



**Apostila — Material de apoio**

# **Estrutura e Semântica do HTML**

Desenvolvimento Front-End I

**Período:** Semana 3

**Autor:** Cleziel Franzoni da Costa

# Sumário

<b>1 Estrutura e Semântica do HTML</b>	<b>1</b>
1.1 A Anatomia de um Documento HTML5 . . . . .	1
1.1.1 Estrutura Essencial e Metadados . . . . .	1
1.2 Aprofundando nas Tags Essenciais . . . . .	2
1.2.1 Links que Vão Além de URLs . . . . .	2
1.2.2 Imagens com Contexto: <figure> e <figcaption> . . . . .	2
1.3 Estudo de Caso Prático: O Card de Produto . . . . .	2
1.3.1 Abordagem 1: A "Sopa de Divs"(Não Semântico) . . . . .	3
1.3.2 Abordagem 2: A Estrutura Semântica (Profissional) . . . . .	3
1.3.3 Análise Comparativa . . . . .	3
1.4 Boas Práticas e Qualidade de Código . . . . .	4
1.4.1 A Hierarquia de Títulos como Sumário Invisível . . . . .	4
1.4.2 Acessibilidade com WAI-ARIA: As Placas de Sinalização da Web . . . . .	4
1.5 Comentários em HTML: Regras e Boas Práticas . . . . .	5
1.5.1 Regra 1: A Sintaxe de Abertura e Fechamento . . . . .	5
1.5.2 Regra 2: A Armadilha do Hífen Duplo . . . . .	6
1.5.3 Regra 3: Comentários Não Podem Ser Aninhados . . . . .	6
1.5.4 Contexto Histórico: Comentários Condicionais (Obsoletos) . . . . .	6

# Capítulo 1

## Estrutura e Semântica do HTML

### Introdução

Dominar o HTML vai muito além de memorizar tags. Trata-se de aprender a pensar como um arquiteto da informação, construindo estruturas que sejam não apenas visualmente corretas, mas também lógicas, acessíveis e otimizadas. Um HTML bem escrito é o alicerce de um projeto front-end de excelência. Ele reduz a complexidade do CSS, simplifica a manipulação com JavaScript e garante que sua aplicação seja compreendida por todos, desde um usuário com leitor de tela até o robô de busca do Google. Esta apostila aprofunda esses conceitos, oferecendo a base para um desenvolvimento verdadeiramente profissional.

### 1.1 A Anatomia de um Documento HTML5

#### 1.1.1 Estrutura Essencial e Metadados

O `<head>` é o centro de controle da página. Além do básico, metadados de qualidade podem transformar a forma como seu site é visto e compartilhado na internet.

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Café Especial Montanhas Capixabas | Grãos Premiados</title>

    <link rel="stylesheet" href="css/estilos.css">
    <link rel="icon" href="favicon.ico" type="image/x-icon">
</head>
<body>
    </body>
</html>
```

#### Metadados para Compartilhamento (Open Graph)

As tags `og:*` (Open Graph Protocol) são usadas pelo Facebook, LinkedIn, WhatsApp e outras redes sociais para criar pré-visualizações ricas quando alguém compartilha seu link. Ignorá-las significa que a rede social tentará "adivinar" qual imagem e texto usar,

muitas vezes com resultados ruins. Preenchê-las corretamente é uma forma de controlar a narrativa da sua marca.

As quatro tags mais importantes são:

- **og:title**: O título principal que aparece no card de compartilhamento.
- **og:description**: O pequeno resumo que aparece abaixo do título.
- **og:image**: A imagem de miniatura que ilustra o link. Essencial para o apelo visual.
- **og:url**: O link "oficial" da página, para centralizar as interações (curtidas, etc.).

Veja como elas são aplicadas no código:

```
<head>
  ...
  <meta property="og:title" content="Café Especial Montanhas Capixabas">
  <meta property="og:description" content="Grãos premiados com sabor e aroma
    inigualáveis.">
  <meta property="og:image" content="https://seusite.com/imagens/cafe-social.
    jpg">
  <meta property="og:url" content="https://seusite.com/produto/cafe-especial">
  ...
</head>
```

## 1.2 Aprofundando nas Tags Essenciais

### 1.2.1 Links que Vão Além de URLs

A tag `<a>` é mais versátil do que parece. Ela pode iniciar ações no dispositivo do usuário:

- **Ligar para um número**: `<a href="tel:+5528999998888>Ligue para nós</a>`
- **Enviar um e-mail**: `<a href="mailto:contato@cafe.com>Envie um e-mail</a>`

### 1.2.2 Imagens com Contexto: `<figure>` e `<figcaption>`

Quando uma imagem precisa de uma legenda explicativa, o par `<figure>` e `<figcaption>` é semanticamente superior a um `<img>` e um `<p>` soltos. Ele agrupa a imagem e sua legenda como uma unidade de informação.

