

# Construção de Software para Web

Estilo

Prof. Guilherme Zucatelli  
2º Período - 2022

# Programação Web

Relembrando...



- **HTML (*Hypertext Markup Language*):** Linguagem principal, responsável pela estruturação dos elementos existentes nas páginas Web.
- **CSS (*Cascading Style Sheets*):** Ferramenta de design, estilo, adotada para descrever/modificar a apresentação dos elementos da página.
- **JS (*JavaScript*):** Linguagem de script para páginas Web.

# Elementos de Estilo

- **CSS (*Cascading Style Sheets*):**

O CSS é uma tecnologia que nos permite especificar o **estilo dos elementos visuais** (por vezes chamada de interface do usuário, interface do cliente) de uma aplicação Web.

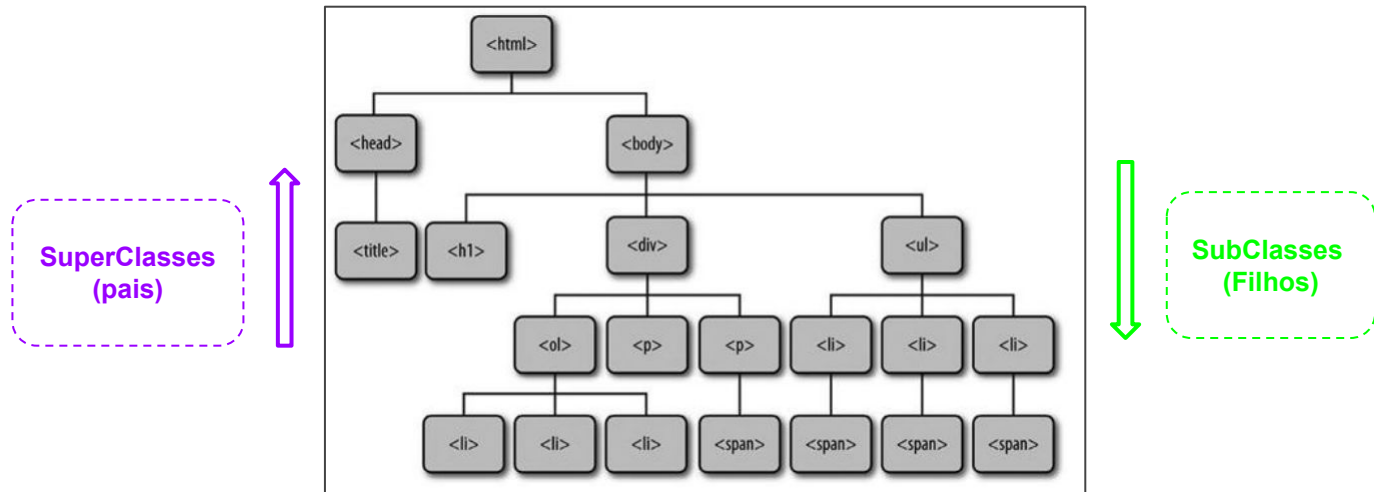
Os arquivos de CSS são organizados a partir da **estrutura hierárquica** do site por meio da definição das **características das classes** e pelos **fundamentos de herança**.

# Elementos de Estilo

- **CSS (Cascading Style Sheets):**

O CSS é uma tecnologia que nos permite especificar o **estilo dos elementos visuais** (por vezes chamada de interface do usuário, interface do cliente) de uma aplicação Web.

Os arquivos de CSS são organizados a partir da **estrutura hierárquica** do site por meio da definição das **características das classes** e pelos **fundamentos de herança**.



# Elementos de Estilo

## Estilizando Elementos Estruturais Básicos:

Para o estudo de nossas configurações de estilo, vamos considerar uma aplicação Web fictícia com alguns elementos estilizados.

Observe o arquivo “index.css” a seguir, com alguns parâmetros de estilo para TAGs:

```
1 /* Arquivo de Estilo */
2 body{
3     background: lightblue;
4     width: 800px;
5     margin: auto;
6 }
7 h1{
8     color: maroon;
9     text-align: center;
10 }
11 p{
12     color: gray;
13     border: 1px solid gray;
14     padding: 10px;
15 }
16
```

# Elementos de Estilo

## Estilizando Elementos Estruturais Básicos:

Para o estudo de nossas configurações de estilo, vamos considerar uma aplicação Web fictícia com alguns elementos estilizados.

