



Desenvolvimento de Sistemas

Estrutura semântica: linguagens de marcação de conteúdo (HTML) para aplicações web

Introdução

A linguagem de marcação de hipertexto, ou somente HTML (*hypertext markup language*), como é amplamente conhecida, é uma linguagem padrão para o desenvolvimento de páginas *web*, em que é possível, por meio das chamadas *tags*, descrever a estrutura e o conteúdo de uma página. Veja que o HTML não é considerado uma linguagem de programação, mas, sim, uma linguagem de marcação.

O HTML é uma linguagem interpretada por todos os navegadores de código aberto e considerada de simples aprendizagem, uma vez que ela não depende da implementação de lógica de programação, porém tem fácil interação com outras linguagens, como PHP (*hypertext preprocessor*), JavaScript e CSS (*cascading style sheets*). Atualmente, o HTML encontra-se na versão 5, o chamado HTML5, com diversas novas funcionalidades e compatibilidades.

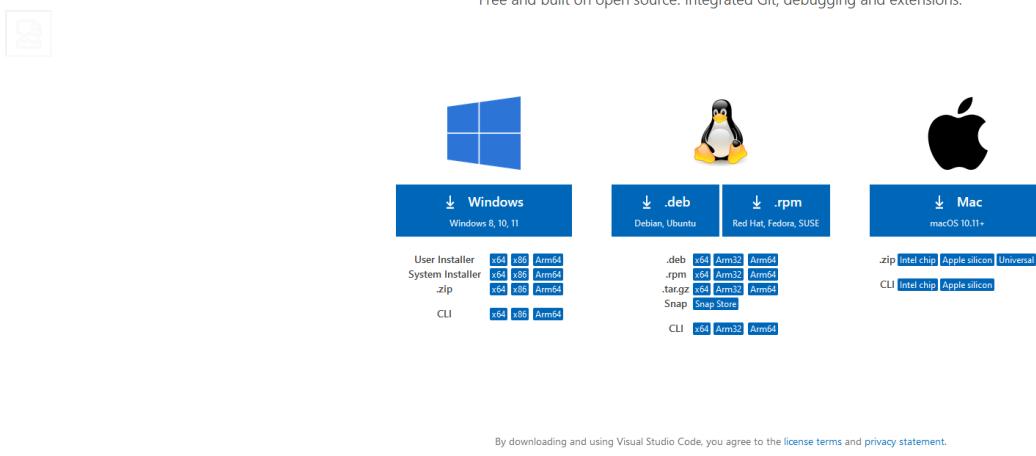
Você pode desenvolver um arquivo com os padrões HTML5 no editor de texto de sua preferência, porém é indicado e será utilizado o Visual Studio Code (VSCode) como editor padrão para desenvolver os exemplos apresentados ao longo do conteúdo.

Editor para criação de arquivos HTML5

Para desenvolver e editar códigos em HTML, é necessário o auxílio de um editor de texto ou um ambiente de desenvolvimento integrado (IDE). Para trabalhar com HTML nesta unidade, será utilizado o VSCode, um editor leve e poderoso que marca a sintaxe para a linguagem.

A seguir, veja um passo a passo para a instalação e configuração desse editor.

1. Realizar o *download* do editor de acordo com seu sistema operacional. Para isso, basta digitar em seu buscar “*Download Visual Studio Code*”.



By downloading and using Visual Studio Code, you agree to the [license terms](#) and [privacy statement](#).

Figura 1 – Site para download do VSCode

Fonte: Senac EAD (2023)

1. Agora, realize a instalação padrão do editor, clicando no executável que acabou de ser baixado.
2. Com o VSCode aberto, clique em **Arquivo** e depois em **Adicionar pasta ao Workspace**.
3. Escolha a localização e o nome da pasta que você trabalhará.
4. Agora, com a pasta de trabalho já configurada, crie seu primeiro arquivo HTML. Para isso, clique sobre a pasta com o botão direito e depois selecione **Novo arquivo**, como mostra a figura a seguir. Não se esqueça de nomear o arquivo com a extensão HTML (exemplo: **teste.html**).

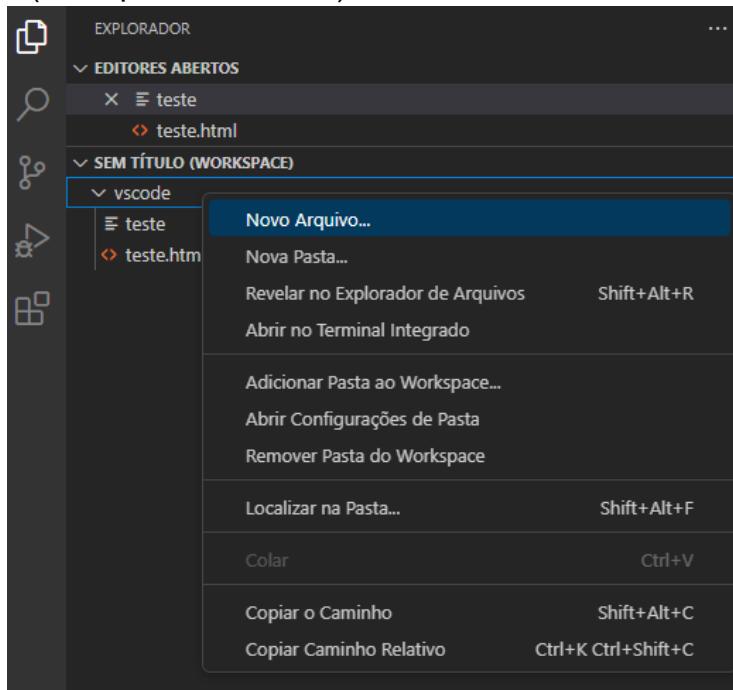


Figura 2 – Criação de arquivos no VSCode

Fonte: Senac EAD (2023)

5. Com seu arquivo HTML já criado, instale uma extensão no VSCode para interpretar esses códigos. Para isso, clique no ícone da lateral esquerda do editor, denominado

Extensões.

