

# Profissão: Engenheiro Front-End



# Boas práticas de CSS

# GUIA DA AULA 1



# Determine regras de CSS

● **Introdução**

● **Escrita do  
Código CSS**

● **Navegador  
e o seletor**



Acompanhe aqui  
os temas que  
serão tratados  
na videoaula



# Introdução

A aplicação de estilos no CSS ocorre do mais específico para o menos específico:

1. Estilos inline;
2. Ids;
3. classes, pseudoclasses e atributos;
4. Elementos e pseudo elementos.



Neste módulo vamos conhecer alguns temas avançados de CSS, como a **especificidade** e **metodologias** para escrita de código CSS: **BEM** e **SMACSS**.

Consulte o código que será escrito durante este módulo no *link*:

[https://github.com/ogiansouza/curso\\_frontend\\_ebac/tree/boas\\_praticas\\_css](https://github.com/ogiansouza/curso_frontend_ebac/tree/boas_praticas_css)



## Escrita do código CSS

Escrever o código CSS é relativamente simples, porém precisamos ter alguns cuidados e entender alguns conceitos, como a **especificidade**, que no CSS diz o quanto específica, forte, um seletor é.

O efeito cascata do CSS pode fazer com que uma regra CSS sobrescreva outra, por exemplo:

```
h1 { color: red; }
```

```
h1 { color: blue; }
```

A cor aplicada ao elemento H1 será azul, por essa ser a última regra para este seletor.



Porém podemos contornar esse comportamento deixando o seletor mais específico:

```
h1.título { color: red; }  
h1 { color: blue; }
```

Com isso, o H1 que tiver a classe “título” terá a cor vermelha, por essa regra ser mais específica.



Existe uma regra para calcularmos a especificidade de uma regra CSS.

A especificidade é definida pelas contagens de Ids –classes e afins – elementos e afins.

Por exemplo:

**h1 // 001 – possui 0 Ids, 0 classes e 1 elemento**

**h1.title // 011 possui 0 Ids, 1 classe e 1 elemento**

A segunda seleção é mais específica pois 011 é maior que 001.



Existe uma exceção na especificidade, que é o uso da palavra **!importante** logo após o valor da propriedade. Por exemplo:

```
h1 { color: red !important; }
```

```
h1.titulo { color: blue; }
```

Neste caso, a primeira regra será aplicada, por possuir o !importante.

Porém, isso é uma má prática que deve ser evitada e contornada com a escrita de um código CSS escalável e de boa manutenibilidade.

Acesse a ferramenta **Specificity Calculator** que auxilia no cálculo da especificidade de um seletor, disponível no *link* <https://specificity.keegan.st/>.



## Navegador e o seletor

O navegador lê um seletor CSS da **direita** para a **esquerda**, leve em consideração o seletor: **nav**. Como a leitura é feita da direita para à esquerda, primeiro o navegador irá selecionar todos os elementos A que existirem na página, após isso, ele irá filtrar por aqueles que estiverem dentro do elemento NAV.

Pode existir centenas de *links* na página e o navegador irá verificar cada um deles para ver se estão dentro do elemento NAV, isso faz com que o nosso seletor não seja performático. A melhor abordagem seria criar uma classe `.nav-link` e aplicá-la aos elementos que desejamos estilizar. Assim o navegador terá que verificar menos elementos no documento, sendo mais performático.