Observe o arquivo “index.css” a seguir, com alguns parâmetros de estilo para TAGs:

```
1 /* Arquivo de Estilo */
2 body{
3     background: lightblue;
4     width: 800px;
5     margin: auto;
6 }
7 h1{
8     color: maroon;
9     text-align: center;
10 }
11 p{
12     color: gray;
13     border: 1px solid gray;
14     padding: 10px;
15 }
16
```

# Elementos de Estilo

## Estilizando Elementos Estruturais Básicos:

Para o estudo de nossas configurações de estilo, vamos considerar uma aplicação Web fictícia com alguns elementos estilizados.

Observe o arquivo “index.css” a seguir, com alguns parâmetros de estilo para TAGs:

```
1 /* Arquivo de Estilo */
2 body{
3     background: lightblue;
4     width: 800px;
5     margin: auto;
6 }
7 h1{
8     color: maroon;
9     text-align: center;
10 }
11 p{
12     color: gray;
13     border: 1px solid gray;
14     padding: 10px;
15 }
16
```

**background:** Plano de Fundo  
**width:** Espessura  
**margin:** Margem

# Elementos de Estilo

## Estilizando Elementos Estruturais Básicos:

Para o estudo de nossas configurações de estilo, vamos considerar uma aplicação Web fictícia com alguns elementos estilizados.

Observe o arquivo “index.css” a seguir, com alguns parâmetros de estilo para TAGs:

```
1 /* Arquivo de Estilo */
2 body{
3   background: lightblue;
4   width: 800px;
5   margin: auto;
6 }
7 h1{
8   color: maroon;
9   text-align: center;
10 }
11 p{
12   color: gray;
13   border: 1px solid gray;
14   padding: 10px;
15 }
16
```

**background:** Plano de Fundo  
**width:** Espessura  
**margin:** Margem

**color:** Cor  
**text-align:** Alinhamento Texto



# Elementos de Estilo

## Estilizando Elementos Estruturais Básicos:

Para o estudo de nossas configurações de estilo, vamos considerar uma aplicação Web fictícia com alguns elementos estilizados.

Observe o arquivo “index.css” a seguir, com alguns parâmetros de estilo para TAGs:

```
1 /* Arquivo de Estilo */
2 body{
3   background: lightblue;
4   width: 800px;
5   margin: auto;
6 }
7 h1{
8   color: maroon;
9   text-align: center;
10 }
11 p{
12   color: gray;
13   border: 1px solid gray;
14   padding: 10px;
15 }
16
```

**background:** Plano de Fundo  
**width:** Espessura  
**margin:** Margem

**color:** Cor  
**text-align:** Alinhamento Texto

**color:** Cor  
**border:** Bordas  
**padding:** Espaçamento

# Elementos de Estilo

## Link com Folha de Estilo:

Para associar as regras de estilo desejadas à nossa página HTML precisamos **vincular** o arquivo HTML com o estilo definido em nosso arquivo CSS.

Para tanto adicionamos a TAG **<link>** em nosso arquivo HTML com os atributos de especificação:

```
<link rel="stylesheet" href="style.css" type="text/css">
```

```
1 /* Arquivo de Estilo */
2 body{
3   background: lightblue;
4   width: 800px;
5   margin: auto;
6 }
7 h1{
8   color: maroon;
9   text-align: center;
10 }
11 p{
12   color: gray;
13   border: 1px solid gray;
14   padding: 10px;
15 }
16
```

