



# Desenvolvimento de Sistemas

---

*Web standards*: padrões sugeridos pelo World Wide Web Consortium (W3C) e recomendações de boas práticas

## *Web standards*: W3C

A história da criação do World Wide Web Consortium (W3C) começa em 1989, quando Tim Berners-Lee, um cientista da computação britânico que trabalhava no CERN, o laboratório de física de partículas na Suíça, propôs a criação de um sistema de hipertexto que permitisse o compartilhamento de informações entre cientistas em diferentes instituições. Ele desenvolveu o que hoje se conhece como a World Wide Web (WWW), que é a base da Internet moderna.

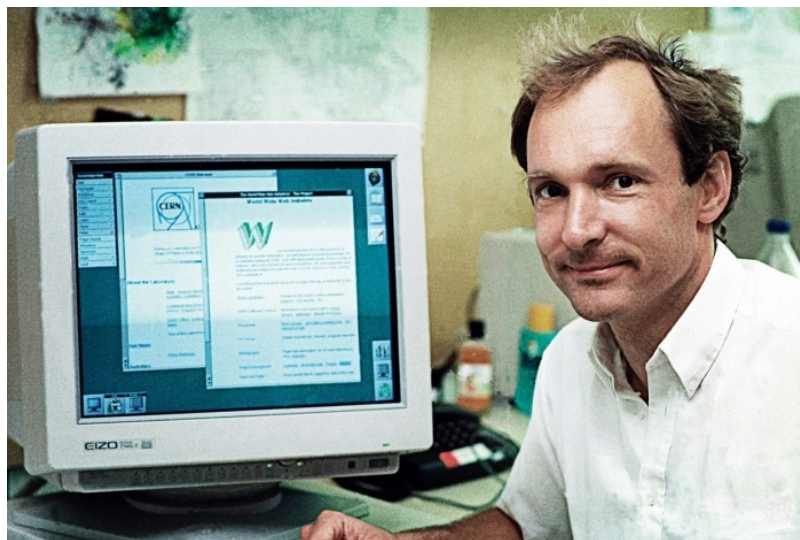


Figura 1 – Tim Berners-Lee, criador da WWW

Fonte: Tietz (2018)

No início dos anos 1990, a popularidade da *web* cresceu rapidamente e surgiram muitos navegadores e servidores *web* diferentes, cada um com as próprias tecnologias e os próprios recursos. Isso tornou difícil para os desenvolvedores criar *sites* que funcionassem bem em todos os navegadores e dispositivos.

Para abordar esse problema, Tim Berners-Lee fundou o **W3C**, em 1994, consórcio composto de empresas, organizações e indivíduos que trabalham juntos para desenvolver padrões abertos e tecnologias para a *web*. A missão do W3C é liderar a *web* para a sua plena potencialidade, desenvolvendo protocolos e diretrizes que promovam sua evolução e garantam sua acessibilidade a todos.



Figura 2 – O primeiro servidor *web* (no adesivo, é possível ver a mensagem “This machine is a server, do not power down”, em uma tradução livre, “Esta máquina é um servidor, não desligar”).

Fonte: Wikipédia (2005)

Ao longo dos anos, o W3C desenvolveu muitas especificações técnicas importantes, como o HTML (*hypertext markup language*), CSS (*cascading style sheets*), XML (*extensible markup language*), entre outros. Esses padrões ajudaram a tornar a *web* mais acessível e aprimoraram a experiência do usuário. A W3C também é uma organização ativa em questões de privacidade e segurança na *web*.

Em resumo, a W3C é uma organização líder no desenvolvimento de padrões para a *web*. Por meio de suas especificações técnicas e recomendações de boas práticas, a W3C ajuda a garantir que a *web* seja acessível, usável e interoperável em todos os dispositivos e navegadores.

*Web standards* (ou padrões *web*) são um conjunto de especificações técnicas e diretrizes que definem as melhores práticas para a criação de *sites* e aplicativos *web*.

Veja, a seguir, alguns dos padrões mais importantes desenvolvidos pela W3C.

## HTML

A linguagem de marcação de hipertexto (HTML) é usada para estruturar e exibir conteúdo na *web*. O W3C estabelece as especificações para a versão mais recente do HTML, que é o HTML5.