6. Agora pesquise por “HTML Preview” e clique em **Instalar**.

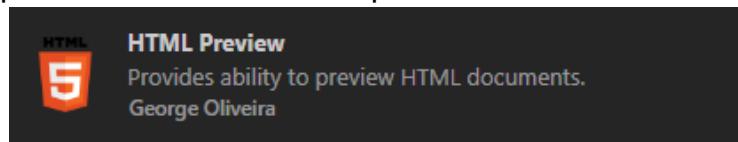


Figura 3 – Extensão HTML Preview VSCode

Fonte: Senac EAD (2023)

7. Essa extensão permite visualizar a página que for desenvolvida como em um navegador.
8. Para fazer a visualização do arquivo desenvolvido, utilize o atalho (com o arquivo em questão aberto) **Ctrl + Shift + V**.

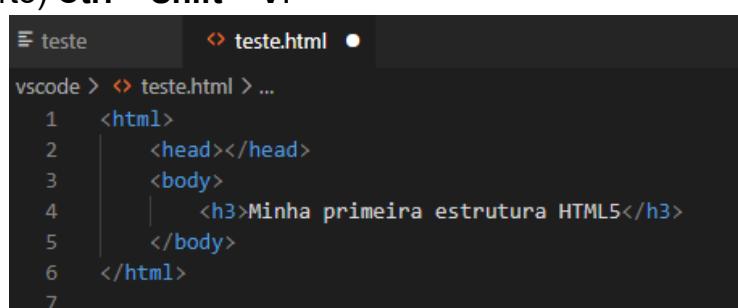


Figura 4 – Primeiro arquivo HTML5 VSCode

Fonte: Senac EAD (2023)

Para cada exemplo que vem a seguir no texto, crie um novo arquivo HTML, digite o código e teste a página abrindo-a em um navegador (na pasta onde o arquivo foi salvo, dê um duplo clique no arquivo e ele abrirá no *browser* padrão).

Tags

Em sua estrutura básica, o HTML trabalha com as chamadas *tags*, sendo estas responsáveis por comunicar o navegador sobre a estrutura de um site HTML. As *tags* são interpretadas pelo navegador e sempre são declaradas dentro de sinais de maior e menor, da seguinte forma: **<tag>**.

As *tags* podem, ou não, necessitar de fechamento, atendendo à seguinte característica:

Tag com fechamento:

<tag> </tag>

onde **<tag>** é a abertura e **</tag>** é o fechamento.

Tag sem fechamento (ou autofechamento):

<tag>

 Veja, na figura a seguir, exemplo de *tags* com e sem fechamento. Não se preocupe, pois ao longo do conteúdo serão abordadas as funcionalidades de cada tag dentro da estrutura do HTML.

```
<html>
  <head></head>
  <body>
    <h3>Minha primeira estrutura HTML5</h3> <!-- Tag com fechamento -->
    <br/> <!--Tag sem fechamento (auto fechamento)-->
  </body>
</html>
```

Figura 5 – Tags básicas HTML5

Fonte: Senac EAD (2023)

Comentários

Observando a figura anterior, é possível ver também os comentários (letras em verde) feitos dentro do código HTML. Cada linguagem tem a própria maneira de declarar um comentário. No caso do HTML, isso ocorre da seguinte forma:

<!-- (este é o sinal de abertura de um comentário)

--> (este é o sinal de fechamento de um comentário)

<!-- Texto do comentário -->

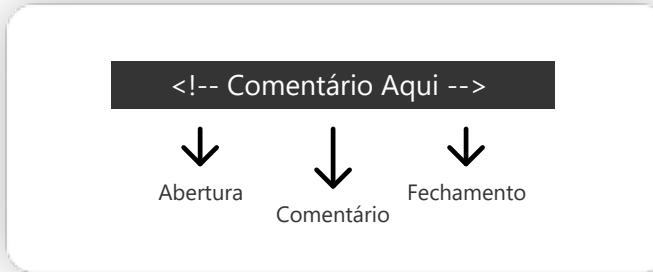


Figura 6 – Comentários HTML5

Fonte: Senac EAD (2023)

Adiante, serão abordadas as *tags* básicas do HTML5.

Tags básicas do HTML

Agora que você já conhece o conceito básico das *tags*, veja quais são as mais básicas

(porém importantes) que formam a estrutura do HTML.



<html>

Define a página HTML. É a *tag* raiz, ou seja, todas as outras *tags* da página devem estar entre <html> e </html>.

<head>

É a área onde se inclui o título da página (que aparecerá no navegador) e as informações de metadados, necessárias para recursos da página, mas que não aparecem na tela.

<body>

É o corpo da página. É entre <body> e </body> que estará presente o conteúdo da página.

No próximo trecho, veja a estrutura HTML de forma básica, utilizando as tags citadas

```
<!DOCTYPE html> <!-- Tag especial que define o HTML5 -->
<html>
  <head>
    <!-- Cabeçalho -->
  </head>
  <body>
    <!-- Conteúdo -->
  </body>
</html>
```

Atributos

É possível entender os atributos como uma informação adicional que se passa para a *tag*. Dessa maneira, a *tag* pode interpretar configurações de forma correta. Por exemplo, na *tag* de , sabe-se que ela é responsável por exibir uma imagem, porém é preciso de um atributo, no caso o **src**, para indicar onde essa imagem está armazenada, da seguinte forma:

```
<img src = "local/imagem.png"/>;
```

Note que os atributos são sempre definidos dentro da *tag* de abertura, como mostra a figura

do exemplo a seguir (não se preocupe com as funcionalidades de cada *tag* ainda, pois isso será visto ao longo do conteúdo).