index.css

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link rel="stylesheet" href="index.css">
5     <title>LifeStyle Capixaba</title>
6   </head>
7   <body>
54 </body>
55 </html>
56
```

# Elementos de Estilo

## Inclusão de Estilo Básico em TAGs:

As regras de estilo na linguagem CSS (*Cascading Style Sheet*) são definidas em cascata. Isto significa que as regras são sobrepostas umas sobre as outras, dependendo de suas definições.

Além disso, é importante observar a hierarquia dos elementos presentes em um arquivo HTML, pois regras de estilo podem ser herdadas do elemento pai.

Atributo	Descrição
background:	Definição do plano de fundo
color:	Definição de cores
width:	Largura do elemento
margin:	Margem: tipo de espaçamento para outros elementos
left-/righth-margin:	Margens à esquerda e à direita
text-align:	Alinhamento do texto: <i>left</i> , <i>right</i> , <i>center</i> , <i>justify</i>
font-family:	Família de fontes: serif, sans-serif, monospace, cursive, fantasy

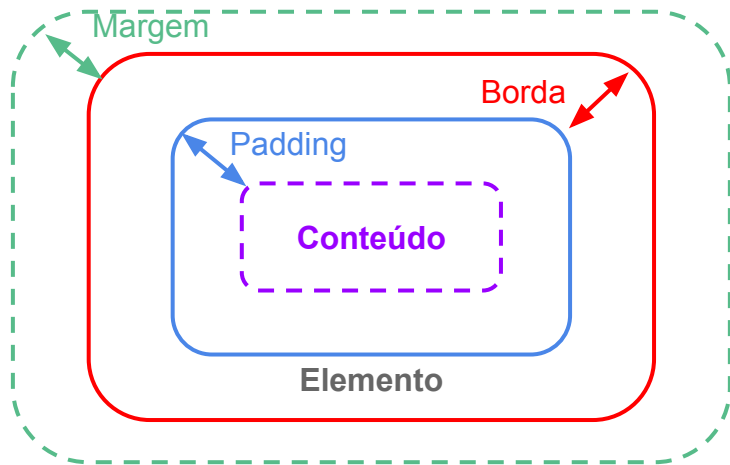
## [Tabela de Cores]

# Elementos de Estilo

## Espaçamento em Elementos de Bloco:

Alguns dos mais importantes elementos de construção de aplicações Web são definidos por uma estrutura em blocos. Na prática, esses elementos são normalmente incluídos na página em uma nova linha.

Alguns exemplos de **elementos de bloco** são: `<p>`, `<h1>`-`<h6>`, `<ol>`, `<ul>`, `<dl>`, `<hr>`, `<img>`, `<div>`



Atributo	Descrição
margin:	Espaçamento de Margem
left-margin:	Margens à esquerda
right-margin:	Margens à direita
padding:	Espaçamento <i>Padding</i>
border:	Espaçamento de Borda
border-color:	Cor da Borda
border-width:	Largura da Borda
border-style:	Estilo da Borda: <i>dashed</i> , <i>solid</i> , ...

# Elementos de Estilo

## Inclusão Básica de Estilo em Classes:

É muito comum que o atributo `class=""` seja adotado para a definição de regras de estilo. Nesse caso, o estilo deve ser definido como `.nome_da_classe`, como no exemplo a seguir:

```
20  a {  
21    padding: 15px;  
22  }  
23  
24  .header-conteudo {  
25    width: 700px;  
26    margin: auto;  
27  }  
28  
29  .main-conteudo {  
30    width: 700px;  
31    margin-left: auto;  
32    margin-right: auto;  
33    text-align: justify;  
34    font-family: cursive;  
35  }
```

# Elementos de Estilo

## Inclusão Básica de Estilo em Classes:

É muito comum que o atributo `class=""` seja adotado para a definição de regras de estilo. Nesse caso, o estilo deve ser definido como `.nome_da_classe`, como no exemplo a seguir:

```
20  a {  
21    padding: 15px;  
22  }  
23  
24  .header-conteudo {  
25    width: 700px;  
26    margin: auto;  
27  }  
28  
29  .main-conteudo {  
30    width: 700px;  
31    margin-left: auto;  
32    margin-right: auto;  
33    text-align: justify;  
34    font-family: cursive;  
35  }
```

Uma das principais importâncias de **regras definidas em classes** pode ser observado para o **elemento genérico <div>**.

Como esse elemento é genérico, podemos definir várias seções diferentes de nossa aplicação com regras de estilo diversificadas.

**Nota:** Caso existam classes com mesmo nome com comportamentos diferentes, deve-se definir a classe com base na TAG a ser adotada, como:

```
h2.primeiro-paragrafo{...  
    }
```

```
h3.primeiro-paragrafo{...  
    }
```

# Elementos de Estilo

## Atividade:

Com base em seus conhecimentos, estilize a página do blog “LifeStyle Capixaba”.

- (a) Utilize as organizações de conteúdo da TAG <body>: <header>, <main> e <footer>.
- (b) Adicione elementos visuais, **<img>** e **<video>**
- (c) Adote a TAG **<div>** para definir alinhamento central para o conteúdo no site com o atributos *width*.
- (d) Defina regras de estilo para suas TAGs e classes diversas.

**Exemplo  
Prático**

**Atividade**

# Elementos de Estilo

## Inclusão de Estilo em Textos:

- Google Fonts API [ [Link](#) ]: Variedade de Fontes (e Ícones) com licença gratuita que podem ser adotadas para estilização das fontes de sua aplicação Web (font-family).

```
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=FAMILY_NAME">
```



# Elementos de Estilo

## Inclusão de Estilo em Textos:

- Google Fonts API [ [Link](#) ]: Variedade de Fontes (e Ícones) com licença gratuita que podem ser adotadas para estilização das fontes de sua aplicação Web (font-family).

```
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=FAMILY_NAME">
```

- Atributos Adicionais:

Atributo	Descrição
text-align:	<b>Alinhamento do texto:</b> <i>left, right, center, justify</i>
font-family:	<b>Família de fontes:</b> serif, sans-serif, monospace, cursive, fantasy + <b>Google API</b>
font-size:	Tamanho da fonte
font-style:	<b>Estilo Fonte:</b> normal, italic, oblique
font-weight:	<b>“Peso”:</b> bold (negrito), bolder, valor (100-900)

# Elementos de Estilo

## Inclusão de Estilo em Textos:

- Google Fonts API [ [Link](#) ]: Variedade de Fontes (e Ícones) com licença gratuita que podem ser adotadas para estilização das fontes de sua aplicação Web (font-family).

`<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=FAMILY_NAME">`

