do documento
16       como um cinza escuro (#333), proporcionando uma
17       leitura fácil sem ser tão dura quanto o preto puro. */
18
19    display: flex;
20    /* Ativa o Flexbox, um método de layout que permite o alinhamento
21       fácil e a distribuição espacial de itens dentro
22       de um contêiner. */
23
24    flex-direction: column;
25    /* Define a direção principal do flexbox como coluna, o que
26       significa que os filhos do corpo serão organizados
27       verticalmente. */
28
29    align-items: center;
30    /* Centraliza os itens filho do corpo na direção transversal
```

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Gráfico de Área com Dados de Excel
- Icons:** Explorer, Search, Find, Refresh, Outline, Timeline.
- Left Sidebar (EXPLORER):** Shows project files: dados.xlsx, # estilos.css, index.html, scripts.js.
- Current File:** # estilos.css
- Code Content:** CSS styles for a container element.

```
index.html # estilos.css X
# estilos.css > body
31     margin: 0;
32     /* Remove todas as margens externas do elemento <body>, ajudando a
33      garantir que o layout não seja afetado por espaços
34      indesejados em torno das bordas da página. */
35
36     padding: 20px;
37     /* Adiciona um preenchimento interno de 20px ao redor do
38      conteúdo dentro do <body>, criando espaço entre o
39      conteúdo e as bordas do navegador. */
40
41 }
42
43 .container {
44
45     text-align: center;
46     /* Alinha o texto dentro do contêiner ao centro, o que é
47      útil para títulos, botões e outros elementos que
48      devem ser destacados visualmente no meio. */
49
50     background-color: #fff;
51     /* Define a cor de fundo do contêiner como branco (#fff),
52      garantindo que qualquer conteúdo dentro dele se
53      destaque contra o fundo mais escuro do corpo. */
54
55     padding: 30px;
56     /* Adiciona um preenchimento de 30px ao redor do conteúdo
57      dentro do contêiner, proporcionando espaço entre o
58      conteúdo e as bordas do contêiner. */
59
60 }
```

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Gráfico de Área com Dados de Excel
- Toolbar:** Includes icons for file operations like Open, Save, and Close, along with zoom and refresh symbols.
- Left Sidebar (EXPLORER):** Shows a project structure under "P41 - GRÁFICO DE ÁREA COM DADOS DE EXCEL":
  - dados.xlsx
  - # estilos.css
  - index.html
  - scripts.js
- Central Editor Area:** Displays the content of the "estilos.css" file. The code defines styles for a container element and an h1 element.
- Bottom Status Bar:** Shows file statistics: 0 0 0, Ln 61, Col 1, Spaces: 4, UTF-8, CRLF, CSS, Port : 5500, and a refresh icon.

```
< > P41 - Gráfico de Área com Dados de Excel
File Edit Selection View Go Run ...
< > Ln 61, Col 1 Spaces: 4 UTF-8 CRLF CSS ⚡ Port : 5500

EXPLORER ...
P41 - GRÁFICO DE ÁREA COM DADOS DE EXCEL ...
dados.xlsx
# estilos.css
index.html
scripts.js

<> index.html # estilos.css X
# estilos.css > .container
61 border-radius: 10px;
62 /* Arredonda os cantos do contêiner, dando-lhe um aspecto
63 | | mais suave e moderno com um raio de 10px. */
64 |
65 box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
66 /* Aplica uma sombra sutil ao contêiner, elevando-o visualmente
67 | | da página. A sombra se estende horizontalmente em 0px,
68 | | verticalmente em 2px, e desfoca em 10px com uma
69 | | cor preta translúcida. */
70 |
71 max-width: 800px;
72 /* Define uma largura máxima para o contêiner de 800px, garantindo
73 | | que ele não se torne demasiadamente largo em telas grandes. */
74 |
75 width: 100%;
76 /* Define a largura do contêiner para ocupar 100% da largura
77 | | de seu elemento pai, mas não excederá a largura
78 | | máxima de 800px. */
79 |
80 }
81 
82 h1 {
83 margin-bottom: 20px;
84 /* Define uma margem inferior de 20px para o elemento <h1>. Isso
85 | | cria um espaço entre o título e qualquer conteúdo que o
86 | | siga, ajudando a separar visualmente o título de
87 | | outros elementos na página. */
88 |
89 |
90 
```

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

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Gráfico de Área com Dados de Excel
- Sidebar (EXPLORER):** Shows a project structure under "P41 - GRÁFICO DE ÁREA COM DADOS DE EXCEL":
  - dados.xlsx
  - # estilos.css
  - index.html
  - scripts.js
- Code Editor:** The current file is "estilos.css". The code defines styles for an "h1" element and a "canvas" element.
- Code Content:**

```
index.html # estilos.css
# estilos.css > h1
91     font-size: 24px;
92     /* Define o tamanho da fonte do título para 24px, o que ajuda a
93      garantir que o título seja grande e claro o suficiente
94      para chamar atenção e ser facilmente legível. */
95
96     font-weight: bold;
97     /* Aplica um peso de fonte 'negrito' ao texto do título. Isso faz
98      com que o texto fique mais espesso e mais proeminente,
99      ajudando a destacar o título como uma informação
100     importante na página. */
101
102    color: #333;
103    /* Define a cor do texto do título para um cinza escuro (#333).
104       Essa cor é escura o suficiente para excelente legibilidade
105       sem ser tão dura quanto o preto puro, reduzindo
106       o cansaço visual. */
107
108 }
109
110 canvas {
111
112     margin-top: 20px;
113     /* Adiciona uma margem superior de 20px ao elemento <canvas>.
114       Isso cria um espaço entre o elemento <canvas> e qualquer
115       conteúdo acima dele, como um título ou outro componente
116       visual, assegurando que o gráfico não esteja visualmente
117       apertado contra outros elementos. */
118
119 }
```
- Status Bar:** Shows file information: index.html, Ln 91, Col 3, Spaces: 4, UTF-8, CRLF, CSS, Port: 5500.

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Gráfico de Área com Dados de Excel
- Toolbar:** Includes icons for Explorer, Search, Find, Refresh, and Help.
- Left Sidebar (EXPLORER):** Shows a project structure:
  - P41 - GRÁFICO DE ÁREA COM...
  - dados.xlsx
  - # estilos.css
  - index.html
  - scripts.js** (selected)
- Central Area:** Displays the contents of the selected file, scripts.js:

```
1 document.addEventListener('DOMContentLoaded', function() {  
2     // Este evento é disparado quando todo o conteúdo do  
3     // DOM (Document Object Model) foi completamente carregado,  
4     // incluindo todos os scripts, imagens e estilos. É o ponto  
5     // em que é seguro manipular o DOM porque você tem certeza  
6     // de que todos os elementos estão presentes.  
7  
8     // Inicia um pedido para buscar o arquivo 'dados.xlsx' do servidor.  
9     fetch('dados.xlsx')  
10  
11    // A função 'fetch' retorna uma promessa que resolve  
12    // quando a resposta do servidor é recebida.  
13    .then(response => response.arrayBuffer())  
14  
15    // Converte a resposta, que é um arquivo Excel, em um ArrayBuffer.  
16    // Um ArrayBuffer é uma estrutura de dados genérica que  
17    // representa uma matriz de bytes.  
18    .then(data => {  
19  
20        // Usa a biblioteca XLSX para ler os dados do arquivo Excel.  
21        // O tipo 'array' indica que os dados estão sendo  
22        // lidos de um ArrayBuffer.  
23        var workbook = XLSX.read(data, { type: 'array' });  
24  
25        // 'workbook.SheetNames' é um array que contém os nomes de  
26        // todas as planilhas no arquivo Excel.  
27        // Aqui, selecionamos o nome da primeira planilha.  
28        var nomeDaPlanilha = workbook.SheetNames[0];  
29  
30        // 'workbook.Sheets[nomeDaPlanilha]' acessa os dados da
```
- Bottom Status Bar:** Includes icons for file status (0), search (0), and other system information like Ln 1, Col 1, Spaces: 8, UTF-8, CRLF, {}, JavaScript, Port: 5500, 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 - Gráfico de Área com Dados de Excel
- Explorer:** Shows a project structure under "P41 - GRÁFICO DE ÁREA COM DADOS DE EXCEL" containing files: dados.xlsx, # estilos.css, index.html, and scripts.js.
- Code Editor:** The "scripts.js" file is open, showing the following code:

```
// planilha pelo seu nome, retornando um objeto
// que representa a planilha.
var planilha = workbook.Sheets[nomeDaPlanilha];

// Converte os dados da planilha, que estão no formato
// específico de Excel, para um array de objetos JSON.
// Cada objeto representa uma linha da planilha, e as
// propriedades do objeto correspondem às colunas.
var dadosJson = XLSX.utils.sheet_to_json(planilha);

// Chama a função 'atualizarGraficoAreas' passando os
// dados JSON. Esta função é definida em
// outro lugar do código JavaScript.
// Ela é responsável por atualizar o gráfico na página
// usando os dados extraídos do arquivo Excel.
atualizarGraficoAreas(dadosJson);

}

.catch(error => {
    // Captura e registra qualquer erro que ocorra durante o
    // processo de fetch ou processamento do arquivo.
    console.error('Erro ao carregar o arquivo Excel:', error);
});

};

}

function atualizarGraficoAreas(dados) {
    // Esta função é responsável por atualizar ou criar um
}
```

The code uses XLSX.js to read an Excel file and update a chart based on the data.

The screenshot shows a browser window with a code editor interface. The title bar reads "P41 - Gráfico de Área com Dados de Excel". The left sidebar contains icons for Explorer, Search, Find, and Outline/Timeline. The Explorer section shows a project folder "P41 - GRÁFICO DE ÁREA COM DADOS DE EXCEL" containing files: "dados.xlsx", "# estilos.css", "index.html", and "scripts.js". The "scripts.js" file is currently selected and open in the main editor area.

```
// gráfico de áreas com dados específicos.
// 'dados' é um array de objetos, onde cada objeto contém
// informações como Produto, Vendas e Meta.

// Obtém o contexto de renderização 2D do elemento
// canvas com ID 'graficoAreas'.
// Esse contexto fornece as funções necessárias para
// desenhar no canvas.
var contextoAreas = document.getElementById('graficoAreas').getContext('2d');

// Extrai os nomes dos produtos de cada item do array de
// dados e armazena no array 'produtos'.
// Isso é feito usando o método 'map', que cria um novo
// array com os resultados da chamada de uma função
// para cada elemento do array original.
var produtos = dados.map(item => item['Produto']);

// Extrai os valores de vendas de cada item do array de
// dados e armazena no array 'vendas'.
var vendas = dados.map(item => item['Vendas']);

// Extrai os valores de metas de cada item do array de
// dados e armazena no array 'metas'.
var metas = dados.map(item => item['Meta']);

// Verifica se já existe um gráfico criado e armazenado
// em 'window.meuGraficoAreas'.
// Se existir, o gráfico anterior é destruído antes
// de criar um novo.
// Isso é necessário para evitar sobreposições de
```

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Gráfico de Área com Dados de Excel
- Toolbar:** Includes icons for Explorer, Search, Open, Save, and Help.
- Left Sidebar (EXPLORER):** Shows a tree view with the following items:
  - P41 - GRÁFICO DE ÁREA COM DADOS DE EXCEL
    - dados.xlsx
    - # estilos.css
    - index.html
    - scripts.js** (selected)
- Code Editor:** The selected file is "scripts.js". The code is as follows:

```
// gráficos antigos com novos dados ou duplicação
// de dados visuais.
if (window.meuGraficoAreas) {
    window.meuGraficoAreas.destroy();
}

window.meuGraficoAreas = new Chart(contextoAreas, {
    // Inicializa uma nova instância de um gráfico no
    // contexto 2D do elemento canvas.

    type: 'line',
    // Define o tipo de gráfico como 'line', que é um
    // gráfico de linhas. Gráficos de linhas são
    // úteis para mostrar tendências ao longo do tempo.

    data: {

        labels: produtos,
        // 'labels' define os rótulos do eixo x do gráfico.
        // Aqui, são os nomes dos produtos extraídos dos dados.

        datasets: [
            // 'datasets' contém um array de objetos que definem os
            // conjuntos de dados a serem representados no gráfico.

            {
                label: 'Vendas',
                // 'label' é a legenda para o conjunto de dados,
                // que neste caso representa as vendas.
            }
        ]
    }
})
```

**Status Bar:** Shows file number 0, line 91, column 7, spaces: 8, encoding: UTF-8, CRLF, and port: 5500.

EXPLORER

P41 - GRÁFICO DE ÁREA COM DADOS DE EXCEL

- dados.xlsx
- # estilos.css
- index.html
- scripts.js

> OUTLINE

> TIMELINE

index.html # estilos.css JS scripts.js X

scripts.js > atualizarGraficoAreas > data > datasets

```
121
122     data: vendas,
123         // 'data' contém os valores numéricos para o gráfico,
124             // que neste caso são os valores de vendas dos produtos.
125
126     backgroundColor: 'rgba(54, 162, 235, 0.5)',
127         // 'backgroundColor' define a cor de fundo para a área
128             // sob a linha do gráfico, com opacidade de 50%.
129
130     borderColor: 'rgba(54, 162, 235, 1)',
131         // 'borderColor' define a cor da linha do gráfico.
132
133     borderWidth: 2,
134         // 'borderWidth' define a espessura da linha do gráfico.
135
136     fill: true,
137         // 'fill' especifica se a área sob a linha deve ser
138             // preenchida, o que é verdadeiro neste caso.
139
140     pointBackgroundColor: 'rgba(54, 162, 235, 1)',
141         // 'pointBackgroundColor' define a cor de fundo dos pontos no gráfico.
142
143     pointBorderColor: '#fff',
144         // 'pointBorderColor' define a cor da borda dos pontos no gráfico.
145
146     pointBorderWidth: 2,
147         // 'pointBorderWidth' define a espessura da borda dos pontos no gráfico.
148
149     pointRadius: 5,
150         // 'pointRadius' define o raio dos pontos no gráfico.
```

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Toolbar:** Back, Forward, Search icon, P41 - Gráfico de Área com Dados de Excel title bar, and window control buttons.
- Left Sidebar (EXPLORER):** Shows a tree view with the following items:
  - P41 - GRÁFICO DE ÁREA COM DADOS DE EXCEL
    - dados.xlsx
    - # estilos.css
    - index.html
    - scripts.js** (selected)
- Code Editor:** The selected file is "scripts.js". The code defines configuration options for a chart's data labels:

```
151 pointHoverRadius: 7,  
152     // 'pointHoverRadius' define o raio dos pontos no gráfico  
153     // quando o mouse passa sobre eles.  
154  
155  
156     dataLabels: {  
157         // 'dataLabels' é uma configuração do plugin 'chartjs-plugin-datalabels'  
158         // que permite adicionar rótulos de dados aos pontos do gráfico.  
159  
160         display: true,  
161         // 'display' habilita a exibição dos rótulos.  
162  
163         align: 'top',  
164         // 'align' define a posição dos rótulos em relação aos  
165         // pontos, neste caso, acima dos pontos.  
166  
167         backgroundColor: '#333',  
168         // 'backgroundColor' define a cor de fundo dos rótulos de dados.  
169  
170         borderRadius: 3,  
171         // 'borderRadius' define o raio da borda dos rótulos de dados.  
172  
173         color: 'white',  
174         // 'color' define a cor do texto dos rótulos de dados.  
175  
176         font: {  
177             size: 10,  
178             weight: 'bold'  
179         },  
180         // 'font' define as propriedades da fonte usada nos rótulos de dados.
```
- Bottom Status Bar:** Includes icons for file operations (New, Open, Save, etc.), status indicators (0 0 0 0), and a search bar.

EXPLORER

...

✓ P41 - GRÁFICO DE ÁREA COM DADOS DE EXCEL

- dados.xlsx
- # estilos.css
- index.html
- scripts.js

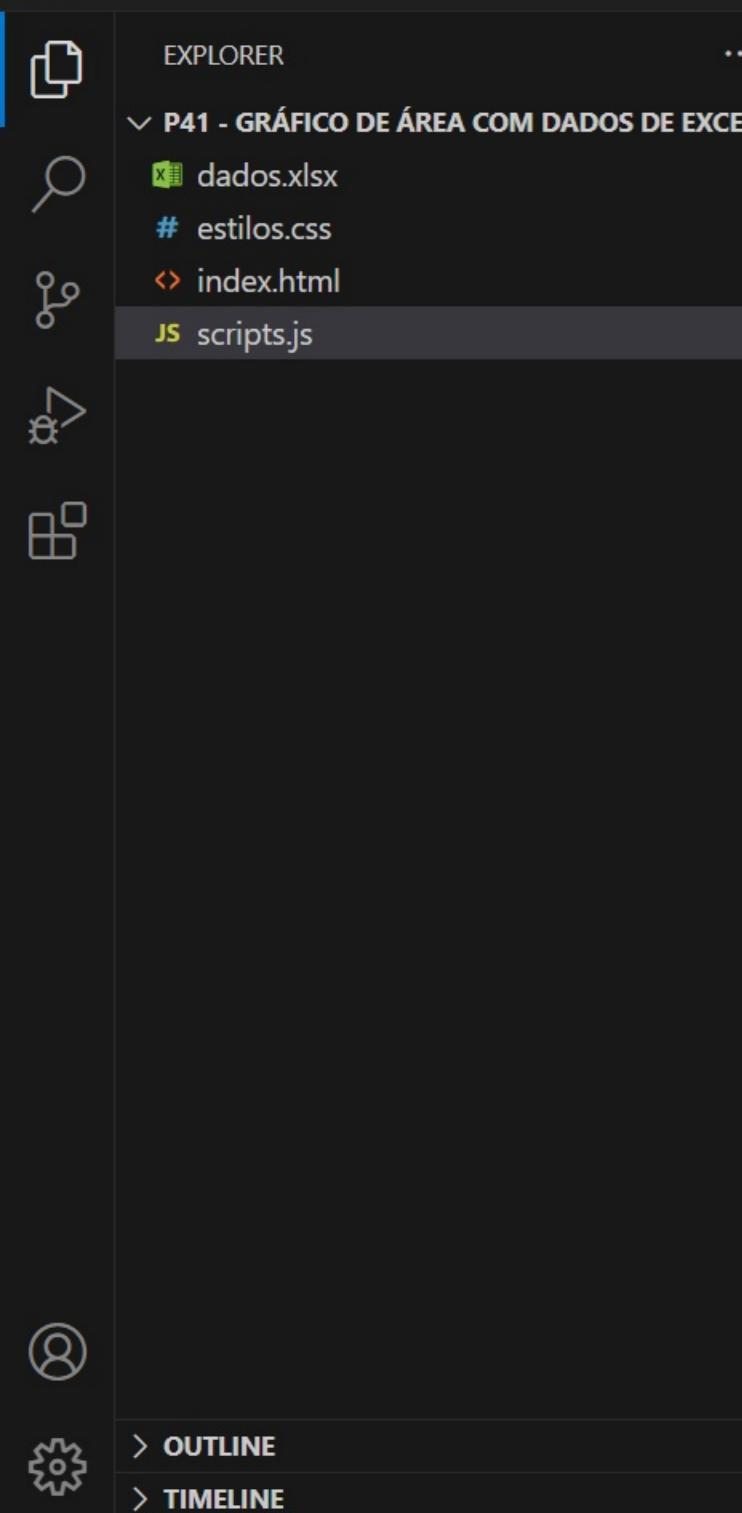
> OUTLINE

> TIMELINE

index.html # estilos.css JS scripts.js X

JS scripts.js > atualizarGraficoAreas > data > datasets > datalabels

```
181     formatter: function(value) {  
182         // 'formatter' é uma função que permite formatar o  
183         // conteúdo do rótulo de dados.  
184         return value; // Retorna o valor como está.  
185     }  
186 },  
187 {  
188     label: 'Meta',  
189     // 'label' é a legenda para este conjunto de dados, que  
190     // neste caso representa as metas estabelecidas.  
191     data: metas,  
192     // 'data' contém os valores numéricos para o gráfico, que  
193     // neste caso são as metas de vendas para cada produto.  
194     backgroundColor: 'rgba(255, 99, 132, 0.2)',  
195     // 'backgroundColor' define a cor de fundo para a área sob a  
196     // linha do gráfico, usando uma cor rosa com opacidade de 20%.  
197     borderColor: 'rgba(255, 99, 132, 1)',  
198     // 'borderColor' define a cor da linha do gráfico, usando uma  
199     // cor rosa mais intensa sem opacidade.  
200     borderWidth: 2,  
201     // 'borderWidth' define a espessura da linha do gráfico.
```



```
index.html          # estilos.css      JS scripts.js X
JS scripts.js > ⚙ atualizarGraficoAreas > ⚙ data > ⚙ datasets
211               borderDash: [5, 5],
212               // 'borderDash' configura a linha para ser tracejada, com
213               // segmentos de linha e espaços de 5 pixels alternadamente.
214
215               fill: true,
216               // 'fill' especifica se a área sob a linha deve ser preenchida, o
217               // que é verdadeiro neste caso.
218
219               pointBackgroundColor: 'rgba(255, 99, 132, 1)',
220               // 'pointBackgroundColor' define a cor de fundo dos pontos no
221               // gráfico, correspondente à cor da linha.
222
223               pointBorderColor: '#fff',
224               // 'pointBorderColor' define a cor da borda dos pontos no
225               // gráfico, que neste caso é branca.
226
227               pointBorderWidth: 2,
228               // 'pointBorderWidth' define a espessura da borda dos
229               // pontos no gráfico.
230
231               pointRadius: 5,
232               // 'pointRadius' define o raio dos pontos no gráfico.
233
234               pointHoverRadius: 7,
235               // 'pointHoverRadius' define o raio dos pontos no gráfico
236               // quando o mouse passa sobre eles, fazendo com que
237               // eles se destaquem ao serem focados.
238
239               datalabels: {
```

File Edit Selection View Go Run ... ← → P41 - Gráfico de Área com Dados de Excel

EXPLORER ...

P41 - GRÁFICO DE ÁREA COM DADOS DE EXCEL

- dados.xlsx
- # estilos.css
- index.html
- scripts.js

scripts.js > atualizarGraficoAreas > data > datasets > datalabels

```
241     display: false
242         // 'datalabels' define as configurações do plugin 'chartjs-plugin-datalabels'
243         // para este conjunto de dados.
244         // 'display: false' desativa a exibição de rótulos de
245         // dados para este conjunto de dados,
246         // o que pode ser útil se quisermos manter o gráfico
247         // visualmente mais limpo ou se os rótulos não adicionarem
248         // informações essenciais.
249
250     }
251 }
252 ],
253 },
254
255 options: {
256     // 'options' é um objeto que contém configurações globais para o
257     // gráfico, afetando seu comportamento e aparência.
258
259     plugins: {
260         // 'plugins' define configurações para plugins
261         // específicos usados no gráfico.
262
263         title: {
264             // 'title' configura o título do gráfico.
265
266             display: true,
267             // 'display: true' ativa a exibição do título no gráfico.
268
269             text: 'Vendas vs Meta por Produto',
270             // 'text' define o conteúdo do texto do título, que neste
```

...

Ln 241, Col 7 Spaces: 8 UTF-8 CRLF {} JavaScript ⚡ Port : 5500

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

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Gráfico de Área com Dados de Excel
- Icons:** Explorer, Search, Find, Refresh, Outline, Timeline.
- Active File:** scripts.js
- Code Content:** The code defines a title object for a chart. It includes properties for font (size 20, Roboto bold, #333 color), padding (top 10), and a note about comparing 'Vendas' and 'Meta' by product.

```
// caso compara "Vendas" e "Meta" por produto.

font: {
    // 'font' define as propriedades da fonte do título.
    size: 20,
    // 'size: 20' configura o tamanho da fonte para 20 pixels,
    // tornando o título claramente visível.

    family: 'Roboto',
    // 'family: "Roboto"' especifica a família de fontes Roboto,
    // que é uma fonte sans-serif moderna e legível.

    weight: 'bold',
    // 'weight: "bold"' aplica um peso de fonte negrito ao título,
    // destacando-o visualmente na visualização do gráfico.

    color: '#333'
    // 'color: "#333"' define a cor do texto do título para um
    // cinza escuro, proporcionando contraste adequado com o
    // fundo enquanto mantém a legibilidade.

},
padding: {
    // 'padding' define o espaçamento em torno do
    // título dentro do espaço de título.

    top: 10,
    // 'top: 10' adiciona um espaçamento de 10
```

File Edit Selection View Go Run ... ← → P41 - Gráfico de Área com Dados de Excel

EXPLORER

P41 - GRÁFICO DE ÁREA COM DADOS DE EXCEL

- dados.xlsx
- # estilos.css
- index.html
- scripts.js

scripts.js > atualizarGraficoAreas > options > plugins > title > padding

```
301 // pixels acima do título.  
302  
303 bottom: 30  
304 // 'bottom: 30' adiciona um espaçamento de 30 pixels  
305 // abaixo do título, criando espaço entre o título e  
306 // o corpo do gráfico.  
307  
308 }  
309 },  
310  
311 legend: {  
312 // 'legend' configura a legenda do gráfico, que ajuda a  
313 // identificar o que cada conjunto de dados representa.  
314  
315 display: true,  
316 // 'display: true' ativa a exibição da legenda.  
317  
318 position: 'top',  
319 // 'position: "top"' posiciona a legenda no topo do gráfico,  
320 // onde é facilmente visível.  
321  
322 labels: {  
323 // 'labels' define a estilização para os rótulos da legenda.  
324  
325 font: {  
326 // 'font' define as propriedades da fonte dos  
327 // rótulos da legenda.  
328  
329 size: 14,  
330 // 'size: 14' estabelece o tamanho da fonte para os
```

OUTLINE

TIMELINE

Ln 301, Col 4 Spaces: 8 UTF-8 CRLF {} JavaScript ⚡ Port: 5500

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Gráfico de Área com Dados de Excel
- Toolbar:** Includes icons for file operations like Open, Save, and Close, and other standard editor functions.
- Left Sidebar (EXPLORER):** Shows a tree view of files:
  - P41 - GRÁFICO DE ÁREA COM DADOS DE EXCEL
    - dados.xlsx
    - # estilos.css
    - index.html
    - scripts.js**
- Code Editor:** The current file is scripts.js. The code is as follows:

```
index.html # estilos.css JS scripts.js X
JS scripts.js > atualizarGraficoAreas > options > plugins > legend > labels > font
331 // rótulos da legenda, garantindo que sejam legíveis
332 // sem dominar o gráfico.
333
334 family: 'Roboto'
335 // 'family: "Roboto"' usa a mesma família de fontes do
336 // título, mantendo a consistência visual no gráfico.
337
338 },
339
340 padding: 20
341 // 'padding: 20' adiciona espaçamento interno de 20 pixels
342 // em torno dos rótulos, separando-os visualmente para clareza.
343
344 }
345 },
346
347 tooltip: {
348 // 'tooltip' é um objeto que define as configurações das
349 // dicas de ferramentas (tooltips) que aparecem quando o
350 // usuário passa o mouse sobre pontos de dados no gráfico.
351
352 backgroundColor: 'rgba(0, 0, 0, 0.7)',
353 // 'backgroundColor' define a cor de fundo dos tooltips.
354 // Neste caso, usa-se preto com uma opacidade de 70% (0.7),
355 // criando um fundo semitransparente que melhora a
356 // visibilidade do texto sem ocultar completamente o
357 // gráfico por trás.
358
359 titleFont: {
360 // 'titleFont' especifica as propriedades da fonte usada
```

The code is written in JavaScript and defines styles for a chart's legend and tooltip. It uses the Roboto font for the legend and a semi-transparent black background for the tooltip to ensure readability.



EXPLORER

#### ✓ P41 - GRÁFICO DE ÁREA COM DADOS DE EXCEL



 dados.xlsx

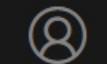
## # estilos.cs



<> index.htm

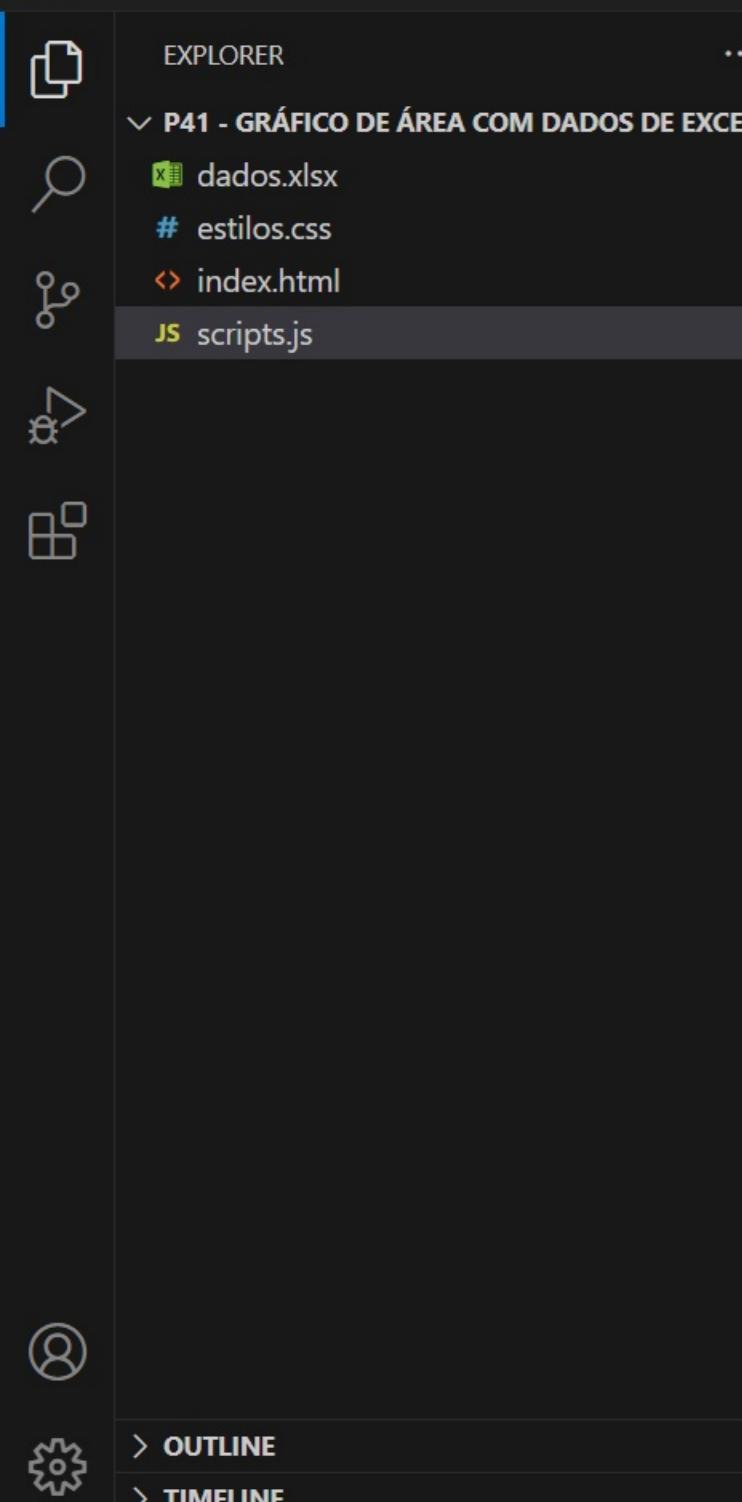


JS scripts.js



> OUTLINE

> TIMELINE



EXPLORER

P41 - GRÁFICO DE ÁREA COM DADOS DE EXCEL

- dados.xlsx
- # estilos.css
- index.html
- scripts.js

> OUTLINE

> TIMELINE

index.html # estilos.css JS scripts.js X

scripts.js > atualizarGraficoAreas > options > plugins > tooltip > callbacks > label

```
421     return label;
422     // Retorna o rótulo completo, que inclui o nome do conjunto
423     // de dados e o valor do ponto de dados.
424 }
425 }
426 }
427 }
428 },
429
430 scales: {
431     // 'scales' define as configurações das escalas do gráfico,
432     // especificamente como os eixos são configurados e apresentados.
433
434 y: {
435     // Configurações específicas para o eixo y, que geralmente
436     // representa a escala vertical em gráficos de linhas, barras, etc.
437
438     beginAtZero: true,
439     // 'beginAtZero: true' garante que a escala do eixo y comece em zero.
440     // Isso é útil para gráficos de quantidades, pois oferece uma
441     // perspectiva clara do início da contagem de valores.
442
443     title: {
444         // 'title' define as configurações do título do eixo y.
445
446         display: true,
447         // 'display: true' habilita a exibição do título do eixo y.
448
449         text: 'Quantidade de Vendas',
450         // 'text' especifica o texto do título, que neste
```

EXPLORER

...

✓ P41 - GRÁFICO DE ÁREA COM DADOS DE EXCEL

dados.xlsx

# estilos.css

index.html

scripts.js

...

OUTLINE

TIMELINE

index.html # estilos.css JS scripts.js X

JS scripts.js > atualizarGraficoAreas > options > scales > y > title

```
451 // caso é "Quantidade de Vendas".
452 // Esse título fornece uma clara indicação do que os
453 // valores no eixo y representam.
454
455 font: {
456     // 'font' configura a aparência do texto do título.
457
458     size: 14,
459     // 'size: 14' define o tamanho da fonte do título do
460     // eixo y em 14 pixels, o que é adequado para
461     // legibilidade sem ser excessivamente grande.
462
463     family: 'Roboto',
464     // 'family: "Roboto"' especifica a família de fonte
465     // como Roboto, proporcionando uma aparência limpa e moderna.
466
467     weight: 'bold'
468     // 'weight: "bold"' aplica um peso de fonte negrito ao
469     // título, destacando-o visualmente no gráfico.
470
471 }
472
473
474 grid: {
475     // 'grid' define as configurações da grade do eixo y.
476
477     color: 'rgba(0, 0, 0, 0.1)',
478     // 'color' estabelece a cor das linhas da grade, que
479     // neste caso são um preto muito suave (opacidade de 10%).
480     // Isso permite que as linhas da grade sejam visíveis para
```

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

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Gráfico de Área com Dados de Excel
- Explorer:** Shows a project structure under "P41 - GRÁFICO DE ÁREA COM DADOS DE EXCEL":
  - dados.xlsx
  - # estilos.css
  - index.html
  - scripts.js** (selected)
- Code Editor:** The "scripts.js" file is open, showing the following code:

```
index.html
# estilos.css
scripts.js

scripts.js > atualizarGraficoAreas > options > scales > y > grid
    // orientação sem dominar o gráfico visualmente.

    borderDash: [5, 5]
        // 'borderDash' configura as linhas da grade para
        // serem tracejadas, com segmentos de 5 pixels
        // alternando com espaços de 5 pixels.
        // Isso adiciona uma estética diferenciada para o
        // plano de fundo do gráfico, facilitando a
        // distinção entre os pontos de dados e o plano de fundo.