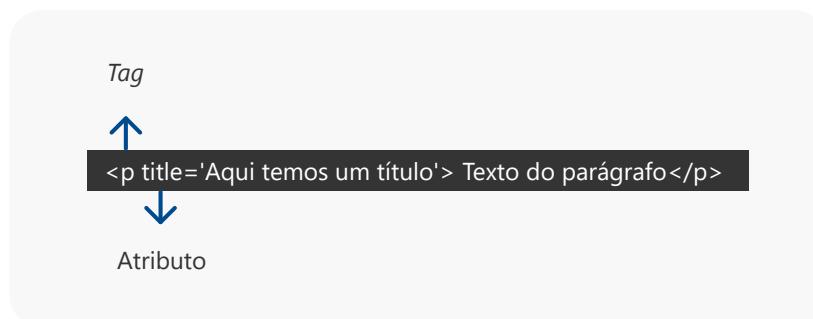


Figura 7 – Exemplo de atributos

Fonte: Senac EAD (2023)

Os atributos podem, ainda, ser separados em dois grupos, os chamados atributos globais e os atributos específicos.

Atributos globais

São aqueles a que todas as *tags* têm acesso. Em outras palavras, podem ser implementados em qualquer *tag*, por exemplo, o atributo **title**. Os principais atributos globais são os listados a seguir.

Class

Define uma classe CSS para estilização do elemento.

Id

Identifica um elemento na página. Recomenda-se que o valor para esse atributo seja único para cada elemento na página.

Style

Define um estilo em CSS.

Title

Define informação extra para o elemento (geralmente um texto que aparece quando se passa o mouse sobre o elemento na página).

Tabindex



Ordem de alcance da tecla Tab para navegar entre os elementos da página.

Atributos específicos

São aqueles que só podem ser implementados em *tags* específicas, por exemplo, o atributo **action**, presente na *tag* de formulário que você verá mais adiante.

O valor de um atributo é sempre delimitado por aspas duplas ("") ou aspa simples ('), e a aspa de abertura deve ser igual à de fechamento (ou seja, algo como "valor' não é válido, mas "valor" e 'valor' sim).

Tags de conteúdos

Da mesma forma que se conta com *tags* básicas e estruturais, há também as *tags* específicas para a introdução de conteúdo dentro do HTML, sendo elas as mencionadas a seguir.

Tags para títulos

Essas *tags* são representadas dentro do HTML5 pela letra “h” com variações que vão de **<h1></h1>** até **<h6></h6>**, sendo o maior número inversamente proporcional ao tamanho da fonte, ou seja, **h1** é título com maior fonte, enquanto **h6** é o título com menor fonte. A imagem a seguir mostra a utilização das *tags* de títulos, partindo da maior para a menor.

Título h1

Título h2

Título h3

Título h4

Título h5



Título h6

Figura 8 – Tags de títulos HTML5

Fonte: Senac EAD (2023)

Na figura a seguir, é possível ver como a imagem anterior fica quando implementada em termos de código HTML5.

```
<h1>Título h1</h1>
<h2>Título h2</h2>
<h3>Título h3</h3>
<h4>Título h4</h4>
<h5>Título h5</h5>
<h6>Título h6</h6>
```

Figura 9 – Tags de títulos HTML5 código

Fonte: Senac EAD (2023)

Tags para textos

As *tags* dedicadas a estruturas textuais dentro do HTML5 são amplamente utilizadas e têm a funcionalidade de definir características para os textos. Agora, veja quais são essas *tags*.

- | | |
|-----------------------|--|
| <p> | É responsável por definir um parágrafo, recebendo o texto que o compõe. |
| | É a <i>tag</i> que transforma em negrito o texto que se encontra dentro dela. |
| | É semelhante à <i>tag</i> , porém, por apresentar um estado lógico, permite que diversas outras configurações sejam feitas dentro das características dela. |

 <i>

É a *tag* que transforma em itálico o texto que se encontra

dentro dela.

É a *tag* que, assim como a <i>, mantém o texto em itálico,

porém dá ênfase no conteúdo.

É um *container* de texto que, diferente de <p>, não quebra

linha (e por isso é classificado como *tag inline*). É geralmente
utilizado para marcar parte de um texto ou de um documento.

É uma *tag* com autofechamento que é responsável por

aplicar uma quebra de linha no texto.

<hr/>

Cria uma linha divisória horizontal.

<a>

Define que determinado texto é um *link* e deve conter o

atributo **href**, o qual informa o endereço desse *link*

No trecho a seguir, veja a implementação das *tags* de texto em uma estrutura básica de HTML5. Leia a descrição de cada *tag* e observe no código como elas estão aplicadas na estrutura.

```
<html>
  <head>
    <title>Título</title>
  </head>
  <body>
    <h3>Este é o título do nosso texto</h3>
    <p>
      <!-- aqui temos um parágrafo -->
    </p>
    <p>
      <strong>Agora um texto em negrito</strong>
      <em>Agora um texto em itálico
    </p>
    <hr/> <!-- linha de separação -->
    <span>
      Texto Final.
    </span>
  </body>
</html>
```

```
</span>
</body>
</html>
```

Na próxima figura, veja como esse trecho de código é apresentado ao usuário final, quando interpretado pelo navegador.

Este é o título do nosso texto

Aqui temos um texto do parágrafo
Agora um texto em negrito
Agora um texto em itálico

Texto final.

Figura 10 – Tags de texto HTML5

Fonte: Senac EAD (2023)

Tag de hyperlink

A expressão “hipertexto” está relacionada com a capacidade de ligar um conteúdo a outro por meio de *hyperlinks*. Assim, a tag **<a>** é de grande importância não só no HTML como em toda a Internet, pois, por meio dela, pode-se definir a ligação a outra página. O atributo mais importante é o **href**, no qual se define a qual página se quer que o *hyperlink* vá. Veja, no exemplo a seguir, um *link* com **<a>** que aponta para o *site* do Senac EAD.