```
<figure>
  
  <figcaption>Figura 1: Nossas plantações nas montanhas do Espírito Santo.</
    figcaption>
</figure>
```

## 1.3 Estudo de Caso Prático: O Card de Produto

Imagine que precisamos construir o card de um produto para uma loja de cafés. Este é um componente extremamente comum em qualquer e-commerce.

### 1.3.1 Abordagem 1: A "Sopa de Divs"(Não Semântico)

Esta abordagem foca apenas na aparência, usando `<div>` para tudo. É funcional, mas péssimo para acessibilidade e SEO.

```
<div class="card-produto">
    <div class="imagem-container">
        
    </div>
    <div class="info-container">
        <span class="categoria-texto">Cafés Especiais</span>
        <div class="titulo Produto-grande">Blend Montanhas Capixabas</div>
        <p class="descricao-curta">Notas de chocolate e caramelo, 84 pontos.</p>

        <span class="preco-destaque">R$ 59,90</span>
        <div class="botao-comprar-azul">Comprar</div>
    </div>
</div>
```

### 1.3.2 Abordagem 2: A Estrutura Semântica (Profissional)

Esta versão usa as tags corretas para descrever o **significado** de cada parte do conteúdo.

```
<article class="card-produto">
    <figure>
        
    </figure>
    <section class="info-container">
        <p class="categoria">Cafés Especiais</p>
        <h3>
            <a href="/produto/blend-montanhas-capixabas">
                Blend Montanhas Capixabas
            </a>
        </h3>
        <p>Notas de chocolate e caramelo, 84 pontos.</p>
        <p class="preco">R$ 59,90</p>
        <a href="/carrinho/adicionar?id=123" class="botao-comprar">Comprar</a>
    </section>
</article>
```

### 1.3.3 Análise Comparativa

`<div>` vs. `<article>` Um card de produto é um item autocontido, que faria sentido mesmo fora do contexto da página (por exemplo, em um resultado de busca). `<article>` transmite essa independência. `<div>` não significa nada.

`<div>` vs. `<h3><a>...</a>` O nome do produto não é apenas um texto grande; é o **título** daquele item e um link para sua página. Usar uma hierarquia de títulos correta (`<h2>`, `<h3>`, etc.) permite que usuários de leitores de tela naveguem de produto em produto rapidamente, apenas ouvindo os títulos.

**<div> vs. <section>** O bloco de informações textuais forma uma seção lógica dentro do card, o que justifica o uso de **<section>**.

**<div> para botão vs. <a>** Um botão "Comprar" representa uma **ação**, uma navegação para outra página (o carrinho). Um link (**<a>**) é a tag correta para isso. Usar uma **<div>** exigiria JavaScript para funcionar e não seria acessível por padrão.

A versão semântica é mais verbosa, mas infinitamente superior. Ela constrói uma estrutura lógica que serve como uma base sólida para CSS, JavaScript e, mais importante, para a experiência de **todos** os usuários.

## 1.4 Boas Práticas e Qualidade de Código

### 1.4.1 A Hierarquia de Títulos como Sumário Invisível

Pense nos títulos (**<h1>** a **<h6>**) não como uma forma de deixar o texto maior ou menor (isso é trabalho do CSS), mas como a criação de um sumário lógico para a sua página.

Mecanismos de busca e, principalmente, usuários de tecnologias assistivas (como leitores de tela) navegam por este "sumário invisível" para entender a estrutura do conteúdo e pular diretamente para a seção que lhes interessa. Pular um nível (ex: de **<h2>** para **<h4>**) quebra essa lógica e torna a navegação confusa.

### 1.4.2 Acessibilidade com WAI-ARIA: As Placas de Sinalização da Web

#### O que é ARIA?

WAI-ARIA (*Web Accessibility Initiative – Accessible Rich Internet Applications*) é um conjunto de atributos que podemos adicionar ao HTML para torná-lo mais acessível. Pense no ARIA como "placas de sinalização" extras que ajudam os leitores de tela a entender componentes complexos e dinâmicos que o HTML padrão não consegue descrever sozinho.

#### A Primeira e Mais Importante Regra do ARIA

Não use ARIA se um elemento HTML nativo já faz o trabalho! Se você precisa de um botão, use **<button>**. Se precisa de um link, use **<a>**. Usar ARIA em um elemento nativo que já é acessível é redundante e pode até confundir.