    }

x: {
    // Configurações específicas para o eixo x, que geralmente
    // representa a escala horizontal em gráficos de linhas, barras, etc.

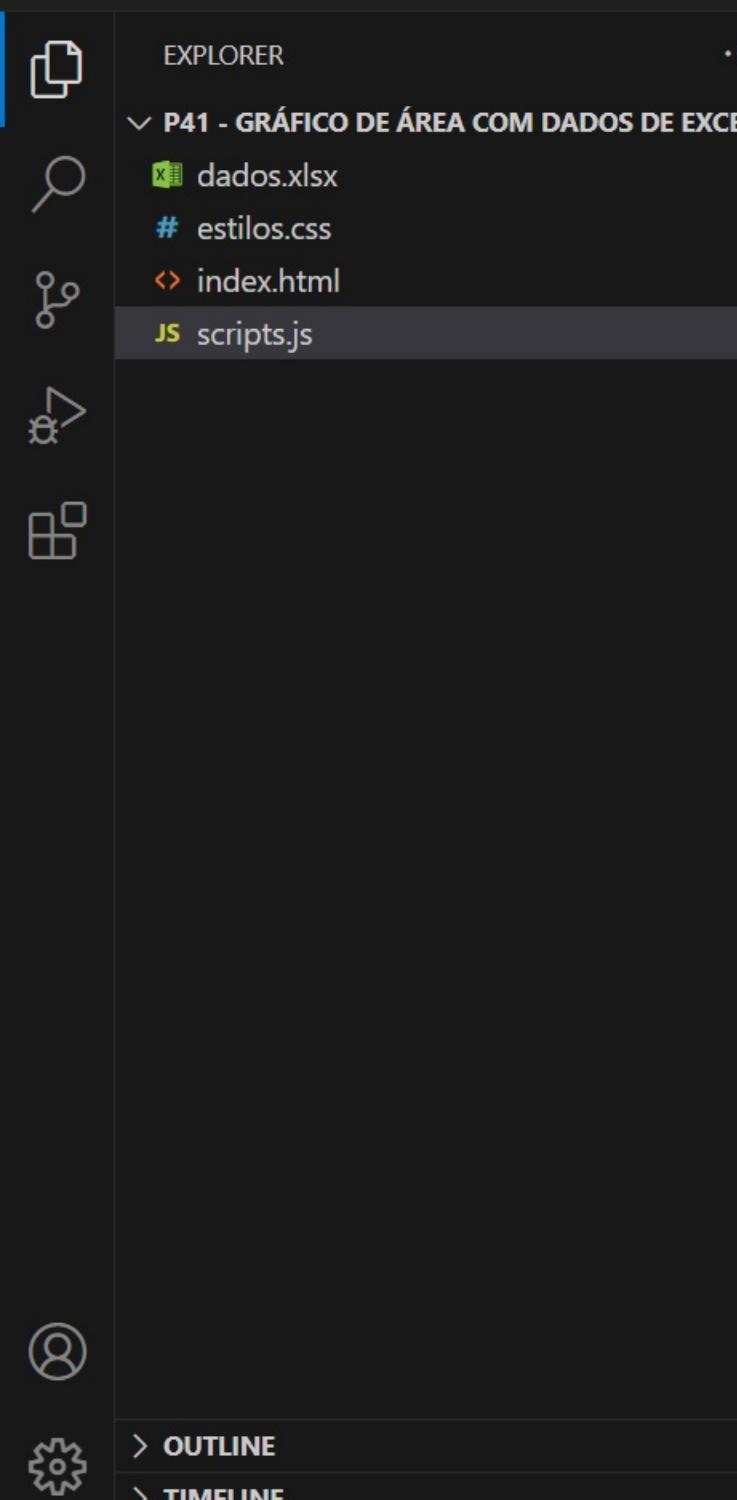
    title: {
        // 'title' define as configurações do título do eixo x.

        display: true,
        // 'display: true' habilita a exibição do título do eixo x.
        // Isso é importante para indicar claramente o que os
        // dados ao longo deste eixo representam.

        text: 'Produto',
        // 'text' especifica o texto do título, que neste caso é "Produto".
        // Este título ajuda a identificar que os dados ao longo do
        // eixo x correspondem a diferentes produtos.

    }
}

481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
```
- Bottom Status Bar:** Shows file information: index.html, Line 481, Column 2, Spaces: 8, UTF-8, CRLF, and a port indicator: Port: 5500.



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

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P41 - Gráfico de Área com Dados de Excel
- Explorer:** Shows a project structure under "P41 - GRÁFICO DE ÁREA COM DADOS DE EXCEL":
  - dados.xlsx
  - # estilos.css
  - index.html
  - scripts.js** (selected)
- Code Editor:** The "scripts.js" file is open, showing the following code:

```
541 borderDash: [5, 5]
542 // 'borderDash' configura as linhas da grade para serem
543 // tracejadas, com segmentos de linha e espaços
544 // de 5 pixels alternadamente.
545 // O estilo tracejado adiciona uma textura sutil ao
546 // fundo do gráfico, ajudando a distinguir as áreas
547 // do gráfico sem distrair dos dados.
548
549
550 }
551 }
552 }
553 },
554
555 plugins: [ChartDataLabels]
556 // Esta linha especifica os plugins a serem usados com o gráfico.
557 // Aqui, 'ChartDataLabels' é incluído no array de plugins.
558 // 'ChartDataLabels' é um plugin Chart.js que permite mostrar
559 // rótulos de dados nos gráficos, proporcionando informações
560 // adicionais diretamente nos pontos de dados ou barras do gráfico.
561 // Isso é útil para fornecer um contexto instantâneo e melhorar a
562 // legibilidade do gráfico, mostrando valores numéricos ou
563 // descrições diretamente nos elementos visuais.
564
565 });
566 }
```
- Bottom Status Bar:** Shows file information: index.html, Line 541, Column 7, Spaces: 8, UTF-8, CRLF, {} JavaScript, Port: 5500.

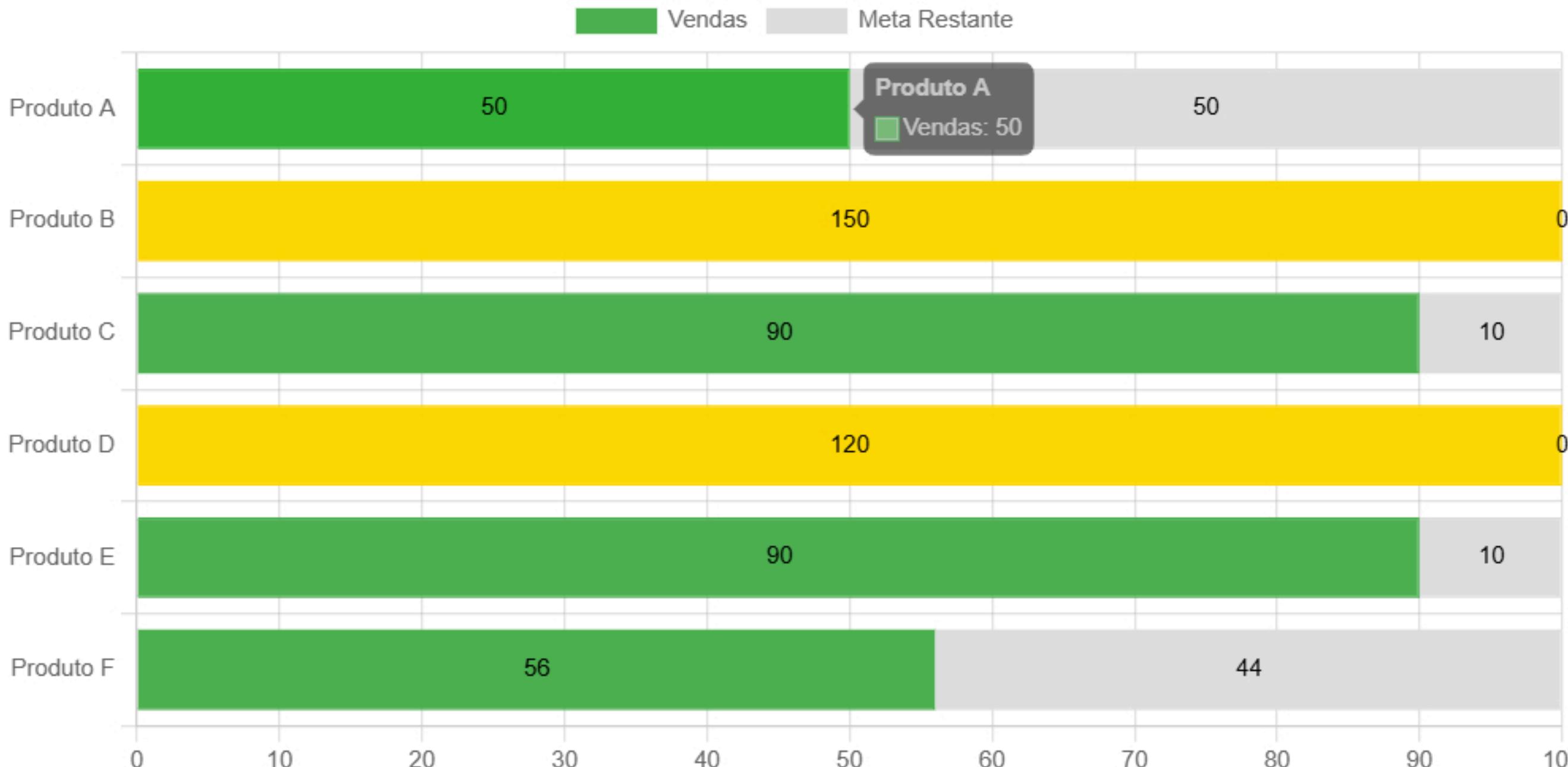
# **UC: Projetos**

## **Assunto:**

### **Small Projects**

### **P42 – Gráfico de Barras Horizontais**

## Gráfico de Barras Horizontais



EXPLORER

P42 - GRÁFICO DE BARRAS HORIZONTAIS

- dados.xlsx
- estilos.css
- index.html
- scripts.js

OUTLINE

TIMELINE

index.html

```
<!DOCTYPE html>
<!-- Declaração do tipo de documento. Esta linha é
     essencial para ativar o modo de padrões completos
     nos navegadores, assegurando que o HTML5 seja
     corretamente interpretado. --&gt;

&lt;html lang="pt-BR"&gt;
<!-- A tag &lt;html&gt; define o início do documento HTML. O atributo 'lang="pt-BR"' especifica que o idioma principal do conteúdo do documento é o português do Brasil, o que ajuda os leitores de tela e os mecanismos de busca a entenderem a língua do conteúdo. --&gt;

&lt;head&gt;
<!-- A tag &lt;head&gt; encapsula metadados (informações sobre dados) do documento HTML, que não são exibidos diretamente na página web, mas são utilizados pelos navegadores e outros serviços web. --&gt;

&lt;meta charset="UTF-8"&gt;
<!-- A tag &lt;meta&gt; com o atributo 'charset="UTF-8"' especifica a codificação de caracteres UTF-8 para o documento HTML. Isso garante que qualquer caractere em qualquer idioma possa ser exibido corretamente no navegador. --&gt;

&lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt;
<!-- Outra tag &lt;meta&gt;, que configura a janela de visualização dos dispositivos móveis para controlar a escala e as dimensões da página.
     - 'width=device-width' ajusta a largura da página para seguir a largura da tela do dispositivo.
     - 'initial-scale=1.0' define o nível inicial de zoom quando a página é carregada pela primeira vez. --&gt;</pre>
```

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P42 - Gráfico de Barras Horizontais
- Explorer:** Shows files in the project: dados.xlsx, estilos.css, index.html (selected), and scripts.js.
- Code Editor:** The current file is index.html, specifically the head section. The code is as follows:

```
<title>Gráfico de Vendas - Gráfico de Barras Horizontais</title>
<!-- A tag &lt;title&gt; define o título do documento, que é mostrado na aba do navegador e é usado pelos mecanismos de busca para determinar o título da página nos resultados de pesquisa. --&gt;

&lt;link rel="stylesheet" href="estilos.css"&gt;
<!-- A tag &lt;link&gt; é usada aqui para conectar uma folha de estilos externa, localizada no arquivo 'estilos.css'. Isso permite que todos os estilos CSS sejam definidos separadamente do HTML, facilitando a manutenção e a reutilização do CSS. --&gt;

&lt;script src="https://cdnjs.cloudflare.com/ajax/libs/xlsx/0.18.5/xlsx.full.min.js"&gt;&lt;/script&gt;
<!-- Inclui o script 'xlsx.full.min.js' de uma CDN (Content Delivery Network). Esta biblioteca JavaScript é usada para ler arquivos Excel (XLSX) diretamente no navegador, permitindo manipular e utilizar dados de arquivos Excel sem a necessidade de um servidor backend. --&gt;

&lt;script src="https://cdn.jsdelivr.net/npm/chart.js"&gt;&lt;/script&gt;
<!-- Inclui o script 'Chart.js' de uma CDN. Chart.js é uma biblioteca leve que permite criar gráficos interativos e responsivos em HTML5. --&gt;

&lt;script src="https://cdn.jsdelivr.net/npm/chartjs-plugin-datalabels"&gt;&lt;/script&gt;
<!-- Adiciona o plugin 'chartjs-plugin-datalabels' para Chart.js, que permite mostrar rótulos de dados nos gráficos, oferecendo mais informações diretamente nas visualizações. --&gt;

&lt;/head&gt;</pre>

Bottom Bar: Icons for Outline, Timeline, and status indicators (0 0 0 0 Port: 5500).


```

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P42 - Gráfico de Barras Horizontais
- Left Sidebar (EXPLORER):** Shows a project structure under "P42 - GRÁFICO DE BARRAS HORIZONTAIS":
  - dados.xlsx
  - # estilos.css
  - index.html (selected)
  - scripts.js
- Right Sidebar:** Icons for File, Edit, Selection, View, Go, Run, ...
- Code Editor:** The "index.html" file is open, showing the following code:

```
<!-- A tag <body> contém todo o conteúdo da página que é visível para o usuário, como texto, imagens, links, entre outros elementos. -->
<div class="container">
    <!-- A tag <div> com a classe 'container' serve como um contêiner para outros elementos HTML. A classe 'container' é geralmente usada para aplicar estilos específicos como largura máxima, margens automáticas para centralização, e estilos de background e borda, definidos no arquivo CSS. -->
<h1>Gráfico de Barras Horizontais</h1>
    <!-- A tag <h1> representa um cabeçalho de primeiro nível. É usado aqui para fornecer um título visível na página, descrevendo o conteúdo ou o propósito da página, neste caso, indicando que se trata de um gráfico de vendas com barras horizontais. Os cabeçalhos são importantes para a acessibilidade e SEO (Search Engine Optimization). -->
<div id="grafico">
    <!-- Uma segunda <div> com o id 'grafico' serve como contêiner específico para o gráfico. O ID é único e pode ser usado para aplicar estilos específicos ou manipulações via JavaScript. -->
    <canvas id="graficoKPI"></canvas>
        <!-- A tag <canvas> com o id 'graficoKPI' é usada como o elemento onde o gráfico será renderizado. O Canvas fornece um espaço de desenho para gráficos (via JavaScript), como gráficos de barras, linhas, tortas, etc. O ID 'graficoKPI' permite que este canvas seja facilmente acessado e manipulado através de scripts. -->
</div>
```
- Bottom Status Bar:** Icons for File, Edit, Selection, View, Go, Run, ... and status: Ln 61, Col 8, Spaces: 7, UTF-8, CRLF, HTML, Port: 5500.

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P42 - Gráfico de Barras Horizontais
- Explorer:** Shows a project structure with files: dados.xlsx, estilos.css, index.html, and scripts.js.
- Code Editor:** The current file is index.html, line 91. The code includes a script tag pointing to scripts.js, which is described as containing code to manipulate data and render a chart. The code ends with a closing html tag at line 100 and a closing body tag at line 99.
- Bottom Status Bar:** Shows file count (0), line count (0), character count (0), and other status indicators.

```
</div>
<script src="scripts.js"></script>
<!-- A tag &lt;script&gt; com o atributo 'src' que aponta para 'script.js' é usada
     para incluir um arquivo JavaScript externo. Este script contém o código
     necessário para manipular os dados e renderizar o gráfico no elemento &lt;canvas&gt;
     mencionado acima. --&gt;
&lt;/body&gt;
&lt;/html&gt;
<!-- Fim do documento HTML. A tag &lt;/html&gt; indica que todos os elementos
     do documento HTML foram fechados corretamente. --&gt;</pre>
```





EXPLORER

▽ P42 - GRÁFICO DE BARRAS HORIZONTAL

- dados.xlsx
- # estilos.css
- <> index.html
- JS scripts.js

> OUTLINE

> TIME LINE

```
< index.html # estilos.css >

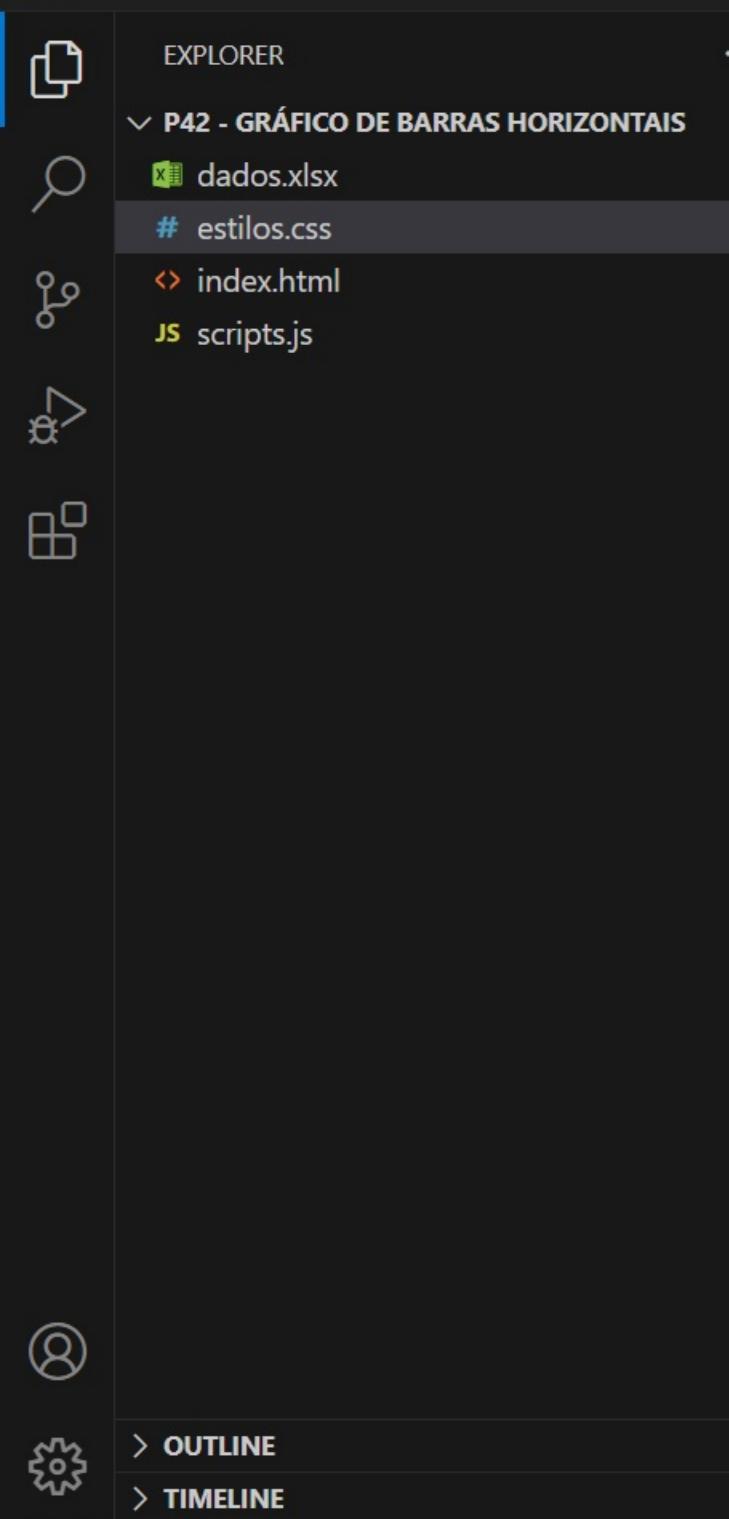
# estilos.css > .container
31    |     contêiner pode se expandir até 1200px de largura, mas não mais
32    |     que isso, garantindo que o conteúdo não fique muito estendido
33    |     em monitores grandes. */
34
35    margin: 50px auto;
36    /* Configura a margem superior e inferior para 50px e as margens laterais
37    |     para 'auto'. Isso centraliza o contêiner horizontalmente dentro
38    |     de seu contêiner pai (neste caso, o <body>), pois as margens
39    |     automáticas empurram igualmente de ambos os lados. */
40
41    padding: 20px;
42    /* Adiciona um preenchimento de 20px a todos os lados do contêiner, criando
43    |     espaço entre os bordos do contêiner e seu conteúdo, melhorando a
44    |     legibilidade e a estética. */
45
46    background-color: #ffffff;
47    /* Define a cor de fundo do contêiner para branco. Isso ajuda a destacar o
48    |     contêiner do fundo da página e torna o conteúdo mais legível. */
49
50    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
51    /* Aplica uma sombra ao redor do contêiner. A sombra é suítil (preta com 10%
52    |     de opacidade) e se espalha 10px em todas as direções. Isso adiciona
53    |     profundidade ao layout, fazendo o contêiner parecer elevado em
54    |     relação ao restante da página. */
55
56    border-radius: 8px;
57    /* Arredonda os cantos do contêiner com um raio de 8px. Cantos arredondados
58    |     são uma escolha de design comum que pode fazer a interface parecer
59    |     mais amigável e moderna. */
60
```

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P42 - Gráfico de Barras Horizontais
- Explorer:** Shows a project folder named "P42 - GRÁFICO DE BARRAS HORIZONTAIS" containing files: dados.xlsx, # estilos.css, index.html, and scripts.js.
- Code Editor:** The file "# estilos.css" is open, showing the following CSS code:

```
index.html # estilos.css X
# estilos.css > .container
61     text-align: center;
62     /* Alinha o texto dentro do contêiner ao centro. Isso é frequentemente
63      usado em cabeçalhos ou outros elementos importantes para chamar
64      atenção e melhorar a estética. */
65
66 }
67
68
69 h1 {
70     color: #333333;
71     /* Define a cor do texto para os elementos <h1> para um cinza escuro (#333333).
72      Esta cor é frequentemente usada para texto porque fornece um alto
73      contraste com fundos claros sem ser tão duro aos olhos quanto o
74      preto puro. É ideal para títulos devido à sua legibilidade e
75      destaque sutil. */
76 }
77
78
79 #grafico {
80     width: 100%;
81     /* Define a largura do contêiner do gráfico para 100% da largura de seu
82      contêiner pai. Isso faz com que o contêiner do gráfico se ajuste à
83      largura disponível, tornando-o responsivo e adaptável a
84      diferentes tamanhos de tela. */
85
86
87     max-width: 800px;
88     /* Estabelece uma largura máxima de 800px para o contêiner do gráfico.
89      Isso significa que, mesmo que o contêiner pai possa se estender
90      além de 800px, o contêiner do gráfico não excederá essa largura,
```

**Bottom Bar:** Icons for File, Edit, Selection, View, Go, Run, ...; Status bar showing Line 61, Column 5, Spaces: 4, UTF-8, CRLF, CSS, Port: 5500.



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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P42 - Gráfico de Barras Horizontais
- Toolbar:** Includes icons for file operations like Open, Save, Find, and Run.
- Left Sidebar (EXPLORER):** Shows a project structure under "P42 - GRÁFICO DE BARRAS HORIZONTAIS" with files: dados.xlsx, # estilos.css, index.html, and scripts.js (selected).
- Central Area:** Displays the contents of the selected file, "scripts.js".

```
document.addEventListener('DOMContentLoaded', () => {
    /* Adiciona um ouvinte de evento ao objeto 'document' para o evento 'DOMContentLoaded'. Este evento é disparado quando todo o conteúdo HTML foi completamente carregado e analisado, sem esperar pelo CSS, imagens e subframes para terminar de carregar. O uso desse evento garante que o script não tentará manipular elementos do DOM antes de estarem disponíveis na página.
    A função é definida como uma arrow function, que é uma maneira concisa de escrever funções em JavaScript. */

    const contextoGrafico = document.getElementById('graficoKPI').getContext('2d');
    /* A constante 'contextoGrafico' é definida para armazenar o contexto de renderização 2D do elemento canvas com o ID 'graficoKPI'.
    - 'document.getElementById('graficoKPI')' seleciona o elemento canvas pelo seu ID.
    - '.getContext('2d')' é um método que retorna um contexto de desenho no canvas, que é usado para desenhar gráficos, fazer manipulações de imagem, etc. O parâmetro '2d' especifica que o contexto de desenho será em duas dimensões. */

    let dados = [];
    /* Declara uma variável 'dados' usando 'let' e inicializa com um array vazio. Esta variável será usada para armazenar os dados que serão extraídos do arquivo Excel. A utilização de 'let' permite que essa variável possa ser modificada ou atualizada mais tarde no código. */

    let grafico;
    /* Declara uma variável 'grafico' com 'let', mas não a inicializa
```

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P42 - Gráfico de Barras Horizontais
- Explorer:** Shows a project named "P42 - GRÁFICO DE BARRAS HORIZONTAIS" containing files: dados.xlsx, # estilos.css, index.html, and scripts.js.
- Code Editor:** The "scripts.js" file is open, showing the following code:

```
index.html # estilos.css JS scripts.js X JS scripts.js > document.addEventListener('DOMContentLoaded') callback

31 com um valor. Esta variável será utilizada para referenciar o
32 gráfico que será criado usando a biblioteca Chart.js, permitindo
33 manipulações posteriores, como atualizações ou destruição do gráfico. */
34
35 // Função para ler o arquivo Excel
36 function lerArquivoExcel() {
37     /* Define a função 'lerArquivoExcel' que não recebe argumentos e é
38     responsável por carregar e processar um arquivo Excel. Essa
39     função é usada para extrair dados do arquivo e disponibilizá-los
40     para outras partes do script. */
41
42     const url = 'dados.xlsx'; // URL ou caminho do arquivo Excel
43     /* Declara uma constante 'url' e atribui a ela o caminho do arquivo
44     Excel a ser lido. Essa URL pode ser um caminho local
45     (como mostrado) ou um link para um arquivo hospedado remotamente. */
46
47     fetch(url)
48     /* Utiliza a função 'fetch' para fazer uma requisição HTTP GET ao
49     caminho especificado pela variável 'url'. A função 'fetch'
50     retorna uma promessa que resolve um objeto de resposta (Response). */
51
52     .then(response => response.arrayBuffer())
53     /* Utiliza o método 'then' para manipular a resposta da requisição.
54     O 'response.arrayBuffer()' é chamado para converter a resposta
55     em um ArrayBuffer, um tipo de objeto usado para representar
56     dados binários genéricos. */
57
58     .then(data => {
59         /* Outro método 'then' que é usado para manipular os dados
60         recebidos do passo anterior. 'data' contém o ArrayBuffer resultante. */
```

The code is a JavaScript script that reads an Excel file ("dados.xlsx") using the Fetch API to get its contents as an ArrayBuffer. It then uses the ArrayBuffer to create a horizontal bar chart using the Chart.js library.



EXPLORER



✓ P42 - GRÁFICO DE BARRAS HORIZONTAIS



 dados.xlsx



## # estilos.cs



<> index.htm



## JS scripts.js

```
JS scripts.js > ⚡ document.addEventListener('DOMContentLoaded') callback > ⚡ lerArquivoExcel > ⚡ then() callback
61
62         const workbook = XLSX.read(data, { type: 'array' });
63         /* Utiliza a biblioteca XLSX para ler os dados do ArrayBuffer
64             como um livro de Excel. A opção '{ type: 'array' }'
65             especifica que os dados são fornecidos como um array. */
66
67         const sheet = workbook.Sheets['Dados'];
68         /* Acessa a primeira planilha do livro Excel, que é esperada
69             para ser nomeada 'Dados'. */
70
71         const jsonData = XLSX.utils.sheet_to_json(sheet, { header: 1 });
72         /* Converte a planilha Excel em JSON, utilizando a função 'sheet_to_json'
73             da biblioteca XLSX. A opção '{ header: 1 }' indica que a
74             primeira linha da planilha deve ser tratada como cabeçalho. */
75
76         dados = jsonData.slice(1); // Remover cabeçalhos
77         /* Atualiza a variável 'dados' para conter somente as linhas de
78             dados, excluindo a primeira linha que contém os cabeçalhos.
79             O método 'slice(1)' é usado para criar uma nova array que começa
80             do segundo elemento da array original, descartando o primeiro
81             elemento (cabeçalho). */
82
83         criarGrafico();
84         /* Chama a função 'criarGrafico' para processar os dados atualizados e
85             gerar o gráfico com base neles. */
86
87     })
88
89     .catch(error => console.error('Erro ao ler o arquivo Excel:', error));
90     /* O método 'catch' é usado para capturar e lidar com qualquer erro
```



> OUTLINE



> TIMELINE

File Edit Selection View Go Run ... ← → P42 - Gráfico de Barras Horizontais

EXPLORER

P42 - GRÁFICO DE BARRAS HORIZONTAIS

- dados.xlsx
- # estilos.css
- index.html
- scripts.js

scripts.js > document.addEventListener('DOMContentLoaded') callback > lerArquivoExcel

```
91     que ocorra durante qualquer uma das etapas anteriores do
92     processo de fetch. Os erros são registrados no console para
93     diagnóstico e correção. */
94
95 }
96
97
98 // Função para criar o gráfico
99 function criarGrafico() {
100     /* Define a função 'criarGrafico', que não recebe argumentos.
101        Esta função utiliza os dados processados anteriormente para
102        construir um gráfico visual usando a biblioteca Chart.js. */
103
104     const produtos = dados.map(row => row[0]);
105     /* Extrai o nome dos produtos dos dados. 'dados' é uma matriz onde
106        cada linha representa um registro diferente e cada coluna um
107        campo diferente. Aqui, 'row[0]' refere-se ao primeiro campo de
108        cada linha, que neste caso é o nome do produto. */
109
110     const vendas = dados.map(row => row[1]);
111     /* Extrai o número de vendas de cada produto. 'row[1]' refere-se ao
112        segundo campo de cada linha, que neste caso é o número
113        de vendas do produto. */
114
115     const metas = dados.map(row => row[2]);
116     /* Extrai a meta de vendas para cada produto. 'row[2]' refere-se
117        ao terceiro campo de cada linha, que neste caso é a meta
118        de vendas estabelecida para o produto. */
119
120     const vendasDataset = vendas.map((venda, index) => Math.min(venda, metas[index]));
```

Ln 91, Col 2 Spaces: 8 UTF-8 CRLF {} JavaScript ⚡ Port: 5500

File Edit Selection View Go Run ... ← → P42 - Gráfico de Barras Horizontais

EXPLORER ...

P42 - GRÁFICO DE BARRAS HORIZONTAIS

- dados.xlsx
- # estilos.css
- index.html
- scripts.js

scripts.js > document.addEventListener('DOMContentLoaded') callback > criarGrafico