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Usando Hyperlink
    <a href="https://www.ead.senac.br">Link para a página do Senac EAD</a>
  </body>
</html>
```

Escreva o código usando o editor VSCode, salve o arquivo, abra-o em um navegador e experimente clicar no *link* que aparece na página.

O *link* pode se referir a uma página local, como no exemplo a seguir, em que se tem uma página HTML chamada **teste.html** na mesma pasta da página criada.

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Apontando para outra página da pasta</h1>
    <a href="teste.html">Vamos para a página Teste</a>
  </body>
</html>
```

Outra possibilidade é usar **<a>** para apontar para uma seção de uma página com bastante conteúdo (um texto com termos de uso de um *software*, por exemplo). Para isso, usa-se “#” no endereço.

```
<a href="#secao2">Ir para a seção 2</a>
<p>
  <!-- algum texto bem extenso-->
</p>
<h2 id="secao2">Seção 2</h2>
<p>
  <!-- outro texto bem extenso-->
</p>
```

Por padrão, *links* textuais ficam sublinhados e em azul; quando visitados, tornam-se roxos.

Imagens podem ser usadas como *hyperlinks*, como mostra o trecho a seguir:

```
<a href="teste.html">
  
</a>
```

Tags de listas

Quando se fala em organização de listas em HTML5, tem-se à disposição três tipos de listas, sendo elas: não ordenadas, ordenadas e de definição, cada uma sendo representada por suas respectivas *tags*. Veja a seguir.

Lista não ordenada

Tag **** para construir a lista e *tag* **** para cada linha dentro da lista. Veja o exemplo no trecho a seguir.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Lista não ordenada HTML5</title>
    </head>
    <body>
        <ul>
            <li>Funcionário 1</li>
            <li>Funcionário 2</li>
        </ul>
    </body>
</html>
```

Lista ordenada

Tag **** para construir a lista e tag **** para cada linha dentro da lista. Veja o exemplo no trecho a seguir.

```
<!DOCTYPE html>
<html>
    <head>
        <title> Lista ordenada HTML5
    </head>
    <body>
        <ol>
            <li>Funcionário 1</li>
            <li>Funcionário 2</li>
        </ol>
    </body>
</html>
```

Lista de definição

Tag **<dl>** para construir a lista, tag **<dt>** para definir o termo e tag **<dd>** para a definição. Veja o exemplo no trecho a seguir.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Lista de definição HTML5</title>
    </head>
    <body>
        <dl>
```

```
<dt>Gerente</dt>
<dd>Funcionário 1</dd>
<dt>Atendente</dt>
<dd>Funcionário 2</dd>
</dl>

<body>
</html>
```

A próxima figura mostra como as três listas ficam quando interpretadas pelo navegador, na seguinte ordem: não ordenadas, ordenadas e de definição.

- Funcionário 1
 - Funcionário 2
-
1. Funcionário 1
 2. Funcionário 2

Gerente
Funcionário 1
Atendente
Funcinário 2

Figura 11 – Tipos de listas HTML5
Fonte: Senac EAD (2023)

O desafio a seguir está disponível de forma interativa no conteúdo *on-line*. Acesse o conteúdo *on-line* e pratique! Nesta versão (impressão), o desafio está proposto de forma estática.

Desafio interativo 1



Utilizando tags **h**, **p**, **ul**, **li** e **a**, monte uma página semelhante à da imagem a seguir. Para o link, a sugestão é usar o endereço <<https://www.ead.senac.br>> e fazer com que abra em nova aba ou janela.

Fatos sobre HTML

A seguir algumas características de HTML

- Não é uma linguagem de programação
- Espaços em texto escrito não necessariamente será visualizado como espaço na página
- Quebras de linha apenas usando tags específicas, como "p" e "br"
- O nome da tag "a" vem de "anchor", ou "âncora"
- O atributo "target=_blank" da tag "a" abre o link em nova aba ou janela

Faça cursos de programação no [Senac EAD](#)



Monte uma página HTML que resulte em algo similar à figura a seguir:

Meu Currículo

Apresentação

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus ante neque, scelerisque nec fermentum vitae, tincidunt vitae diam.

Experiências profissionais

- *Lorem ipsum (2020-2022)*
- *Dolor sit amet (2019-2020)*
- *Consectetur adipiscing (2016-2019)*

Contatos

Email: fulano@mail.com

Site: [Meu Portfolio](#)

Figura 12 – Exemplo de currículo

Fonte: Senac EAD (2023)

Use *tags* de texto e *hyperlink* de (para o *link*, use qualquer endereço da *web* ou de outra página que esteja em sua pasta). No lugar do texto “*Lorem ipsum*” de teste, redija algo significativo.

Tags multimídia

Essas *tags* são funcionalidades implementadas no HTML5 que permitem a inserção direta de conteúdo multimídia dentro do HTML. Veja, a seguir, quais são elas.



Essa *tag* é utilizada para inserir uma imagem dentro da estrutura HTML, sempre acompanhada do atributo **src**, o qual permite dizer onde a imagem está localizada. Observe o trecho a seguir.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Imagen HTML5</title>
    </head>
    <body>
        
    </body>
</html>
```

Note que alguns atributos compõem a *tag* **** descrita no exemplo acima, e é possível destacar alguns detalhes sobre eles. O **src** é o atributo responsável por informar onde a imagem está localizada. Note que quando se utiliza somente o nome do arquivo (**imagem.jpg**), a estrutura entende que a imagem em questão está na mesma pasta onde o arquivo HTML está sendo armazenado. Caso essa imagem esteja em um diretório diferente ou em um nível antes do arquivo em questão, é possível demonstrar o caminho das seguintes formas, como mostra o trecho a seguir.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Imagen HTML5</title>
    </head>
    <body>
        
    </body>
</html>
```

Note que, no trecho citado, o arquivo **imagem.jpg** encontra-se dentro do diretório: **arquivos/imagens/imagem.jpg**. Já no próximo trecho, utiliza-se “..” para dizer que a imagem se encontra no diretório anterior ao do arquivo HTML. Veja:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Imagem HTML5
  </head>
  <body>
    
  </body>
</html>
```

Ainda é possível observar os atributos **width** e **height**, os quais definem, respectivamente, a largura e a altura da imagem em *pixels*. E não menos importante, há o atributo **alt**, o qual exibe um texto alternativo caso a imagem não possa ser exibida.

No trecho a seguir, pode-se observar um exemplo de uma página acessando uma imagem em um repositório remoto. Note que este exemplo pode ser testado em seu editor de texto, sem a necessidade da presença do arquivo (imagem) de forma local.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Imagem HTML5
  </head>

  <body>
    <p>
      <h3>Exemplo de imagem remota </h3>
      
    </p>
  </body>
</html>
```

<video>

Essa *tag*, como propriamente dita, é responsável pela inserção de um vídeo ao conteúdo, porém ela necessita de alguns parâmetros para funcionar corretamente, como uma *tag* interna denominada **<source>**, que é responsável por anexar os conteúdos multimídia. Essa *tag* tem atributos como o local onde o vídeo está localizado (atributo **src**) e o tipo de vídeo (atributo **type**).