## CSS

As folhas de estilo em cascata (CSS) são usadas para definir o *layout* e o estilo de um *site*. O W3C estabelece as especificações para a versão mais recente do CSS, que é o CSS3.

## JavaScript

A linguagem de programação JavaScript é usada para adicionar interatividade a um *site*. O W3C estabelece as especificações para o JavaScript, incluindo o padrão ECMAScript.

Além desses padrões, a W3C também desenvolve especificações para outras tecnologias *web*, como XML, SVG (*scalable vector graphics*), e *web accessibility* (acessibilidade na *web*).

A importância dos *web standards* reside na sua capacidade de permitir que desenvolvedores criem *sites* e aplicativos *web* acessíveis, usáveis e interoperáveis. Ao seguir esses padrões, os desenvolvedores podem garantir que seus *sites* sejam mais compatíveis com diferentes navegadores e dispositivos, além de melhorar a experiência do usuário.

## Boas práticas de uso para os padrões W3C

1. Use HTML semântico – O HTML semântico é uma prática que envolve o uso de elementos HTML para fornecer significado e estrutura ao conteúdo de uma página da *web*. Isso torna o conteúdo mais fácil de ler e entender para os usuários e também ajuda os mecanismos de busca a entenderem o que é importante em uma página. Sempre que possível, use *tags* HTML apropriadas, como **<header>**, **<main>**, **<section>**, **<article>**, **<nav>** e **<footer>** para estruturar o conteúdo da sua página.
2. Valide o seu código – A validação de código é um processo que verifica se o HTML, CSS e JavaScript são escritos de acordo com as especificações da W3C. Isso ajuda a garantir que seus *sites* sejam acessíveis e compatíveis com

diferentes navegadores e dispositivos. Você pode usar ferramentas *on-line*, como o validador W3C, para verificar se o seu código é válido.

3. Use CSS para estilizar seu *site* – O CSS é uma linguagem de estilo que permite definir a aparência do seu *site*. Use CSS para criar *layouts* responsivos, tornar seu *site* acessível para usuários com deficiência visual e para manter a consistência visual em todo ele.
4. Certifique-se de que seu *site* seja acessível – A acessibilidade na *web* é importante para garantir que todos os usuários possam acessar e interagir com seu *site*. Siga as diretrizes de acessibilidade da W3C, como o *Web Content Accessibility Guidelines* (WCAG), para garantir que seu *site* seja acessível para usuários com deficiências visuais, auditivas ou motoras.
5. Teste seu *site* em diferentes navegadores e dispositivos – Para garantir que seu *site* seja compatível com diferentes navegadores e dispositivos, teste-o em diferentes plataformas e dispositivos. Isso ajuda a identificar problemas e corrigi-los antes que os usuários encontrem problemas ao acessar seu *site*.
- 6.