```
121     /* Calcula o valor efetivo de vendas que será exibido no gráfico. Utiliza a
122      função 'Math.min' para escolher o menor valor entre as vendas
123      reais e a meta, garantindo que o valor de vendas no gráfico
124      não exceda a meta. */
125
126     const metasDataset = metas.map((meta, index) => meta - vendasDataset[index]);
127     /* Calcula o valor restante para alcançar a meta para cada produto. Subtrai
128      as vendas efetivas (do 'vendasDataset') da meta total, resultando na
129      quantidade que ainda falta para alcançar a meta. */
130
131     const coresVendas = vendas.map((venda, index) => venda > metas[index] ? '#FFD700' : '#4CAF50');
132     /* Define a cor das barras de vendas no gráfico. Se as vendas de um
133      produto excederem a meta, a barra é colorida de dourado ('#FFD700');
134      caso contrário, é colorida de verde ('#4CAF50'). Essa visualização
135      ajuda rapidamente a identificar quais produtos estão acima ou
136      abaixo das metas. */
137
138     if (grafico) {
139         grafico.destroy();
140         /* Verifica se um gráfico já existe (se 'grafico' não é nulo). Se
141          existir, destrói o gráfico anterior antes de criar um novo.
142          Isso é necessário para evitar sobreposições de gráficos antigos e
143          novos quando a função é chamada novamente, como ao atualizar os dados. */
144     }
145
146
147
148     grafico = new Chart(contextoGrafico, {
149         /* Instancia um novo gráfico utilizando a biblioteca Chart.js.
150             O objeto 'grafico' é atribuído a uma nova instância de 'Chart',
```

Ln 121, Col 4 Spaces: 8 UTF-8 CRLF {} JavaScript Port: 5500



EXPLORER

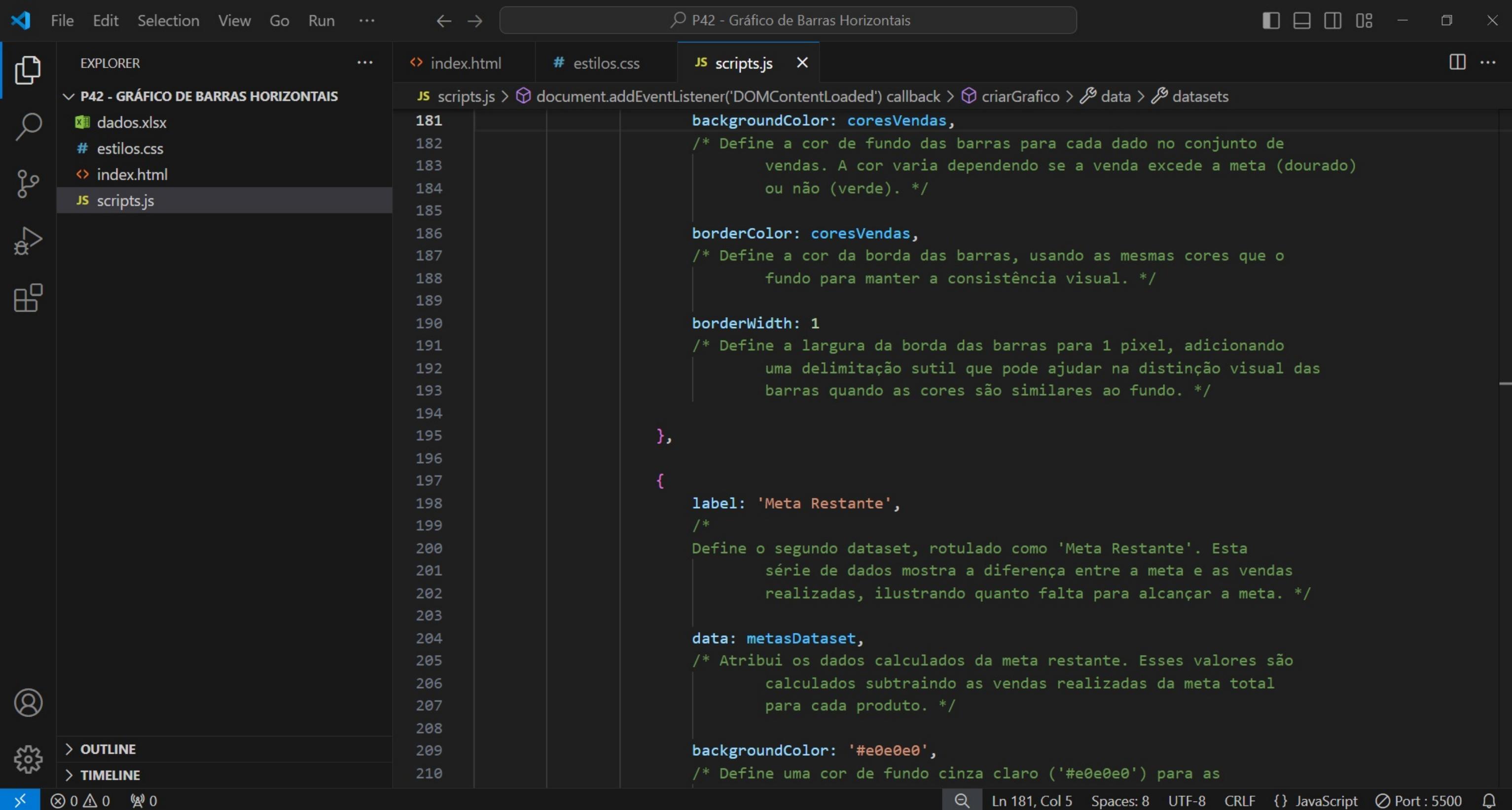
#### ✓ P42 - GRÁFICO DE BARRAS HORIZONTAIS

 dados.xlsx

# estilos.cs

<> index.htm

index.html | estilos.css | scripts.js



The screenshot shows a Microsoft Edge browser window with the following details:

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Address Bar:** P42 - Gráfico de Barras Horizontais
- Right Side:** Minimize, Maximize, Close buttons.
- Left Sidebar (Explorer):**
  - EXPLORER
  - P42 - GRÁFICO DE BARRAS HORIZONTAIS
    - dados.xlsx
    - # estilos.css
    - index.html
    - scripts.js**
  - OUTLINE
  - TIMELINE
- Content Area:** A code editor showing the contents of the scripts.js file. The code defines two datasets for a horizontal bar chart, setting colors based on sales performance (gold for above target, green for below).
- Bottom Status Bar:** Shows file number (0), document status (0), and other system information.

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P42 - Gráfico de Barras Horizontais
- Icons:** Explorer, Search, Open, Save, Help, Outline, Timeline.
- File Explorer (Left):** P42 - GRÁFICO DE BARRAS HORIZONTAIS (selected), dados.xlsx, # estilos.css, index.html, scripts.js.
- Editor Area (Center):** The file scripts.js is open, showing the following code:

```
index.html    # estilos.css    JS scripts.js    X
JS scripts.js > document.addEventListener('DOMContentLoaded') callback > criarGrafico > data > datasets
211           barras que representam a meta restante. Esta cor foi escolhida
212           para ser discreta, destacando as barras de vendas enquanto
213           ainda mostra claramente a diferença até a meta. */
214
215           borderColor: '#e0e0e0',
216           /* Define a cor da borda das barras da meta restante como o
217           mesmo cinza claro, mantendo a uniformidade no design
218           visual do gráfico. */
219
220           borderWidth: 1
221           /* Define a largura da borda das barras da meta restante
222           para 1 pixel, similar ao conjunto de dados de vendas. */
223
224       }
225   ]
226 },
227
228 options: {
229     /* A propriedade 'options' dentro do objeto de configuração do
230     gráfico contém uma série de subpropriedades que definem o
231     comportamento e a apresentação visual do gráfico. */
232
233     responsive: true,
234     /* Define o gráfico como responsivo. Isso significa que o gráfico
235     se ajustará automaticamente ao tamanho do container em
236     que está sendo renderizado. Essencial para garantir uma
237     boa visualização em dispositivos com diferentes tamanhos de tela. */
238
239     indexAxis: 'y',
240     /* Define o eixo de índice como 'y', alterando a orientação do gráfico
```

The code is a configuration object for a chart. It includes properties for data, options (responsive, indexAxis), and styling (borderColor, borderWidth).



EXPLORER



✓ P42 - GRÁFICO DE BARRAS HORIZONTAIS



 dados.xlsx

## # estilos.cs



< index.htm



JS scripts.js



> OUTLINE



> TIMELINE

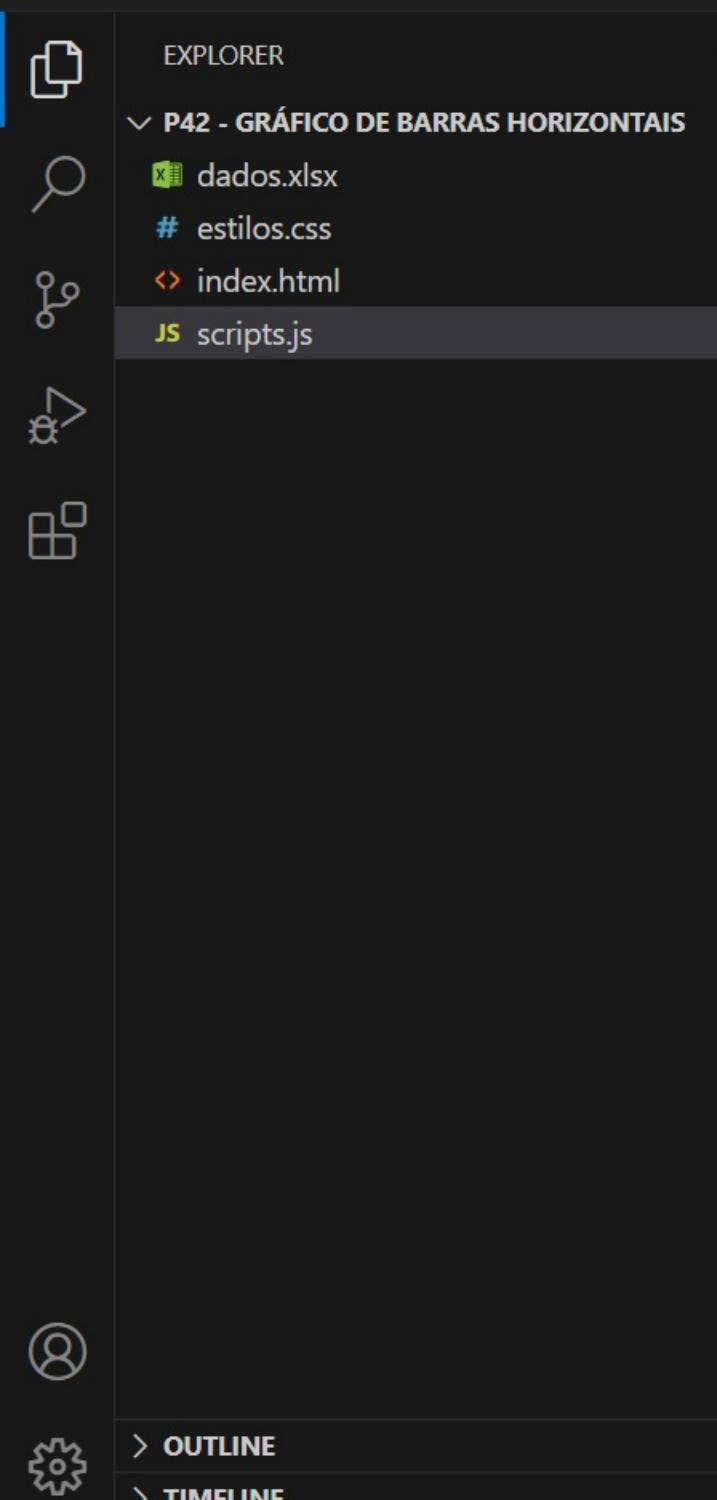
The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P42 - Gráfico de Barras Horizontais
- Toolbar:** Includes icons for file operations like Open, Save, Find, and Close.
- Left Sidebar (EXPLORER):** Shows a project structure under "P42 - GRÁFICO DE BARRAS HORIZONTAIS" with files: dados.xlsx, # estilos.css, index.html, and scripts.js. "scripts.js" is currently selected.
- Code Editor:** The "scripts.js" tab is active, displaying the following code:

```
index.html    # estilos.css  JS scripts.js  X

JS scripts.js > document.addEventListener('DOMContentLoaded') callback > criarResultado > options > scales > x > ticks
271           eixo 'x' serão de 10 unidades, ajudando a manter uma escala
272           uniforme e legível. */
273
274       }
275   },
276
277   y: {
278
279     /*
280      Configurações específicas para o eixo 'y', que neste gráfico
281      horizontal representa as categorias, ou seja, os produtos. */
282
283     stacked: true
284     /* Ativa o empilhamento de barras no eixo 'y'. Isso permite que cada
285      conjunto de dados (vendas e metas restantes) seja mostrado
286      na mesma barra, mas segmentado visualmente dentro dessa barra. */
287
288   }
289 },
290
291   plugins: {
292     /*
293       A seção 'plugins' do objeto de configuração do gráfico permite definir e
294       customizar os comportamentos adicionais através de plugins externos
295       ou incorporados, como legendas e tooltips. */
296
297   legend: {
298     /*
299       A propriedade 'legend' configura a legenda do gráfico, que fornece
300       informação sobre o que cada cor no gráfico representa. */
301
302       position: 'top',
303
304   }
305 }
```

The code is written in JavaScript and defines a configuration object for a horizontal bar chart. It includes sections for x-axis ticks, y-axis configurations (with stacked bars), plugins, and a legend.



File Edit Selection View Go Run ... ← → P42 - Gráfico de Barras Horizontais

EXPLORER ...

P42 - GRÁFICO DE BARRAS HORIZONTAIS

- dados.xlsx
- # estilos.css
- index.html
- scripts.js

scripts.js > document.addEventListener('DOMContentLoaded') callback > criarGrafico > options > plugins > tooltip > callbacks > label

```
331     }
332     }
333     }
334     }
335     },
336
337     dataLabels: {
338         /* 'dataLabels' é um plugin que permite exibir rótulos de dados
339         diretamente nos elementos do gráfico, como barras ou
340         pontos. A configuração abaixo define como esses rótulos
341         são apresentados e formatados. */
342
343         display: true,
344             /* Define a exibição dos rótulos de dados como ativa ('true'). Isso
345             significa que os rótulos serão mostrados automaticamente em
346             todos os dados visíveis no gráfico. */
347
348         color: 'black',
349             /* Define a cor dos rótulos de dados para preto. Esta escolha
350             garante que os rótulos sejam claramente visíveis contra um
351             fundo geralmente mais claro das barras do gráfico. */
352
353         anchor: 'center',
354             /* Define o ponto de ancoragem dos rótulos de dados para 'center', o
355             que significa que os rótulos serão centralizados no meio dos
356             elementos de dados (por exemplo, no centro de uma barra no
357             gráfico de barras). */
358
359         align: 'center',
360             /* Define o alinhamento do texto dos rótulos para 'center',
```

...

OUTLINE

TIMELINE

Ln 331, Col 1 Spaces: 8 UTF-8 CRLF {} JavaScript ⚡ Port: 5500



EXPLORER

✓ P42 - GRÁFICO DE BARRAS HORIZONTAIS



 dados.xlsx

## # estilos.cs



<> index.htm



... 80 p.wj



`index.html`    `# estilos.css`    `js scripts.js`



> OUTLINE

> TIMELINE

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P42 - Gráfico de Barras Horizontais
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Print, Help, ...
- Sidebar (EXPLORER):** Shows a project structure under "P42 - GRÁFICO DE BARRAS HORIZONTAIS":
  - dados.xlsx
  - # estilos.css
  - index.html
  - scripts.js** (selected)
- Code Editor:** The "scripts.js" file is open, showing the following code:

```
index.html
# estilos.css
scripts.js

scripts.js > document.addEventListener('DOMContentLoaded') callback > criarGrafico > options > plugins > datalabels > formatter
sem modificação. Isso inclui todos os dados do segundo
conjunto de dados (metas restantes) e as vendas que não
excedem as metas. */

391
392
393
394
395
396
397
398
399
400
401     plugins: [ChartDataLabels]
402     /* Especifica um array de plugins que serão utilizados pelo gráfico.
403      Neste caso, 'ChartDataLabels' é incluído no array para ser
404      ativado no gráfico.
405      - 'ChartDataLabels' é o nome do plugin que foi configurado anteriormente
406      nas opções para mostrar rótulos de dados nas barras. Incluir
407      este plugin no array garante que ele será aplicado ao gráfico,
408      permitindo que os rótulos de dados sejam exibidos conforme
409      especificado nas configurações de 'datalabels'. */
410   });
411 }
412
413 // Inicializar leitura do arquivo Excel
414 lerArquivoExcel();
415
416 );
```
- Bottom Status Bar:** Shows file information: Ln 391, Col 4, Spaces: 8, UTF-8, CRLF, {}, JavaScript, Port : 5500.

**UC:** Projetos

**Assunto:**

Small Projects

P43 – Consulta de Endereço por CEP

## Consulta de Endereço por CEP

Digite o CEP

Buscar

EXPLORER

P43 - CONSULTA ENDEREÇO POR CEP

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

OUTLINE

TIMELINE

index.html # script.js

```
1 <!DOCTYPE html>
2 <!-- A linha acima define o tipo do documento como HTML5,
3     que é a versão mais recente do HTML. -->
4
5 <html lang="pt-BR">
6 <!-- A tag <html> inicia o documento HTML. O atributo 'lang'
7     especifica o idioma do conteúdo do documento, neste
8     caso, 'pt-BR' para português do Brasil. -->
9
10 <head>
11 <!-- A tag <head> contém metadados (dados sobre dados) que
12     não são exibidos diretamente na página web, mas são
13     importantes para o navegador. -->
14
15 <meta charset="UTF-8">
16 <!-- A tag <meta> com o atributo 'charset' define o conjunto
17     de caracteres usado no documento, neste caso, UTF-8, que
18     inclui a maioria dos caracteres de todos os idiomas escritos e
19     é uma boa prática para evitar problemas de exibição de texto. -->
20
21 <meta name="viewport" content="width=device-width, initial-scale=1.0">
22 <!-- Outra tag <meta>, desta vez configurando a 'viewport' (área
23     de visualização) da página. 'width=device-width' faz a largura da
24     página seguir a largura da tela do dispositivo, e 'initial-scale=1.0'
25     define o nível inicial de zoom quando a página é carregada pela
26     primeira vez. Essas configurações ajudam a garantir que o site
27     seja responsivo, ou seja, se ajuste bem em dispositivos móveis. -->
28
29 <title>Consulta de Endereço por CEP</title>
30 <!-- A tag <title> define o título da página, que é mostrado na aba
```

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P43 - Consulta Endereço por CEP
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Print, Help, ...
- Left Sidebar (EXPLORER):** Shows a project folder "P43 - CONSULTA ENDEREÇO POR CEP" containing files: index.html, script.js, and # styles.css.
- Code Editor:** The file "index.html" is open, showing the following code with explanatory comments:

```
<!DOCTYPE html>
<html>
    <head>
        do navegador. Este título também é usado por motores de busca e
        quando a página é favoritada. -->
        <link rel="stylesheet" href="# styles.css">
        <!-- A tag <link> é usada aqui para vincular um arquivo de folha de
            estilo CSS externo ao documento HTML. 'rel="stylesheet"' indica
            que o tipo de documento vinculado é uma folha de estilo e 'href="# styles.css"''
            especifica o caminho para o arquivo CSS que será aplicado
            para estilizar a página. -->
    </head>
    <!-- A tag </head> indica o fim da seção head do documento HTML. -->
    <body>
        <!-- A tag <body> contém todo o conteúdo da página que é
            visível para o usuário, como texto, imagens, links, etc. -->
        <div class="container">
            <!-- A tag <div> é um container genérico para conteúdo
                de fluxo, que por si só não representa nada, mas pode
                ser estilizado com CSS ou manipulado com JavaScript.
                A classe "container" é geralmente usada para definir um
                estilo específico que centraliza o conteúdo e limita sua
                largura dentro da página, facilitando o design responsivo. -->
            <h1>Consulta de Endereço por CEP</h1>
            <!-- A tag <h1> representa um cabeçalho de nível um, o mais
                importante na hierarquia de títulos de uma página.
                Usado aqui para dar título à função principal da
                página, que é a consulta de endereço por CEP. -->
        </div>
    </body>
</html>
```

The code editor also displays status information at the bottom: Line 31, Column 7, Spaces: 4, UTF-8, CRLF, HTML, Port: 5500.



EXPLORER

...

## P43 - CONSULTA ENDEREÇO POR CEP

index.html

script.js

# styles.css



&gt; OUTLINE

&gt; TIMELINE

index.html X # styles.css JS script.js

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

```
61      <div class="caixa-busca">
62          <!-- Outra &lt;div&gt;, desta vez com a classe "caixa-busca",
63              que pode ser estilizada especificamente para conter
64              elementos de formulário de busca. --&gt;
65
66          &lt;input type="text" id="campo-cep" placeholder="Digite o CEP" aria-label="Digite o CEP"&gt;
67          <!-- A tag &lt;input&gt; é usada para criar um elemento
68              interativo para o usuário inserir dados.
69              'type="text"' define o tipo de entrada como texto.
70              'id="campo-cep"' é um identificador único para o elemento,
71                  que pode ser referenciado em JavaScript e CSS.
72              'placeholder="Digite o CEP"' fornece um texto de exemplo que
73                  desaparece quando o campo é focado.
74              'aria-label="Digite o CEP"' melhora a acessibilidade, fornecendo
75                  uma descrição dos leitores de tela, indicando o que
76                  deve ser inserido no campo. --&gt;
77
78          &lt;button id="botao-buscar"&gt;Buscar&lt;/button&gt;
79          <!-- A tag &lt;button&gt; cria um botão clicável. Neste caso, ele é
80              usado para submeter a consulta de CEP.
81              'id="botao-buscar"' identifica o botão unicamente, permitindo
82                  mais controle com JavaScript ou CSS. --&gt;
83
84      &lt;/div&gt;
85
86
87      &lt;div id="resultado" class="resultado"&gt;
88          <!-- Uma &lt;div&gt; destinada a exibir os resultados da busca,
89              identificada por 'id="resultado"' para manipulações
90              específicas via JavaScript e 'class="resultado"' para</pre>
```



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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P43 - Consulta Endereço por CEP
- Icons:** Explorer, Search, Find, Split, Outline, Timeline.
- Left Sidebar:** EXPLORER, P43 - CONSULTA ENDEREÇO POR CEP (index.html, script.js, # styles.css).
- Active Tab:** # styles.css
- Code Content:** CSS rules for styling the application. The code includes rules for the entire document, body/html elements, and a general rule for all elements.

```
/* Aplica regras a todos os elementos no documento */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: Arial, sans-serif;
}

/* Aplica regras aos elementos <body> e <html> */
body, html {
    height: 100%;
    display: flex;
    align-items: center;
    justify-content: center;
}
```

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P43 - Consulta Endereço por CEP
- Icons:** Explorer, Search, Find, Open, Save, Outline, Timeline.
- Left Sidebar:** EXPLORER, P43 - CONSULTA ENDEREÇO POR CEP (index.html, script.js, # styles.css).
- Active Tab:** # styles.css
- Code Content:** CSS styles for index.html. The code defines styles for the body element, including horizontal and vertical centering, a blue background color (#007bff), and a container class. The container class ensures the element occupies 100% of its parent's width and has a maximum width of 800px. It also sets a white background color (#ffffff) and adds 20px of internal padding.
- Bottom Status Bar:** Line 31, Col 3, Spaces: 8, UTF-8, CRLF, CSS, Port: 5500.

```
index.html # styles.css script.js

# styles.css > body
31     e <html> horizontalmente ao centro */
32     justify-content: center;
33
34     /* Alinha os elementos filhos de <body>
35     e <html> verticalmente ao centro */
36     align-items: center;
37
38     /* Define a cor de fundo de <body> e <html> como azul */
39     background-color: #007bff;
40
41 }
42
43
44 /* Define estilos para elementos com a classe "container" */
45 .container {
46
47     /* Define a largura do elemento para ocupar 100% da
48     | largura do seu contêiner pai. */
49     width: 100%;
50
51     /* Define uma largura máxima para o elemento, garantindo
52     | que ele não ultrapasse 800 pixels.
53     Isso é útil para garantir que o layout não fique
54     | muito largo em telas grandes. */
55     max-width: 800px;
56
57     /* Define a cor de fundo do elemento para branco (#ffffff). */
58     background-color: #ffffff;
59
60     /* Adiciona um preenchimento interno de 20 pixels
```

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P43 - Consulta Endereço por CEP
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Print, Help, ...
- Sidebar (EXPLORER):** Shows the project structure:
  - P43 - CONSULTA ENDEREÇO POR CEP
    - index.html
    - script.js
    - # styles.css
- Code Editor:** The # styles.css file is open, containing the following CSS code:

```
/* Estilos para o contêiner centralizado com padding e border-radius */
.container {
    padding: 20px;
    border-radius: 8px;
    /* Aplica uma sombra abafada ao contêiner */
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
    /* Centraliza o texto dentro do contêiner */
    text-align: center;
}

/* Estilos aplicados à tag <h1> */
h1 {
    color: #000000;
}
```
- Bottom Status Bar:** Shows file status (0), line count (Ln 61, Col 4), spaces used (Spaces: 8), encoding (UTF-8), line endings (CRLF), file type (CSS), and port information (Port : 5500).

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P43 - Consulta Endereço por CEP
- Icons:** Explorer, Search, Find, Open, Close, Help, Outline, Timeline.
- File List (EXPLORER):** P43 - CONSULTA ENDEREÇO POR CEP (expanded), index.html, script.js, styles.css.
- Active Tab:** styles.css
- Code Content:** CSS styles for an application. The styles include:
  - For h1 element:
    - Line 91: border-radius: 8px;
    - Line 96: margin-bottom: 20px;
  - For .caixa-busca class:
    - Line 102: /\* Estilos aplicados a elementos com a classe "caixa-busca" \*/
    - Line 103: .caixa-busca {
    - Line 105: display: flex;
    - Line 109: justify-content: center;
    - Line 113: gap: 10px;
    - Line 117: margin-bottom: 40px;
- Bottom Status Bar:** Line 91, Col 4, Spaces: 8, UTF-8, CRLF, CSS, Port: 5500.

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P43 - Consulta Endereço por CEP
- Toolbar:** Icons for Explorer, Search, Find, Open, Save, and Close.
- Left Sidebar:** EXPLORER, P43 - CONSULTA ENDEREÇO POR CEP, index.html, script.js, # styles.css.
- Current File:** # styles.css (highlighted in grey)
- Code Editor:** Displays the following CSS code:

```
121 }  
122  
123  
124 /* Estilos aplicados a todos os elementos <input> do tipo texto */  
125 input[type="text"] {  
126     /* Define a largura do input como 70% da largura  
         do seu contêiner pai. */  
127     width: 70%;  
128  
129     /* Adiciona um preenchimento de 10 pixels em todos os  
         lados do input para melhorar a legibilidade e a estética. */  
130     padding: 10px;  
131  
132     /* Arredonda os cantos do input com um raio de 5 pixels  
         para um visual mais suave e moderno. */  
133     border-radius: 5px;  
134  
135     /* Aplica uma borda sólida com 2 pixels de espessura e cor cinza claro (#ccc),  
         ajudando a definir claramente os limites do campo de entrada. */  
136     border: 2px solid #ccc;  
137  
138     /* Define o tamanho da fonte do texto digitado no input para 20 pixels,  
         tornando-o facilmente legível para os usuários. */  
139     font-size: 20px;  
140  
141 }  
142  
143  
144 }  
145  
146  
147 }  
148  
149  
150 /* Estilos aplicados a todos os elementos <button> */
```

The code defines styles for text input fields, including width, padding, rounded corners, a solid border, and a font size of 20px. It also includes a closing brace for a previous rule and a comment for button elements.

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P43 - Consulta Endereço por CEP
- Sidebar (Left):**
  - EXPLORER
  - P43 - CONSULTA ENDEREÇO POR CEP
    - index.html
    - script.js
    - # styles.css
  - OUTLINE
  - TIMELINE
- Code Editor (Main Area):**
  - File: index.html, # styles.css, script.js
  - Current File: # styles.css
  - Content:

```
151 button {  
152     /* Adiciona um preenchimento de 10 pixels na parte  
153      | superior e inferior e 20 pixels nos lados,  
154      | garantindo um espaço adequado dentro do botão para o texto. */  
155     padding: 10px 20px;  
156  
157     /* Define a cor de fundo do botão para amarelo (#ffc107),  
158      | o que faz com que ele se destaque na página como um  
159      | elemento interativo importante. */  
160     background-color: #ffc107; /* Botão amarelo para destaque */  
161  
162     /* Define a cor do texto dentro do botão para um cinza escuro (#333),  
163      | garantindo um bom contraste com o fundo amarelo. */  
164     color: #333;  
165  
166     /* Remove qualquer borda do botão, optando por  
167      | um design limpo e moderno. */  
168     border: none;  
169  
170     /* Arredonda os cantos do botão com um raio de 5  
171      | pixels para um toque final estético. */  
172     border-radius: 5px;  
173  
174     /* Muda o cursor para uma mão quando o usuário passa o mouse sobre o botão,  
175      | indicando que o elemento é clicável. */  
176     cursor: pointer;  
177  
178 }  
179  
180 }
```
- Bottom Status Bar:** Ln 151, Col 1, Spaces: 8, UTF-8, CRLF, CSS, Port: 5500

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P43 - Consulta Endereço por CEP
- Left Sidebar (EXPLORER):** Shows a project structure under "P43 - CONSULTA ENDEREÇO POR CEP": index.html, script.js, and styles.css.
- Central Area:** A tab bar with index.html, # styles.css (selected), and JS script.js.
- Code Editor Content:** The "# styles.css" file contains the following CSS code:

```
181  /* Estilos aplicados ao estado hover de todos os
182   | elementos <button> */
183 button:hover {
184
185   /* Altera a cor de fundo do botão para um tom de azul
186   | mais escuro (#210766) quando o mouse está sobre ele.
187   | Isso fornece um feedback visual imediato ao usuário,
188   | indicando que o botão é interativo. */
189   background-color: #210766;
190
191   /* cor da fonte como branco */
192   color: #ffffff;
193
194 }
195
196
197 /* Estilos aplicados a elementos com a classe "resultado" */
198 .resultado {
199
200   /* Define a cor de fundo para um cinza muito claro (#f8f9fa),
201   | que ajuda a diferenciar a área de resultados
202   | do restante da interface, proporcionando uma leitura
203   | visual clara dos resultados apresentados. */
204   background-color: #f8f9fa; /* Fundo claro para resultados */
205
206   /* Adiciona um preenchimento de 10 pixels em todos os
207   | lados dentro do contêiner de resultados,
208   | criando espaço entre o conteúdo interno e as
209   | bordas do contêiner. */
210   padding: 10px;
```

The code is styled with green numbers for line numbers, blue for CSS properties, and orange for comments. The selected file is "# styles.css".

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P43 - Consulta Endereço por CEP
- Toolbar:** Icons for Explorer, Search, Find, Open, Save, and others.
- Left Sidebar (EXPLORER):** Shows the project structure: P43 - CONSULTA ENDEREÇO POR CEP with files index.html, script.js, and styles.css.
- Current File:** styles.css (selected tab)
- Code Content:** CSS rules for styling results. The selected rule is line 211:

```
211 # resultado {
212     /* Arredonda os cantos do contêiner de resultados com um
213      | raio de 5 pixels, suavizando a aparência visual. */
214     border-radius: 5px;
215
216     /* Define a largura do contêiner de resultados para
217      | ocupar 100% da largura do seu contêiner pai,
218      | assegurando que ele se estenda completamente na
219      | área disponível horizontalmente. */
220     width: 100%;
221
222     /* Ajusta o tamanho da fonte do texto dentro do contêiner
223      | de resultados para 20 pixels,
224      | garantindo que os dados sejam grandes o suficiente
225      | para serem lidos confortavelmente. */
226     font-size: 20px;
227
228 }
```

**Bottom Status Bar:** Icons for file operations (New, Open, Save, etc.), status (Ln 211, Col 3, Spaces: 8, UTF-8, CRLF), and port (Port : 5500).

EXPLORER

...

▼ P43 - CONSULTA ENDEREÇO POR CEP

↳ index.html

JS script.js

# styles.css

...

OUTLINE

TIMELINE

index.html # styles.css JS script.js X

JS script.js > ...

```
1 // Acessa o elemento da página HTML com o ID 'botao-buscar' e
2 // adiciona um ouvinte de evento que responde a cliques neste botão.
3 document.getElementById('botao-buscar').addEventListener('click', function() {
4
5     // Inicia acessando o elemento do HTML pelo seu ID 'campo-cep'.
6     // O ID é um identificador único usado para localizar e retornar o elemento para o script.
7     var cep = document.getElementById('campo-cep').value.trim();
8
9     // ↑ Declaração da variável 'cep'. Aqui, várias ações estão sendo realizadas:
10    // 1. `document.getElementById('campo-cep')`: Acessa o elemento de
11    //      // entrada onde os usuários digitam o CEP.
12    //      - 'document' é o objeto que representa todo o documento HTML.
13    //      - ` getElementById()` é um método que retorna o elemento que
14    //          // possui o ID especificado no documento.
15    //      - 'campo-cep' é o ID do elemento de entrada HTML que estamos acessando.
16    // 2. `.value`: Esta propriedade retorna o valor atual do campo de
17    //      // entrada (o que o usuário digitou).
18    // 3. `trim()`: Este método é chamado no valor retornado. `trim()` é
19    //      // um método de String em JavaScript que remove espaços em
20    //      // branco do início e do fim da string.
21    //      - Espaços em branco incluem espaços, tabs, saltos de linha e outros
22    //          // caracteres de espaço invisíveis.
23    //      - A utilização de `trim()` é importante para assegurar que
24    //          // espaços extras não afetem a consulta ao CEP, como por
25    //          // exemplo, um usuário acidentalmente digitando espaços
26    //          // antes ou depois dos números.
27
28    // O valor processado (CEP sem espaços extras) é armazenado na
29    // variável 'cep', que agora contém a string limpa para ser
30    // usada na solicitação à API.
```

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P43 - Consulta Endereço por CEP
- Icons:** Explorer, Search, Open, Find, Outline, Timeline.
- Left Sidebar (EXPLORER):** P43 - CONSULTA ENDEREÇO POR CEP (index.html, script.js, styles.css)
- Active Tab:** script.js
- Code Content:** A script for address lookup by CEP using the ViaCEP API. The code includes comments in Portuguese explaining the logic, such as checking if a CEP is provided and handling errors if the CEP is not found.
- Bottom Status Bar:** Line 31, Column 1, Spaces: 4, UTF-8, CRLF, {} JavaScript, Port: 5500, Bell icon.