- Atributos Adicionais:

Atributo	Descrição
text-align:	<b>Alinhamento do texto:</b> <i>left, right, center, justify</i>
font-family:	<b>Família de fontes:</b> serif, sans-serif, monospace, cursive, fantasy + <b>Google API</b>
font-size:	Tamanho da fonte
font-style:	<b>Estilo Fonte:</b> normal, italic, oblique
font-weight:	<b>“Peso”:</b> bold (negrito), bolder, valor (100-900)

- BackUp:** Pode-se definir múltiplas fontes para um determinado elemento/classe, estipulando uma prioridade para o caso de alguma fonte não estar acessível.

# Elementos de Estilo

## Definindo Cores em CSS:

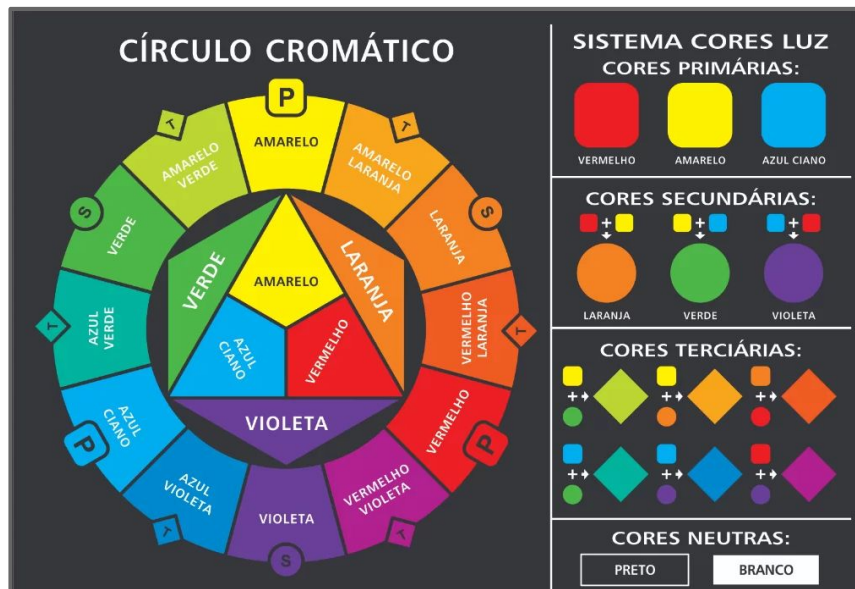
- Teoria das Cores:



# Elementos de Estilo

## Definindo Cores em CSS:

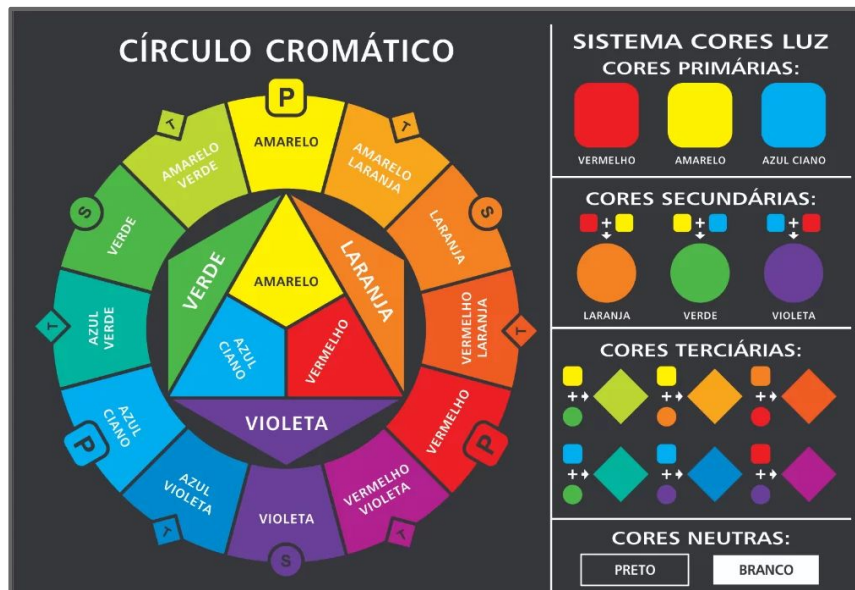
- Teoria das Cores:



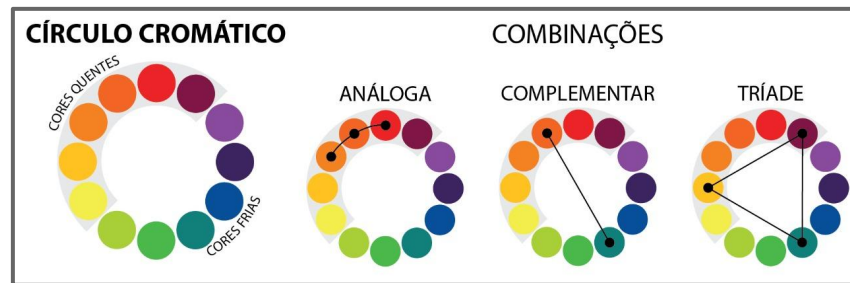
# Elementos de Estilo

## Definindo Cores em CSS:

- Teoria das Cores:



- Combinações e Paletas:



A escolha das cores é essencial para a definição de uma identidade gráfica do seu site.

Para tanto, é muito comum adotar uma seleção de cores que são **harmônicas entre si**, denominada **Paleta de Cores**.

Veja algumas opções: <https://colorhunt.co/>

# Elementos de Estilo

## Definindo Cores em CSS:

- RGB:

Uma forma comum de definir cores é pela mistura das componentes **Red-Green-Blue**.

Em CSS (e inúmeras outras aplicações) as partes de cada cor RGB são definidas considerando uma escala de 0-255 (total de  $256=2^8=16^2$  opções).

```
color: rgb(0, 0, 255);
```

```
color: rgb(150, 0, 0);
```

```
color: rgb(15, 122, 68);
```

# Elementos de Estilo

## Definindo Cores em CSS:

- RGB:

Uma forma comum de definir cores é pela mistura das componentes **Red-Green-Blue**.

Em CSS (e inúmeras outras aplicações) as partes de cada cor RGB são definidas considerando uma escala de 0-255 (total de  $256=2^8=16^2$  opções).