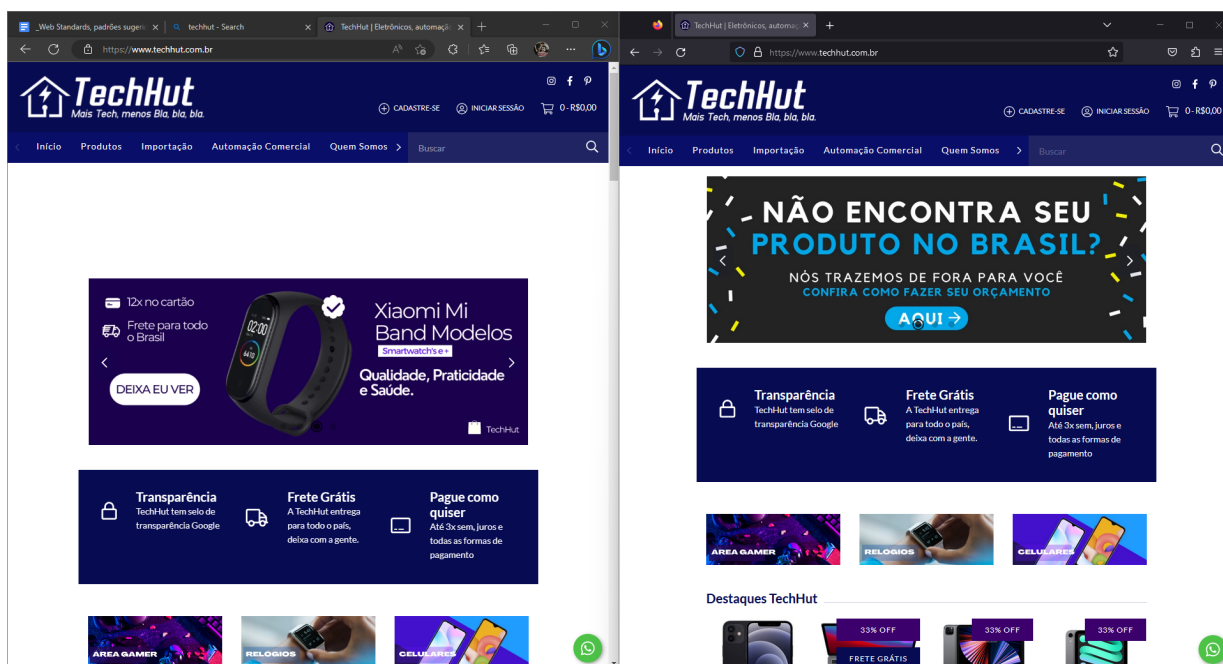


Figura 3 – Um mesmo *site* aberto em dois navegadores diferentes  
Fonte: adaptado de TechHut (c2023)

Seguir as recomendações de boas práticas acima pode ajudar a garantir que seu *site* seja acessível, usável e compatível com diferentes navegadores e dispositivos. Além disso, essas práticas podem melhorar a experiência do usuário e tornar seu *site* mais atraente e profissional.



Ao escrever seus códigos usando as recomendações da W3C, é possível levar em consideração as seguintes boas práticas de escrita de código e indentação para HTML, CSS e JavaScript.

## Boas práticas em HTML

- ◆ Use *tags* semânticas apropriadas para descrever o conteúdo. Por exemplo, use **<header>**, **<nav>**, **<main>** e **<footer>** em vez de **<div>**.
- ◆ Use aspas duplas para atributos de *tag* em vez de aspas simples.
- ◆ Inclua um **alt** em todas as imagens, para garantir que usuários com deficiência visual possam entender o conteúdo da imagem.
- ◆ Use indentação consistente para tornar o código mais fácil de ler.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Minha página</title>
  </head>
  <body>
    <header>
      <h1>Minha Página</h1>
    </header>
    <nav>
      <ul>
        <li>
          <a href="#">Página Inicial</a>
        </li>
        <li>
          <a href="#">Sobre Nós</a>
        </li>
        <li>
          <a href="#">Contato</a>
        </li>
      </ul>
    </nav>
    <main>
      <h2>Sobre Nós</h2>
      <p>...</p>
    </main>
    <footer>
      <p>© 2023 - Todos os direitos reservados</p>
    </footer>
  </body>
</html>
```

## Boas práticas em CSS

- ◆ Use nomes descritivos para classes e IDs, em vez de nomes genéricos.
- ◆ Use a cascata do CSS para evitar repetição de estilos.
- ◆ Comente seu código para ajudar a descrever o que faz cada seletor.

Exemplo:

```
/* Estilos para a navegação */
nav {
  background-color: #333;
  color: #fff;
  padding: 10px;
}

navul {
  list-style: none;
  margin: 0;
  padding: 0;
}

navli {
  display: inline-block;
}

nava {
  color: #fff;
  text-decoration: none;
  padding: 10px;
}

/* Estilos para o cabeçalho */
header {
  background-image: url('header-bg.jpg');
  height: 200px;
  background-size: cover;
}
```

## Boas práticas em JavaScript

- ◆ Use nomes descritivos para variáveis e funções.
- ◆ Use **const** e **let** em vez de **var** ao declarar variáveis e constantes para evitar problemas com escopo.
- ◆ Use espaços em branco para tornar o código mais legível.

Exemplo:

```
const lista = [1, 2, 3, 4];

function somarLista(lista) {
  let soma = 0;
  for (let i = 0; i < lista.length; i++) {
    soma += lista[i];
  }
  return soma;
}
```

## Validador de código W3C

O validador de código HTML do W3C é uma ferramenta muito útil para verificar se o seu código HTML está seguindo os padrões recomendados e se está livre de erros. Aqui está um guia passo a passo para usar o validador:

1. Acesse o *site* do validador de código HTML do W3C (para isso, pesquise por “Validator W3” em seu buscador).
2. Na seção **Validated By Direct Input**, cole seu código HTML no campo de texto. Para testar nesse momento, use o seguinte código HTML, que contém inconsistências (você consegue apontar algumas delas antes da análise?):

```
<!DOCTYPE html>
<html>
  <head>
    <title>Testes em páginas</title>
    <meta charset="utf-8">
  </head>
  <body>
    <main>
```





```

<center><h1>Artigos</h1></center>
<article id="artigo">
    <h2>Título do artigo 1</h2>
    
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus sodales elit vel massa cursus, at varius nisl feugiat. Quisque vitae justo vitae nulla sollicitudin ullamcorper. Fusce eu dui vulputate, aliquet mauris vel, ultricies mauris.
    </p>
</article>
<article id="artigo">
    <h2>Título do artigo 2</h2>
    
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus sodales elit vel massa cursus, at varius nisl feugiat. Quisque vitae justo vitae nulla sollicitudin ullamcorper. Fusce eu dui vulputate, aliquet mauris vel, ultricies mauris.
    </p>
</article>
</body>
</html>

```

W3C<sup>®</sup> Markup Validation Service  
Check the markup (HTML, XHTML, ...) of Web documents

Validate by URI   Validate by File Upload   **Validate by Direct Input**

Validate by direct input

Enter the Markup to validate:

```

<!DOCTYPE html>
<html>
  <head>
    <title>Testes em páginas</title>
    <meta charset="utf-8">
  </head>
  <body>
    <main>
      <center><h1>Artigos</h1></center>
      <article id="artigo">
        <h2>Título do artigo 1</h2>
        