```
// A estrutura condicional 'if' verifica se a variável 'cep' contém algum valor.
if(cep) {

    // Se 'cep' não está vazio, faz uma solicitação à API do ViaCEP
    // usando 'fetch', que é uma função que permite fazer solicitações
    // de rede, como acessar uma URL.
    fetch(`https://viacep.com.br/ws/${cep}/json/`)

        // Após receber a resposta, converte os dados de JSON
        // para um objeto JavaScript.
        .then(resposta => resposta.json())

        // Com os dados convertidos, esse bloco de código é executado.
        .then(dados => {

            // Verifica se os dados recebidos contêm a propriedade 'erro'.
            // Isso acontece se o CEP não for encontrado.
            if(dados.erro) {

                // A condição 'if' checa se a propriedade 'erro' existe no objeto 'dados'.
                // Se 'dados.erro' é verdadeiro, significa que a API retornou
                // um erro, indicando que o CEP fornecido não foi encontrado.

                // Se o CEP não for encontrado, atualiza o conteúdo do elemento
                // com ID 'resultado' para mostrar uma mensagem de erro.
                document.getElementById('resultado').innerHTML = '<p>CEP não encontrado.</p>';

                // ↑ Utiliza o método 'getElementById' para selecionar o elemento HTML
            }
        })
}
```

File Edit Selection View Go Run ... ← → P43 - Consulta Endereço por CEP

EXPLORER

P43 - CONSULTA ENDEREÇO POR CEP

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

script.js

```
JS script.js > ⚡ addEventListener('click') callback > ⚡ then() callback
```

```
61 // cujo ID é 'resultado'.
62 // A propriedade 'innerHTML' desse elemento é então definida para
63 // incluir um parágrafo com o texto "CEP não encontrado."
64 // 'innerHTML' é uma forma de manipular o conteúdo HTML de um
65 // elemento diretamente, permitindo inserir novos elementos HTML ou
66 // alterar o conteúdo existente.
67 // A mensagem de erro serve para informar ao usuário que o CEP
68 // digitado não retornou resultados válidos.
69
70 } else {
71
72     // Se não houver erro, o conteúdo do elemento 'resultado' é
73     // atualizado com detalhes do endereço.
74     document.getElementById('resultado').innerHTML =
75         <p><strong>Endereço:</strong> ${dados.logradouro}</p>
76         <p><strong>Bairro:</strong> ${dados.bairro}</p>
77         <p><strong>Cidade:</strong> ${dados.localidade}</p>
78         <p><strong>Estado:</strong> ${dados.uf}</p>
79         <p><strong>CEP:</strong> ${dados.cep}</p>
80     ;
81
82     // Se a propriedade 'erro' não estiver presente no objeto
83     // 'dados', significa que um endereço válido foi retornado.
84     // A propriedade 'innerHTML' do mesmo elemento 'resultado' é
85     // usada novamente, desta vez para apresentar os detalhes do endereço.
86     // Os detalhes incluem logradouro, bairro, cidade, estado e o CEP consultado.
87     // Cada um desses detalhes é acessado diretamente do objeto 'dados'
88     // que foi convertido de JSON e inserido em elementos HTML para melhor apresentação.
89     // Esta parte do código é crucial para exibir ao usuário as
90     // informações retornadas pela API de forma clara e estruturada.
```

OUTLINE

TIMELINE

Ln 61, Col 1 Spaces: 4 UTF-8 CRLF {} JavaScript ⚡ Port : 5500

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

- File Bar:** File Edit Selection View Go Run ...
- Search Bar:** P43 - Consulta Endereço por CEP
- Toolbars:** Standard icons for file operations (New, Open, Save, Print, Find, Replace, etc.)
- Sidebar (Left):**
  - EXPLORER
  - P43 - CONSULTA ENDEREÇO POR CEP
    - index.html
    - script.js (selected)
    - styles.css
- Code Editor (Main Area):**
  - File: script.js
  - Content:

```
91      }
92      }
93      })
94
95      // Em caso de qualquer erro na solicitação de rede, este bloco será executado.
96      .catch(() => {
97
98          // Atualiza o elemento 'resultado' para mostrar uma mensagem
99          // indicando que ocorreu um erro ao buscar o CEP.
100         document.getElementById('resultado').innerHTML = '<p>Erro ao buscar o CEP.</p>';
101
102     });
103 });
104 });
105
106
```
  - Callout: A callout box highlights the code block from line 91 to line 104, indicating the execution flow from an event listener to a catch block.
- Bottom Status Bar:** Includes icons for file operations, a gear icon for settings, and status indicators for outline, timeline, and ports.
- Bottom Navigation:** Icons for file operations (New, Open, Save, Print, Find, Replace, etc.)

# **UC: Projetos**

## **Assunto:**

### Small Projects

### P44 – Leitor de QR Code

## Leitor de QR Code

Selecionar Arquivo

File Edit Selection View Go Run ... ← → P44 - Leitor de QR Code

EXPLORER ...

P44 - LEITOR DE QR CODE

- # estilos.css
- index.html
- qrcode.png
- qrcode+(1).png
- qrcode+(2).png
- qrcode+(3).png
- script.js

index.html X qrcode.png # estilos.css JS script.js

index.html > ...

```
1 <!DOCTYPE html>
2 <!-- A linha <!DOCTYPE html> é uma declaração (não uma tag)
3 que informa ao navegador a versão do HTML que a
4 página está escrita. Neste caso, especifica HTML5.
5 Esta declaração é necessária para ativar o modo de
6 padrões completos do navegador, garantindo que o HTML
7 seja interpretado de acordo com as especificações do HTML5. -->
8
9 <html lang="pt-BR">
10 <!-- A tag <html> é o elemento raiz de um documento HTML,
11     contendo todos os outros elementos (exceto o <!DOCTYPE>).
12     O atributo 'lang' especifica o idioma do conteúdo do
13     documento, que é Português do Brasil (pt-BR) neste caso.
14     Isso é útil para acessibilidade, SEO e navegadores que
15     podem oferecer tradução automática de páginas. -->
16
17 <head>
18 <!-- A tag <head> contém metadados (informações sobre os dados) que
19     não são exibidos diretamente na página web, mas são usados
20     pelo navegador para processar a página. Inclui links para
21     scripts, instruções de estilo, metadados e outras informações. -->
22
23 <meta charset="UTF-8">
24 <!-- A tag <meta> com o atributo 'charset' define o conjunto de
25     caracteres usado no documento, que é 'UTF-8' neste caso. UTF-8 é
26     um padrão de codificação de caracteres que inclui quase
27     todos os caracteres de todos os idiomas do mundo, garantindo
28     que o texto seja exibido corretamente independentemente do
29     idioma ou plataforma. -->
30
```

Ln 1, Col 1 Spaces: 8 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:** P44 - Leitor de QR Code
- Sidebar:** EXPLORER, P44 - LEITOR DE QR CODE (containing # estilos.css, index.html, qrcode.png, qrcode+(1).png, qrcode+(2).png, qrcode+(3).png, script.js), OUTLINE, and TIMELINE.
- Code Editor:** The main area displays the content of index.html under the head section. The code is annotated with comments explaining its purpose.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<!-- Outra tag &lt;meta&gt; que controla a janela de visualização (viewport) do navegador. O 'content' desta tag é definido para garantir que a largura da janela de visualização seja igual à largura do dispositivo (width=device-width) e que o nível de zoom inicial seja 1 (initial-scale=1.0). Isso é crucial para otimizar sua página para dispositivos móveis, garantindo que a página seja exibida de forma adequada em diferentes tamanhos de tela. --&gt;

&lt;title&gt;Leitor de QR Code&lt;/title&gt;
<!-- A tag &lt;title&gt; define o título da página, que é mostrado na aba do navegador e é usado pelos motores de busca. Aqui, o título é 'Leitor de QR Code', que descreve a funcionalidade principal da página web. --&gt;

&lt;link rel="stylesheet" href="estilos.css"&gt;
<!-- A tag &lt;link&gt; é usada aqui para vincular uma folha de estilo externa, especificada pelo atributo 'href'. Este link aponta para o arquivo 'estilos.css', que contém as definições de estilo CSS para a página. O 'rel' atributo especifica o relacionamento entre o documento atual e o recurso vinculado, que neste caso é uma folha de estilo ('stylesheet'). --&gt;

&lt;/head&gt;
<!-- Fim da tag &lt;head&gt;, indicando que todos os metadados necessários foram incluídos. --&gt;</pre>

Bottom Status Bar: X 0 △ 0 ⌂ 0 Q Ln 31, Col 3 Spaces: 8 UTF-8 CRLF HTML ⌂ Go Live ⌂


```

File Edit Selection View Go Run ... ← → P44 - Leitor de QR Code

EXPLORER ...

P44 - LEITOR DE QR CODE

- # estilos.css
- index.html
- qrkode.png
- qrkode+(1).png
- qrkode+(2).png
- qrkode+(3).png
- script.js

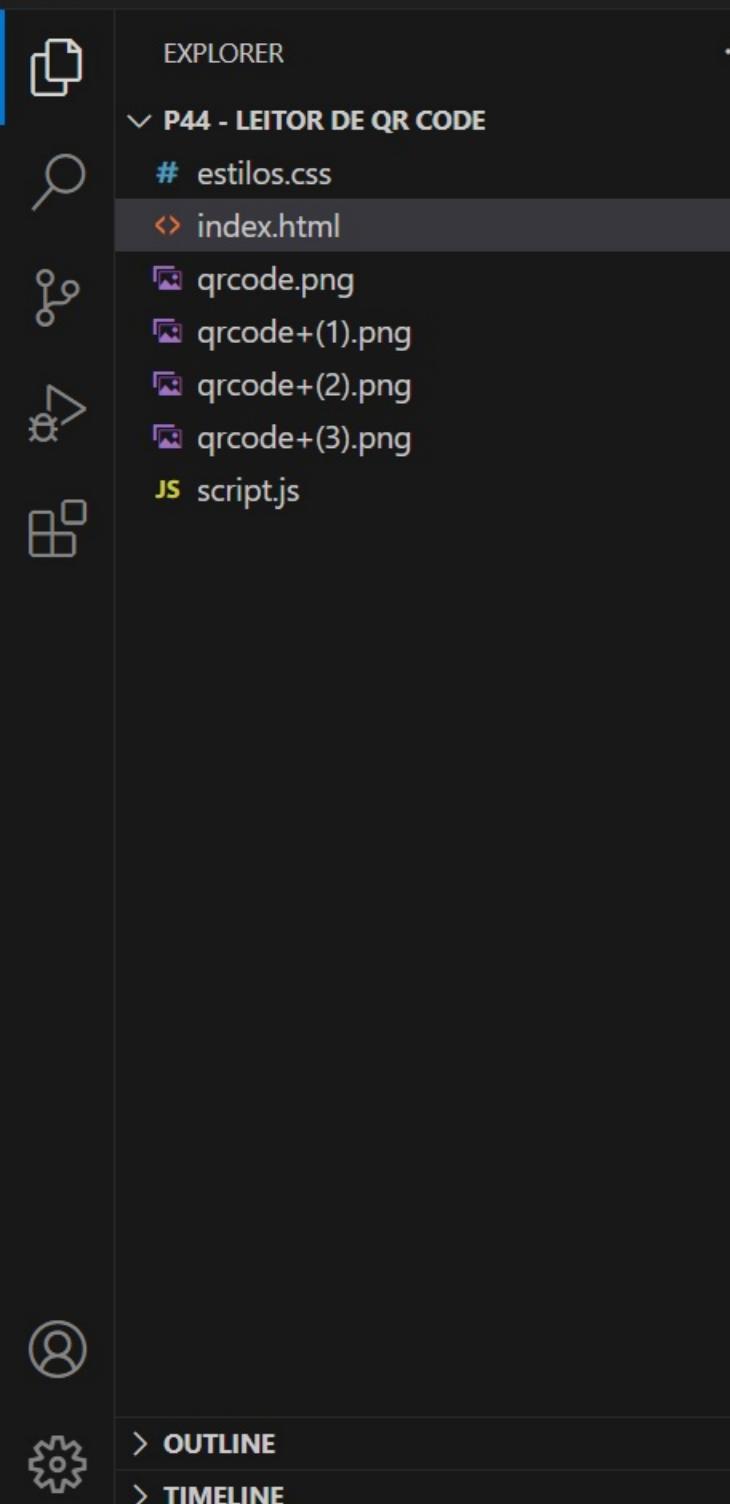
index.html X qrkode.png # estilos.css JS script.js

index.html > html > body

```
61 <body>
62     <!-- A tag &lt;body&gt; marca o início do corpo do documento HTML, que
63         contém todo o conteúdo visível da página como texto,
64         hiperlinks, imagens, tabelas, etc. --&gt;
65
66     &lt;div class="container"&gt;
67         <!-- A tag &lt;div&gt; com a classe 'container' serve como um contêiner
68             para agrupar elementos relacionados. Aqui, agrupa todos os
69             elementos da interface do leitor de QR Code. As propriedades
70             de estilo são definidas pela classe para padronizar a
71             aparência e o layout. --&gt;
72
73     &lt;h1&gt;Leitor de QR Code&lt;/h1&gt;
74         <!-- A tag &lt;h1&gt; define o cabeçalho principal da página. É usado
75             para descrever o tópico ou propósito da página, neste caso,
76             indicando que a página é um 'Leitor de QR Code'. Os
77             cabeçalhos também são importantes para SEO (Search
78             Engine Optimization). --&gt;
79
80     &lt;label for="selecionarArquivo" class="botao-upload"&gt;Selecionar Arquivo&lt;/label&gt;
81         <!-- A tag &lt;label&gt; associa-se ao elemento &lt;input&gt; por meio do
82             atributo 'for', que deve corresponder ao 'id' do &lt;input&gt;.
83             Isso melhora a acessibilidade, permitindo que os usuários
84             cliquem no rótulo para ativar o controle de entrada.
85             A classe 'botao-upload' é usada para aplicar estilos
86             específicos. --&gt;
87
88     &lt;input type="file" id="selecionarArquivo" accept="image/*" style="display:none;"&gt;
89         <!-- A tag &lt;input&gt; do tipo 'file' permite que os usuários escolham um
90             ou mais arquivos de seus dispositivos de armazenamento para</pre>

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


```



```
< index.html > qrcode.png # estilos.css JS script.js

< index.html > html > body > div.container
91     serem carregados ou processados. O atributo 'accept'
92     limita os tipos de arquivos que podem ser selecionados
93     para qualquer tipo de imagem. O estilo 'display:none;'
94     esconde o <input> real e utiliza um <label> para
95     interação visual. -->
96
97     <canvas id="canvas" style="display:none;"></canvas>
98     <!-- A tag <canvas> é usada para desenhar gráficos via scripting.
99         Aqui, é utilizada para processar a imagem do QR Code. Está
100        oculta inicialmente com 'display:none;' até que
101        seja necessário. -->
102
103    <div id="resultado"></div>
104    <!-- Uma <div> com o id 'resultado' é usada para exibir o resultado
105        do processamento do QR Code. Inicialmente, não tem conteúdo
106        até que um QR Code seja lido e processado. -->
107
108    </div>
109    <!-- Fim da <div> 'container'. -->
110
111    <script src="https://cdn.jsdelivr.net/npm/jsqr@1.4.0/dist/jsQR.min.js"></script>
112    <!-- Inclui o script jsQR, uma biblioteca JavaScript para decodificação
113        de QR Codes. O script é carregado do CDN, garantindo alta
114        disponibilidade e desempenho. -->
115
116    <script src="script.js"></script>
117    <!-- Referencia um arquivo JavaScript externo chamado 'script.js',
118        que contém a lógica específica para o funcionamento da
119        página de leitura de QR Code. Este script manipula a
120        interacão do usuário e a lógica de processamento do QR Code. -->
```

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P44 - Leitor de QR Code
- Icons:** Explorer, Search, Find, Open, Save, Outline, Timeline.
- Left Sidebar:** EXPLORER, P44 - LEITOR DE QR CODE (containing # estilos.css, <> index.html, qrkode.png, qrkode+(1).png, qrkode+(2).png, qrkode+(3).png, JS script.js), OUTLINE, and TIMELINE.
- Central Area:** A code editor window titled "index.html".
  - File Structure: index.html > html > body
  - Content:

```
121
122 </body>
123 <!-- Fim da tag <body>, que encerra o conteúdo visível da página. -->
124
125 </html>
126 <!-- Fim da tag <html>, que encerra o documento HTML. -->
```
- Bottom Status Bar:** < 0 △ 0 ⌂ 0, Line 121, Column 1, Spaces: 8, UTF-8, CRLF, HTML, Port : 5500, 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:** P44 - Leitor de QR Code
- Toolbar:** Minimize, Maximize, Close, Fullscreen, ...
- Left Sidebar (EXPLORER):**
  - P44 - LEITOR DE QR CODE
    - # estilos.css
    - index.html
    - qrcode.png
    - qrcode+(1).png
    - qrcode+(2).png
    - qrcode+(3).png
  - script.js
- Current File:** # estilos.css
- Code Content:** CSS styles for the body element, defining font-family, display, justify-content, align-items, height, and comments explaining the purpose of each rule.
- Bottom Status Bar:** X, 0 0 0, Q, Ln 1, Col 1, Spaces: 4, UTF-8, CRLF, CSS, Port : 5500, Bell

```
body {  
    font-family: Arial, sans-serif;  
    /* Define as famílias de fontes do corpo do documento  
       para Arial como a primeira escolha e sans-serif  
       como fallback. Arial é uma fonte sem serifa,  
       amplamente disponível e legível, enquanto sans-serif é  
       um genérico para qualquer fonte sem serifa caso o  
       Arial não esteja disponível. */  
    display: flex;  
    /* Aplica o modelo de layout Flexbox ao corpo do documento.  
       O Flexbox é um método de layout que permite espaçar e  
       alinhar itens de forma eficiente, mesmo em tamanhos  
       de container dinâmicos. */  
    justify-content: center;  
    /* Alinha os itens filhos do body (neste caso, a <div>  
       container) horizontalmente ao centro do elemento  
       pai (body), proporcionando um alinhamento centralizado  
       no eixo principal. */  
    align-items: center;  
    /* Alinha os itens filhos do body verticalmente ao centro do  
       elemento pai. Isso garante que o conteúdo dentro de <body>  
       esteja centralizado tanto vertical quanto horizontalmente. */  
    height: 100vh;  
    /* Define a altura do <body> para ser 100vh (viewport height), o  
       que significa 100% da altura da janela de visualização. */
```

File Edit Selection View Go Run ... ← → P44 - Leitor de QR Code

EXPLORER ...

P44 - LEITOR DE QR CODE

- # estilos.css
- index.html
- qrcode.png
- qrcode+(1).png
- qrcode+(2).png
- qrcode+(3).png
- script.js

# estilos.css > body

```
31     Isso ajuda a garantir que o conteúdo ocupe toda a
32     altura visível da tela, criando uma experiência de
33     tela cheia. */
34
35     background-color: #e0f7fa;
36     /* Define a cor de fundo do corpo do documento para um azul
37     claro (#e0f7fa). Isso fornece um plano de fundo suave
38     que não distrai, aprimorando a legibilidade e a
39     estética geral da página. */
40
41     margin: 0;
42     /* Remove todas as margens externas do <body>, garantindo que
43     não haja espaço extra entre as bordas do navegador e o
44     conteúdo da página, maximizando o uso do espaço disponível. */
45
46 }
47
48
49 .container {
50
51     background-color: #ffffff;
52     /* Define a cor de fundo do elemento com a classe 'container'
53     para branco (#ffffff). A escolha do branco proporciona um
54     fundo limpo e neutro que não distrai, favorecendo a
55     visibilidade de outros elementos dentro do container. */
56
57     padding: 20px;
58     /* Aplica um preenchimento de 20 pixels em todas as direções
59     dentro do container. Isso cria espaço entre o conteúdo
60     interno e as bordas do container, melhorando a
```

Ln 31, Col 2 Spaces: 4 UTF-8 CRLF CSS ⚡ Port : 5500

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

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P44 - Leitor de QR Code
- Icons:** Explorer, Search, Find, Open, Save, Outline, Timeline.
- Project Explorer:** P44 - LEITOR DE QR CODE (expanded) containing files: # estilos.css, index.html, qrcode.png, qrcode+(1).png, qrcode+(2).png, qrcode+(3).png, script.js.
- Code Editor:** The # estilos.css file is open, showing CSS code for a container element. The code includes comments explaining the purpose of each rule, such as border-radius, box-shadow, text-align, max-width, and width.
- Status Bar:** Includes icons for file operations (Save, Undo, Redo), status (0 0 0), and system information (Ln 61, Col 1, Spaces: 4, UTF-8, CRLF, CSS, Port: 5500).

```
index.html    qrcode.png  # estilos.css  script.js

# estilos.css > .container
61     legibilidade e a estética visual do conteúdo. */
62
63     border-radius: 10px;
64     /* Arredonda as bordas do container com um raio de 10 pixels.
65      Este efeito suaviza as bordas retas padrão do
66      container, dando-lhe uma aparência mais suave e moderna. */
67
68     box-shadow: 0 0 10px rgba(0, 0, 0, 0.5);
69     /* Aplica uma sombra ao redor do container. O valor '0 0 10px
70      rgba(0, 0, 0, 0.5)' significa que a sombra não tem
71      deslocamento horizontal ou vertical, tem um desfoco
72      de 10 pixels, e uma cor preta com 50% de transparência.
73      Isso cria um efeito de profundidade, destacando o
74      container do resto da página. */
75
76     text-align: center;
77     /* Alinha o texto e, por extensão, todos os elementos inline
78      dentro do container, ao centro. Isso é particularmente
79      útil para centralizar cabeçalhos, botões e outros
80      controles dentro do container. */
81
82     max-width: 800px;
83     /* Define a largura máxima do container como 800 pixels.
84      Isso significa que o container não pode se estender
85      além de 800 pixels, independentemente do tamanho da
86      tela, o que ajuda a manter a legibilidade em
87      telas maiores. */
88
89     width: 100%;
90     /* Define a largura do container para ser 100% do seu
```

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P44 - Leitor de QR Code
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Split, ...
- Left Sidebar (EXPLORER):**
  - P44 - LEITOR DE QR CODE
    - # estilos.css
    - index.html
    - qrcode.png
    - qrcode+(1).png
    - qrcode+(2).png
    - qrcode+(3).png
  - script.js
- Central Area:** Code Editor showing `estilos.css`. The code defines styles for an `h1` element and a `.botao-upload` class.

```
index.html      qrcode.png  # estilos.css  script.js
# estilos.css > .container
# estilos.css > .container
elemento pai. Isso garante que o container se
adapte à largura do espaço disponível, até o
máximo definido pela propriedade 'max-width'. */
91
92
93
94
95 }
96
97
98 h1 {
99
100    color: #00796b;
101   /* Define a cor do texto para os elementos <h1> como um
102      tom específico de verde azulado (#00796b).
103      Esta cor pode ajudar a estabelecer uma
104      identidade visual consistente para a página e é
105      suficientemente escura para garantir boa
106      legibilidade contra um fundo claro. */
107
108 }
109
110
111 .botao-upload {
112
113   display: inline-block;
114   /* Altera o método de exibição do elemento para 'inline-block',
115      o que permite que o elemento tenha propriedades de
116      bloco (como padding e width), mas seja disposto na
117      linha como um elemento inline. Isso é útil para
118      permitir que os elementos se alinhem em uma linha
119      enquanto mantêm propriedades de dimensionamento
120      de bloco. */
```

- Bottom Bar:** Icons for Save, Undo, Redo, Find, Replace, etc., and status information: Ln 91, Col 1, Spaces: 4, UTF-8, CRLF, CSS, Port: 5500, and a bell icon.



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

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P44 - Leitor de QR Code
- Explorer:** Shows files in the project: # estilos.css, index.html, qrcode.png, script.js.
- Current File:** # estilos.css (highlighted)
- Code Editor Content:**

```
151  /* Define a cor do texto dentro do elemento para branco (#ffffff),  
152   garantindo alto contraste com o fundo verde azulado  
153   para excelente legibilidade. */  
154  
155   margin: 20px 0;  
156   /* Aplica uma margem de 20 pixels acima e abaixo do elemento,  
157   ajudando a separá-lo de outros conteúdos na página e  
158   melhorando o layout geral. */  
159  
160 }  
161  
162  
163 #resultado {  
164  
165   margin-top: 20px;  
166   /* Define a margem superior do elemento com o ID 'resultado'  
167   como 20 pixels. Isso cria um espaço entre o  
168   elemento 'resultado' e quaisquer elementos que  
169   estejam acima dele na página, ajudando a separar  
170   visualmente o conteúdo e a melhorar o layout geral. */  
171  
172   font-size: 48px;  
173   /* Estabelece o tamanho da fonte para os textos dentro do  
174   elemento 'resultado' como 48 pixels. Esse tamanho  
175   grande torna o texto altamente visível e legível,  
176   adequado para exibir informações importantes como o  
177   resultado da leitura de um QR Code. */  
178  
179   color: #004d40;  
180   /* Define a cor do texto para um tom escuro de verde (#004d40).
```
- Bottom Status Bar:** 0 0 0 0, Ln 151, Col 2, Spaces: 4, UTF-8, CRLF, CSS, Port: 5500,

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P44 - Leitor de QR Code
- Toolbar:** Icons for Explorer, Search, Find, Open, Save, and others.
- Left Sidebar (EXPLORER):** Shows files in the project: # estilos.css, index.html, qrcode.png, qrcode+(1).png, qrcode+(2).png, qrcode+(3).png, and script.js.
- Current File:** # estilos.css (highlighted)
- Code Editor:** Displays the following CSS code:

```
# estilos.css > #resultado  
181     Esta cor não apenas se harmoniza com a paleta de  
182     cores geral do site, mas também oferece excelente  
183     contraste contra um fundo claro, melhorando a  
184     acessibilidade e a legibilidade. */  
185  
186  
187 }
```
- Bottom Status Bar:** Icons for file status (0), search (0), and other metrics (0), Ln 181, Col 5, Spaces: 4, UTF-8, CRLF, CSS, Port : 5500, and a bell icon.



EXPLORER

...

index.html

qrCode.png

# estilos.css

JS script.js

X

...

## P44 - LEITOR DE QR CODE

# estilos.css

&lt; index.html

qrCode.png

qrCode+(1).png

qrCode+(2).png

qrCode+(3).png

JS script.js

JS script.js &gt; ...

```
1 // Adiciona um ouvinte de evento 'change' ao elemento de
2 // entrada de arquivo com o ID 'selecionarArquivo'.
3 document.getElementById('selecionarArquivo').addEventListener('change', function(evento) {
4     // Esta linha de código seleciona o elemento DOM com o
5     // ID 'selecionarArquivo', que é um campo de entrada
6     // para arquivos. Em seguida, um ouvinte de evento é
7     // adicionado a este elemento. O tipo de evento que
8     // estamos ouvindo é 'change', que é disparado quando o
9     // valor do campo de entrada (neste caso, o arquivo
10    // selecionado pelo usuário) muda. A função passada como
11    // segundo argumento será chamada sempre que esse evento ocorrer.
12
13    // Armazena o primeiro arquivo selecionado pelo usuário na variável 'arquivo'.
14    // Aqui, 'evento.target' se refere ao elemento DOM que
15    // disparou o evento, ou seja, nosso campo de
16    // entrada de arquivo. 'files' é uma propriedade desse
17    // elemento que contém uma lista dos arquivos selecionados.
18    // '[0]' acessa o primeiro arquivo dessa lista.
19    // A variável 'arquivo' agora contém uma referência ao arquivo
20    // selecionado pelo usuário, permitindo que seu conteúdo
21    // seja acessado e manipulado pelo script.
22    const arquivo = evento.target.files[0];
23
24    // Se não houver arquivo selecionado, encerra a função prematuramente.
25    // Esta linha verifica se a variável 'arquivo' é falsa, o que
26    // seria o caso se nenhum arquivo estivesse selecionado.
27    // Se for verdadeiro que 'arquivo' não existe (ou seja, é falsa),
28    // a palavra-chave 'return' encerra a execução da função
29    // imediatamente. Isso é uma medida de segurança para evitar
30    // erros ao tentar ler um arquivo inexistente.
```



&gt; OUTLINE

&gt; TIMELINE



⊗ 0 △ 0 ⌂ 0

⊗ 0 △ 0 ⌂ 0



Ln 1, Col 1

Spaces: 8

UTF-8

CRLF

{ } JavaScript

⊗ Port: 5500



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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P44 - Leitor de QR Code
- File Explorer:** Shows files in the current project:
  - # estilos.css
  - <> index.html
  - qrkode.png
  - # estilos.css
  - JS script.js
- Code Editor:** Displays the content of script.js. The code uses JavaScript to read a file selected from an input field and display its content as a QR code image.

```
index.html qrkode.png # estilos.css JS script.js X

JS script.js > addEventListener('change') callback
31     if (!arquivo) return;
32
33     // Cria um novo objeto FileReader, que permite a leitura do
34     // conteúdo de arquivos armazenados no dispositivo do usuário.
35     // 'FileReader' é uma classe JavaScript que fornece
36     // funcionalidades para ler o conteúdo de arquivos Blob e
37     // File (como os arquivos selecionados num campo de
38     // entrada de arquivo). Ao instanciar 'FileReader', criamos um
39     // novo objeto 'leitor' que pode ser usado para iniciar a
40     // leitura do arquivo e definir funções de callback que
41     // reagem aos resultados dessa leitura (por exemplo,
42     // quando a leitura é concluída ou ocorre um erro).
43 const leitor = new FileReader();
44
45
46 // Define uma função que será chamada quando o
47 // FileReader concluir a leitura do arquivo.
48 leitor.onload = function(evento) {
49
50     // Cria um novo objeto Image, que será usado para
51     // carregar e renderizar a imagem.
52     const img = new Image();
53
54     // Define uma função que será chamada quando a
55     // imagem for carregada com sucesso.
56     // Esta linha estabelece uma função de callback para o
57     // evento 'load' do objeto 'img'. Esse evento é
58     // disparado quando a imagem termina de carregar
59     // completamente. Isso é crucial porque a imagem precisa
60     // ser completamente carregada antes de qualquer processamento.
```

- Bottom Status Bar:** Shows file count (0), line count (Ln 31, Col 5), spaces (Spaces: 8), encoding (UTF-8), line endings (CRLF), file type ({} JavaScript), port (Port: 5500), and a refresh icon.