```
<div role="button" tabindex="0">Clique aqui</div>  
  
<button>Clique aqui</button>
```

#### Quando Usar ARIA?

Use ARIA quando você cria "widgets" com JavaScript, como menus sanfona (accordions), abas (tabs), caixas de diálogo (dialogs), etc., que não possuem uma tag HTML específica. Os três pilares do ARIA são:

**Roles (Funções - **role**)** : Definem o que um elemento é. Ex: **role="dialog"**.

**Properties (Propriedades - aria-\*)** : Descrevem as **características** de um elemento.  
Ex: `aria-required="true"`.

**States (Estados - aria-\*)** : Descrevem a **condição atual** de um elemento. Ex:  
`aria-expanded="false"`.

### Exemplo Prático: Um Menu Sanfona (Accordion) Acessível

Um menu sanfona é um exemplo perfeito. O HTML por si só não consegue dizer a um leitor de tela que um título está "controlando"um painel e se ele está "expandido"ou "recolhido".

```
<h3>
  <button
    aria-expanded="false"
    aria-controls="painel-1"
    id="titulo-1">
    Qual a diferença entre café Arabica e Robusta?
  </button>
</h3>

<div
  id="painel-1"
  role="region"
  aria-labelledby="titulo-1"
  hidden>
  <p>O café Arabica é conhecido por sua acidez e complexidade aromática,
  enquanto o Robusta tem mais cafeína e um sabor mais amargo e intenso.</p>
</div>
```

**Análise do Exemplo:** `aria-expanded` informa se o painel está aberto ou fechado. `aria-controls` cria uma conexão direta entre o botão e o painel que ele controla. O JavaScript é responsável por alterar o estado (`aria-expanded`) e a visibilidade (`hidden`) do painel.

## 1.5 Comentários em HTML: Regras e Boas Práticas

Comentários são fundamentais para documentar o código, explicar decisões e facilitar a manutenção de projetos. No entanto, em HTML eles seguem regras específicas que, se ignoradas, podem comprometer o funcionamento da página.

### 1.5.1 Regra 1: A Sintaxe de Abertura e Fechamento

A forma correta de criar um comentário é abrir com `<!--` e fechar com `-->`. Tudo que estiver dentro desse bloco será ignorado pelo navegador.

```
<p>Este texto aparece na página.</p>
<!-- Este texto é um comentário e não será exibido -->
<p>Este também aparece normalmente.</p>
```

### 1.5.2 Regra 2: A Armadilha do Hífen Duplo

\*\*Nunca use dois hifens consecutivos ('–') dentro de um comentário.\*\* O navegador interpreta a sequência – como parte do fechamento e pode encerrar o comentário antes da hora, quebrando seu HTML.

```
<!-- Correto -->
<!-- Evite usar -- dentro de um comentário -->

<!-- Incorreto -->
<!-- Este comentário -- vai gerar problemas -->
```

### 1.5.3 Regra 3: Comentários Não Podem Ser Aninhados

Não é possível abrir um comentário dentro de outro. O navegador fecha o comentário assim que encontra a primeira sequência -->, ignorando o restante da intenção original.

```
<!-- Comentário externo
    <!-- Comentário interno (inválido) -->
    Este texto vai aparecer na página sem querer!
-->
```

Se precisar desativar um bloco de código que já contém comentários, apague temporariamente os comentários internos ou use recursos do seu editor (como atalhos de comentar/descomentar).

### 1.5.4 Contexto Histórico: Comentários Condicionais (Obsoletos)

Em códigos antigos você pode encontrar estruturas como esta:

```
<!--[if IE]>
  <p>Este texto só aparecia no Internet Explorer.</p>
<![endif]-->
```

Isso **não era um comentário comum**. Tratava-se de uma funcionalidade específica do Internet Explorer (IE), usada para carregar trechos de HTML ou CSS exclusivos para certas versões do navegador. **Atenção:** essa técnica está obsoleta e não deve mais ser utilizada.

## Encerramento da Semana

Ao longo desta semana exploramos a base estrutural do HTML5 e a importância de adotar uma marcação semântica consistente. Vimos como elementos aparentemente simples, como `<header>`, `<main>` e `<footer>`, influenciam diretamente a acessibilidade, a otimização em mecanismos de busca e a clareza do código para outros desenvolvedores.

Também aprendemos a utilizar boas práticas em links, imagens e comentários, entendendo não apenas a sintaxe, mas os impactos reais que essas escolhas geram na experiência do usuário.

**Lembre-se:** escrever HTML não é apenas “fazer aparecer na tela”. É construir documentos estruturados, legíveis por máquinas e pessoas, que podem evoluir com facilidade e se conectar a diferentes tecnologias (CSS, JavaScript, frameworks e APIs).

## Próximos Passos

Na próxima semana, avançaremos para o estudo do **CSS**, entendendo como a camada de estilo complementa a estrutura semântica que você já domina. Combinando HTML e CSS, será possível criar páginas não só corretas do ponto de vista estrutural, mas também visualmente atrativas e profissionais.

**Mensagem Final:** Escrever código semântico é investir em qualidade, acessibilidade e futuro. Cada tag escolhida com cuidado aproxima o seu projeto de um padrão profissional.