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus sodales elit vel massa cursus, at varius nisl feugiat. Quisque vitae justo vitae nulla sollicitudin ullamcorper. Fusce eu dui vulputate, aliquet mauris vel, ultricies mauris.
        </p>
      </article>
      <article id="artigo">
        <h2>Título do artigo 2</h2>

```

► More Options

Check

Figura 4 – Validando código no site do W3C

Fonte: Senac EAD (2023)

### 3. Clique no botão **Check**.



#### 4. O validador analisará o seu código e mostrará uma lista de erros, avisos e informações no painel de resultados.

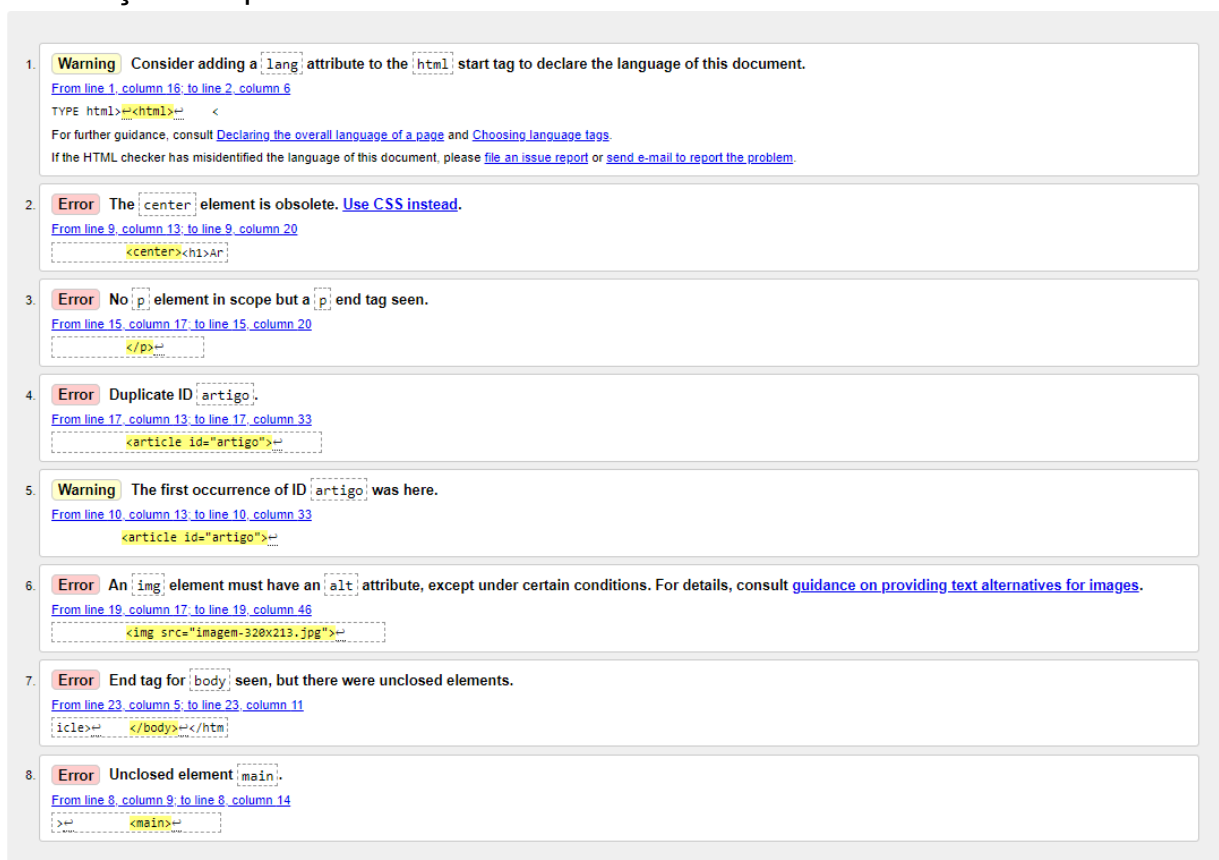


Figura 5 – Erros encontrados pelo validador

Fonte: Senac EAD (2023)

Veja algumas dicas para ajudar você a entender os resultados:

- ◆ Os erros são problemas que devem ser corrigidos, pois impedem que o seu código seja interpretado corretamente pelos navegadores.
- ◆ Os avisos são problemas que devem ser analisados, mas podem ser ignorados em alguns casos.
- ◆ As informações fornecem dicas e sugestões para melhorar o seu código.

Os erros que o validador gera com base no código de teste de HTML apresentado anteriormente são os seguintes (em tradução livre):

Alerta: “Considere adicionar atributo **lang** à *tag html* para declarar a linguagem deste documento”. Incluir **<html lang="pt-br">**, por exemplo, ajuda na indexação das páginas, mas não é algo obrigatório.

Erro: “Elemento **center** é obsoleto”. A *tag <center>* era usada no HTML4, mas na versão nova foi abolida.

Erro: “Nenhum elemento **p** no escopo, mas um fechamento de **p** detectado”. O validador refere-se à linha 15 do código, em que há um `</p>` sem um `<p>` correspondente.

Erro: “ID **artigo** duplicado”. De fato, note que ambos os elementos `<article>` do código estão com o mesmo valor para o atributo **id**.

Alerta: “A primeira ocorrência de **artigo** é aqui”. O validador está apenas indicando em qual linha está o primeiro elemento com o id “artigo”.

Erro: “Um elemento **img** deve ter um atributo **alt**, exceto em certas condições”. Está se referindo à tag `<img>` da linha 19, em que não há atributo **alt** preenchido.

Erro: “Tag de fechamento **body** detectada, mas há elementos não fechados”. O problema aqui acontece diretamente por conta do erro seguinte.

Erro: “Elemento **main** não fechado”. Repare que, no código, omitiu-se `</main>`, o que gerou este erro e o anterior.

Corrija os erros e alertas apontados pelo validador W3C para o código HTML proposto e submeta novamente o código ao validador.

Quando tiver um código livre de qualquer erro e de qualquer observação, você verá a seguinte mensagem na tela:

**Document checking completed. No errors or warnings to show.**

Traduzindo a mensagem: “Checagem do documento completa. Sem erros ou alertas para mostrar”.

Lembre-se de que o validador é uma ferramenta importante, mas não é a única que você deve usar para garantir a qualidade do seu código. É importante sempre escrever código claro e organizado, seguindo as boas práticas recomendadas.

## *Search engine optimization*

O SEO (*search engine optimization*) é um conjunto de técnicas que tem como objetivo melhorar a visibilidade e o posicionamento de um *site* nos resultados de busca do Google e de outros buscadores. Uma das técnicas de SEO mais importantes é o uso de *metatags* no código HTML do *site*, especialmente a *metatag description* e a *metatag keywords*.

A **metatag description** é um resumo do conteúdo da página, que é exibido nos resultados de busca. É importante escrever uma descrição concisa, clara e atraente, com até 160 caracteres. Para incluir a **metatag description** em seu *site*, adicione o seguinte código HTML entre as tags **<head>** e **</head>**:

```
<meta name="description" content="Insira aqui uma descrição clara e concisa do conteúdo da sua página.">
```

A **metatag keywords** é uma lista de palavras-chave relevantes para o conteúdo da página. Embora ela não seja mais usada pelos principais mecanismos de busca, ainda pode ser útil para algumas plataformas de busca menos conhecidas.

Além do conhecido Google, há algumas plataformas de busca menos conhecidas, por exemplo: Ecosia, DuckDuckGo, Bing, Yahoo e Ask.com.

Para incluir a **metatag keywords** em seu *site*, adicione o seguinte código HTML entre as tags **<head>** e **</head>**:

```
<meta name="keywords" content="Insira aqui uma lista de palavras-chave relevantes para o conteúdo da sua página."></metaname>
```

Além das *metatags*, há algumas dicas de otimização de busca para um *site* da *web*. Veja a seguir.

## Use palavras-chave relevantes

Escolha palavras-chave relevantes para o conteúdo da sua página e inclua-as em títulos, texto, imagens e *links* internos.

## Escreva conteúdo de qualidade

Crie conteúdo de qualidade, útil e relevante para o seu público-alvo e atualize-o regularmente.

## Otimize as imagens

Use imagens de alta qualidade e otimize-as para carregar rapidamente, incluindo *tags alt* descritivas.

## Use URLs descritivas

Escolha URLs (*uniform resource locator*) que descrevam claramente o conteúdo da página e use hifens para separar palavras.

## Use *links* internos e externos

Use *links* internos para ajudar os visitantes a navegar no seu *site* e *links* externos para aumentar a credibilidade do seu *site* e melhorar o SEO.

## Otimize o carregamento da página

Certifique-se de que o seu *site* carrega rapidamente, otimizando imagens, reduzindo o tamanho do arquivo, usando *caches*, entre outras técnicas.

## Use redes sociais

Promova o seu *site* nas redes sociais para aumentar a sua visibilidade e direcionar tráfego para ele.

Lembre-se de que o SEO é um processo contínuo e que é importante monitorar e ajustar sua estratégia de acordo com as mudanças nos algoritmos dos mecanismos de busca.

# Encerramento



Em resumo, você viu, neste conteúdo, vários assuntos relacionados ao desenvolvimento *web*, incluindo os padrões do W3C para HTML, CSS e JavaScript, as melhores práticas para escrever código bem estruturado e indentado e, ainda, a importância da otimização para mecanismos de busca para um *site*. Você também viu que aprender a seguir as diretrizes do W3C pode ajudar a garantir que o código do seu *site* seja consistente, acessível e compatível com vários navegadores e dispositivos.

Além disso, você viu que seguir as melhores práticas de codificação, incluindo a boa estruturação e indentação do código, pode tornar o processo de desenvolvimento mais fácil e eficiente. Por fim, você aprendeu que otimizar seu *site* para mecanismos de busca é essencial para aumentar sua visibilidade e seu alcance na *web*. Utilizando estratégias de SEO, como a adição de *metatags* ao código HTML, é possível melhorar o posicionamento do *site* nos resultados de busca e aumentar o tráfego orgânico. No geral, seguir essas práticas e recomendações pode levar a um *site* mais acessível, eficiente e eficaz na *web*.