

# Atividades

## <!-- Atividade 01-->

Em CSS, a especificidade é uma medida usada para determinar quais estilos serão aplicados a um elemento quando houver regras conflitantes. Quando há várias regras que se aplicam a um mesmo elemento, o navegador usa a especificidade para decidir qual regra prevalece. A especificidade é calculada com base em um sistema de pontos que depende dos seletores usados na regra CSS.

### Como calcular a especificidade

A especificidade de um seletor é geralmente representada por quatro valores (A, B, C, D), onde cada letra corresponde a um tipo de seletor:

1. A: Seletor de estilo embutido (aplicado diretamente ao elemento com o atributo `style=""`). Este seletor tem uma especificidade muito alta.
2. B: Seletor de ID (`#id`).
3. C: Seletor de classe (`.classe`), seletor de pseudo-classe (`:hover`, `:nth-child`) e seletor de atributo (`[type="text"]`).
4. D: Seletor de elemento (`p`, `h1`, etc.) e pseudo-elemento (`::before`, `::after`).

A especificidade é calculada atribuindo um valor a cada categoria de seletor. O valor é contado em cada posição da hierarquia e lido da esquerda para a direita.

### Exemplo Prático de Cálculo de Especificidade

Vamos analisar três regras CSS com diferentes seletores:

1. `h1` → especificidade: (0, 0, 0, 1)
  - Só há um seletor de elemento (`h1`), então a especificidade é 1 no último dígito.
2. `.header h1` → especificidade: (0, 0, 1, 1)
  - Temos uma classe (`.header`) e um elemento (`h1`), resultando em 1 na posição de classes e 1 para o seletor de elemento.
3. `#main .header h1` → especificidade: (0, 1, 1, 1)
  - Temos um ID (`#main`), uma classe (`.header`) e um elemento (`h1`). A especificidade é, portanto, 1 na posição de IDs, 1 na posição de classes, e 1 na posição de elementos.

Nesse caso, se essas três regras fossem aplicadas ao mesmo elemento h1 com ID main e classe header, o navegador aplicaria os estilos de #main .header h1, pois tem a especificidade mais alta

## <!-- Atividade 02 -->

Aqui estão cinco seletores CSS úteis com exemplos de uso:

### 1. Seletor de Atributo

Aplica estilos com base em atributos específicos de um elemento.

```
<a href="https://example.com" target="_blank">Externo</a>
```

```
<a href="https://example.com">Interno</a>
```

```
<style>
```

```
a[target="_blank"] {  
    color: rgb(217, 0, 0);  
}
```

```
</style>
```

Explicação: Este seletor [target="\_blank"] seleciona todos os links com o atributo target igual a \_blank, aplicando a cor vermelha.

### 2. Seletor de Irmão Adjacente

Estiliza um elemento que vem imediatamente após um elemento especificado.

```
<h2>Título 1</h2>
```

```
<p>Este parágrafo será estilizado.</p>
```

```
<h2>Título 2</h2>
```

```
<p>Este parágrafo não será estilizado.</p>
```

```
<style>
```

```
h2 + p {  
  color: blue;  
}
```

```
</style>
```

Explicação: O seletor h2 + p aplica estilos ao parágrafo que vem imediatamente após um h2, definindo a cor azul.

### 3. Seletor de Filho Direto

Seleciona apenas os filhos diretos de um elemento pai específico.

```
<div class="container">
```

```
<p>Filho direto</p>
```

```
<div>
```

```
<p>Não é filho direto</p>
```

```
</div>
```

```
</div>
```

```
<style>
```

```
.container > p {  
  font-weight: bold;  
}
```

```
</style>
```

Explicação: O seletor .container > p aplica estilo apenas aos parágrafos que são filhos diretos de .container, deixando o parágrafo dentro da div aninhada sem a formatação.

### 4. Seletor de Pseudo-classe :nth-child()

Seleciona o enésimo elemento filho de um tipo específico.

```
<ul>

  <li>Item 1</li>

  <li>Item 2</li>

  <li>Item 3</li>

  <li>Item 4</li>

</ul>
```

```
<style>

  li:nth-child(2) {

    color: green;

  }

</style>
```

Explicação: O seletor `li:nth-child(2)` aplica o estilo apenas ao segundo item da lista, mudando a cor para verde.

## 5. Seletor de Pseudo-elemento `::after`

Adiciona conteúdo após um elemento.

```
<p class="nota">Nota:</p>

<style>

  .nota::after {

    content: " Esta é uma nota importante.";

    font-style: italic;

  }

</style>
```

Explicação: O seletor `.nota::after` insere texto ao final do elemento com a classe `.nota`, adicionando uma observação em itálico.

Esses seletores permitem criar uma ampla gama de estilos, ajudando a tornar o design das páginas mais versátil e dinâmico. Você pode testar esses exemplos em um editor HTML para ver os estilos aplicados.