File Edit Selection View Go Run ... ← → P44 - Leitor de QR Code

EXPLORER ... index.html qrcode.png # estilos.css JS script.js X

✓ P44 - LEITOR DE QR CODE

# estilos.css  
index.html  
qrcode.png  
qrcode+(1).png  
qrcode+(2).png  
qrcode+(3).png  
JS script.js

JS script.js > addEventListener('change') callback > onload

```
61     img.onload = function() {  
62         // Obtém o elemento <canvas> pelo seu ID 'canvas'.  
63         // Acessa o elemento <canvas> do DOM usando seu ID.  
64         // O canvas é usado para manipulação gráfica, como  
65             // desenhar e processar imagens.  
66         const canvas = document.getElementById('canvas');  
67  
68         // Obtém o contexto 2D do canvas, que permite  
69             // realizar operações de desenho.  
70         // 'getContext('2d')' é um método do elemento <canvas> que  
71             // retorna um contexto de desenho no canvas, ou  
72                 // seja, um objeto que fornece funções e propriedades  
73                     // para desenhar e manipular gráficos no canvas.  
74         // Aqui, estamos trabalhando com gráficos 2D.  
75         const contexto = canvas.getContext('2d');  
76  
77         // Define a largura e altura do canvas com base nas dimensões da imagem carregada.  
78         // Estas linhas ajustam a largura e a altura do canvas para  
79             // serem iguais às da imagem carregada. Isso garante que a  
80                 // imagem seja desenhada no canvas sem distorção de escala.  
81         canvas.width = img.width;  
82         canvas.height = img.height;  
83  
84         // Desenha a imagem no canvas, ocupando todo o espaço disponível.  
85         // O método 'drawImage' do contexto 2D desenha a imagem 'img' no  
86             // canvas. Os parâmetros '0, 0' definem a posição inicial no  
87                 // canvas onde a imagem começará a ser desenhada,  
88                     // e 'img.width, img.height' são as dimensões da imagem,  
89                         // garantindo que ela ocupe todo o canvas.
```

OUTLINE  
TIMELINE

✖ 0 △ 0 ⌂ 0

Q Ln 61, Col 9 Spaces: 8 UTF-8 CRLF {} JavaScript ⚡ Port : 5500

File Edit Selection View Go Run ... ← → P44 - Leitor de QR Code

EXPLORER

P44 - LEITOR DE QR CODE

- # estilos.css
- index.html
- qrcode.png
- qrcode+(1).png
- qrcode+(2).png
- qrcode+(3).png
- script.js

JS script.js > addEventListener('change') callback > onload > onload

```
91     contexto.drawImage(img, 0, 0, img.width, img.height);

92

93     // Obtém os dados da imagem desenhada no canvas.
94     // 'getImageData' é um método que retorna um objeto
95         // ImageData representando os dados de pixels para a área
96         // definida do canvas. Aqui, capturamos os dados de toda a área do canvas.
97 const dadosImagen = contexto.getImageData(0, 0, canvas.width, canvas.height);

98

99     // Usa a função jsQR para tentar decodificar um QR Code a partir dos dados da imagem.
100    // 'jsQR' é uma função de uma biblioteca externa que tenta
101        // decodificar um QR Code dado um conjunto de dados de
102            // imagem (pixels). Se um QR Code válido for encontrado,
103                // retorna seus dados; caso contrário, retorna null.
104 const codigo = jsQR(dadosImagen.data, canvas.width, canvas.height);

105

106    // Verifica se um QR Code foi decodificado com sucesso.
107 if (codigo) {

108

109    // Atualiza o texto do elemento com ID 'resultado' com
110        // os dados do QR Code lido.
111    // Se um código QR foi decodificado com sucesso, o
112        // elemento com ID 'resultado' é atualizado para
113            // mostrar os dados do QR Code.
114 document.getElementById('resultado').innerText = `Código QR lido: ${codigo.data}`;

115

116 } else {

117

118    // Atualiza o texto do elemento com ID 'resultado' para
119        // informar que a leitura falhou.
120        // Se nenhum QR Code válido foi decodificado, uma
```

Ln 91, Col 7 Spaces: 8 UTF-8 CRLF {} JavaScript ⚡ Port: 5500



# **UC: Projetos**

## **Assunto:**

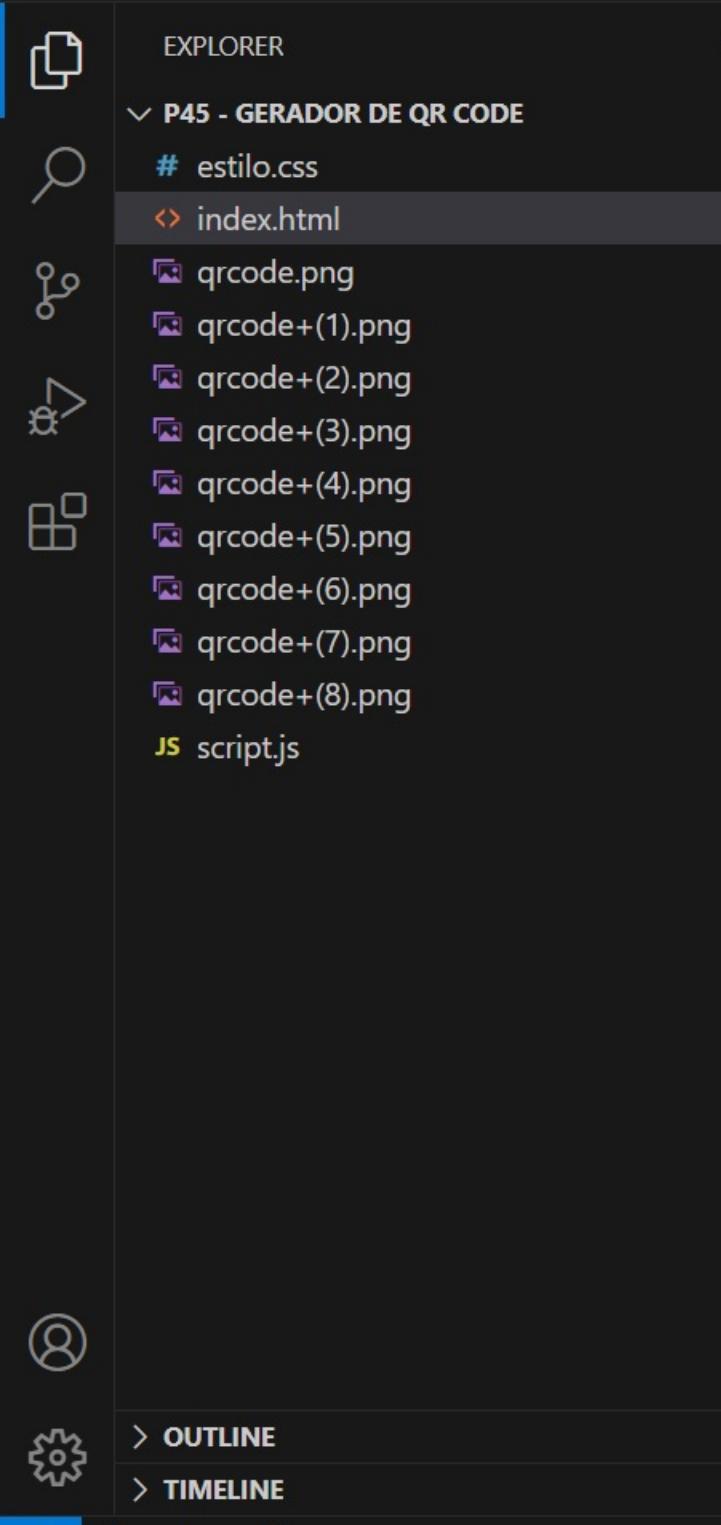
### Small Projects

### P45 – Gerador de QR Code

# Gerador de QR Code

Digite o texto ou URL

Gerar QR Code



```
↳ index.html X  
↳ index.html > ...  
1  <!DOCTYPE html>  
2  <!-- Declara que o documento é um documento HTML5, o que  
3  | ajuda os navegadores a renderizar o conteúdo  
4  | corretamente seguindo as regras do HTML5. -->  
5  
6  <html lang="pt-br">  
7  <!-- Abre a tag HTML definindo o idioma do documento como  
8  | português do Brasil, o que é útil para acessibilidade e  
9  | motores de busca. -->  
10  
11 <head>  
12 <!-- Início do cabeçalho do documento HTML, contém metadados e  
13 | links para scripts e folhas de estilo. -->  
14  
15 <meta charset="UTF-8">  
16 <!-- Define a codificação de caracteres para UTF-8, que inclui a  
17 | maioria dos caracteres usados globalmente, garantindo a  
18 | correta exibição de caracteres especiais e acentos. -->  
19  
20 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
21 <!-- Configura a visualização para ser responsiva em diferentes  
22 | dispositivos. Esta linha garante que a largura da  
23 | viewport (área de visualização) do dispositivo seja  
24 | igual à largura do dispositivo e que a escala inicial  
25 | seja 1, ou seja, sem zoom inicial. -->  
26  
27 <title>Gerador de QR Code</title>  
28 <!-- Define o título da página, que aparece na aba do  
29 | navegador e é usado em SEO e favoritos. -->  
30
```

File Edit Selection View Go Run ... ⏪ ⏩ 🔎 P45 - Gerador de QR Code

EXPLORER ...

P45 - GERADOR DE QR CODE

- # estilo.css
- index.html
- qrCode.png
- qrCode+(1).png
- qrCode+(2).png
- qrCode+(3).png
- qrCode+(4).png
- qrCode+(5).png
- qrCode+(6).png
- qrCode+(7).png
- qrCode+(8).png
- JS script.js

index.html X

index.html > html > head

```
31     <link rel="stylesheet" href="estilo.css">
32     <!-- Link para a folha de estilos externa chamada 'estilo.css', que
33         |     contém os estilos CSS para estilizar a página. --&gt;
34
35 &lt;/head&gt;
36     <!-- Fim do cabeçalho do documento. --&gt;
37
38
39 &lt;body&gt;
40     <!-- Início do corpo do documento HTML, onde todo o
41         |     conteúdo visível da página é inserido. --&gt;
42
43     &lt;div class="container"&gt;
44         <!-- Divisão ou seção na página, definida com uma classe 'container'
45             |     para aplicar estilos específicos através do CSS. --&gt;
46
47         &lt;h1&gt;Gerador de QR Code&lt;/h1&gt;
48         <!-- Cabeçalho principal (h1) da página, utilizado aqui para
49             |     dar um título visível na parte superior da aplicação. --&gt;
50
51         &lt;input type="text" id="entradaTexto" placeholder="Digite o texto ou URL"&gt;
52         <!-- Campo de entrada de texto onde o usuário pode digitar o
53             |     texto ou URL que será convertido em QR Code.
54             |     O atributo 'placeholder' fornece um texto guia dentro do campo. --&gt;
55
56         &lt;button id="botaoGerar"&gt;Gerar QR Code&lt;/button&gt;
57         <!-- Botão que, quando clicado, acionará a geração do QR Code
58             |     baseado no texto inserido no campo de entrada acima. --&gt;
59
60         &lt;div id="qrCode"&gt;&lt;/div&gt;</pre>

Ln 31, Col 4 Spaces: 8 UTF-8 CRLF HTML ⚡ Port : 5500


```

File Edit Selection View Go Run ... ← → 🔍 P45 - Gerador de QR Code

EXPLORER

P45 - GERADOR DE QR CODE

- # estilo.css
- index.html
- qrcode.png
- qrcode+(1).png
- qrcode+(2).png
- qrcode+(3).png
- qrcode+(4).png
- qrcode+(5).png
- qrcode+(6).png
- qrcode+(7).png
- qrcode+(8).png
- JS script.js

index.html

```
<!-- Divisão onde o QR Code gerado será exibido. Inicialmente vazia, será preenchida dinamicamente pelo script. -->
<button id="botaoBaixar" style="display: none;">Baixar QR Code</button>
<!-- Botão para baixar o QR Code gerado. Inicialmente oculto (display: none), será exibido após a geração do QR Code pelo JavaScript. -->
</div>
<!-- Fim da divisão 'container'. -->
<script src="https://cdn.jsdelivr.net/npm/qrcodejs/qrcode.min.js"></script>
<!-- Inclusão do script externo 'qrcode.min.js' da biblioteca QRCodeJS que permite a geração de QR Codes. Este script é necessário para criar os códigos a partir do texto fornecido. -->
<script src="script.js"></script>
<!-- Inclusão do script externo 'script.js', que contém a lógica JavaScript específica para esta aplicação, como a manipulação de eventos de clique e a lógica para gerar e baixar QR Codes. -->
</body>
<!-- Fim do corpo do documento HTML. -->
</html>
<!-- Fim do documento HTML. -->
```

OUTLINE

TIMELINE

Ln 61, Col 7 Spaces: 8 UTF-8 CRLF HTML ⚡ Port : 5500

EXPLORER

...

✓ P45 - GERADOR DE QR CODE

- # estilo.css
- index.html
- qrcode.png
- qrcode+(1).png
- qrcode+(2).png
- qrcode+(3).png
- qrcode+(4).png
- qrcode+(5).png
- qrcode+(6).png
- qrcode+(7).png
- qrcode+(8).png

JS script.js

> OUTLINE

> TIMELINE

index.html # estilo.css X

# estilo.css > body

```
1 body {  
2     font-family: Arial, sans-serif;  
3     /* Define as famílias de fontes do corpo do documento  
4         como Arial e, caso Arial não esteja disponível,  
5         usa uma fonte sans-serif genérica. */  
6  
7     display: flex;  
8     /* Estabelece o modelo de layout Flexbox para o elemento body, o  
9         que permite um layout mais flexível e responsivo. */  
10  
11    justify-content: center;  
12    /* Alinha os itens filhos do body (no eixo principal, que é  
13        horizontal por padrão no Flexbox) ao centro.  
14        Isso centraliza horizontalmente todo o conteúdo  
15        dentro do body. */  
16  
17    align-items: center;  
18    /* Alinha os itens filhos do body (no eixo cruzado, que é  
19        vertical por padrão no Flexbox) ao centro. Isso  
20        centraliza verticalmente todo o conteúdo dentro do body. */  
21  
22    height: 100vh;  
23    /* Define a altura do body como 100% da altura da janela de  
24        visualização (viewport), garantindo que o body  
25        ocupe toda a altura visível da tela. */  
26  
27  
28    background-color: #f0f0f0;  
29    /* Define a cor de fundo do corpo do documento como um  
30        cinza claro (#f0f0f0). */
```



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

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P45 - Gerador de QR Code
- Explorer:** Shows files and folders under "P45 - GERADOR DE QR CODE".
  - # estilo.css
  - index.html
  - qrcode.png
  - qrcode+(1).png
  - qrcode+(2).png
  - qrcode+(3).png
  - qrcode+(4).png
  - qrcode+(5).png
  - qrcode+(6).png
  - qrcode+(7).png
  - qrcode+(8).png
  - script.js
- Code Editor:** The "# estilo.css" file is open.

```
61 # container {  
62     box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
63     /* Aplica uma sombra ao redor dos elementos com a  
64      classe 'container'. A sombra é sutil (10px de  
65      desfoque, e com uma transparência de 10%) e  
66      não se desloca do elemento (0px horizontalmente e  
67      verticalmente), criando um efeito de elevação sutil. */  
68 }  
69  
70  
71 h1 {  
72     margin-bottom: 20px;  
73     /* Define a margem inferior para o elemento <h1> como 20 pixels.  
74      Isso adiciona espaço abaixo dos cabeçalhos <h1>,  
75      separando-os visualmente dos elementos que os  
76      seguem na página. */  
77 }  
78  
79  
80 }  
81  
82  
83 input {  
84     width: 80%;  
85     /* Define a largura dos elementos <input> para 80% da largura  
86      do seu elemento pai, permitindo um design responsivo  
87      que se ajusta ao tamanho do contêiner pai. */  
88 }  
89  
90     padding: 10px;
```
- Status Bar:** Shows file information (Ln 61, Col 1), spaces used (Spaces: 4), encoding (UTF-8), line endings (CRLF), CSS, and port number (Port : 5500).

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P45 - Gerador de QR Code
- Explorer:** Shows a tree view of files and folders under "P45 - GERADOR DE QR CODE". The files listed are: #estilo.css, index.html, qrcode.png, qrcode+(1).png, qrcode+(2).png, qrcode+(3).png, qrcode+(4).png, qrcode+(5).png, qrcode+(6).png, qrcode+(7).png, qrcode+(8).png, and script.js.
- Code Editor:** The main area displays the contents of the "estilo.css" file. The code defines styles for an input element (#estilo.css > input). The styles include:
  - /\* Define um preenchimento interno de 10 pixels em todas as direções dentro dos elementos <input>, aumentando a área clicável e melhorando a legibilidade ao distanciar o texto das bordas. \*/
  - margin-bottom: 10px;
  - /\* Define a margem inferior para os elementos <input> como 10 pixels, criando espaço abaixo de cada input antes do próximo elemento na página. \*/
  - border: 1px solid #ccc;
  - /\* Adiciona uma borda sólida de 1 pixel em torno dos elementos <input>, com uma cor cinza clara (#ccc), delineando claramente a área de input no design da interface. \*/
  - border-radius: 5px;
  - /\* Define o raio da borda dos elementos <input> para 5 pixels, suavizando as bordas para um visual mais moderno e menos agressivo. \*/
  - font-size: 16px;
  - /\* Define o tamanho da fonte dentro dos elementos <input> para 16 pixels, garantindo que o texto seja suficientemente grande para ser lido facilmente sem precisar de zoom. \*/
- Status Bar:** Shows file information (Ln 91, Col 2), space usage (Spaces: 4), encoding (UTF-8), line endings (CRLF), and CSS support.
- Bottom Icons:** Includes icons for search, refresh, and other editor functions.

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P45 - Gerador de QR Code
- Toolbar:** Minimize, Maximize, Close, Fullscreen, ...
- Left Sidebar (EXPLORER):**
  - P45 - GERADOR DE QR CODE
    - # estilo.css
    - index.html
    - qrcode.png
    - qrcode+(1).png
    - qrcode+(2).png
    - qrcode+(3).png
    - qrcode+(4).png
    - qrcode+(5).png
    - qrcode+(6).png
    - qrcode+(7).png
    - qrcode+(8).png
  - script.js
- Central Area:** Shows the content of the #estilo.css file, specifically the styles for the button element.
- Bottom Status Bar:** Includes icons for file operations (Save, Undo, Redo), status indicators (0 0 0), and system information (Ln 121, Col 1, Spaces: 4, UTF-8, CRLF, CSS, Port: 5500).

```
< > index.html # estilo.css X  
# estilo.css > button  
121 button {  
122     padding: 10px 20px;  
123     /* Aplica um preenchimento interno de 10 pixels na  
124      vertical e 20 pixels na horizontal em todos os  
125      elementos <button>. Isso aumenta o espaço  
126      dentro do botão, tornando-o mais fácil de  
127      clicar e visualmente agradável. */  
128  
129     background-color: #007bff;  
130     /* Define a cor de fundo dos botões como um azul  
131      brilhante (#007bff), que é visualmente atraente e  
132      destaca o botão como um elemento interativo na  
133      interface do usuário. */  
134  
135     color: #fff;  
136     /* Define a cor do texto dentro dos botões para  
137      branco (#fff), garantindo alto contraste com a  
138      cor de fundo azul para melhor legibilidade. */  
139  
140     border: none;  
141     /* Remove qualquer borda padrão do botão, criando uma  
142      aparência mais limpa e moderna. */  
143  
144     border-radius: 5px;  
145     /* Arredonda as bordas do botão com um raio de 5 pixels,  
146      suavizando as bordas para um estilo mais  
147      amigável e moderno. */  
148  
149     font-size: 16px;
```

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P45 - Gerador de QR Code
- Toolbar:** Minimize, Maximize, Close, Fullscreen, ...
- Left Sidebar (EXPLORER):**
  - P45 - GERADOR DE QR CODE
    - # estilo.css
    - index.html
    - qrcode.png
    - qrcode+(1).png
    - qrcode+(2).png
    - qrcode+(3).png
    - qrcode+(4).png
    - qrcode+(5).png
    - qrcode+(6).png
    - qrcode+(7).png
    - qrcode+(8).png
  - script.js
- Code Editor:** The current file is # estilo.css. The code defines styles for a button and a #qrcode element.

```
/* Define o tamanho da fonte dos textos dentro dos
   botões para 16 pixels, o que ajuda na
   legibilidade e acessibilidade. */
button {
    cursor: pointer;
    /* Muda o cursor para um ponteiro quando sobrevoa os
       botões, indicando que o elemento é clicável. */
    margin-top: 10px;
    /* Adiciona uma margem superior de 10 pixels aos botões,
       separando-os dos elementos acima e ajudando a
       organizar visualmente a interface. */
}

button:hover {
    background-color: #0056b3;
    /* Altera a cor de fundo dos botões para um azul mais
       escuro (#0056b3) quando o mouse está sobre
       eles, proporcionando um feedback visual imediato
       que o botão é interativo e está focado. */
}

#qrcode {
    margin-top: 20px;
    /* Define uma margem superior de 20 pixels para o elemento
```
- Bottom Status Bar:** Line 151, Column 4, Spaces: 4, UTF-8, CRLF, CSS, Port: 5500, Bell icon

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P45 - Gerador de QR Code
- Toolbar:** Standard window control icons.
- Left Sidebar (EXPLORER):**
  - P45 - GERADOR DE QR CODE
    - # estilo.css
    - <> index.html
    - qrkode.png
    - qrkode+(1).png
    - qrkode+(2).png
    - qrkode+(3).png
    - qrkode+(4).png
    - qrkode+(5).png
    - qrkode+(6).png
    - qrkode+(7).png
    - qrkode+(8).png
  - JS script.js
- Central Editor Area:**
  - File: # estilo.css
  - Line: # estilo.css > #qrkode
  - Line 181: com o id 'qrkode'. Isso adiciona espaço entre o elemento 'qrkode' e quaisquer elementos que estejam acima dele na página, ajudando a separar visualmente o conteúdo. \*/
  - Line 186: display: flex;
  - Line 187: /\* Estabelece o modelo de layout Flexbox para o elemento com o id 'qrkode'. O Flexbox é um modelo de layout projetado para estruturar espaços uniformes entre itens em um contêiner e alinhá-los eficientemente, mesmo quando seus tamanhos são desconhecidos ou dinâmicos. \*/
  - Line 194: justify-content: center;
  - Line 195: /\* Alinha os itens filhos do elemento com o id 'qrkode' (no eixo principal, que é horizontal por padrão no Flexbox) ao centro. Este estilo é utilizado para centralizar o conteúdo do QR code horizontalmente dentro do contêiner 'qrkode'. \*/
  - Line 201: }
- Bottom Status Bar:** < 0 △ 0 ⌂ 0 Ln 181, Col 12 Spaces: 4 UTF-8 CRLF CSS ⚡ Port : 5500

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P45 - Gerador de QR Code
- Explorer:** Shows a folder named "P45 - GERADOR DE QR CODE" containing files: # estilo.css, index.html, qrcode.png, qrcode+(1).png, qrcode+(2).png, qrcode+(3).png, qrcode+(4).png, qrcode+(5).png, qrcode+(6).png, qrcode+(7).png, qrcode+(8).png, and script.js.
- Code Editor:** The "script.js" tab is active, displaying the following JavaScript code:

```
// Seleciona o elemento botão com o ID 'botaoGerar' e
// adiciona um ouvinte de evento para reagir a cliques.
document.getElementById('botaoGerar').addEventListener('click', function() {
    // Obtém o valor atual do elemento de entrada de
    // texto com o ID 'entradaTexto' e o armazena na variável 'texto'.
    var texto = document.getElementById('entradaTexto').value;

    // Verifica se a variável 'texto' não está vazia.
    if (texto) {
        // Seleciona o elemento com o ID 'qrcode' e armazena a
        // referência na variável 'contenedorQrCode'.
        var contenedorQrCode = document.getElementById('qrcode');

        // Limpa qualquer conteúdo HTML anterior dentro do
        // elemento 'qrcode', preparando-o para um novo QR Code.
        contenedorQrCode.innerHTML = ''; // Limpa o QR Code anterior, se existir

        // Cria uma nova instância do QRCode dentro do 'contenedorQrCode'
        // usando a biblioteca QRCodeJS.
        var qrCode = new QRCode(contenedorQrCode, {
            text: texto, // Define o texto que será transformado em QR Code
            width: 256, // Define a largura do QR Code como 256 pixels
            height: 256, // Define a altura do QR Code como 256 pixels
        });

        // Seleciona o botão de download com o ID 'botaoBaixar' e
        // muda o estilo de 'display' para 'block', tornando-o visível.
        document.getElementById('botaoBaixar').style.display = 'block';
    }
});
```

**Status Bar:** Shows file count (0), line count (0), character count (0), and other system information like Ln 1, Col 1, Spaces: 8, UTF-8, CRLF, {} JavaScript, Port: 5500, 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:** P45 - Gerador de QR Code
- Explorer:** Shows files in the project: index.html, #estilo.css, script.js, and several qrcode.png files (qrcode+(1).png to qrcode+(8).png).
- Code Editor:** The script.js file is open, containing the following code:

```
// O método 'toDataURL' é chamado sobre o elemento <canvas> para
// obter a imagem em formato PNG como uma string
// de URL codificada em base64.

// Define o atributo 'download' do elemento de âncora,
// especificando o nome do arquivo que será
// usado quando o usuário salvar o download.
link.download = 'qrcode.png';
// O valor 'qrcode.png' é o nome de arquivo proposto para o
// arquivo baixado, indicando ao navegador que o
// recurso deve ser baixado ao invés de navegado.

// Dispara um clique programaticamente no elemento de âncora.
// Isso inicia o download do arquivo
// definido no atributo 'href'.
link.click();
// Esse método 'click' simula um clique do usuário, que é o
// que efetivamente inicia o processo de download.

});
```

**Bottom Status Bar:** Shows file information (Ln 61, Col 2), spaces used (Spaces: 8), encoding (UTF-8), line endings (CRLF), and a port number (Port : 5500).

# **UC: Projetos**

## **Assunto:**

### Small Projects

### P46 – Jogo Quebra-Cabeça

Carregar imagem

9 Peças ▾

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P46 - Jogo Quebra-Cabeça
- Icons:** Explorer, Search, Open, Save, Help, Outline, Timeline.
- Left Sidebar (Explorer):** P46 - JOGO QUEBRA-CABEÇA (selected), index.html, script.js, styles.css.
- Active Editor:** index.html
- Code Content:**

```
1 <!DOCTYPE html>
2  <!-- Declaração do tipo de documento:
3      &lt;!DOCTYPE html&gt; é usada para informar ao
4          navegador que este documento está
5              escrito em HTML5. --&gt;
6
7 &lt;html lang="pt-br"&gt;
8  <!-- Abertura da tag &lt;html&gt; que define o início de
9      um documento HTML.
10     O atributo 'lang="pt-br"' define o idioma
11         principal do conteúdo como Português do Brasil,
12             ajudando navegadores e mecanismos de busca a
13                 entenderem o conteúdo. --&gt;
14
15 &lt;head&gt;
16  <!-- Abertura da tag &lt;head&gt; que contém metadados sobre o
17      documento, como título, links para estilos, etc. --&gt;
18
19 &lt;meta charset="UTF-8"&gt;
20  <!-- Define a codificação de caracteres do documento como UTF-8,
21      que é uma codificação padrão para representar a
22          maioria dos caracteres usados na web.
23          Isso garante que caracteres especiais e
24              acentos sejam exibidos corretamente. --&gt;
25
26 &lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt;
27  <!-- Define a configuração da viewport para melhorar a
28      exibição em dispositivos móveis.
29          'width=device-width' ajusta a largura da página
30              para a largura da tela do dispositivo.</pre>
```

File Edit Selection View Go Run ... ⏪ ⏩ 🔎 P46 - Jogo Quebra-Cabeça

EXPLORER ... < index.html X

P46 - JOGO QUEBRA-CABEÇA

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

index.html > html > head

```
31     'initial-scale=1.0' define o nível inicial de zoom
32         quando a página é carregada pela primeira vez. -->
33
34     <link rel="stylesheet" href="styles.css">
35     <!-- Vincula um arquivo CSS externo para estilizar a página.
36         'href="styles.css"' especifica o caminho para o arquivo CSS.
37         Isso permite que o navegador carregue e aplique
38             os estilos definidos no arquivo 'styles.css'. --&gt;
39
40     &lt;title&gt;Jogo de Quebra-Cabeça&lt;/title&gt;
41     <!-- Define o título da página, que será exibido na aba
42         do navegador e usado pelos mecanismos de busca.
43         O texto entre as tags &lt;title&gt; e &lt;/title&gt; será o
44             nome da aba do navegador. --&gt;
45
46 &lt;/head&gt;
47     <!-- Fecha a tag &lt;head&gt;. Todo o conteúdo que define as
48         configurações e informações da página foi
49             inserido acima. --&gt;
50
51
52
53 &lt;body&gt;
54     <!-- Abertura da tag &lt;body&gt; que contém todo o
55         conteúdo visível da página da web.
56         Tudo o que está dentro desta tag será
57             exibido no navegador. --&gt;
58
59     &lt;label for="entradaImagem" class="botao-personalizado"&gt;Carregar imagem&lt;/label&gt;
60     <!-- Cria um rótulo (label) que será clicável</pre>

OUTLINE



TIMELINE



✖ 0 ▲ 0 ⌂ 0 🔍 Ln 31, Col 2 Spaces: 6 UTF-8 CRLF HTML ⚡ Port : 5500 📡


```

P46 - Jogo Quebra-Cabeça

```
index.html
index.html > html > body
61     para carregar uma imagem.
62     O atributo 'for="entradaImagem"' associa o
63         rótulo ao input com id 'entradaImagem'.
64     A classe 'botao-personalizado' aplica
65         estilos CSS personalizados ao rótulo,
66         transformando-o em um botão estilizado para
67             carregar a imagem. -->
68
69     <input type="file" id="entradaImagem" accept="image/*">
70     <!-- Cria um campo de entrada de arquivo (input)
71         para carregar uma imagem.
72     'type="file"' define o tipo de entrada como arquivo.
73     'id="entradaImagem"' é um identificador único para o elemento,
74         permitindo que ele seja acessado e
75             manipulado via JavaScript.
76     'accept="image/*"' restringe os tipos de
77         arquivos aceitos a imagens, garantindo que
78             apenas arquivos de imagem possam ser selecionados. -->
79
80     <select id="selecaoTamanho">
81         <!-- Cria um menu suspenso (select) para o
82             usuário escolher o tamanho do quebra-cabeça.
83         'id="selecaoTamanho"' é um identificador único
84             que permite a manipulação deste
85                 elemento via JavaScript.
86             Isso será usado para selecionar a quantidade de
87                 peças do quebra-cabeça. -->
88
89     <option value="9">9 Peças</option>
90     <!-- Cria uma opção no menu suspenso com o valor '9' (9 peças).
```

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P46 - Jogo Quebra-Cabeça
- Explorer:** Shows a project folder named "P46 - JOGO QUEBRA-CABEÇA" containing files: index.html, script.js, and styles.css.
- Code Editor:** The file "index.html" is open, showing the following code with annotations:

```
<html>
  <body>
    <select id="selecaoTamanho">
      <option value="9">9 Peças</option>
      <!-- Cria uma opção no menu suspenso com o valor '9' (9 peças). -->
      <option value="16">16 Peças</option>
      <!-- Cria uma opção no menu suspenso com o valor '16' (16 peças). -->
      <option value="25">25 Peças</option>
      <!-- Cria uma opção no menu suspenso com o valor '25' (25 peças). -->
    </select>

    <div class="container-quebra-cabeca" id="containerQuebraCabeca"></div>
    <!-- Cria uma div com a classe 'container-quebra-cabeca' para conter o quebra-cabeça.
        'id="containerQuebraCabeca"' é um identificador único que permite a manipulação deste elemento via JavaScript.
        Essa div será usada para exibir o quebra-cabeça, configurando a grade onde as peças serão posicionadas. -->

    <script src="script.js"></script>
    <!-- Vincula um arquivo JavaScript externo para adicionar funcionalidades à página.
        'src="script.js"' especifica o caminho para o arquivo JavaScript.
        Esse script conterá o código que manipula a lógica do jogo de quebra-cabeça. -->

  </body>
<!-- Fecha a tag <body>. Todo o conteúdo visível da página foi inserido acima. -->
```

The code is annotated with Portuguese comments explaining the purpose of each section, such as creating a dropdown menu for puzzle sizes (9, 16, or 25 pieces), defining a container div for the puzzle, and linking an external JavaScript file to handle the logic.

A screenshot of a code editor interface, likely Visual Studio Code, displaying an HTML file named `index.html`. The code editor has a dark theme with light-colored syntax highlighting. The top bar includes standard menu items like File, Edit, Selection, View, Go, Run, and a search bar containing the text "P46 - Jogo Quebra-Cabeça". The left sidebar features icons for Explorer, Search, Problems, and other tools, with the "PROBLEMS" icon currently active. The Explorer sidebar shows a project folder named "P46 - JOGO QUEBRA-CABEÇA" containing files: `index.html`, `script.js`, and `styles.css`. The main editor area shows the following code:

```
121    </html>
122    <!-- Fecha a tag &lt;html&gt;, indicando o fim do documento HTML. --&gt;</pre>

The status bar at the bottom provides information about the current file: Line 121, Column 1, Spaces: 6, UTF-8 encoding, CRLF line endings, and HTML content type. It also shows a port number of 5500 and a small bell icon.


```

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P46 - Jogo Quebra-Cabeça
- Icons:** Explorer, Search, Find, Open, Save, Outline, Timeline.
- Left Sidebar:** EXPLORER, P46 - JOGO QUEBRA-CABEÇA (index.html, script.js), # styles.css (selected).
- Central Area:** Editor showing CSS code for a grid container.

```
.container-quebra-cabeça {
    /* Seleciona o elemento com a classe
       "container-quebra-cabeça".
    Este contêiner será usado para posicionar as
       peças do quebra-cabeça em uma grade. */

    display: grid;
    /* Define o modo de exibição do contêiner como uma grade.
       O layout de grade permite organizar os
       itens em linhas e colunas. */

    grid-template-columns: repeat(3, 150px);
    /* Define a estrutura das colunas da grade.
       "repeat(3, 150px)" cria três colunas, cada
       uma com 150 pixels de largura. */

    grid-template-rows: repeat(3, 150px);
    /* Define a estrutura das linhas da grade.
       "repeat(3, 150px)" cria três linhas, cada
       uma com 150 pixels de altura. */

    gap: 5px;
    /* Define um espaço (gap) de 5 pixels entre
       cada célula da grade.
       Isso cria um espaçamento uniforme entre as
       peças do quebra-cabeça. */

    margin-top: 20px;
    /* Adiciona uma margem superior de 20 pixels ao contêiner.
       Isso cria um espaço de 20 pixels acima do contêiner. */
}
```
- Bottom Status Bar:** ⌂ 0 ▲ 0 ⌂ 0, Ln 1, Col 1, Spaces: 3, UTF-8, CRLF, CSS, ⌂ Port : 5500, ⌂

A screenshot of a code editor interface, likely Visual Studio Code, displaying CSS code for a jigsaw puzzle 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 "P46 - Jogo Quebra-Cabeça".

The left sidebar features icons for Explorer, Search, Problems, and Outline/Timeline.

The Explorer panel shows a project structure under "P46 - JOGO QUEBRA-CABEÇA" containing "index.html", "script.js", and the current file, "# styles.css".

