



escola
britânica de
artes criativas
& tecnologia

Profissão: Engenheiro Front-End



Boas práticas de CSS

GUIA DA AULA 2



Aplique a metodologia **SMACSS**



Acompanhe aqui
os temas que
serão tratados
na videoaula

- **Sobre o SMACSS**
- **Base**
- **Layout**
- **Módulos**
- **Estados**
- **Temas**
- **Nomenclaturas**



Sobre o SMACSS

O **SMACSS** (Scalable and Modular Architecture for CSS) é uma arquitetura utilizada para construção de código CSS. Nela, temos a divisão do CSS nas categorias:

- Base;
- *Layout*;
- Módulos;
- Estados;
- Temas.

Acesse a documentação completa do SMACSS no link <https://smacss.com/>.



Base

Nos arquivos base estilizamos as regras mais básicas da página. É essencialmente um *reset*, onde podemos remover as margens, espaçamentos e definir outras propriedades que seguirão um padrão na página.

Exemplo de conteúdo de um arquivo base:

```
h1, h2, h3, h4, h5, h6 {  
    margin: 0;  
}  
  
body {  
    font-family: Roboto, sans-serif;  
}
```



Layout

No SMACSS os componentes são divididos entre componentes maiores e menores, e um *layout* é um componente maior, que tem a função de conter outros componentes, podemos considerá-los como *containers*.

Exemplo de conteúdo de um arquivo *layout*:

```
#header {  
  padding: 16px;  
  display: flex;  
}
```



No exemplo anterior utilizamos o ID como seletor do elemento, na documentação do SMACSS vemos o uso de Ids na estilização de *layouts*, porém não há nenhum problema em utilizar classes.

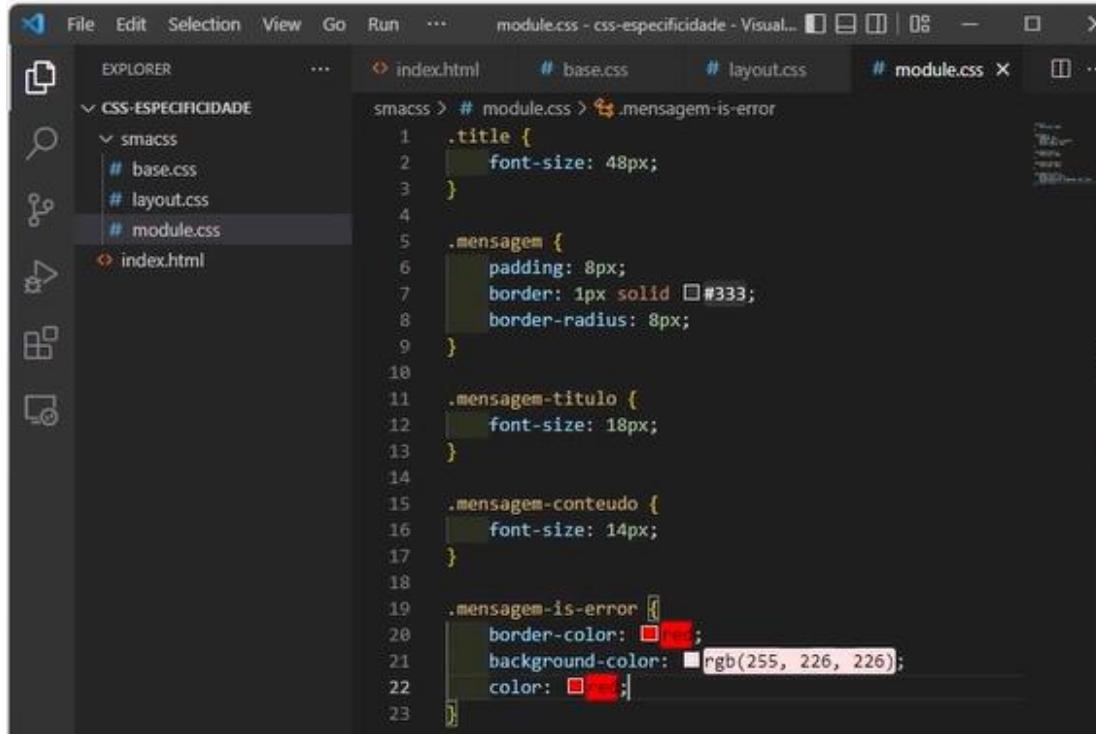


Módulos

Um módulo é um componente estilizado. Geralmente estão localizados dentro de um *layout*. O módulo deve ser independente e ter a mesma aparência em qualquer lugar que seja inserido. Por exemplo, um *card* de produto, deve possuir o mesmo estilo dentro de uma listagem de produtos ou quando utilizado dentro de uma página de *post*.

Um módulo pode conter outros módulos, como no caso de um *card* de produto, podemos ter um módulo de detalhes do produto.





The screenshot shows a Visual Studio Code interface with a dark theme. The left sidebar has icons for Explorer, Search, Open, and File Manager. The Explorer panel shows a folder named 'CSS-ESPECIFICIDADE' containing a 'smacss' folder with files 'base.css', 'layout.css', and 'module.css', and an 'index.html' file. The 'module.css' file is currently selected and open in the main editor area. The code in 'module.css' is as follows:

```
smacss > # module.css > .mensagem-is-error
1 .title {
2     font-size: 48px;
3 }
4
5 .mensagem {
6     padding: 8px;
7     border: 1px solid #333;
8     border-radius: 8px;
9 }
10
11 .mensagem-titulo {
12     font-size: 18px;
13 }
14
15 .mensagem-contudo {
16     font-size: 14px;
17 }
18
19 .mensagem-is-error {
20     border-color: #f00;
21     background-color: #fff;
22     color: #f00;
23 }
```



Estados

No SMACSS, um estado é utilizado para sobre escrever um estilo. Por convenção utilizamos o prefixo “is-” no nome da classe. Por exemplo, em um campo de formulário inválido:

```
.is-invalid{  
    border-color: red;  
}
```

```
<input class="form-input is-invalid" type="text"  
placeholder="Nome completo" />
```



Temas

Um tema no SMACSS tem o objetivo de afetar a aparência das páginas com cores e até mesmo diferentes fontes. Exemplo:

```
.button{ // modules/button.css  
    padding: 8px;  
    font-weight: bold;  
}
```

```
.button{ // themes/private.css  
    background-color: #f0932b;  
    color: #fff;  
}
```



Podemos ver esse caso de uso em *sites* de bancos tradicionais, onde há seguimentos de clientes, como *convencional*, *private*, *exclusive* etc.

O site é o mesmo, a estrutura é a mesma, o que muda é o tema, a identidade visual. Um exemplo prático são os *sites* da Globo, onde temos apenas a mudança do logo e cor de fundo nos cabeçalhos:



Nomenclaturas

O SMACSS nos propõe a utilização de alguns prefixos para nomear as classes:

l-ou **layout-** para a construção de layouts;

is- para estados, como: is-open, is-close, is-hidden;

Obs.: Não utilize prefixos para o módulo, visto que essas classes serão a maioria, ficaria muito verboso escrever module- antes do nome da classe de qualquer módulo.