```
color: rgb(0, 0, 255);  
color: rgb(150, 0, 0);  
color: rgb(15, 122, 68);
```

- Escala Hexadecimal (Hex):

A escala hexadecimal adota a base 16 para representação dos números.

Um número **XY** na base 10:  $X*10^1+Y*10^0$

Um número **XY** na base 16:  $X*16^1+Y*16^0$

Algarismos da **Base Hex**: 0,1,2,...,9,A,B,C,D,E,F

# Elementos de Estilo

## Definindo Cores em CSS:

- RGB:

Uma forma comum de definir cores é pela mistura das componentes **Red-Green-Blue**.

Em CSS (e inúmeras outras aplicações) as partes de cada cor RGB são definidas considerando uma escala de 0-255 (total de  $256=2^8=16^2$  opções).

```
color: rgb(0, 0, 255);  
color: rgb(150, 0, 0);  
color: rgb(15, 122, 68);
```

- Escala Hexadecimal (Hex):

A escala hexadecimal adota a base 16 para representação dos números.

Um número **XY** na base 10:  $X*10^1+Y*10^0$

Um número **XY** na base 16:  $X*16^1+Y*16^0$

Algarismos da **Base Hex**: 0,1,2,...,9,A,B,C,D,E,F

```
color: #0000FF;  
color: #960000;  
color: #0F7A44;
```



# Elementos de Estilo

## Definindo Cores em CSS:

- RGB:

Uma forma comum de definir cores é pela mistura das componentes **Red-Green-Blue**.

Em CSS (e inúmeras outras aplicações) as partes de cada cor RGB são definidas considerando uma escala de 0-255 (total de  $256=2^8=16^2$  opções).

```
color: rgb(0, 0, 255);  
color: rgb(150, 0, 0);  
color: rgb(15, 122, 68);
```

- Escala Hexadecimal (Hex):

A escala hexadecimal adota a base 16 para representação dos números.

Um número **XY** na base 10:  $X*10^1+Y*10^0$

Um número **XY** na base 16:  $X*16^1+Y*16^0$

Algarismos da **Base Hex**: 0,1,2,...,9,A,B,C,D,E,F

```
color: #0000FF;  
color: #960000;  
color: #0F7A44;
```

Sugestão para escolha de Cores: <https://www.webfx.com/web-design/color-picker/>

Fator translúcido (**alpha**): Use **rgba**(15, 122, 68, 202) e **#0F7A44CA**;

# Elementos de Estilo

## Variáveis CSS e Identificador (id):

Definir **variáveis** é muito importante para especificações pertinentes da página.

As variáveis podem ser declaradas dentro de um **escopo de classe**, com acesso a todas as suas subclasses, ou em todo o documento (root).

Para adotá-las: `var(nome_variavel);`

```
5  :root{
6    --cor_paleta1: #69BEF5;
7  }
8
9  body {
10   background: white;
11   font-family: cursive;
12   --cor_paleta2: #E669F5;
13 }
14
```

index.css

# Elementos de Estilo

## Variáveis CSS e Identificador (id):

Definir **variáveis** é muito importante para especificações pertinentes da página.

As variáveis podem ser declaradas dentro de um **escopo de classe**, com acesso a todas as suas subclasses, ou em todo o documento (root).

Para adoptá-las: `var(nome_variavel);`

```
5  :root{
6    --cor_paleta1: #69BEF5;
7  }
8
9  body {
10   background: white;
11   font-family: cursive;
12   --cor_paleta2: #E669F5;
13 }
14
```

index.css

- Identificador (id):

Além de classes, os elementos de HTML podem ser identificados por “id”s.

Cada elemento pode possuir um **único id**, o qual apresenta **prioridade sobre as regras de classes**.

Para definições de regras de estilo para identificadores adota-se a **sintaxe com #**, exemplificado a seguir:

```
54  #primeiro_paragrafo{
55    text-align: justify;
56    color: #78F569;
57  }
```

# Elementos de Estilo

## Atividade:

Com base em seus conhecimentos, estilize a página do blog “LifeStyle Capixaba”.

- (a) Utilize uma família de fontes (Google API) específica para o nome do site.
- (b) Defina diferentes cores (paletas) que julgue interessantes para o seu site **em variáveis**.
- (c) Insira outras informações e adote os identificadores **ID** para compor a organização do seu site.



Prof. Guilherme Zucatelli

PhD on Defense Engineering

e-mail: [guilherme.zucatelli@uvv.br](mailto:guilherme.zucatelli@uvv.br)