The main code editor area shows the following CSS:

```
index.html # styles.css < .container-quebra-cabeça
31 }
32 }
33
34 .peca-quebra-cabeça {
35     /* Seleciona o elemento com a classe "peca-quebra-cabeça".
36     | Cada peça do quebra-cabeça será estilizada com estas regras. */
37
38     width: 150px;
39     /* Define a largura de cada peça do quebra-cabeça como 150 pixels.
40     | Isso corresponde à largura das colunas da grade. */
41
42     height: 150px;
43     /* Define a altura de cada peça do quebra-cabeça como 150 pixels.
44     | Isso corresponde à altura das linhas da grade. */
45
46     background-size: 450px 450px;
47     /* Define o tamanho da imagem de fundo de cada peça.
48     | A imagem de fundo é redimensionada para 450 pixels por 450 pixels,
49     | que é o tamanho total da imagem usada no quebra-cabeça. */
50
51     cursor: grab;
52     /* Define o cursor do mouse como uma mão que
53     | agarra (grab) ao passar sobre a peça.
54     | Isso indica que a peça pode ser arrastada. */
55
56 }
57
58 #entradaImagem {
59     /* Seleciona o elemento com o id "entradaImagem".
60     | Este é o campo de entrada de arquivo para
```

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P46 - Jogo Quebra-Cabeça
- Icons:** Explorer, Search, Find, Refresh, Outline, Timeline.
- Left Sidebar (EXPLORER):** P46 - JOGO QUEBRA-CABEÇA (index.html, script.js, # styles.css)
- Active File:** # styles.css
- Code Content:**

```
# styles.css > #entradaDalmagem
61     | carregar a imagem. */
62
63     display: none;
64     /* Oculta o campo de entrada de arquivo.
65      Isso permite usar um rótulo estilizado (label)
66      como botão para carregar a imagem. */
67
68 }
69
70 .botao-personalizado {
71     /* Seleciona o elemento com a classe "botao-personalizado".
72     Este rótulo será estilizado para se parecer com um botão. */
73
74     display: inline-block;
75     /* Define o modo de exibição do rótulo como inline-block.
76     Isso permite que ele seja estilizado como um
77     botão, mantendo o comportamento inline. */
78
79     padding: 10px 20px;
80     /* Adiciona um espaçamento interno de 10 pixels
81     na parte superior e inferior,
82     e 20 pixels nas laterais. Isso aumenta o
83     tamanho do botão e melhora a aparência. */
84
85     background-color: #007bff;
86     /* Define a cor de fundo do botão como azul (#007bff).
87     Isso dá ao botão uma aparência distinta e chamativa. */
88
89     color: white;
90     /* Define a cor do texto dentro do botão
```

- Bottom Status Bar:** ⌂ 0 △ 0 ⌂ 0 Ln 61, Col 1 Spaces: 3 UTF-8 CRLF CSS ⌂ Port: 5500



A screenshot of a code editor interface, likely Visual Studio Code, displaying a JavaScript file named `script.js`. The file contains code for a jigsaw puzzle game, specifically for setting up event listeners and selecting game elements.

```
// Adiciona um ouvinte de eventos para o
// evento 'DOMContentLoaded'.
// Este evento é disparado quando o documento HTML foi
// completamente carregado e analisado, sem esperar
// que estilos, imagens e subframes terminem de carregar.
// A função fornecida será executada assim que o DOM estiver pronto.
document.addEventListener('DOMContentLoaded', () => {

    // Seleciona o elemento com o id 'containerQuebraCabeça' e
    // o armazena na variável containerQuebraCabeça.
    // Este elemento é o contêiner onde as peças do
    // quebra-cabeça serão exibidas.
    const containerQuebraCabeça = document.getElementById('containerQuebraCabeça');

    // Seleciona o elemento com o id 'entradaImagen' e
    // o armazena na variável entradaImagen.
    // Este é o campo de entrada de arquivo usado
    // para carregar a imagem do quebra-cabeça.
    const entradaImagen = document.getElementById('entradaImagen');

    // Seleciona o elemento com o id 'selecaoTamanho' e
    // o armazena na variável selecaoTamanho.
    // Este é o menu suspenso onde o usuário seleciona o
    // número de peças do quebra-cabeça.
    const selecaoTamanho = document.getElementById('selecaoTamanho');

    // Cria um novo elemento <div> que será usado para
    // mostrar a pontuação do jogador.
    const pontuacaoDisplay = document.createElement('div');
```

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P46 - Jogo Quebra-Cabeça
- Icons:** Explorer, Search, Open, Find, Help, Outline, Timeline.
- Left Sidebar (EXPLORER):** P46 - JOGO QUEBRA-CABEÇA (index.html, script.js, # styles.css)
- Active Tab:** script.js
- Code Content:** A snippet of JavaScript code for a jigsaw puzzle game. The code includes comments in Portuguese explaining its functionality, such as inserting a score display element before the puzzle container and defining font sizes and margins for the display. It also declares variables for image URLs and pieces, and adds an event listener for file selection changes.
- Bottom Status Bar:** Line 31, Col 7, Spaces: 8, UTF-8, CRLF, {} JavaScript, Port: 5500

```
// Insere o novo elemento pontuacaoDisplay no corpo do
// documento, antes do containerQuebraCabeça.
// Isso garante que a pontuação seja exibida acima do quebra-cabeça.
document.body.insertBefore(pontuacaoDisplay, containerQuebraCabeça);

// Define o tamanho da fonte do pontuacaoDisplay como 20 pixels.
pontuacaoDisplay.style.fontSize = '20px';

// Define uma margem superior de 10 pixels para o pontuacaoDisplay.
pontuacaoDisplay.style.marginTop = '10px';

// Declara uma variável urlImagen para armazenar
// a URL da imagem carregada.
// Inicialmente, esta variável é uma string vazia.
let urlImagen = '';

// Declara uma variável pecas que será usada
// para armazenar as peças do quebra-cabeça.
// Inicialmente, esta variável é nula.
let pecas = null;

// Adiciona um ouvinte de eventos para o evento 'change' no
// campo de entrada de arquivo (entradaImagen).
// A função fornecida será executada sempre que o
// usuário selecionar um novo arquivo.
entradaImagen.addEventListener('change', function(e) {

    // Verifica se nenhum arquivo foi selecionado.
    // Se a lista de arquivos (e.target.files) estiver
    // vazia, o comprimento será 0.
```

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P46 - Jogo Quebra-Cabeça
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Print, Help, ...
- Left Sidebar (EXPLORER):** Shows a tree view with a node expanded: P46 - JOGO QUEBRA-CABEÇA, containing index.html, script.js, and # styles.css.
- Current File:** script.js (highlighted in yellow)
- Code Content:** A snippet of JavaScript code for a file selection and reading process. The code uses event listeners and FileReader to handle file uploads and read their contents.

```
// Neste caso, a função retorna imediatamente,  
// impedindo o processamento adicional.  
// Impede processamento se nenhum arquivo for selecionado.  
if (e.target.files.length === 0) return;  
  
// Obtém o primeiro arquivo selecionado pelo usuário.  
// A lista de arquivos selecionados é acessível  
// via e.target.files, onde e é o evento disparado.  
// O primeiro arquivo na lista (e.target.files[0]) é  
// armazenado na variável arquivo.  
const arquivo = e.target.files[0];  
  
// Cria uma nova instância de FileReader.  
// FileReader é um objeto que permite que aplicativos  
// leiam o conteúdo de arquivos (ou buffers de dados brutos)  
// armazenados no cliente de forma assíncrona.  
const leitor = new FileReader();  
  
// Define uma função para ser executada quando o  
// arquivo for lido com sucesso.  
// Esta função é atribuída ao evento 'onload' do FileReader.  
// Quando o arquivo for carregado, a função será executada.  
leitor.onload = function(e) {  
  
    // Armazena a URL da imagem carregada na variável urlImagen.  
    // e.target.result contém o resultado da  
    // leitura do arquivo, que é uma URL de dados.  
    urlImagen = e.target.result;  
  
    // Chama a função inicializarQuebraCabeça para
```

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P46 - Jogo Quebra-Cabeça
- Explorer:** P46 - JOGO QUEBRA-CABEÇA (index.html, script.js, # styles.css)
- Code Editor:** Script.js (selected tab)
- Code Content:** A script for a 4x6 jigsaw puzzle. It uses event listeners to handle image loading and configuration.

```
// configura o quebra-cabeça com a nova imagem.
inicializarQuebraCabeca();

};

// Lê o conteúdo do arquivo como uma URL de dados.
// Isso converte o arquivo de imagem em uma URL que
// pode ser usada como fonte de uma imagem no HTML.
// O método readAsDataURL é assíncrono e, quando a
// leitura é concluída, o evento 'onload' é disparado.
leitor.readAsDataURL(arquivo);

});

// Define a função inicializarQuebraCabeca que
// configura o quebra-cabeça com a imagem carregada.
function inicializarQuebraCabeca() {

    // Limpa o conteúdo do containerQuebraCabeca,
    // removendo qualquer quebra-cabeça anterior.
    // Limpa o container para a nova imagem
    containerQuebraCabeca.innerHTML = '';

    // Obtém o valor selecionado no menu suspenso (selecaoTamanho),
    // converte-o para um número inteiro,
    // e o armazena na variável tamanho. Isso define o
    // número de peças do quebra-cabeça.
    const tamanho = parseInt(selecaoTamanho.value);

    // Cria o container para o quebra-cabeça
    const containerQuebraCabeca = document.createElement('div');
    containerQuebraCabeca.id = 'containerQuebraCabeca';
    document.body.appendChild(containerQuebraCabeca);

    // Cria as peças do quebra-cabeça
    const peças = criarPeças(tamanho);
    const quebraCabecaHTML = gerarHTMLParaPeças(peças);
    containerQuebraCabeca.innerHTML = quebraCabecaHTML;

    // Adiciona o quebra-cabeça ao DOM
    const quebraCabecaElement = document.querySelector('#quebraCabeca');
    quebraCabecaElement.innerHTML = '';
    quebraCabecaElement.appendChild(containerQuebraCabeca);
}
```

**Bottom Bar:** X, ⌂ 0 △ 0 ⌂ 0, Q, Ln 91, Col 12, Spaces: 8, UTF-8, CRLF, {}, JavaScript, ⌂ Port: 5500, ⌂

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P46 - Jogo Quebra-Cabeça
- Toolbars:** Standard toolbar icons for file operations.
- Sidebar:** Explorer sidebar showing files: index.html, script.js, and styles.css under the folder P46 - JOGO QUEBRA-CABEÇA.
- Code Editor:** The script.js file is open, containing the following code:

```
// script.js
// document.addEventListener('DOMContentLoaded') callback > inicializarQuebraCabeça

121 // Calcula a dimensão do quebra-cabeça (por exemplo, 3 para
122 // 9 peças, 4 para 16 peças) com base no tamanho.
123 const dimensao = Math.sqrt(tamanho);
124
125 // Define o tamanho da imagem como 450 pixels por 450 pixels.
126 // Assume que a imagem tem um tamanho de 450px por 450px
127 const tamanhoImagen = 450;
128
129 // Calcula o tamanho de cada peça do quebra-cabeça
130 // dividindo o tamanho da imagem pela dimensão.
131 const tamanhoPeca = tamanhoImagen / dimensao;
132
133 // Define o número e o tamanho das colunas da
134 // grade do containerQuebraCabeça.
135 // A propriedade gridTemplateColumns define o
136 // layout das colunas da grade.
137 // `repeat(${dimensao}, ${tamanhoPeca}px)` cria um
138 // número de colunas igual à dimensão (número de peças por linha),
139 // e cada coluna terá a largura de uma peça
140 // do quebra-cabeça (tamanhoPeca).
141 containerQuebraCabeça.style.gridTemplateColumns = `repeat(${dimensao}, ${tamanhoPeca}px)`;
142
143 // Define o número e o tamanho das linhas da grade do
144 // containerQuebraCabeça.
145 // A propriedade gridTemplateRows define o
146 // layout das linhas da grade.
147 // `repeat(${dimensao}, ${tamanhoPeca}px)` cria
148 // um número de linhas igual à dimensão (número de peças por coluna),
149 // e cada linha terá a altura de uma peça do
150 // quebra-cabeça (tamanhoPeca).
```

The code implements a function to initialize a 16x16 jigsaw puzzle by calculating piece dimensions and setting up a grid layout using CSS Grid.



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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P46 - Jogo Quebra-Cabeça
- Toolbar:** Back, Forward, Search icon, Refresh icon, Minimize, Maximize, Close.
- Left Sidebar (EXPLORER):**
  - P46 - JOGO QUEBRA-CABEÇA
    - index.html
    - script.js**
    - # styles.css
- Current File:** script.js
- Code Content:** The code is a script for a 4x6 Jigsaw puzzle. It uses event listeners to handle DOM content loading and initializes the puzzle pieces. The code includes comments explaining the logic for calculating the position of each piece based on its index and the size of the puzzle grid. It sets the background image, position, width, height, and size for each piece's element.

```
// Isso faz com que a peça exiba uma parte da
// imagem do quebra-cabeça.
peca.style.backgroundImage = `url(${urlImagen})`;

// Calcula a posição de fundo da peça do quebra-cabeça.
// A posição horizontal (x) é calculada com base na
// coluna em que a peça deveria estar,
// multiplicada pelo tamanho da peça e negativa para
// mover a imagem na direção correta.
// Posição horizontal da peça na imagem de fundo
const x = -(posicao % dimensao) * tamanhoPeca;

// A posição vertical (y) é calculada com base na
// linha em que a peça deveria estar,
// multiplicada pelo tamanho da peça e negativa para
// mover a imagem na direção correta.
// Posição vertical da peça na imagem de fundo
const y = -Math.floor(posicao / dimensao) * tamanhoPeca;
peca.style.backgroundPosition = `${x}px ${y}px`;

// Define a largura e a altura da peça do quebra-cabeça.
// Isso garante que cada peça tenha o tamanho correto.
peca.style.width = `${tamanhoPeca}px`;
peca.style.height = `${tamanhoPeca}px`;

// Define o tamanho de fundo da peça do quebra-cabeça
// para corresponder ao tamanho total da imagem.
// Isso faz com que a imagem de fundo na peça se
// encaixe corretamente com outras peças.
peca.style.backgroundSize = `${tamanhoImagen}px ${tamanhoImagen}px`;
```

- Bottom Status Bar:** Line 181, Col 8, Spaces: 8, UTF-8, CRLF, {} JavaScript, Port: 5500.

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P46 - Jogo Quebra-Cabeça
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Print, Help, ...
- Left Sidebar (EXPLORER):**
  - P46 - JOGO QUEBRA-CABEÇA
    - index.html
    - script.js**
    - # styles.css
- Code Editor:** The current file is script.js. The code is written in Portuguese and handles the initialization of a 4x6 jigsaw puzzle. It includes comments explaining the purpose of each step:
  - Lines 211-227: Adds a piece to the container, defining custom attributes 'data-posicao' and 'data-indice'.
  - Lines 228-240: Selects all puzzle pieces, adds drag listeners, and initializes the score.
- Bottom Status Bar:** Includes icons for file operations, a search bar, and status information: Line 211, Column 1, Spaces: 8, UTF-8, CRLF, {}, JavaScript, Port: 5500.

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P46 - Jogo Quebra-Cabeça
- Toolbar:** Back, Forward, Search icon, Refresh icon, Minimize, Maximize, Close.
- Sidebar (EXPLORER):** Shows a tree view with a node expanded: P46 - JOGO QUEBRA-CABEÇA, containing index.html, script.js, and # styles.css.
- Code Editor:** The current file is script.js. The code is as follows:

```
index.html    # styles.css    JS script.js    X
JS script.js > document.addEventListener('DOMContentLoaded') callback > inicializarQuebraCabeça

241     atualizarPontuacao();
242
243 }
244
245
246 // Define a função embaralhar que recebe um
247 // array como argumento e retorna o array embaralhado.
248 function embaralhar(array) {
249
250     // Itera sobre os elementos do array de trás para frente.
251     // 'i' começa no último índice do array (array.length - 1)
252     // e decrementa até 1.
253     for (let i = array.length - 1; i > 0; i--) {
254
255         // Gera um índice aleatório 'j' entre 0 e 'i' (inclusive).
256         // Math.random() gera um número decimal aleatório
257         // entre 0 (inclusive) e 1 (exclusive).
258         // Multiplicar por (i + 1) ajusta o intervalo para [0, i].
259         // Math.floor() arredonda o número decimal para baixo,
260         // convertendo-o em um número inteiro.
261         const j = Math.floor(Math.random() * (i + 1));
262
263         // Troca os elementos no índice 'i' e no índice 'j'.
264         // Utiliza a desestruturação de array para fazer a troca.
265         // [array[i], array[j]] = [array[j], array[i]]; faz a
266         // troca dos valores entre array[i] e array[j].
267         [array[i], array[j]] = [array[j], array[i]];
268     }
269
270     // Retorna o array embaralhado.
```

The code implements a shuffle function for arrays, using a Fisher-Yates algorithm. It defines a function `embaralhar` that takes an array as input and returns it shuffled. The function iterates from the last index to the first, generates a random index `j` between 0 and `i` (inclusive), and swaps the elements at indices `i` and `j` using destructuring assignment.

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P46 - Jogo Quebra-Cabeça
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Print, Help, ...
- Left Sidebar (EXPLORER):** Shows a tree view with the project folder "P46 - JOGO QUEBRA-CABEÇA" expanded, containing files: index.html, script.js, and styles.css.
- Current File:** script.js is the active tab, showing the following code:

```
// script.js > document.addEventListener('DOMContentLoaded') callback > embaralhar
271     // Após a conclusão do loop, o array original foi
272     // modificado e seus elementos estão em ordem aleatória.
273     return array;
274 }
275
276
277
278 // Define a função adicionarOuvintesArrastar que
279 // adiciona ouvintes de eventos de arrastar e
280 // soltar às peças do quebra-cabeça.
281 function adicionarOuvintesArrastar() {
282
283     // Itera sobre cada elemento na lista de
284     // peças do quebra-cabeça (pecas).
285     pecas.forEach(peca => {
286
287         // Adiciona um ouvinte de evento para o início
288         // do arrastar (dragstart) em cada peça.
289         // Quando o arrastar começa, a função lidarInicioArrastar é chamada.
290         peca.addEventListener('dragstart', lidarInicioArrastar);
291
292         // Adiciona um ouvinte de evento para quando uma
293         // peça arrastada estiver sobre outra peça (dragover).
294         // A função lidarSobreArrastar é chamada para
295         // permitir que o elemento seja um alvo válido para o drop.
296         peca.addEventListener('dragover', lidarSobreArrastar);
297
298         // Adiciona um ouvinte de evento para o momento em
299         // que uma peça arrastada é solta (drop) sobre outra peça.
300         // A função lidarSolta é chamada para lidar com a troca das peças.
```

The code implements a shuffle function and a drag-and-drop event listener for a jigsaw puzzle pieces array. It adds event listeners for 'dragstart', 'dragover', and 'drop' events to handle piece movement and placement.



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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P46 - Jogo Quebra-Cabeça
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Print, Help, ...
- Left Sidebar (EXPLORER):** Shows a tree view with a node expanded: P46 - JOGO QUEBRA-CABEÇA, containing index.html, script.js, and # styles.css.
- Current File:** script.js (selected tab)
- Code Content:** A snippet of JavaScript code for a 4x6 Jigsaw puzzle. It includes functions to handle dragging and dropping pieces. The code uses comments to explain its purpose, such as preventing default behavior and checking for specific attributes on DOM elements.
- Bottom Status Bar:** Includes icons for search, refresh, and file operations, along with status information: Line 331, Column 2, Spaces: 8, UTF-8, CRLF, {}, JavaScript, Port: 5500, and a refresh icon.

```
// Define a função lidarSobreArrastar que lida com o
// evento de uma peça arrastada estar sobre outra peça.
function lidarSobreArrastar(e) {
    // Previne o comportamento padrão do evento.
    // Isso permite que a peça arrastada seja
    // solta (drop) sobre o alvo válido.
    e.preventDefault();

}

function lidarSoltar(e) {
    e.preventDefault();
    // Previne o comportamento padrão do evento de soltar (drop).
    // Isso é necessário para permitir que o drop aconteça.

    if (pecaArrastada) {
        // Verifica se existe uma peça que está sendo arrastada.

        const indiceArrastado = parseInt(pecaArrastada.getAttribute('data-indice'));
        // Obtém o índice da peça arrastada a partir
        // do atributo 'data-indice'.
        // 'data-indice' é um atributo personalizado que
        // armazena a posição atual da peça no quebra-cabeça.
        // parseInt converte o valor do atributo, que é
        // uma string, para um número inteiro.

        const indiceAlvo = parseInt(this.getAttribute('data-indice'));
        // Obtém o índice da peça alvo (sobre a qual a peça
```

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P46 - Jogo Quebra-Cabeça
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Print, Help
- Left Sidebar (EXPLORER):**
  - P46 - JOGO QUEBRA-CABEÇA
    - index.html
    - script.js
    - # styles.css
- Current File:** script.js
- Code Content:** The code is written in Portuguese and handles the logic for a 15-puzzle game. It includes functions for handling piece dragging, swapping pieces, and checking for a win condition.

```
index.html    # styles.css    JS script.js    X
JS script.js > document.addEventListener('DOMContentLoaded') callback > lidarSolta
361           | // arrastada é solta) a partir do atributo 'data-indice'.
362           | // 'this' refere-se ao elemento sobre o qual a
363           | // peça arrastada foi solta.
364           | // parseInt converte o valor do atributo, que é uma
365           | // string, para um número inteiro.
366
367           trocarPecas(indiceArrastado, indiceAlvo);
368           // Chama a função trocarPecas para trocar a peça arrastada com a peça alvo.
369           // A função trocarPecas troca as posições e atributos das duas peças no DOM.
370
371       }
372   }
373
374
375 function lidarFimArrastar() {
376
377     this.classList.remove('esconder');
378     // Remove a classe 'esconder' da peça, tornando-a
379     // visível novamente após o arrastar.
380     // 'this' refere-se à peça que foi arrastada.
381     // A classe 'esconder' é adicionada no início do
382     // arrastar para ocultar a peça arrastada
383     // durante o movimento.
384
385     verificarVitoria();
386     // Chama a função verificarVitoria para verificar se o
387     // quebra-cabeça foi resolvido após a troca.
388     // A função verificarVitoria verifica se todas as
389     // peças estão na posição correta.
390     // Se todas as peças estiverem na posição correta, o
```

Bottom status bar: ⌂ 0 △ 0 ⌂ 0 Ln 361, Col 6 Spaces: 8 UTF-8 CRLF {} JavaScript ⌂ Port : 5500 ⌂

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P46 - Jogo Quebra-Cabeça
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Print, Help, ...
- Left Sidebar (EXPLORER):** Shows the project structure:
  - P46 - JOGO QUEBRA-CABEÇA
    - index.html
    - script.js (selected)
    - # styles.css
- Code Editor:** Displays the script.js file content. The code is written in Portuguese and handles a 4x6 jigsaw puzzle game.
- Bottom Status Bar:** Shows file information (index.html), line number (Ln 391, Col 1), spaces (Spaces: 8), encoding (UTF-8), line endings (CRLF), language (JavaScript), and port (Port : 5500).

```
// jogador ganha o jogo.
}

function verificarVitoria() {
    const estaResolvido = Array.from(pecas).every(peca, indice) => parseInt(peca.getAttribute('data-posicao')) === indice;
    // Verifica se todas as peças estão na posição correta.
    // Array.from(pecas) cria uma nova array a partir
    // da NodeList de peças.
    // every() verifica se todos os elementos no array
    // passam no teste implementado pela função fornecida.
    // A função verifica se o valor de 'data-posicao' de
    // cada peça é igual ao seu índice no array.

    if (estaResolvido) {
        // Se todas as peças estiverem na posição
        // correta (estaResolvido for true), o
        // jogador ganha o jogo.

        alert("Parabéns! Você resolveu o quebra-cabeça!");
        // Exibe um alerta parabenizando o jogador.

        incrementarPontuacao();
        // Chama a função incrementarPontuacao para
        // aumentar a pontuação do jogador.

    }
}
```

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P46 - Jogo Quebra-Cabeça
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Help
- Sidebar (EXPLORER):** Shows a project named "P46 - JOGO QUEBRA-CABEÇA" containing files: index.html, script.js, and styles.css.
- Code Editor:** The "script.js" tab is active, displaying the following code:

```
index.html styles.css script.js (active)
JS script.js > document.addEventListener('DOMContentLoaded') callback > verificarVitoria

420
421 }
422
423
424 function incrementarPontuacao() {
425
426     let pontuacao = parseInt(localStorage.getItem('pontuacao') || '0');
427     // Obtém a pontuação atual do jogador a partir do localStorage.
428     // localStorage.getItem('pontuacao') retorna a
429     //     // pontuação armazenada como string.
430     // parseInt converte a pontuação para um número inteiro.
431     // Se não houver pontuação armazenada, usa '0' como valor padrão.
432
433     pontuacao++;
434     // Incrementa a pontuação do jogador em 1.
435
436     localStorage.setItem('pontuacao', pontuacao.toString());
437     // Armazena a nova pontuação no localStorage.
438     // localStorage.setItem('pontuacao', pontuacao.toString())
439     //     // converte a pontuação para string e a armazena.
440
441     atualizarPontuacao();
442     // Chama a função atualizarPontuacao para atualizar a
443     //     // exibição da pontuação na página.
444
445 }
446
447 function trocarPecas(indiceArrastado, indiceAlvo) {
448
449     const pecaAlvo = pecas[indiceAlvo];
```

The code implements a function to increment player score and another to swap puzzle pieces.

File Edit Selection View Go Run ... ← → P46 - Jogo Quebra-Cabeça

EXPLORER ... <> index.html # styles.css JS script.js X

JS script.js > document.addEventListener('DOMContentLoaded') callback > trocarPecas

```
450 // Obtém a peça alvo (a peça sobre a qual a peça
451 // arrastada foi solta) a partir do índice.
452
453 const posicaoArrastada = pecaArrastada.style.backgroundPosition;
454 // Armazena a posição de fundo da peça arrastada em
455 // uma variável temporária.
456
457 pecaArrastada.style.backgroundPosition = pecaAlvo.style.backgroundPosition;
458 // Define a posição de fundo da peça arrastada para a
459 // posição de fundo da peça alvo.
460
461 pecaAlvo.style.backgroundPosition = posicaoArrastada;
462 // Define a posição de fundo da peça alvo para a
463 // posição de fundo da peça arrastada
464 // armazenada anteriormente.
465
466 const tempPos = pecaArrastada.getAttribute('data-posicao');
467 // Armazena a posição original da peça arrastada em
468 // uma variável temporária.
469
470 pecaArrastada.setAttribute('data-posicao', pecaAlvo.getAttribute('data-posicao'));
471 // Define a posição original da peça arrastada para a
472 // posição original da peça alvo.
473
474 pecaAlvo.setAttribute('data-posicao', tempPos);
475 // Define a posição original da peça alvo para a
476 // posição original da peça arrastada
477 // armazenada anteriormente.
478
479 }
```

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P46 - Jogo Quebra-Cabeça
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Print, Help, ...
- Left Sidebar (EXPLORER):**
  - P46 - JOGO QUEBRA-CABEÇA
    - index.html
    - script.js**
    - # styles.css
- Central Area:** Script.js tab selected.

```
index.html    # styles.css    JS script.js    X
JS script.js > document.addEventListener('DOMContentLoaded') callback

480
481     function atualizarPontuacao() {
482
483         let pontuacao = parseInt(localStorage.getItem('pontuacao') || '0');
484         // Obtém a pontuação atual do jogador a partir do localStorage.
485         // localStorage.getItem('pontuacao') retorna a
486         // pontuação armazenada como string.
487         // parseInt converte a pontuação para um número inteiro.
488         // Se não houver pontuação armazenada, usa '0' como valor padrão.
489
490         pontuacaoDisplay.textContent = `Pontuação: ${pontuacao}`;
491         // Atualiza o conteúdo de texto do elemento pontuacaoDisplay
492         // para exibir a pontuação atual.
493         // `Pontuação: ${pontuacao}` é uma string template
494         // que inclui a pontuação atual.
495
496     }
497
498 });

```
- Bottom Left Sidebar:**
  - OUTLINE
  - TIMELINE
- Bottom Status Bar:** Line 480, Col 1, Spaces: 8, UTF-8, CRLF, {}, JavaScript, Port : 5500

# **UC: Projetos**

## **Assunto:**

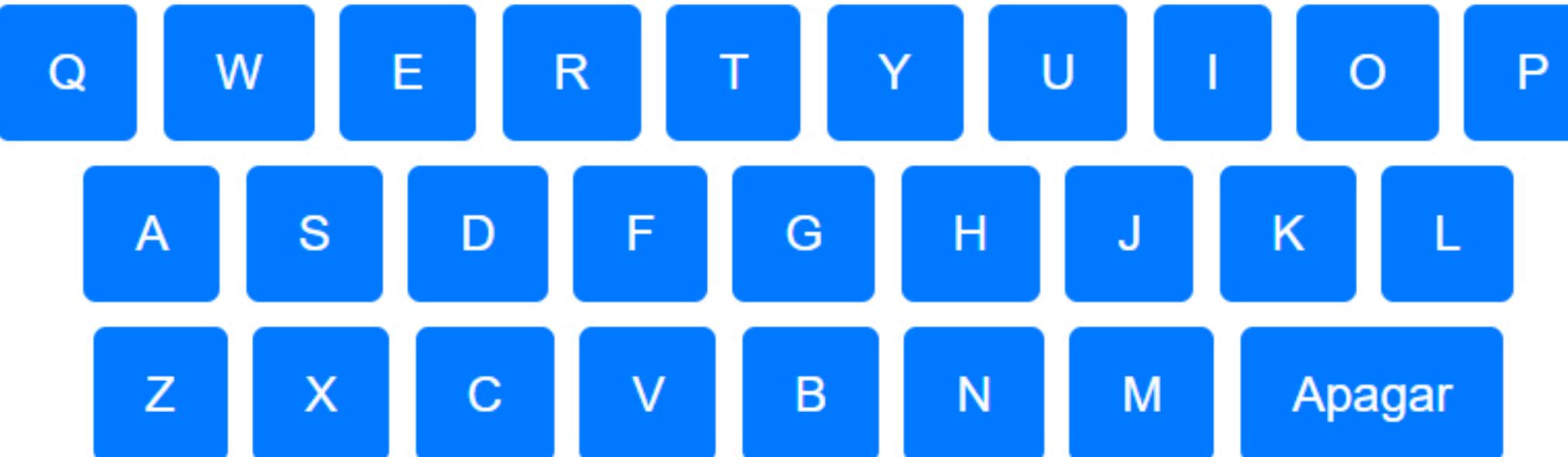
### Small Projects

### P47 – Jogo de Anagrama

# Jogo de Anagrama

Pontuação: 0

# SACA



Verificar

Dica

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P47 - Jogo de Anagrama
- Explorer:** Shows a project named "P47 - JOGO DE ANAGRAMA" containing files: index.html, script.js, and style.css.
- Code Editor:** The "index.html" file is open, displaying the following code with annotations in Portuguese:

```
<!DOCTYPE html>
<!-- Declara que este documento é um documento HTML5. --&gt;
&lt;html lang="pt-BR"&gt;
    <!-- O elemento raiz &lt;html&gt; inicia o documento HTML,
        especificando o idioma (português do Brasil)
        com o atributo 'lang'. --&gt;
    &lt;head&gt;
        <!-- O elemento &lt;head&gt; contém metadados (dados sobre dados)
            que geralmente não são exibidos diretamente
            na página web. --&gt;
        &lt;meta charset="UTF-8"&gt;
            <!-- A tag &lt;meta&gt; define a codificação de caracteres
                para o documento como UTF-8, que inclui quase
                todos os caracteres e símbolos do mundo. --&gt;
        &lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt;
            <!-- Esta tag &lt;meta&gt; configura a viewport (área de
                visualização) para controlar o layout em dispositivos
                móveis, ajustando a largura da viewport para a
                largura do dispositivo e definindo o nível
                inicial de zoom como 1.0. --&gt;
        &lt;title&gt;Jogo de Anagrama&lt;/title&gt;
            <!-- Define o título do documento, que é exibido
                na aba do navegador. --&gt;
        &lt;link rel="stylesheet" href="style.css"&gt;
    &lt;/head&gt;
    &lt;body&gt;
        &lt;div&gt;</pre>

The code editor has a dark theme and includes line numbers and syntax highlighting for HTML, CSS, and JavaScript.