Alguns outros atributos da *tag* <video> são:

**controls**

Exibição de botões de controle no vídeo, como **Play** e **Pause**.

autoplay

Vídeo inicia automaticamente com a página.

loop

Vídeo fica em repetição.

muted

Vídeo fica sem áudio.

Note que o trecho a seguir apresenta o código responsável pela inserção de um vídeo dentro da estrutura do HTML, em que os atributos **width** e **height** são responsáveis por definir as medidas de tamanho do vídeo (largura e altura, respectivamente). Ainda há atributos para exibir os controles do vídeo (**controls**) e para inicialização automática (**autoplay**).

```
<!DOCTYPE html>
<html>
  <head>
    <title>Videos HTML5
  </head>
  <body>
    <video width="600" height="350" controls autoplay>
      <source src="video.mp4" type="video/mp4">
      <source src="video.ogg" type="video/ogg">
    </video>
  </body>
</html>
```

<audio>

Com funcionamento semelhante à *tag* de vídeo, essa *tag* é responsável por adicionar arquivos de áudio à estrutura HTML. Ela também conta com o auxílio da *tag* <source>. Veja:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Áudio HTML5</title>
  </head>
  <body>
    <audio>
      <source src="audio.mp3" type="audio/mpeg">
    </audio>
  </body>
</html>
```

Crie uma página com uma galeria de quatro figuras. Baixe imagens da Internet na pasta onde foi salvar a página HTML. Separe as imagens usando **
** ou **<hr/>**. Use o atributo **width** para que todas fiquem com a mesma largura.

Tags de formulário

Formulários são estruturas formadas por campos de texto, botões, *selects*, *checkboxes*, entre outros. Um formulário tem como objetivos a obtenção e a transferência de informações dentro de uma estrutura HTML, podendo interagir com outras linguagens.

Um formulário dentro do HTML é representado pela tag **<form>**, a qual indica a abertura de um novo formulário. Essa tag ainda recebe alguns atributos, os quais você verá a seguir.

Method

Esse atributo recebe o método HTTP (*hyper text transfer protocol*), que será usado pelo formulário, sendo ele GET ou POST.

GET – Quando usado, realiza a passagem dos valores do formulário diretamente pela URL (*uniform resource locator*). Esses valores ficam facilmente visíveis no navegador. Veja o exemplo em que os campos **nome** e **idade** são passados por meio da URL (para a página **recebedados**) utilizando o método GET:

<<https://www.meusite.com/recebedados?nome=Luis&idade=25>>

POST – Diferentemente do método GET, o método POST não utiliza a URL, pois envia os dados de forma interna, ou seja, não são visíveis primariamente para o usuário.

Action



Esse atributo indica para qual página os dados do formulário serão enviados. Note que essa página geralmente representa um serviço de *back-end* que receberá os valores e os utilizará para operações.

Method

Esse atributo recebe o método HTTP (*hyper text transfer protocol*), que será usado pelo formulário, sendo ele GET ou POST.

GET – Quando usado, realiza a passagem dos valores do formulário diretamente pela URL (*uniform resource locator*). Esses valores ficam facilmente visíveis no navegador. Veja o exemplo em que os campos **nome** e **idade** são passados por meio da URL (para a página **recebedados**) utilizando o método GET:

```
<https://www.meusite.com/recebedados?nome=Luis&idade=25>
```

POST – Diferentemente do método GET, o método POST não utiliza a URL, pois envia os dados de forma interna, ou seja, não são visíveis primariamente para o usuário.

Action

Esse atributo indica para qual página os dados do formulário serão enviados. Note que essa página geralmente representa um serviço de *back-end* que receberá os valores e os utilizará para operações.

Dentro de uma *tag* de formulário é possível receber outras *tags*, que são responsáveis por estruturar e receber os dados digitados pelo usuário. Veja, a seguir, as principais.

```
<input>
```

Essa é a principal tag dentro de um formulário. Ela recebe os dados digitados pelo usuário e conta com alguns atributos:

- ◆ **type** – Demonstra o tipo de campo e os mais comuns são: **text**, **date**, **file**, **button**, **hidden**, **password** e **submit**.
- ◆ **value** – Define o valor do campo, ou seja, o dado que o campo recebe.

- ◆ **name** – É uma identificação do campo, muitas vezes utilizada para o reconhecimento dos campos na página que recebe o formulário.
- ◆ **id** – É a identificação padrão para uso e interação de um campo com outras linguagens, por exemplo: JavaScript.

<select>

É uma estrutura que abriga diversos outros campos, a fim de gerar uma seleção. As opções presentes dentro de <select> são disponibilizadas pela tag <option>.

<textarea>

Campo semelhante ao <input>, porém tem uma entrada de texto e o tamanho visual é variável.

<label>

Define um rótulo ou uma descrição para um campo de formulário.

No trecho a seguir, é possível ver a estrutura de um formulário básico para cadastro de funcionários. Note que, aqui, implementou-se, inicialmente, o atributo **action**, dentro da tag <form>, sendo esse atributo responsável por informar para qual página os dados que estão sendo submetidos no formulário serão enviados.

Em seguida, é possível ver o método de envio (**method**). Nesse caso, está sendo utilizado o método POST, o qual já foi descrito anteriormente. Veja que o **input** é do tipo **text**, ou seja, você receberá dados em texto por meio dele. Este ainda tem o atributo **name**, o qual é responsável por identificar o campo. Há ainda o **required**, que informa que o campo (**input**) é de preenchimento obrigatório, ou seja, não é possível submeter o formulário caso esse campo não esteja preenchido. Sugere-se o teste do trecho a seguir em seu editor de texto, para que você observe melhor o funcionamento do formulário.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Meu Site</title>
  </head>
  <body>
    <h3>Primeiro exemplo de form</h3>
    <form action="paginaderecebimento" method="POST">
      <label>Nome:</label>
```

```
<input type="text" name="nome_funcionario" required />
<br />
<input type="submit" value="Cadastro" />
</form>
</body>
</html>
```

