

# **Apostila de HTML**

**Professora: Vanessa Lago Machado**  
**Disciplina: Criação de Páginas Web**

**Passo Fundo**  
**2018**

## Sumário

1.	INTRODUÇÃO A WEB .....	3
1.1.	História da Internet .....	3
1.2.	Linguagens <i>Client-side</i> e <i>Server-side</i> .....	6
1.2.1.	Linguagens Client-side (Front-end) .....	6
1.2.2.	Linguagens Server-side (Back-end) .....	6
1.3.	Tipos de Páginas Web .....	7
2.	HTML .....	8
2.1.	Estrutura básica do HTML .....	8
2.1.1.	Atributos .....	9
2.1.2.	Tags textuais .....	9
2.1.3.	Formatação de caracteres especiais: .....	10
2.1.4.	Comentário .....	11
2.2.	Listas .....	11
2.2.1.	Listas Ordenadas .....	11
2.2.2.	Listas Não Ordenadas: .....	12
2.2.3.	Listas de Definição: .....	14
2.3.	Link .....	15
2.3.1.	Atributos .....	15
2.3.2.	Voltar e Avançar: .....	16
2.3.3.	Âncora: .....	16
2.4.	Imagem: .....	17
2.4.1.	Atributos: .....	17
2.5.	Tabela: .....	20
2.5.1.	Mesclando Células .....	21
2.6.	Formulários .....	23
2.6.1.	Interação com servidor .....	24
2.6.2.	Atributos da tag input .....	25
2.6.3.	Outros campos válidos para um formulário: .....	26
2.6.4.	HTML5 em formulários .....	28
	Referências .....	31

## 1. INTRODUÇÃO A WEB

A Web refere-se a um enorme sistema de informação baseado no modelo cliente-servidor, o qual se encontra distribuído pelo mundo, para compreendê-la melhor é importante conhecer um pouco de sua história.

### 1.1. História da Internet

Em meados de 1969, durante a guerra fria, surgiu à necessidade de realizar a comunicação entre as bases militares dos EUA, para isso a DARPA, agência nacional de defesa criada pelo exército dos EUA para criação de tecnologias durante a guerra fria, criou uma rede de comunicação chamada ARPANET. Essa rede permitia a comunicação entre as diferentes bases militares, contudo a comunicação entre os computadores ainda era dificultado, devido ao fato de cada base militar possuir um computador diferente, dessa forma surgiu o protocolo de comunicação *Network Control Protocol* (NCP).

Conforme Tanenbaum e Wetherall (2011, p. 18), protocolo trata-se de “um acordo entre as partes que se comunicam, estabelecendo como se dará a comunicação”, em que a violação do protocolo dificultará ou inviabilizará a comunicação entre as partes.

Todavia, com o passar do tempo, e o aumento incontrolável de máquinas ingressando na rede, o protocolo NCP começou a não atender a demanda exigida. Então, em 1973, Robert Kahn criou o protocolo *Transfer Control Protocol* (TCP), utilizado para transferência de dados, o qual possui como princípio segmentar os dados em pacotes.

Além disso, Robert Kahn também foi responsável pela criação do termo *Internetting*, usado para representar redes que se comunicavam entre si.

O protocolo TCP resolveu o problema da transferência de dados, mas não conseguiu tratar o problema de destinação das mensagens, com a identificação das máquinas. Com esse intuito Vinton Cerf criou o protocolo *Internetwork Protocol* (IP). Assim, com a junção dessas soluções surgiu o conjunto de protocolos TCP/IP, utilizado até os dias de hoje para transferência dados e identificação de máquinas.

Com o tempo surgiu novos desafios, em que no *Conseil Européen pour la Recherche Nucléaire* (CERN) iniciou-se um projeto para viabilizar o

compartilhamento de textos científicos entre os cientistas da Europa, pois, até o momento, era possível compartilhar seus textos com pessoas específicas com uso de e-mail, mas a demanda era tornar esses textos públicos. Dessa forma, Tim Berners-Lee, físico do CERN, no projeto *Enquire*, visava à criação de hipertextos. Então, em meados de 1990, ele teve a ideia de unir o conjunto de protocolos TCP/IP e os hipertextos, possibilitando compartilhar os conteúdos de forma pública. Para isso criou um novo protocolo *Hiper Text Transfer Protocol* (HTTP), utilizado para transferência de dados na rede mundial de computadores, a *World Wide Web*.