```









EXPLORER



✓ P47 - JOGO DE ANAGRAMA



<> index.html

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

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P47 - Jogo de Anagrama
- Explorer:** Shows a project named "P47 - JOGO DE ANAGRAMA" containing files: index.html, script.js, and style.css.
- Code Editor:** The file "index.html" is open, showing the following code:

```
<span class="close" onclick="fecharModal('modal-erro')">&times;</span>
<!-- Elemento &lt;span&gt; com a classe 'close', contendo um
     caractere de multiplicação (&amp;times;) que
     funciona como um botão de fechar.
     O atributo 'onclick' chama a função JavaScript 'fecharModal()'
     com o ID deste modal como argumento, permitindo que o
     modal seja fechado dinamicamente. --&gt;
&lt;p&gt;Tente novamente.&lt;/p&gt;
<!-- Tag &lt;p&gt; (parágrafo) contendo a mensagem 'Tente novamente.',
     que é exibida quando o usuário comete um erro no jogo. --&gt;
&lt;/div&gt;
<!-- Fim da &lt;div&gt; com classe 'modal-content'. --&gt;
&lt;/div&gt;
<!-- Fim da &lt;div&gt; do modal 'modal-erro'. --&gt;
&lt;script src="script.js"&gt;&lt;/script&gt;
<!-- A tag &lt;script&gt; com o atributo 'src' que aponta
     para 'script.js'. Este script contém o JavaScript
     necessário para o funcionamento das funcionalidades
     interativas da página, incluindo a lógica do jogo e o
     controle dos modais. --&gt;
&lt;/body&gt;
<!-- Fim do elemento &lt;body&gt;, que contém todos os
     elementos visíveis da página. --&gt;</pre>

The code is annotated with comments explaining the purpose of various HTML elements and their associated JavaScript logic.

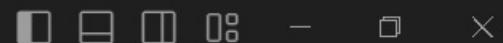

```



File Edit Selection View Go Run

←

🔍 P47 - Jogo de Anagrama



EXPLORER



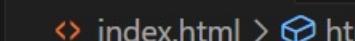
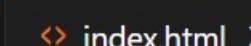
✓ P47 - JOGO DE ANAGRAMA



<> index.htm



JS script.js



181 </htm

182 <!-- Fim do documento HTML. -->



> OUTLINE

> TIMELINE



⊗ 0 △ 0 ⓐ 0

Ln 181, Col 4 Spaces: 6 UTF-8 CRLF HTML ⚡ Port : 5500



A screenshot of a code editor interface, likely Visual Studio Code, displaying a file named `# style.css`. The editor shows the following CSS code:

```
31     |     centro, usado aqui com flexbox. */
32
33     align-items: center;
34     /* Alinha os itens filhos do body verticalmente ao
35      |     centro, também em contexto de flexbox. */
36
37     height: 100vh;
38     /* Define a altura do corpo para 100vh (viewport height),
39      |     garantindo que ele ocupe toda a altura da janela de
40      |     visualização. */
41
42 }
43
44 .container {
45
46     max-width: 900px;
47     /* Define a largura máxima do contêiner para 900 pixels
48      |     para evitar que ele se torne demasiadamente largo
49      |     em telas grandes. */
50
51     width: 100%;
52     /* Define a largura do contêiner para 100% do seu
53      |     contêiner pai (neste caso, body), mas não
54      |     excedendo 900px devido ao max-width. */
55
56     padding: 20px;
57     /* Adiciona um preenchimento de 20 pixels em todas as
58      |     direções dentro do contêiner, criando espaço entre o
59      |     conteúdo interno e as bordas do contêiner. */
60
```

The code uses CSS properties like `align-items: center`, `height: 100vh`, `max-width: 900px`, and `padding: 20px` to style the `body` and `.container` elements.



A screenshot of a code editor interface, likely Microsoft Visual Studio Code, displaying a file named `# style.css`. The editor shows a portion of the CSS code for a game, specifically for styling an `h1` element and a selector for the container when it is being hovered over.

The visible code includes:

```
91
92 }
93
94
95 h1 {
96
97     margin-top: 0;
98     /* Remove a margem superior do elemento <h1>,
99        fazendo com que ele se alinhe mais de perto
100       com os elementos acima dele. */
101
102     color: #333;
103     /* Define a cor do texto do cabeçalho para
104        um cinza escuro (#333). */
105
106     font-size: 2.5em;
107     /* Define o tamanho da fonte para 2.5 vezes o
108        tamanho da fonte base do elemento, aumentando a
109        importância visual do cabeçalho. */
110
111     font-weight: bold;
112     /* Aplica o peso da fonte como negrito,
113        aumentando a ênfase no texto do cabeçalho. */
114
115 }
116
117 #Pontuacao {
118
119     font-size: 1.5em;
120     /* Define o tamanho da fonte do elemento com ID 'Pontuacao'
```

The code editor's sidebar shows other files in the project: `index.html`, `script.js`, and `# style.css`. The bottom status bar indicates the current file is `# style.css`, line 91, column 2, with 6 spaces, using UTF-8 encoding, and a port of 5500 is open.

A screenshot of a code editor interface, likely Visual Studio Code, displaying a file named `# style.css`. The editor shows several lines of CSS code with accompanying comments explaining the styling rules.

```
index.html # style.css X  
# style.css > #pontuacao  
121     | para 1.5 vezes o tamanho da fonte base, fazendo com  
122     | que a pontuação seja destacada visualmente. */  
123  
124     margin-top: 10px;  
125     /* Adiciona uma margem superior de 10 pixels para separar  
126     | visualmente a pontuação dos elementos acima dela. */  
127  
128     color: #555;  
129     /* Define a cor do texto para um cinza médio (#555),  
130     | proporcionando uma leitura fácil sem ser muito  
131     | agressivo visualmente. */  
132  
133 }  
134  
135 #palavra {  
136  
137     font-size: 5.5em;  
138     /* Define o tamanho da fonte para 5.5 vezes o tamanho da  
139     | fonte base, destacando a palavra que os jogadores  
140     | devem adivinhar como uma parte central do jogo. */  
141  
142     margin-top: 20px;  
143     /* Adiciona uma margem superior de 20 pixels, criando  
144     | espaço entre a palavra e os elementos acima dela,  
145     | ajudando a separar visualmente as seções da  
146     | interface do usuário. */  
147  
148     color: #0a3766;  
149     /* Define a cor do texto para um azul escuro (#0a3766),  
150     | que proporciona uma aparência distinta e agradável
```

A screenshot of a code editor interface, likely Visual Studio Code, displaying a file named `# style.css`. The editor shows several lines of CSS code with accompanying comments explaining the styling rules. The code is organized into two main sections: one for the element `#palavra` and another for the element `#resultado`.

```
index.html # style.css X
# style.css > #palavra
151     | visualmente. */
152
153     font-weight: bold;
154     /* Aplica o peso da fonte como negrito, tornando a palavra
155     | mais dominante e visível na interface do usuário. */
156
157     text-transform: uppercase;
158     /* Transforma todo o texto do elemento 'palavra' em
159     | maiúsculas, aumentando a sua visibilidade e impacto. */
160
161 }
162
163 #resultado {
164
165     font-size: 2.5em;
166     /* Define o tamanho da fonte do elemento com ID 'resultado'
167     | para 2.5 vezes o tamanho da fonte base, garantindo
168     | que as mensagens de feedback sejam grandes e claras. */
169
170     margin-top: 20px;
171     /* Adiciona uma margem superior de 20 pixels para distanciar o
172     | feedback dos elementos acima dele, ajudando a segmentar
173     | visualmente a interface. */
174
175     color: #333;
176     /* Define a cor do texto para um cinza escuro (#333), o
177     | que ajuda a manter a legibilidade enquanto fornece
178     | contraste suficiente com o fundo. */
179
180     min-height: 40px;
```

A screenshot of a code editor interface, likely Visual Studio Code, displaying a file named `# style.css`. The editor shows several lines of CSS code with comments explaining the styling rules. The code is organized into two main sections: one for the element `#palavra` and another for the element `#resultado`.

```
index.html # style.css X
# style.css > #palavra
151     | visualmente. */
152
153     font-weight: bold;
154     /* Aplica o peso da fonte como negrito, tornando a palavra
155     | mais dominante e visível na interface do usuário. */
156
157     text-transform: uppercase;
158     /* Transforma todo o texto do elemento 'palavra' em
159     | maiúsculas, aumentando a sua visibilidade e impacto. */
160
161 }
162
163 #resultado {
164
165     font-size: 2.5em;
166     /* Define o tamanho da fonte do elemento com ID 'resultado'
167     | para 2.5 vezes o tamanho da fonte base, garantindo
168     | que as mensagens de feedback sejam grandes e claras. */
169
170     margin-top: 20px;
171     /* Adiciona uma margem superior de 20 pixels para distanciar o
172     | feedback dos elementos acima dele, ajudando a segmentar
173     | visualmente a interface. */
174
175     color: #333;
176     /* Define a cor do texto para um cinza escuro (#333), o
177     | que ajuda a manter a legibilidade enquanto fornece
178     | contraste suficiente com o fundo. */
179
180     min-height: 40px;
```

A screenshot of a code editor interface, likely Visual Studio Code, displaying a CSS file for a game. The top bar shows the title "P47 - Jogo de Anagrama". The left sidebar has icons for Explorer, Search, Problems, and Outline/Timeline. The Explorer panel shows files: index.html, script.js, and style.css (which is selected). The main editor area shows the following CSS code:

```
211 < index.html # style.css X  
212 # style.css > ...  
213  
214 .tecla {  
215     margin: 5px;  
216     /* Adiciona margens de 5 pixels em torno de cada tecla  
217      para garantir espaço entre elas, evitando que  
218      toquem umas nas outras. */  
219     padding: 15px 20px;  
220     /* Aplica um preenchimento interno de 15 pixels na  
221      vertical e 20 pixels na horizontal em cada tecla,  
222      aumentando a área de clique e melhorando a  
223      aparência visual. */  
224     font-size: 1.2em;  
225     /* Define o tamanho da fonte das teclas para 1.2 vezes o  
226      tamanho da fonte base do elemento pai, tornando o  
227      texto nas teclas mais legível. */  
228  
229     background-color: #007bff;  
230     /* Define a cor de fundo das teclas para um azul  
231      brilhante (#007bff), o que ajuda a tecla a se  
232      destacar visualmente. */  
233  
234     color: #fff;  
235     /* Define a cor do texto das teclas para branco (#fff),  
236      garantindo um bom contraste com o fundo azul para  
237      facilidade de leitura. */  
238  
239     border: none;
```

A screenshot of a code editor interface, likely Visual Studio Code, displaying a file named `# style.css`. The editor shows the following CSS code:

```
/* Remove qualquer borda padrão das teclas,  
   contribuindo para um design mais limpo e moderno. */  
  
border-radius: 5px;  
/* Arredonda os cantos das teclas com um raio de 5 pixels,  
   suavizando o design geral e melhorando a estética. */  
  
cursor: pointer;  
/* Muda o cursor para um ponteiro quando sobrevoado,  
   indicando que a tecla é clicável. */  
  
transition: background-color 0.3s, transform 0.3s;  
/* Aplica uma transição suave para mudanças na cor de  
   fundo e transformações da tecla, como quando é  
   sobrevoada, melhorando a resposta visual interativa. */  
  
}  
  
.tecla:hover {  
  
background-color: #078b23;  
/* Muda a cor de fundo das teclas para um verde  
   escuro (#078b23) quando são sobrevoadas, fornecendo um  
   feedback visual imediato que a tecla é interativa. */  
  
transform: scale(1.05);  
/* Aumenta ligeiramente o tamanho da tecla em 5% quando  
   sobrevoada, adicionando um efeito visual sutil que  
   destaca a tecla sob o cursor do usuário. */
```

The code editor has a dark theme. The sidebar on the left includes icons for Explorer, Search, Problems, and Outline. The status bar at the bottom shows file statistics and a port number.

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P47 - Jogo de Anagrama
- Icons:** Explorer, Find, Open, Save, Help, Outline, Timeline.
- Left Sidebar (EXPLORER):** P47 - JOGO DE ANAGRAMA (index.html, script.js, # style.css)
- Current File:** # style.css
- Code Content:** CSS rules for a button class (#btn-verificar, #btn-dica). The code includes:
  - Margin-top: 20px; (margin-top: 20px;)
  - Padding: 15px 30px; (padding: 15px 30px;)
  - Font-size: 1.2em; (font-size: 1.2em;)
  - Color: #fff; (color: #fff;)
  - Border: none; (border: none;)
  - Border-radius: 5px; (border-radius: 5px;)Comments explain the purpose of each rule, such as adding margin for spacing and increasing font size for readability.
- Bottom Status Bar:** Line 271, Column 1, Spaces: 6, UTF-8, CRLF, CSS, Port: 5500.

File Edit Selection View Go Run ... ← → 🔍 P47 - Jogo de Anagrama

EXPLORER ...

P47 - JOGO DE ANAGRAMA

- index.html
- script.js
- # style.css

# style.css #btn-verificar

```
301 /* Arredonda os cantos dos botões com um raio de 5
302 pixels, criando um visual mais suave e
303 esteticamente agradável. */
304
305 cursor: pointer;
306 /* Muda o cursor para um ponteiro quando o usuário
307 passa o mouse sobre os botões, indicando que
308 são clicáveis. */
309
310 transition: background-color 0.3s, transform 0.3s;
311 /* Adiciona efeitos de transição suaves para as
312 mudanças de cor de fundo e escala dos botões,
313 melhorando a interatividade visual ao interagir
314 com eles. */
315
316 }
317
318 #btn-verificar {
319
320 background-color: #28a745;
321 /* Define a cor de fundo do botão 'Verificar' como
322 verde (#28a745), uma cor associada a ações
323 positivas como confirmar ou prosseguir. */
324
325 }
326
327 #btn-verificar:hover {
328
329 background-color: #218838;
330 /* Muda a cor de fundo do botão 'Verificar' para um
```

Ln 301, Col 7 Spaces: 6 UTF-8 CRLF CSS ⚡ Port : 5500

A screenshot of a code editor interface, likely Visual Studio Code, displaying a file named `# style.css`. The editor shows CSS code for a game, specifically for styling buttons. The code includes rules for `#btn-verificar:hover`, `#btn-dica`, and `#btn-dica:hover`. The CSS uses color hex codes (#218838, #ffc107, #e0a800) and transform properties (scale(1.05)) to create visual feedback for user interaction.

```
index.html # style.css < P47 - JOGO DE ANAGRAMA index.html script.js # style.css

# style.css > #btn-verificar:hover
331     tom de verde mais escuro (#218838) quando o mouse
332     está sobre ele, fornecendo um feedback visual
333     de interatividade. */
334
335     transform: scale(1.05);
336     /* Aumenta ligeiramente a escala do botão 'Verificar'
337     em 5% ao passar o mouse, dando um efeito de
338     destaque e aumentando a sensação de interatividade. */
339
340 }
341
342 #btn-dica {
343
344     background-color: #ffc107;
345     /* Define a cor de fundo do botão 'Dica' como
346     amarelo (#ffc107), uma cor que chama atenção e é
347     frequentemente usada para destacar elementos
348     importantes ou ferramentas auxiliares. */
349
350 }
351
352 #btn-dica:hover {
353
354     background-color: #e0a800;
355     /* Altera a cor de fundo do botão 'Dica' para um
356     amarelo mais escuro (#e0a800) quando o mouse
357     está sobre ele, fornecendo um feedback visual
358     de que o botão é interativo. */
359
360     transform: scale(1.05);
```

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P47 - Jogo de Anagrama
- Icons:** Explorer, Search, Find, Open, Save, Help, Outline, Timeline.
- Left Sidebar (EXPLORER):** P47 - JOGO DE ANAGRAMA (index.html, script.js, style.css)
- Active Tab:** # style.css
- Code Content:** A CSS file for styling a modal window. The code includes comments in Portuguese explaining the purpose of various styles like font-size, color, and position.

```
/* Aumenta ligeiramente a escala do botão 'Dica' em 5% ao passar o mouse, aumentando sua visibilidade e interatividade. */
}

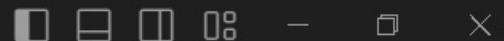
/* Estilos do Modal */
.modal {
    display: none;
    /* Inicialmente esconde o modal, definindo display como none. O modal só será exibido mediante ação específica (como clicar em um botão). */
    position: fixed;
    /* O modal é posicionado em relação à janela do navegador, e não se move ao rolar a página. */
    z-index: 1;
    /* Define a pilha de ordem do modal. Um z-index mais alto traz o modal para frente de outros elementos na página. */
    left: 0;
    top: 0;
    /* Posiciona o modal exatamente no topo e à esquerda da janela do navegador, cobrindo toda a área visível quando exibido. */
    width: 100%;
```



File Edit Selection View Go Run ·

← →

🔍 P47 - Jogo de Anagrama



EXPLORER

✓ P47 - JOGO DE ANAGRAMA



< index.html # style.css >

```
# style.css > .modal

391     height: 100%;  
392     /* Define a largura e a altura do modal para  
393      | cobrir toda a tela, desde que position  
394      | esteja como fixed. */  
395  
396     overflow: auto;  
397     /* Permite que o conteúdo dentro do modal seja  
398      | rolável se o conteúdo exceder o tamanho  
399      | da janela. */  
400  
401     background-color: □rgb(0, 0, 0);  
402     background-color: □rgba(0, 0, 0, 0.4);  
403     /* Define a cor de fundo do modal. Primeiro é  
404      | declarado um fundo preto sólido e então  
405      | substituído por preto com transparência (40% de  
406      | opacidade), permitindo que o conteúdo por  
407      | trás do modal seja parcialmente visível. */  
408  
409     justify-content: center;  
410     align-items: center;  
411     /* Utiliza Flexbox para centralizar o conteúdo do  
412      | modal (definido em .modal-content) tanto  
413      | horizontal quanto verticalmente dentro do  
414      | contêiner .modal. */  
415  
416 }  
417  
418 .modal-content {  
419  
420     background-color: ■#fefefe;
```

-

Ln

91, Col 3 S

aces: 6 1

TF-8 CR

- CSS

② Port : 55

A screenshot of a code editor interface, likely Visual Studio Code, displaying a file named '# style.css' under the 'P47 - JOGO DE ANAGRAMA' project in the Explorer sidebar. The editor shows CSS code for styling a modal content container. The code includes comments explaining the purpose of each rule, such as defining the background color, margin, padding, border, width, max-width, and border-radius.

```
/* Define a cor de fundo do conteúdo do modal  
   para branco quase completo. */  
  
margin: 15% auto;  
/* Centraliza o conteúdo do modal horizontalmente e  
   o posiciona a 15% do topo da página. */  
  
padding: 20px;  
/* Adiciona um preenchimento interno de 20 pixels em  
   todas as direções dentro do contêiner de  
   conteúdo do modal, criando espaço entre o  
   conteúdo e as bordas do modal. */  
  
border: 1px solid #888;  
/* Adiciona uma borda sólida com 1 pixel de  
   espessura e cor cinza (#888) ao redor do  
   contêiner do conteúdo do modal. */  
  
width: 80%;  
/* Define a largura do conteúdo do modal para 80%  
   da largura do seu contêiner pai (.modal),  
   ajudando a manter o conteúdo responsivo e  
   gerenciável em diferentes tamanhos de tela. */  
  
max-width: 500px;  
/* Estabelece uma largura máxima de 500 pixels  
   para o conteúdo do modal, garantindo que  
   não se torne muito largo em telas grandes. */  
  
border-radius: 10px;
```

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 syntax highlighting. The left sidebar shows project files: index.html, script.js, and style.css. The main area shows the content of style.css.

```
index.html # style.css < X
# style.css > .modal-content
451 /* Aplica um raio de borda de 10 pixels para
452     arredondar os cantos do conteúdo do modal,
453     suavizando sua aparência visual. */
454
455     text-align: center;
456     /* Centraliza o texto e outros elementos dentro
457         do conteúdo do modal, proporcionando um
458         layout equilibrado e fácil de ler. */
459
460     font-size: 4.2em;
461     /* Aumenta o tamanho da fonte para 4.2 vezes o
462         tamanho padrão da fonte do documento,
463         fazendo com que o texto dentro seja
464         mais legível. */
465
466 }
467
468
469 .close {
470
471     color: #aaa;
472     /* Define a cor do texto para um cinza
473         claro (#aaa), que é útil mas ainda legível. */
474
475     float: right;
476     /* Posiciona o elemento `.close` à direita do
477         seu container, geralmente usado para botões
478         de fechar em modais ou componentes interativos. */
479
480     font-size: 28px;
```

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P47 - Jogo de Anagrama
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Split, ...
- Left Sidebar (EXPLORER):** Shows a tree view of files under "P47 - JOGO DE ANAGRAMA": index.html, script.js, and style.css. The style.css file is currently selected.
- Code Editor:** The "style.css" tab is active, displaying the following CSS code:

```
/* style.css > .close
 *
481 /* Estabelece o tamanho da fonte para 28 pixels,
482 | garantindo que o ícone de fechamento seja
483 | facilmente clicável e visível. */
484
485 font-weight: bold;
486 /* Aplica um peso de fonte negrito para tornar o
487 | elemento `.close` mais proeminente e claro
488 | para o usuário. */
489
490 }
491
492 .close:hover,
493 .close:focus {
494
495 color: black;
496 /* Muda a cor do texto para preto ao passar o
497 | mouse ou ao focar (por exemplo, via teclado ou
498 | interação de acessibilidade), proporcionando
499 | um feedback visual claro. */
500
501 text-decoration: none;
502 /* Remove qualquer decoração de texto (como
503 | sublinhado) que possa ser aplicada por padrão
504 | pelo navegador ao focar ou passar o mouse. */
505
506 cursor: pointer;
507 /* Muda o estilo do cursor para 'pointer',
508 | indicando que o elemento é clicável. */
509
510 }
```
- Bottom Status Bar:** Shows icons for file operations (Save, Undo, Redo, Find, Replace, etc.) and status information: Ln 481, Col 2, Spaces: 6, UTF-8, CRLF, CSS, Port: 5500, and a refresh icon.

A screenshot of a code editor interface, likely Visual Studio Code, displaying a CSS file named `# style.css`. The file contains CSS code for a game, specifically for mobile devices with a maximum width of 600px. It includes a media query for padding and a rule for `h1` elements with a font size of 2em.

```
index.html # style.css < ... # style.css > ... 511 512 513 @media (max-width: 600px) { 514     /* Esta media query aplica as seguintes regras CSS 515         quando a largura da janela do navegador é igual ou 516         inferior a 600 pixels. É comumente usada para 517         adaptar estilos a telas de dispositivos móveis. */ 518 519     .container { 520         padding: 15px; 521         /* Reduz o preenchimento dentro do elemento com a 522             classe `container` para 15 pixels em todos os 523             lados. Essa redução ajuda a maximizar o espaço 524             disponível para o conteúdo dentro de containers em 525             telas menores, essencial para manter o conteúdo 526             legível e acessível sem ocupar espaço excessivo. */ 527 528     } 529 530     h1 { 531 532         font-size: 2em; 533         /* Ajusta o tamanho da fonte do cabeçalho (elemento <h1>) 534             para 2 vezes o tamanho da fonte base do documento. 535             Essa redução é proporcional e mantém a hierarquia 536             visual importante, garantindo que os cabeçalhos 537             continuem a se destacar sem dominar o espaço 538             visualmente restrito. */ 539 540 }
```

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P47 - Jogo de Anagrama
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Refresh, Help.
- Left Sidebar (EXPLORER):**
  - P47 - JOGO DE ANAGRAMA
    - index.html
    - script.js
    - # style.css (selected)
- Central Area:** Editor showing index.html and # style.css tabs.
  - # style.css:** Contains CSS rules for a word-guessing game.
    - Rule 541: `#palavra { font-size: 2em; }` (Defines the font size for the key word element.)

/\* Define o tamanho da fonte para a palavra-chave que os usuários devem adivinhar (elemento com ID `#palavra`) para 2 vezes o tamanho da fonte base, mantendo sua proeminência na interface do jogo, garantindo que seja facilmente legível mesmo em telas menores. \*/
    - Rule 551: `}`
    - Rule 552: `.tecla {`
    - Rule 553: `font-size: 1em;`  
Rule 554: `padding: 10px;`

/\* Ajusta as teclas (elementos com classe `.tecla`) para terem um tamanho de fonte igual ao tamanho base da fonte do documento e um preenchimento de 10 pixels. Isso as torna acessíveis e fáceis de interagir em telas pequenas, enquanto maximiza o uso do espaço disponível no teclado virtual. \*/
    - Rule 563: `}`
    - Rule 564: `#btn-verificar, #btn-dica {`
    - Rule 565: `font-size: 1em;`  
Rule 566: `padding: 10px 20px;`

/\* Define o tamanho da fonte dos botões 'Verificar'
    - Rule 570:
- Bottom Status Bar:** X ⊗ 0 △ 0 ⌂ 0 Ln 541, Col 1 Spaces: 6 UTF-8 CRLF CSS ⚡ Port : 5500



File Edit Selection View Go Run ·

← →

🔍 P47 - Jogo de Anagrama



EXPLORER

✓ P47 - JOGO DE ANAGRAMA

<> index.htm

JS script.js

```
# style.css
```

<> index.htm

# style.css



> OUTLINE

> TIMELINE

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P47 - Jogo de Anagrama
- Toolbar:** Back, Forward, Home, Stop, Refresh, Minimize, Maximize, Close.
- Left Sidebar (EXPLORER):** Shows a tree view with the project folder "P47 - JOGO DE ANAGRAMA" expanded, containing files: index.html, script.js (selected), and style.css.
- Code Editor:** The "script.js" tab is active, displaying the following code:

```
const palavras = [
    'CASA', 'BOLA', 'GATO', 'CHUVA', 'SOL', 'AMIGO', 'LUA', 'FOGO', 'VENTO', 'PATO',
    'LIVRO', 'MESA', 'TOALHA', 'JANELA', 'CARRO', 'ESCOLA', 'FLORESTA', 'PRAIA', 'COMPUTADOR', 'TELEFONE',
    'CADERNO', 'LAMPADA', 'ESPADA', 'DINHEIRO', 'PERFUME', 'CHAVE', 'QUADRO', 'PLANTA', 'FILME', 'MUSICA',
    'REVISTA', 'BOTAO', 'CAIXA', 'SAPATO', 'CACHORRO', 'PASSARINHO', 'CEBOLA', 'TOMATE', 'CELULAR', 'RELOGIO',
    'GARRAFA', 'COPO', 'CANETA', 'LENCOL', 'TRAVESSEIRO', 'COLHER', 'GARFO', 'PRATO', 'CORTINA', 'ESPONJA'
];

/* Declara uma constante chamada 'palavras' que é um array
   de strings. Cada string é uma palavra que os jogadores
   tentarão adivinhar no jogo de anagrama. */

let palavraAleatoria = '';
/* Declara uma variável 'palavraAleatoria' inicializada
   como uma string vazia. Esta variável será usada para
   armazenar uma das palavras do array 'palavras'
   escolhida aleatoriamente. */

let tentativaAtual = '';
/* Declara uma variável 'tentativaAtual', inicialmente
   vazia, que será utilizada para armazenar as entradas
   do usuário à medida que tentam adivinhar a palavra
   correta. */

let pontuacao = 0;
/* Declara uma variável 'pontuacao' e inicializa-a
   com 0. Esta variável é usada para rastrear a
   pontuação do jogador. */
```

The code defines an array of words, initializes variables for a random word selection and user input, and initializes a score variable.

File Edit Selection View Go Run ... ← → P47 - Jogo de Anagrama

EXPLORER ... index.html # style.css JS script.js X

JS script.js > novaPalavra

```
31 function novaPalavra() {  
32     /* Define uma função chamada 'novaPalavra'. Esta  
33      função é usada para selecionar uma nova palavra  
34      aleatória do array e preparar o jogo para uma  
35      nova rodada. */  
36  
37     palavraAleatoria = palavras[pontuacao % palavras.length];  
38     /* Atribui à variável 'palavraAleatoria' uma palavra  
39      do array 'palavras'. O índice é determinado pelo  
40      resto da divisão da pontuação atual pelo número  
41      de palavras no array. Isso garante que a seleção  
42      de palavras seja cíclica e dependa da pontuação  
43      do jogador. */  
44  
45     document.getElementById('palavra').innerText = embaralharPalavra(palavraAleatoria);  
46     /* Localiza o elemento HTML com o id 'palavra' e  
47      define seu texto interno para o valor retornado  
48      pela função 'embaralharPalavra', passando  
49      a 'palavraAleatoria' como argumento.  
50      'embaralharPalavra' é uma função que presumivelmente  
51      embaralha as letras da palavra para criar um  
52      anagrama que o jogador deve resolver. */  
53  
54     document.getElementById('resultado').innerText = '';  
55     /* Localiza o elemento HTML com o id 'resultado' e  
56      limpa seu conteúdo, definindo-o para uma string  
57      vazia. Isso é útil para remover mensagens ou  
58      resultados de tentativas anteriores quando uma  
59      nova palavra é apresentada. */  
60
```

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P47 - Jogo de Anagrama
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Print, Help, ...
- Sidebar:** EXPLORER, P47 - JOGO DE ANAGRAMA (index.html, script.js, # style.css), OUTLINE, TIMELINE.
- Code Area:** Script.js (selected tab)
- Code Content:** A snippet of JavaScript code for an anagram game. The code includes functions for initializing variables, shuffling words, and manipulating character arrays.

```
index.html # style.css JS script.js X  
JS script.js > novaPalavra  
tentativaAtual = '';  
/* Reinicia a variável 'tentativaAtual', limpando as  
entradas anteriores do usuário ao começar  
uma nova rodada. */  
}  
function embaralharPalavra(palavra) {  
    /* Declaração da função 'embaralharPalavra' com um  
    parâmetro 'palavra', que é a string a  
    ser embaralhada. */  
  
    let palavraEmbaralhada = '';  
    /* Inicializa uma variável 'palavraEmbaralhada' como  
    uma string vazia. Esta variável armazenará a  
    versão embaralhada da palavra original à medida  
    que os caracteres são rearranjados. */  
  
    const palavraArray = palavra.split('');  
    /* Utiliza o método 'split('')' para dividir a string 'palavra'  
    em um array de caracteres. Por exemplo,  
    "CASA" torna-se ['C', 'A', 'S', 'A'].  
    Isso permite manipular cada caractere individualmente. */  
  
    while (palavraArray.length > 0) {  
        /* Inicia um loop 'while' que continua enquanto houver  
        caracteres no array 'palavraArray'. Este loop  
        repetirá até que todos os caracteres sejam removidos  
        do array original e adicionados à string 'embaralhada'. */
```

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

- File Bar:** File Edit Selection View Go Run ...
- Search Bar:** P47 - Jogo de Anagrama
- Icons:** Explorer, Search, Find, Open, Save, Help, Outline, Timeline.
- Project Explorer:** P47 - JOGO DE ANAGRAMA (index.html, script.js, # style.css).
- Editor Tabs:** index.html, # style.css, script.js (active).
- Editor Content:** script.js (JavaScript code for shuffling words).

```
const indiceAleatorio = Math.floor(Math.random() * palavraArray.length);
/* Gera um índice aleatório dentro do intervalo do
   tamanho atual do array 'palavraArray'.
'Math.random()' gera um número decimal entre 0 e 1 (exclusivo),
que é então multiplicado pelo comprimento do
array para criar um número no intervalo necessário.
'Math.floor()' é utilizado para arredondar o número
para baixo, resultando em um índice válido de
array (que começa do 0). */

palavraEmbaralhada += palavraArray.splice(indiceAleatorio, 1)[0];
/* Utiliza o método 'splice()' para remover 1 elemento
   no índice 'indiceAleatorio' do array 'palavraArray'.
'splice()' retorna um array contendo os itens
removidos, e '[0]' acessa o primeiro (e único)
elemento deste array.
Esse elemento é então concatenado à
string 'palavraEmbaralhada'.
Isso garante que cada caractere seja selecionado
aleatoriamente e movido da lista original para
a nova string embaralhada, sem repetição. */

return palavraEmbaralhada;
/* Após o loop terminar, retorna a string 'palavraEmbaralhada',
que agora contém todos os caracteres da palavra
original em uma nova ordem aleatória. */
```
- Status Bar:** Line 91, Col 1, Spaces: 6, UTF-8, CRLF, {} JavaScript, Port: 5500.

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P47 - Jogo de Anagrama
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Print, Help, ...
- Sidebar (EXPLORER):** Shows a tree view with the project folder "P47 - JOGO DE ANAGRAMA" expanded, containing files: index.html, script.js (selected), and style.css.
- Code Editor:** The "script.js" tab is active, displaying the following code:

```
121 }
122
123 function criarTecladoVirtual() {
124     /* Declara a função 'criarTecladoVirtual', utilizada
125      para construir dinamicamente um teclado na
126      interface do usuário, facilitando a entrada
127      de dados pelo jogador. */
128
129     const tecladoContainer = document.getElementById('teclado-virtual');
130     /* Localiza o elemento no DOM com o ID 'teclado-virtual',
131      que servirá como o contêiner principal para o teclado.
132      Este contêiner irá hospedar todas as teclas criadas
133      dinamicamente. */
134
135     const linhasTeclado = [
136         'QWERTYUIOP',
137         'ASDFGHJKL',
138         'ZXCVBNM'
139     ];
140     /* Define um array de strings, onde cada string
141      representa uma linha do teclado QWERTY. Essa
142      estrutura facilita a criação de teclas em linhas que
143      correspondem ao layout familiar de um teclado
144      físico. */
145
146     linhasTeclado.forEach(linha => {
147         /* Inicia um loop para iterar sobre cada linha do
148         teclado. 'linha' representa uma das strings no
149         array 'linhasTeclado'. */
150     }
151 }
```

The code implements a function to create a virtual keyboard by defining key layouts and iterating over them.