Note que, no próximo trecho, há um formulário mais completo e desenvolvido, abordando grande parte dos atributos e das *tags* vistas anteriormente. Veja que agora há **input** do tipo **email**, o qual aceita apenas padrões reconhecidos (email@provedor.com). Há ainda um **input** do tipo **password**, que deixa oculta a senha digitada pelo usuário (não é criptografia). Por fim, há um **select**, com duas opções para o usuário selecionar (escolher apenas uma), seguido de uma área de texto livre (**textarea**). O **input** do tipo **submit** remete a um botão que tem como funcionalidade submeter os dados preenchidos no formulário, para a página definida no **action** (**paginaderecebimento**).

```
<!DOCTYPE html>
<html>
<head>
    <title>Meu Site </title>
</head>
<body>
    <form action="paginaderecebimento" method="POST">
        <label>Nome: </label>
        <input type="text" name="nome_funcionario" />
        <br/>
        <label>Email: </label>
        <input type="email" name="email_funcionario" />
        <br/>
        <label>Senha: </label>
        <input type="password" name="senha_funcionario" />
        <br/>
        <label>Atuação: </label>
        <select name="atuacao">
            <option value="Atendente">Atendente </option>
            <option value="Gerente">Gerente </option>
        </select>
        <br/>
        <label>Comentário <label>
        <textarea name="comentario"> </textarea>
        <br/>
        <input type="submit" value="Cadastro" />
    </form>
</body>
</html>
```

Na imagem a seguir, é possível visualizar o formulário programado anteriormente. Note que ele está em HTML5 puro; em outras palavras, não tem nenhum tipo de estilização.

Nome:

Email:

Senha:

Atuação:

Comentário

Cadastrar

Figura 13 – Tags de formulário HTML5

Fonte: Senac EAD (2023)

O desafio a seguir está disponível de forma interativa no conteúdo *on-line*. Acesse o conteúdo *on-line* e pratique! Nesta versão (impressão), o desafio está proposto de forma estática.

Desafio interativo 2



Utilizando tags **h**, **form**, **input**, **select** e **label**, construa uma página próxima da imagem a seguir. Tag **br** pode ser usada para a separação de linhas.

Informe seu endereço

Rua: Número:

Cidade: Estado:

Tipo: Residencial Comercial

Radio buttons, ou botões de seleção única, têm a peculiaridade de serem apresentados em grupo, e apenas uma das opções pode ser selecionada por vez.

```
<p>Qual o seu curso?</p>

<input type="radio" id="tds" name="opcoes-curso" value="TDS">
<label for="tds">Téc. Desenvolv. Sistemas</label>

<input type="radio" id="ipi" name="opcoes-curso" value="TII">
<label for="ipi">Téc. Informática para a Internet</label>
```



É importante que na tag **<input>** do tipo **radio** sejam informados nome e valor, pois eles serão usados no servidor para processar a informação.

Checkbox comporta-se de maneira semelhante, mas não precisa estar em grupo.

```
<input type="checkbox" id="news" name="news" value="Sim">
<label for="news">Deseja receber nossa newsletter mensal?</label>
```

É importante notar a relação de **radio buttons** e **checkboxes** com o elemento **<label>**: deve-se usar essa *tag* e seu atributo **for** com o valor do atributo **id** do elemento de **input**. Isso permite que o usuário não precise clicar exatamente na caixinha de **checkbox** ou no botão de rádio para selecionar a opção desejada, basta clicar no texto associado.

A seguir, veja um exemplo expandido para os trechos anteriores.

```
<!DOCTYPE html>
<html>
  <body>
    <form>
      <div>
        <label for="mail">Digite seu email:</label>
        <input type="email" id="mail" />
      </div>
      <div>
        <p>Qual o seu curso?</p>
        <input type="radio" id="tds" name="opcoes-curso" value="TDS">
        <label for="tds">Téc. Desenvolv. Sistemas</label>

        <input type="radio" id="ipi" name="opcoes-curso" value="TII">
        <label for="ipi">Téc. Informática para a Internet
      </div>
      <div>
        <input type="checkbox" id="news" name="news" value="Sim">
        <label for="news">Deseja receber nossa newsletter mensal?
      </div>
    </form>
  </body>
</html>
```

Digite seu email:

Qual o seu curso?

- Téc. Desenvolv. Sistemas Téc. Informática para a Internet
 Deseja receber nossa newsletter mensal?



Figura 14 – Página resultante do código anterior
Fonte: Senac EAD (2023)



Crie uma página com um formulário de candidatura a uma vaga de emprego com os seguintes campos:

- ✓ Nome completo (caixa de texto) ✓ Telefone (caixa de texto)
- ✓ Vaga desejada (seleção com opções “Atendente”, “Pedreiro”, “Programador” e “Professor”)
- ✓ Necessidade especial ou não (**checkbox**)

Não se esqueça do botão para submeter os dados.

Tags de tabelas

Essa *tag* denominada `<table>` tem como objetivo criar uma tabela dentro do HTML, sendo formada por outras *tags* que definem a estrutura dessa tabela, sendo elas:

`<tr>` Representa uma linha da tabela.

`<td>` Representa uma coluna da tabela.

**<th>**

Representa um título destacado dentro da tabela.

<thead>

É uma tag estrutural, que representa o cabeçalho de uma tabela.

<tbody>

É uma tag estrutural que representa o corpo da tabela.