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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P47 - Jogo de Anagrama
- Toolbar:** Back, Forward, Home, Stop, Refresh, Minimize, Maximize, Close.
- Left Sidebar (EXPLORER):** Shows a tree view of files under "P47 - JOGO DE ANAGRAMA": index.html, script.js, and # style.css. The "script.js" file is currently selected.
- Code Editor:** The "script.js" tab is active. The code is a snippet from a game application, specifically for creating a virtual keyboard layout. It uses document.createElement('button') to create buttons, appends them to a div, and adds click event listeners to handle character deletion.

```
index.html    # style.css    JS script.js    X
JS script.js > criarLayoutVirtual > linhasTeclado.forEach() callback > forEach() callback
181         linhaDiv.appendChild(botao);
182         /* Adiciona o botão criado como um filho do <div> da
183          | linha atual, inserindo efetivamente a tecla no
184          | layout do teclado. */
185      });
186
187      tecladoContainer.appendChild(linhaDiv);
188      /* Adiciona o <div> que contém a linha completa de
189       | teclas ao contêiner do teclado, montando
190       | progressivamente o layout completo do teclado
191       | na página. */
192
193  });
194
195  });
196
197 const botaoApagar = document.createElement('button');
198 /* Cria um novo botão que será usado para apagar
199   | caracteres inseridos pelo usuário. */
200
201 botaoApagar.innerText = 'Apagar';
202 /* Define o texto do botão como 'Apagar', indicando
203   | sua função. */
204
205 botaoApagar.classList.add('tecla', 'tecla-apagar');
206 /* Adiciona duas classes ao botão: 'tecla' para
207   | estilização geral de teclas e 'tecla-apagar'
208   | para estilização específica do botão de apagar. */
209
210 botaoApagar.onclick = apagarUltimaLetra;
```

- Bottom Status Bar:** Shows file information: Ln 181, Col 1, Spaces: 6, UTF-8, CRLF, and a port number Port : 5500.

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

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P47 - Jogo de Anagrama
- Toolbars:** Standard icons for file operations.
- Left Sidebar (EXPLORER):**
  - P47 - JOGO DE ANAGRAMA
    - index.html
    - script.js
    - # style.css
- Central Area:** Script.js tab selected.

```
index.html    # style.css    JS script.js    X
JS script.js > criarLayoutVirtual

211     /* Define a função que será chamada quando o botão de
212      apagar for clicado. 'apagarUltimaLetra' é uma
213      função que, remove o último
214      caractere da entrada atual do usuário. */
215
216     tecladoContainer.appendChild(botaoApagar);
217     /* Adiciona o botão de apagar ao contêiner principal do
218      teclado, colocando-o no teclado virtual como
219      uma tecla funcional. */
220
221 }
222
223
224 function adicionarLetra(letra) {
225     /* Declaração da função 'adicionarLetra' com 'letra' como
226      parâmetro. Esta função é chamada quando uma
227      tecla do teclado virtual é pressionada. */
228
229     tentativaAtual += letra;
230     /* Adiciona a 'letra' pressionada ao fim da string 'tentativaAtual'.
231      Esta variável acumula as entradas do usuário à
232      medida que elas vão sendo inseridas. */
233
234     atualizarTentativa();
235     /* Chama a função 'atualizarTentativa' para atualizar o
236      display do jogo, mostrando a entrada
237      atualizada do usuário. */
238
239 }
240
```
- Bottom Status Bar:** Icons for file operations, status (Ln 211, Col 4), and settings (Spaces: 6, UTF-8, CRLF, {} JavaScript, Port: 5500).

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P47 - Jogo de Anagrama
- Left Sidebar (EXPLORER):** P47 - JOGO DE ANAGRAMA (expanded), containing index.html, script.js, and # style.css.
- Central Area:** Script.js file open, showing the following code:

```
241 function apagarUltimaLetra() {
242     /* Declaração da função 'apagarUltimaLetra', que é
243      | responsável por remover a última letra
244      | inserida na 'tentativaAtual'. */
245
246     tentativaAtual = tentativaAtual.slice(0, -1);
247     /* Utiliza o método 'slice' para criar uma nova string,
248      | removendo o último caractere da string 'tentativaAtual'.
249      | O parâmetro '-1' indica que o corte deve ser feito até o
250      | penúltimo caractere, efetivamente removendo o último. */
251
252     atualizarTentativa();
253     /* Após modificar 'tentativaAtual', chama 'atualizarTentativa'
254      | para atualizar o display e mostrar a nova
255      | string sem a última letra. */
256
257 }
258
259
260
261 function atualizarTentativa() {
262     /* Declaração da função 'atualizarTentativa', que
263      | atualiza a visualização da tentativa do
264      | usuário na interface do jogo. */
265
266     document.getElementById('resultado').innerText = tentativaAtual;
267     /* Localiza o elemento HTML com o ID 'resultado' e
268      | atualiza seu texto interno para o valor da
269      | variável 'tentativaAtual', que contém as
270      | letras inseridas pelo usuário até o momento.
```

The code implements two functions: `apagarUltimaLetra()` which removes the last character from the user's current guess, and `atualizarTentativa()` which updates the display with the current guess.

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P47 - Jogo de Anagrama
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Print, Help, ...
- Left Sidebar (EXPLORER):** Shows a tree view with a node expanded: P47 - JOGO DE ANAGRAMA, containing index.html, script.js (selected), and # style.css.
- Code Editor:** The current file is script.js, tabbed under JS. The code is as follows:

```
index.html    # style.css    JS script.js    X
JS script.js > atualizarTentativa
271     Isso exibe as letras escolhidas pelo usuário na tela,
272         permitindo-lhes acompanhar visualmente sua
273             progressão na formação da palavra. */
274
275 }
276
277
278 function verificarTentativa() {
279     /* Declaração da função 'verificarTentativa', que é
280         chamada quando o jogador decide verificar se
281             a palavra digitada está correta. */
282
283     if (tentativaAtual === palavraAleatoria) {
284         /* Início de uma estrutura condicional que compara a
285             variável 'tentativaAtual' (a entrada do usuário)
286             com 'palavraAleatoria' (a palavra correta).
287                 Se forem iguais, o bloco de código dentro
288                     do if será executado. */
289
290         pontuacao += 1;
291         /* Incrementa a variável 'pontuacao' em 1, aumentando a
292             pontuação do jogador por acertar a palavra. */
293
294         atualizarPontuacao();
295         /* Chama a função 'atualizarPontuacao', que atualiza a
296             visualização da pontuação na interface do usuário.
297                 Essa função manipula o DOM para
298                     refletir a nova pontuação. */
299
300         document.getElementById('resultado').innerText = 'Parabéns! Você acertou!';
```

The code implements a function to verify user input against a random word, update the score, and display a success message.

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P47 - Jogo de Anagrama
- Toolbar:** Minimize, Maximize, Close, Fullscreen, Print, Help, ...
- Sidebar:** EXPLORER, P47 - JOGO DE ANAGRAMA (index.html, script.js, # style.css), OUTLINE, TIMELINE.
- Code Area:** Script.js (selected tab). The code handles user input validation and feedback.

```
index.html    # style.css  JS script.js  X
JS script.js > verificarTentativa
301     /* Acessa o elemento HTML com o ID 'resultado' e
302         define seu texto para 'Parabéns! Você acertou!', 
303         fornecendo um feedback positivo ao jogador. */
304
305     setTimeout(novaPalavra, 2000);
306     /* Utiliza a função 'setTimeout' para chamar a
307         função 'novaPalavra' após um delay de
308         2000 milissegundos (2 segundos). Isso dá ao
309         jogador um tempo para ver a mensagem de
310         acerto antes de iniciar uma nova rodada. */
311
312 } else {
313     /* Parte do condicional que executa se 'tentativaAtual'
314         não for igual a 'palavraAleatoria' (ou seja, o
315         jogador errou a palavra). */
316
317     mostrarModal('modal-erro');
318     /* Chama a função 'mostrarModal' com o argumento 'modal-erro',
319         que exibe um modal de erro na tela,
320         informando ao jogador que ele não acertou a
321         palavra e incentivando uma nova tentativa. */
322
323 }
324
325 tentativaAtual = '';
326 /* Reinicia a variável 'tentativaAtual', limpando a
327         entrada anterior do usuário, preparando para
328         uma nova tentativa ou nova palavra. */
329
330 atualizarTentativa();
```



EXPLORER



✓ P47 - JOGO DE ANAGRAMA



```
< index.html # style.css JS script.js > X  
JS script.js > verificarTentativa  
331     /* Chama a função 'atualizarTentativa', que  
332         atualiza algum elemento de interface do usuário  
333             relacionado à entrada atual do jogador,  
334                 refletindo o reset da tentativa. */  
335  
336 }  
337  
338 function atualizarPontuacao() {  
339     /* Declara a função 'atualizarPontuacao', que é responsável  
340         por atualizar a visualização da pontuação do  
341             usuário na página web e armazenar essa pontuação  
342                 no localStorage. */  
343  
344     document.getElementById('pontuacao').innerText = `Pontuação: ${pontuacao}`;  
345     /* Acessa o elemento HTML com o ID 'pontuacao' e atualiza  
346         seu conteúdo de texto para mostrar a pontuação  
347             atual do jogador.  
348             Usa-se a interpolação de string (com backticks) para  
349                 inserir dinamicamente o valor da variável 'pontuacao'  
350                     no texto. */  
351  
352     localStorage.setItem('anagramaPontos', pontuacao);  
353     /* Usa o objeto 'localStorage' para armazenar permanentemente a  
354         pontuação atual do usuário sob a chave 'anagramaPontos'.  
355             Isso permite que a pontuação seja persistida mesmo se a  
356                 página for fechada ou recarregada. */  
357  
358 }  
359  
360 function carregarPontuacao() {
```

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P47 - Jogo de Anagrama
- Icons:** Explorer, Search, Find, Open, Close, Help, Outline, Timeline.
- Left Sidebar (EXPLORER):** P47 - JOGO DE ANAGRAMA (expanded) containing index.html, script.js, and style.css.
- Active Tab:** script.js
- Code Content:** A function named carregarPontuacao() is defined at line 360. It retrieves a saved score from localStorage and converts it to an integer using parseInt(). Finally, it calls the atualizarPontuacao() function at line 388.

```
function carregarPontuacao() {
    /* Declara a função 'carregarPontuacao', que é
       responsável por carregar a pontuação do
       jogador a partir do armazenamento local
       quando a página é carregada ou atualizada. */

    const pontuacaoSalva = localStorage.getItem('anagramaPontos');
    /* Usa o método 'getItem' do objeto 'localStorage' para
       recuperar o valor armazenado anteriormente sob a
       chave 'anagramaPontos'.
       A pontuação recuperada é armazenada na
       constante 'pontuacaoSalva'. */

    if (pontuacaoSalva) {
        /* Verifica se existe uma pontuação salva. Se a
           variável 'pontuacaoSalva' contiver algum
           valor (ou seja, se não for null ou undefined), o
           bloco de código dentro do if será executado. */

        pontuacao = parseInt(pontuacaoSalva);
        /* Converte o valor de 'pontuacaoSalva' de uma string
           para um número inteiro usando a função 'parseInt'.
           Isso é necessário porque os valores armazenados no
           localStorage são sempre strings. A variável 'pontuacao' é
           atualizada com esse valor numérico. */

    }

    atualizarPontuacao();
    /* Chama a função 'atualizarPontuacao' para atualizar a
       pontuação exibida na interface. */
}
```

- Bottom Status Bar:** Line 360, Col 3, Spaces: 6, UTF-8, CRLF, {} JavaScript, Port: 5500.

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Left Sidebar:** EXPLORER, P47 - JOGO DE ANAGRAMA (index.html, script.js, style.css), OUTLINE, TIMELINE.
- Central Area:** A tab bar with index.html, # style.css, JS script.js (selected), and a close button. Below it is a search bar with the placeholder "P47 - Jogo de Anagrama".
- Code Editor Content:** The script.js file contains the following code:

```
JS script.js > carregarPontuacao
390     interface do usuário com a pontuação carregada.
391     Isso garante que a pontuação exibida na página seja
392         consistente com o valor recuperado do localStorage. */
393
394 }
395
396 function mostrarModal(modalId) {
397     /* Declaração da função 'mostrarModal' com um
398         parâmetro 'modalId', que é o ID do modal a
399             ser mostrado. */
400
401     const modal = document.getElementById(modalId);
402     /* Obtém o elemento modal do DOM usando seu ID fornecido,
403         permitindo a manipulação do mesmo. */
404
405     modal.style.display = 'flex';
406     /* Altera a propriedade de estilo 'display' do modal
407         para 'flex', tornando-o visível. Isso ativa a
408             exibição do modal, que por padrão está
409                 oculto (display: none), e o configura com
410                     display flexível para centralizar o conteúdo
411                         dentro dele. */
412
413 }
414
415 function fecharModal(modalId) {
416     /* Declaração da função 'fecharModal' com um
417         parâmetro 'modalId', que é o ID do modal a
418             ser fechado. */
419 }
```
- Bottom Status Bar:** Icons for file operations (undo, redo, save, etc.), status indicators (Ln 390, Col 6, Spaces: 6, UTF-8, CRLF), language (JavaScript), port (Port: 5500), and a settings gear icon.

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P47 - Jogo de Anagrama
- Toolbar:** Back, Forward, Home, Refresh, Stop, Minimize, Maximize, Close.
- Sidebar (EXPLORER):** Shows a tree view with a node expanded: P47 - JOGO DE ANAGRAMA, containing index.html, script.js (selected), and # style.css.
- Code Editor:** The script.js file is open, showing the following code:

```
index.html # style.css JS script.js X  
JS script.js > fecharModal  
420 const modal = document.getElementById(modalId);  
421 /* Semelhante à função 'mostrarModal', obtém o  
   elemento modal do DOM usando seu ID. */  
422  
423 modal.style.display = 'none';  
424 /* Configura a propriedade de estilo 'display' do  
   modal para 'none', escondendo-o da visualização.  
   Isso é usado para fechar o modal quando  
   não é mais necessário, como após o usuário ler a  
   dica ou após fechar uma notificação de erro. */  
425  
426 }  
427  
428 function mostrarDica() {  
429 /* Declara a função 'mostrarDica', utilizada para mostrar a  
   palavra correta ao usuário como uma dica. */  
430  
431 document.getElementById('dica-palavra').innerText = palavraAleatoria;  
432 /* Acessa o elemento HTML com o ID 'dica-palavra' e  
   define seu texto interno para a  
   variável 'palavraAleatoria',  
   que é a palavra corrente que o usuário precisa  
   adivinhar. Isso mostra a palavra diretamente  
   ao usuário como uma dica. */  
433  
434 mostrarmodal('modal-dica');  
435 /* Chama a função 'mostrarModal' com o argumento 'modal-dica',  
   que abre o modal onde a dica é exibida. */  
436  
437 }  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449 }
```

The code implements two functions: fecharModal and mostrarDica. The fecharModal function hides a modal by setting its display property to 'none'. The mostrarDica function sets the innerText of an element with ID 'dica-palavra' to the value of the variable 'palavraAleatoria'.

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P47 - Jogo de Anagrama
- Toolbar:** Minimize, Maximize, Close
- Sidebar (Left):**
  - EXPLORER
  - P47 - JOGO DE ANAGRAMA
    - index.html
    - script.js
    - # style.css
  - Search icon
  - File icon
  - Folder icon
  - Help icon
  - Outline icon
  - Timeline icon
- Code Editor (Main Area):**
  - File: script.js
  - Content:

```
450
451 // Chamadas de função para inicializar o jogo
452
453 carregarPontuacao();
454 /* Chama a função 'carregarPontuacao' ao iniciar o jogo
455     para recuperar e mostrar qualquer pontuação
456     anterior que tenha sido salva no localStorage. */
457
458 novaPalavra();
459 /* Chama a função 'novaPalavra' para iniciar o jogo com
460     uma nova palavra aleatória, dando ao usuário a
461     chance de começar ou continuar jogando. */
462
463 criarTecladoVirtual();
464 /* Chama a função 'criarTecladoVirtual' para construir o
465     teclado na interface do usuário, que o jogador usará
466     para inserir suas tentativas de adivinhar a palavra. */
467
```
- Bottom Bar:** X, ⌂ 0 ▲ 0 ⌂ 0, Ln 450, Col 1, Spaces: 6, UTF-8, CRLF, {}, JavaScript, ⚙ Port : 5500

# **UC: Projetos Assunto: Small Projects P48 – Relógio Mundial**

## Relógio Mundial

**Nova York**

06/11/2024 - 20:14:32

**Londres**

07/11/2024 - 01:14:32

**Tóquio**

07/11/2024 - 09:14:32

**Sydney**

07/11/2024 - 10:14:32

**Dubai**

07/11/2024 - 04:14:32

**Moscou**

07/11/2024 - 03:14:32

**São Paulo**

06/11/2024 - 21:14:32

**Pequim**

07/11/2024 - 08:14:32

**Berlim**

07/11/2024 - 02:14:32

**Paris**

07/11/2024 - 02:14:32

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P48 - Relógio Mundial
- Toolbar:** Standard window control icons (minimize, maximize, close) and a refresh icon.
- Sidebar:** Explorer view showing a project folder "P48 - RELÓGIO MUNDIAL" containing "index.html", "script.js", and "# styles.css".
- Code Editor:** The main area displays the content of "index.html".

```
<!DOCTYPE html>
<!-- Define o tipo de documento como HTML5 --&gt;
&lt;html lang="pt-BR"&gt;
    <!-- Abertura da tag &lt;html&gt; com atributo de idioma
        definido como português do Brasil --&gt;
&lt;head&gt;
    &lt;meta charset="UTF-8"&gt;
        <!-- Define a codificação de caracteres do documento como UTF-8 --&gt;
    &lt;meta name="viewport" content="width=device-width, initial-scale=1.0"&gt;
        <!-- Define a configuração do viewport para garantir que a
            página seja responsiva.
            O "width=device-width" define a largura da página para seguir a
            largura da tela do dispositivo.
            O "initial-scale=1.0" define o nível de zoom inicial
            quando a página é carregada. --&gt;
    &lt;title&gt;Relógio Mundial&lt;/title&gt;
        <!-- Define o título da página que aparece na aba do navegador --&gt;
    &lt;link rel="stylesheet" href="styles.css"&gt;
        <!-- Vincula o arquivo de estilos externo (CSS) para a página --&gt;
&lt;/head&gt;
    <!-- Fecha a seção &lt;head&gt; onde são colocadas as meta-informações
        sobre o documento --&gt;</pre>
```
- Status Bar:** Shows file statistics: Line 1, Column 1, Spaces: 4, UTF-8, CRLF, HTML, Port: 5500, and a notification icon.

File Edit Selection View Go Run ... ⏪ ⏩ 🔎 P48 - Relógio Mundial

EXPLORER ...

P48 - RELÓGIO MUNDIAL

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

index.html X

index.html > html

```
31
32 <body>
33     <!-- Abertura da tag &lt;body&gt; onde todo o conteúdo visível da
34     |     |     |     |     página será colocado --&gt;
35
36     &lt;div class="relogio-mundial"&gt;
37         <!-- Div contêiner principal para os relógios mundiais, com a
38         |         |         |         classe "relogio-mundial" para estilização --&gt;
39
40         &lt;h1&gt;Relógio Mundial&lt;/h1&gt;
41         <!-- Título principal da página, que será exibido como um
42         |         |         |         cabeçalho de nível 1 --&gt;
43
44         &lt;div id="relogios"&gt;
45             <!-- Div onde os relógios serão inseridos dinamicamente pelo
46             |             |             |             JavaScript. O id "relogios" é usado para identificar
47             |             |             |             este elemento no script JavaScript --&gt;
48
49             &lt!-- Os relógios serão inseridos aqui pelo JavaScript --&gt;
50
51         &lt;/div&gt;
52         <!-- Fecha a div com id "relogios" --&gt;
53
54     &lt;/div&gt;
55     <!-- Fecha a div com a classe "relogio-mundial" --&gt;
56
57     &lt;script src="script.js"&gt;&lt;/script&gt;
58     <!-- Vincula o arquivo JavaScript externo (script.js) que
59     |     |     |     conterá a lógica para atualizar os relógios --&gt;
60</pre>

Ln 31, Col 1 Spaces: 4 UTF-8 CRLF HTML ⚡ Port: 5500


```

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P48 - Relógio Mundial
- Left Sidebar (Explorer):**
  - EXPLORER
  - P48 - RELÓGIO MUNDIAL
    - index.html
    - script.js
    - styles.css
- Central Area:** A code editor window titled "index.html".
  - Content:

```
</body>
<!-- Fecha a tag &lt;body&gt; --&gt;
63
64 &lt;/html&gt;
65 &lt!-- Fecha a tag &lt;html&gt; e encerra o documento HTML --&gt;</pre>
```
  - Code navigation:
    - ↳ index.html > html > body
- Bottom Status Bar:** ✘ 0 △ 0 ⌂ 0 Ln 61, Col 1 Spaces: 4 UTF-8 CRLF HTML ⚡ Port : 5500

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P48 - Relógio Mundial
- Left Sidebar (EXPLORER):** P48 - RELÓGIO MUNDIAL (index.html, script.js, # styles.css)
- Active Editor:** # styles.css (highlighted with a blue bar)
- Code Content:** CSS rules for the body element, defining font-family, background-color, color, display, justify-content, align-items, height, and margin.
- Bottom Status Bar:** Line 1, Col 1, Spaces: 4, UTF-8, CRLF, CSS, Port: 5500

```
index.html # styles.css X
# styles.css > body
1 body {
2   font-family: 'Arial', sans-serif;
3   /* Define a família de fontes para o texto do corpo. 'Arial' é
4   |   |   a fonte principal e 'sans-serif' é a fonte alternativa */
5
6   background-color: #2C3E50;
7   /* Define a cor de fundo do corpo como um tom escuro
8   |   |   de azul (#2C3E50) */
9
10  color: #ECF0F1;
11  /* Define a cor do texto como um tom claro de cinza (#ECF0F1) */
12
13
14  display: flex;
15  /* Define o modelo de layout do corpo como flexível (flexbox),
16  |   |   permitindo alinhamento e distribuição de itens */
17
18  justify-content: center;
19  /* Alinha horizontalmente os itens do corpo ao centro */
20
21  align-items: center;
22  /* Alinha verticalmente os itens do corpo ao centro */
23
24  height: 100vh;
25  /* Define a altura do corpo como 100% da altura da
26  |   |   janela de visualização (viewport height) */
27
28  margin: 0;
29  /* Remove as margens padrão do corpo */
30
```

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P48 - Relógio Mundial
- Icons:** Minimize, Maximize, Close, Fullscreen, Help.
- Left Sidebar (EXPLORER):**
  - P48 - RELÓGIO MUNDIAL
    - index.html
    - script.js
  - # styles.css (selected)
- Central Area:** Code Editor showing index.html and # styles.css tabs.
  - # styles.css:** Contains the following CSS code:

```
31 }  
32  
33 .relogio-mundial {  
34  
35     text-align: center;  
36     /* Alinha o texto no centro do contêiner */  
37  
38     background: #34495E;  
39     /* Define a cor de fundo do contêiner como um  
40      |   |   tom de azul escuro (#34495E) */  
41  
42     padding: 20px;  
43     /* Adiciona um preenchimento interno de 20 pixels ao  
44      |   |   redor do conteúdo do contêiner */  
45  
46     border-radius: 10px;  
47     /* Define bordas arredondadas com um raio de 10 pixels */  
48  
49     box-shadow: 0 0 20px rgba(0, 0, 0, 0.3);  
50     /* Adiciona uma sombra ao redor do contêiner. A sombra não  
51      |   |   tem deslocamento horizontal (0) nem vertical (0),  
52      |   |   é espalhada em 20 pixels, e é de cor preta com 30%  
53      |   |   de opacidade (rgba(0, 0, 0, 0.3)) */  
54  
55     width: 90%;  
56     /* Define a largura do contêiner como 90% da largura do elemento pai */  
57  
58     max-width: 1200px;  
59     /* Define a largura máxima do contêiner como 1200 pixels */  
60
```
- Bottom Status Bar:** Ln 31, Col 1, Spaces: 4, UTF-8, CRLF, CSS, Port: 5500.

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P48 - Relógio Mundial
- Icons:** Explorer, Search, Repository, Preview, Help, Outline, Timeline.
- Left Sidebar (EXPLORER):** Shows a project structure under P48 - RELÓGIO MUNDIAL: index.html, script.js, and # styles.css (selected).
- Central Area:** A code editor with the following CSS code:

```
index.html # styles.css # styles.css > .relogio-mundial
61 }
62
63 .relogio-mundial h1 {
64
65     margin-bottom: 20px;
66     /* Adiciona uma margem inferior de 20 pixels para
67      |   |   separar o título do conteúdo abaixo */
68
69     font-size: 2em;
70     /* Define o tamanho da fonte do título como 2 vezes o
71      |   |   tamanho padrão da fonte */
72
73     color: #ECF0F1;
74     /* Define a cor do texto do título como um tom claro
75      |   |   de cinza (#ECF0F1) */
76
77 }
78
79 #relógios {
80
81     display: grid;
82     /* Define o modelo de layout do contêiner como grid (grade),
83      |   |   permitindo uma distribuição uniforme dos itens */
84
85     grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
86     /* Define que as colunas da grade devem se repetir
87      |   |   automaticamente para preencher o espaço disponível.
88      |   |   Cada coluna deve ter um tamanho mínimo de 200 pixels e um
89      |   |   tamanho máximo que se ajuste ao espaço disponível (1fr) */
90
```

The code defines styles for a global clock application, including a large title with a 20px bottom margin and a light gray font color (#ECF0F1), and a grid container for displaying multiple time zones.

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P48 - Relógio Mundial
- Icons:** Minimize, Maximize, Close, Fullscreen, Help, More.
- Sidebar (EXPLORER):** Shows project structure under P48 - RELÓGIO MUNDIAL: index.html, script.js, # styles.css.
- Code Area:** The current file is # styles.css. The code defines a class .relogio with various styling rules. The code is annotated with comments explaining the properties and their effects.

```
# styles.css
# relogios
gap: 20px;
/* Define um espaço (gap) de 20 pixels entre as colunas e
linhas da grade */

}

.relogio {
background: #1ABC9C;
/* Define a cor de fundo do contêiner de cada relógio
como um tom de verde claro (#1ABC9C) */

padding: 20px;
/* Adiciona um preenchimento interno de 20 pixels ao
redor do conteúdo do relógio */

border-radius: 8px;
/* Define bordas arredondadas com um raio de 8 pixels */

box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
/* Adiciona uma sombra ao redor do contêiner do relógio.
A sombra não tem deslocamento horizontal (0)
nem vertical (0), é espalhada em 10 pixels,
e é de cor preta com 20% de opacidade (rgba(0, 0, 0, 0.2)) */

transition: transform 0.2s;
/* Define uma transição suave para a propriedade 'transform'
com duração de 0.2 segundos */
```

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

- Top Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P48 - Relógio Mundial
- Icons:** Explorer, Search, Find, Split, Outline, Timeline.
- Left Sidebar (Explorer):** P48 - RELÓGIO MUNDIAL (index.html, script.js, # styles.css)
- Active File:** # styles.css
- Code Content:**

```
121     color: #2C3E50;
122     /* Define a cor do texto dentro do contêiner do relógio
123      |   |   | como um tom escuro de azul (#2C3E50) */
124
125 }
126
127 .relogio:hover {
128
129     transform: scale(1.05);
130     /* Ao passar o mouse sobre o contêiner do relógio,
131      |   |   | aplica uma transformação de escala para
132      |   |   | aumentar o tamanho em 5% */
133
134 }
135
136 .local {
137
138     font-size: 1.2em;
139     /* Define o tamanho da fonte para o nome do local
140      |   |   | como 1.2 vezes o tamanho padrão da fonte */
141
142     font-weight: bold;
143     /* Define a fonte como negrito para o nome do local */
144
145     margin-bottom: 10px;
146     /* Adiciona uma margem inferior de 10 pixels para separar o
147      |   |   | nome do local do conteúdo abaixo */
148
149     color: #ECF0F1;
150     /* Define a cor do texto do nome do local como um tom
```

- Bottom Status Bar:** Ln 121, Col 9, Spaces: 4, UTF-8, CRLF, CSS, Port : 5500





File Edit Selection View Go Run ... ⏪ ⏩ P48 - Relógio Mundial

EXPLORER ... index.html # styles.css JS script.js X

JS script.js > [?] locais

```
31 { nome: "Berlim", fusoHorario: 2 },
32 // Objeto representando Berlim com fuso horário UTC+2
33
34 { nome: "Paris", fusoHorario: 2 }
35 // Objeto representando Paris com fuso horário UTC+2
36
37 ];
38
39
40 function atualizarRelogios() {
41     // Define a função 'atualizarRelogios' que atualizará os
42     // horários dos relógios exibidos na página
43
44     const containerRelogios = document.getElementById("relogios");
45     // Seleciona o elemento HTML com o ID 'relogios' e o armazena
46     // na constante 'containerRelogios'
47
48     containerRelogios.innerHTML = "";
49     // Limpa todo o conteúdo dentro do elemento 'containerRelogios',
50     // removendo quaisquer relógios existentes
51
52     locais.forEach(local => {
53         // Itera sobre cada objeto no array 'locais'
54
55         const agora = new Date();
56         // Cria um novo objeto Date que representa a
57         // data e hora atuais
58
59         // Calcular a hora local considerando o fuso horário e o deslocamento do UTC
60         const horaLocal = new Date(agora.getTime() + (local.fusoHorario * 60 + agora.getTimezoneOffset()) *
```

OUTLINE

TIMELINE

✖ ⊗ 0 △ 0 ⌂ 0 🔍 0 ⏹ Ln 31, Col 4 Spaces: 4 UTF-8 CRLF {} JavaScript ⏹ Port: 5500 ⏹

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

- File Bar:** File Edit Selection View Go Run ...
- Search Bar:** P48 - Relógio Mundial
- Icons:** Explorer, Search, Open, Save, Outline, Timeline.
- Active File:** script.js
- Content:** A script for calculating local time from UTC.

```
index.html styles.css script.js
script.js > [?] locais

/*
Explicação da Correção:
Cálculo do Tempo Local:
agora.getTime() obtém o tempo atual em milissegundos desde a
época (1970-01-01).

(local.fusoHorario * 60 + agora.getTimezoneOffset()) * 60000
calcula o deslocamento total em milissegundos considerando o
fuso horário local e o deslocamento do UTC (horário de verão).

new Date(agora.getTime() + (local.fusoHorario * 60 + agora.getTimezoneOffset()) * 60000)
cria um novo objeto Date ajustado para o fuso horário específico.

*/
// Ajusta a hora atual para o fuso horário específico do local.
// Obtém a hora UTC atual e adiciona o fuso horário do local

const dataFormatada = horaLocal.toLocaleDateString('pt-BR');
// Formata a data ajustada para o formato 'pt-BR' (dd/mm/aaaa)

const horaFormatada = horaLocal.toLocaleTimeString('pt-BR', { hour: '2-digit', minute: '2-digit',
second: '2-digit' });
// Formata a hora ajustada para o formato 'pt-BR' (hh:mm:ss)
// com duas casas para horas, minutos e segundos

const divRelogio = document.createElement("div");
// Cria um novo elemento 'div' para representar um relógio
```

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

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P48 - Relógio Mundial
- Icons:** Explorer, Search, Problems, Outline, Timeline.
- Left Sidebar:** EXPLORE, P48 - RELÓGIO MUNDIAL, index.html, script.js, # styles.css.
- Active Tab:** script.js
- Code Content:** A script for a world clock application. The code creates three div elements: 'relogio' (containing local names), 'local' (containing local names), and 'data-hora' (containing date and time). It uses document.createElement('div') to create the elements and append them to the 'relogio' element.

```
index.html    # styles.css  JS script.js  X
JS script.js > ⚡ atualizarRelogios > ⚡ locais.forEach() callback
90         divRelogio.className = "relogio";
91         // Define a classe CSS do novo elemento 'div'
92         |         // como 'relogio' para estilização
93
94         const divLocal = document.createElement("div");
95         // Cria um novo elemento 'div' para o nome do local
96
97         divLocal.className = "local";
98         // Define a classe CSS do novo elemento 'div'
99         |         // como 'local' para estilização
100
101        divLocal.textContent = local.nome;
102        // Define o texto do novo elemento 'div' como o nome do local
103
104        const divDataHora = document.createElement("div");
105        // Cria um novo elemento 'div' para a data e hora
106
107        divDataHora.className = "data-hora";
108        // Define a classe CSS do novo elemento 'div'
109        |         // como 'data-hora' para estilização
110
111        divDataHora.textContent = `${dataFormatada} - ${horaFormatada}`;
112        // Define o texto do novo elemento 'div' como a
113        |         // data e a hora formatadas
114
115        divRelogio.appendChild(divLocal);
116        // Adiciona o elemento 'divLocal' (nome do local)
117        |         // como filho do elemento 'divRelogio'
118
119        divRelogio.appendChild(divDataHora);
```

- Status Bar:** Ln 90, Col 9, Spaces: 4, UTF-8, CRLF, {} JavaScript, Port: 5500.

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

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** P48 - Relógio Mundial
- Toolbar:** Back, Forward, Home, Stop, Refresh, Minimize, Maximize, Close.
- Sidebar (EXPLORER):** Shows the project structure under P48 - RELÓGIO MUNDIAL, including index.html, script.js, and styles.css.
- Code Editor:** The current file is script.js, containing the following code:

```
// Adiciona o elemento 'divDataHora' (data e hora)
// como filho do elemento 'divRelogio'
containerRelogios.appendChild(divRelogio);
// Adiciona o elemento 'divRelogio' completo (que
// contém o nome do local e a data/hora)
// como filho do 'containerRelogios' na página
};

setInterval(atualizarRelogios, 1000);
// Define um intervalo de tempo para chamar a
// função 'atualizarRelogios' a cada 1000
// milissegundos (1 segundo)

atualizarRelogios();
// Chama a função 'atualizarRelogios' uma vez para garantir
// que os relógios sejam atualizados imediatamente
// ao carregar a página, antes de começar o intervalo de tempo
```

**Status Bar:** Ln 120, Col 7, Spaces: 4, UTF-8, CRLF, {}, JavaScript, Port: 5500.