<tfoot>

É a tag estrutural que representa o rodapé de uma tabela.

No trecho a seguir, há um exemplo mais simples de uma tabela, contendo apenas elementos de linhas (**tr**) e colunas (**td**). Essa tabela tem o objetivo de exibir os salários dos funcionários de acordo com o cargo deles. Note que, em cada coluna (**td**) da primeira linha (**tr**), há a descrição do cargo. Já nas colunas da segunda linha, é possível ver os respectivos salários.

Para visualizar melhor, aconselha-se o teste do código a seguir em seu editor de texto.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Tabela HTML5</title>
  </head>
  <body>
    <h3>Tabela salarial 2023</h3>
    <table border='1'>
      <tr>
        <td>Gerente</td>
        <td>Atendente</td>
        <td>Auxiliar</td>
      </tr>
      <tr>
        <td>R$5040,00</td>
        <td>R$2450,00</td>
        <td>R$1640,00</td>
      </tr>
    </table>
  </body>
</html>
```

O trecho a seguir demonstra a criação de uma tabela mais complexa, com todos os elementos citados anteriormente. Essa tabela tem o objetivo de exibir os salários dos cargos de

uma empresa. É possível notar que a tabela tem alguns atributos aplicados, como: **border**, o qual define a espessura da borda da tabela; **colspan**, que define quantas colunas o campo ocupará; e **rowspan**, que define quantas linhas o campo ocupará.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Tabela HTML5</title>
  </head>
  <body>
    <table border='1'>
      <thead>
        <tr>
          <th colspan="4">Tabela Salarial</th>
        </tr>
      </thead>
      <tfoot>
        <tr>
          <td colspan="4">Atualizada em: 2023
        </tr>
      </tfoot>
      <tbody>
        <tr>
          <td rowspan="2">Cargos </td>
          <td>Gerente </td>
          <td>Atendente </td>
          <td>Auxiliar </td>
        </tr>
        <tr>
          <td>R$5000,00 </td>
          <td>R$1640,00 </td>
          <td>R$1450,00 </td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```

Na próxima figura, veja o resultado da criação da tabela, quando exibida pelo navegador.

Tabela Salarial			
Cargos	Gerente	Atendente	Auxiliar
R\$5000,00	R\$1640,00	R\$1450,00	
Atualizada em: 2023			

Figura 15 – Tag de tabelas HTML5

Fonte: Senac EAD (2023)

<table> já foi usado há muito tempo para organizar elementos na tela. Essa prática, no entanto, é obsoleta e problemática, pois não permite fluidez na página e adaptação dela a diferentes resoluções. Assim, **<table>** deve ser usado apenas para tabular dados. A formatação de *layout* deve ser aplicada com *tags* estruturais ou *containers*, como **<div>**, aplicando estilização com CSS.

O desafio a seguir está disponível de forma interativa no conteúdo *on-line*. Acesse o conteúdo *on-line* e pratique! Nesta versão (impressão), o desafio está proposto de forma estática.

Desafio interativo 3



Usando a **table** e suas *tags*, bem como seus atributos relacionados, monte uma página com uma tabela semelhante à da imagem a seguir.

Alunos Matriculados

Nome	Matrícula	Data de início
Maria Silva	6325	20/10/2022
José Souza	6859	15/12/2022
Ana Siqueira	7854	01/03/2023

Tags estruturais HTML5

Como propriamente dito, as *tags* estruturais servem para estruturar a organização de uma página *web*. São *tags* responsáveis por proporcionar uma melhor visualização do conteúdo de um sistema *web* para o usuário por meio do navegador. As *tags* estruturais do HTML5 são as listadas a seguir.

<header>

Essa *tag* é utilizada para definir o cabeçalho de uma página. Normalmente é bastante reutilizada na maior parte das páginas do sistema.

<nav>

É a *tag* amplamente utilizada para definir a navegação e abrigar a criação de menus e listas.

<main>

Essa *tag* é responsável por apresentar o conteúdo de determinado tópico.

<article>

Essa *tag* é uma estrutura independente dentro da página. Ela tem conteúdo e pode, ou não, estar dentro de uma seção (<section>). Geralmente se refere a um texto independente na página, por exemplo, um *post* de *blog* ou uma notícia.

<section>

Separa a página em seções. Um <section> pode representar um conjunto de elementos, como <article> (por exemplo, uma seção com vários *posts* de *blog* sobre um determinado assunto) ou trechos específicos e significativos de um conteúdo (por exemplo, uma divisão do <article> em introdução e conclusão).

<aside>

É uma *tag* de conteúdo relacionada ao conteúdo ao seu redor.

<div>

É uma das *tags* estruturais mais utilizadas. Ela define uma divisão dentro de uma página. É muito utilizada na organização do conteúdo dentro do HTML5.

<footer>

Define o rodapé da página.

Observação: nenhuma das tags estruturais tem autofechamento.

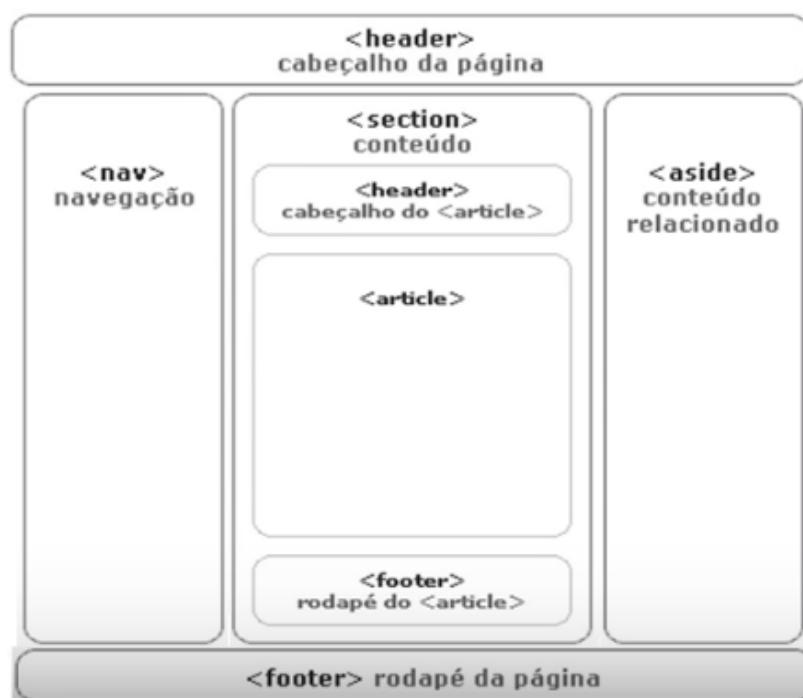


Figura 16 – Tags estruturais HTML5

Fonte: Melo (c2023)

Na figura anterior, é possível ver uma estrutura em HTML5 de uma página *web*. Note que, inicialmente, há a tag **<header>** contendo o cabeçalho da página. Na lateral esquerda, é possível verificar uma estrutura de navegação (*tag* **<nav>**) que poderia abrigar um *menu* lateral. Na parte central da imagem, há a seção de conteúdo (**<section>**), seguida de um novo cabeçalho que precede a *tag* **<article>**. Note que, na lateral direita, há a presença da *tag* **<aside>**, apresentando o conteúdo relacionado. Por fim, há o rodapé do **<article>** e também da página em si (**<footer>**).

A estrutura mostrada nessa figura pode ser representada no seguinte código em HTML5:

```
<!DOCTYPE html>
<html>
  <header>
    <!--Cabeçalho feral da página-->
  </header>
  <nav>
    <!--Navegação-->
```

```
</nav>
<section>
    <header>
        <!--Cabeçalho do article-->
    </header>
    <article>
        <!--Conteúdo article-->
    </article>
    <footer>
        <!--Rodapé article-->
    </footer>
</section>
<aside>
    <!--Conteúdo relacionado-->
</aside>
<footer>
    <!--Rodapé final da página-->
</footer>
</html>
```

É importante notar que, apesar de poderem denotar regiões específicas em uma página, essas *tags* não geram nenhum alinhamento ou posicionamento específico na tela. Isso é alcançado usando CSS.

Veja outro exemplo:

```
<!DOCTYPE html>
<html>
    <body>
        <h1>Nossos cursos
        <nav>
            <ul>
                <li><a href="index.html">Página Principal</a></li>
                <li><a href="sobre.html">Sobre a Escola</a></li>
                <li><a href="contato.html">Entre em Contato</a></li>
            </ul>
        </nav>
        <main>
            <section id="tecnologia">
                <article>
                    <h2>Técnico em Desenvolvimento de Sistemas</h2>
                    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus ante neque, scelerisque nec fermentum vitae, tincidunt vitae diam.</p>
                </article>

                <article>
                    <h2>Técnico em Informática para a Internet</h2>
                    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus ante neque, scelerisque nec fermentum vitae, tincidunt vitae diam.</p>
                </article>
            </section>
        </main>
    </body>
</html>
```

```
<section id="gestao">
    <article>
        <h2>Técnico em Administração</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus ante neque, scelerisque nec fermentum vitae, tincidunt vitae diam.</p>
    </article>

    <article>
        <h2>Técnico em Logística</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus ante neque, scelerisque nec fermentum vitae, tincidunt vitae diam.</p>
    </article>
</section>
</main>

</body>
</html>
```

O resultado no navegador deverá ser algo como o exemplo mostrado na figura a seguir. Note que, visualmente, nada acontece em específico. Porém, semanticamente, a página está mais preparada para buscadores, por exemplo.

Nossos cursos

- [Página Principal](#)
- [Sobre a Escola](#)
- [Entre em Contato](#)

Técnico em Desenvolvimento de Sistemas

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus ante neque, scelerisque nec fermentum vitae, tincidunt vitae diam.

Técnico em Informática para a Internet

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus ante neque, scelerisque nec fermentum vitae, tincidunt vitae diam.

Técnico em Administração

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus ante neque, scelerisque nec fermentum vitae, tincidunt vitae diam.

Técnico em Logística

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus ante neque, scelerisque nec fermentum vitae, tincidunt vitae diam.

Figura 17 – Página usando tags semânticas estruturais

Fonte: Senac EAD (2023)

Sempre que for possível e adequado, use *tags* semânticas para separar conteúdo. Porém, algumas vezes, deseja-se apenas agrupar alguns elementos da página para algum propósito, como aplicar um estilo ou reposicioná-lo na página. Para isso, é possível usar o *container* genérico **<div>**. Veja, no exemplo a seguir, que elementos **<div>**, por padrão, aparecem um abaixo do outro na página.

```
<!DOCTYPE html>
<html>
    <body>
        <span><b>Texto à esquerda</b></span><span><i>Texto à direita</i></span>
```

```
<div>
    <span>Texto no div 1</span>
</div>
<div>
    <span>Texto no div 2</span>
</div>
</body>
</html>
```

Elementos `<div>` serão bastante usados na estilização da página com CSS ou na aplicação de ferramentas como o Bootstrap.

Tag *iframe*

Um *iframe* é uma funcionalidade disponível no HTML5 que permite executar conteúdos de outras páginas dentro de um elemento da página principal. Com isso, é possível exibir documentos e vídeos externos, por exemplo, dentro da página HTML principal. Implementado por meio da tag `<iframe>`, um *iframe* tem alguns atributos básicos, como: **src** (origem), **width** (largura da janela de conteúdo) e **height** (altura da janela de conteúdo). No trecho a seguir, veja um *iframe* básico que exibe um vídeo externo dentro da página principal.

```
<!DOCTYPE html>
<html>
    <head>
        <title>iFrame HTML5</title>
    </head>
    <body>
        <iframe src="https://www.youtube.com/embed/1A7VVPKHqho" width="400" height="300"></iframe>
    </body>
</html>
```

Na figura a seguir, veja o resultado do *iframe*, quando interpretado pelo navegador.





Figura 18 – Visualização no navegador: *iframe* HTML5

Fonte: Senac EAD (2023)