Para dispor o conteúdo na Web os arquivos são identificados por meio de sua *Uniform Resource Locator* (URL), traduzido como Localizador Padrão de Recursos, que se refere a uma forma padronizada de identificar qualquer recurso no domínio da internet, para isso a URL possui como sintaxe:

Protocolo: //servidor/caminho\_do\_arquivo

Desse modo, por meio do projeto *Enquire*, Tim Berners-Lee criou: a linguagem de marcação HTML, para dispor o conteúdo na Web; a URL, para endereçar cada conteúdo disponível; o protocolo HTTP, para transferência de dados. Além disso, criou o primeiro servidor e navegador, para tornar viável a publicação de conteúdos. Dessa forma, Tim Berners-Lee é conhecido como o pai da Web, criando também a *World Wide Web Consortium* (W3C), uma comunidade internacional que tem como objetivo garantir a manutenção e desenvolvimento da Web.

Para entendermos melhor o endereçamento das páginas web, com uso de URL, vamos segmentá-la:

Cada máquina possui um endereço único dentro da rede em que encontra-se, conhecido como IP, o qual é segmentado em 4 blocos de 8 bits cada, exemplo: 192.168.0.100.

Logo, cada página web encontra-se hospedada em uma máquina, assim, cada página Web possui um endereço IP associado, porém para facilitar nossa busca pelas páginas esses endereços são mascarados com o uso do protocolo DNS, o qual atribui um nome de domínio para um endereço IP, exemplo: Google.com.br. Ainda, um domínio pode ser subdividido em subdomínios, exemplo:

- **Domínio:** ifsul.edu.br
- **Subdomínio:** ead.ifsul.edu.br

O domínio é dividido em categorias e localidade (domínio.categoria.local).

**Categorias:** Refere-se ao tipo de conteúdo publicado na Web, as principais são:

- edu – Instituições de ensino
- com – Instituições comerciais
- gov – Instituições governamentais
- adv – advogados
- imo – imobiliárias
- org - organizações

**Localidade:** Refere-se ao local de origem em que encontra-se hospedado tal conteúdo, exemplo:

- br – Brasil
- ar – Argentina
- us – Estados unidos

Desse modo, ao acessar um conteúdo da Web é possível interpretar algumas informações sobre o conteúdo e/ ou empresa. Como exemplo, ao acessar o Website do Google (Google.com.br) sabe-se que Google é o nome de domínio do conteúdo, máscara dada para substituir o endereço IP do mesmo. Além disso, a empresa refere-se a uma instituição comercial e o Website acessado encontra-se hospedado no Brasil.

A linguagem de Marcação *Hyper Text Markup Language* (HTML) trata-se de uma linguagem da Web, utilizada para demarcar os componentes Web por meio de marcadores (Tags).

Dentre as características do HTML verifica-se que não é uma linguagem *case sensitive*, ou seja, não diferencia letras minúsculas de maiúsculas.

Se o HTML é somente uma linguagem de marcação, como podemos fazer uma página Web, visualmente chamativa e que possa interagir com o usuário? Como um portal e-commerce, por exemplo.

A resposta é simples: HTML + CSS + linguagem de programação.

Como linguagem de programação pode-se utilizar diversas linguagens, e, inclusive utilizar mais de uma linguagem em uma mesma página. Como exemplo é possível verificar que para validação de dados e animações é comum utilizar a linguagem de programação JavaScript e para estilização das páginas, tornando-as atrativas aos usuários, utiliza-se folhas de estilo *Cascading Style Sheets* (CSS).

## **1.2. Linguagens *Client-side* e *Server-side***

Para compreender melhor essa união de tecnologias é importante compreender como elas funcionam, existem dois tipos de linguagens para desenvolvimento Web: *client-side* e *server-side*.

### **1.2.1. Linguagens *Client-side* (Front-end)**

As linguagens *client-side*, também conhecido como *front-end*, são aquelas linguagens que são interpretadas pelo navegador, sem necessidade de envio dos dados a um servidor para obter uma resposta, ou seja, todo processamento ocorre diretamente no *browser* do usuário. Exemplo: HTML, CSS e Javascript.

Essa área de atuação destina-se ao desenvolvimento da interface do usuário, por isso profissionais que trabalham com essa área são chamados de programadores *front-end*.

### **1.2.2. Linguagens *Server-side* (Back-end)**

As linguagens de programação *server-side*, também conhecidas como *back-end*, são aquelas linguagens que são interpretadas pelo servidor, ou seja, ao realizar uma requisição utilizando uma linguagem de programação *back-end* essa requisição é enviada ao servidor, o qual processa a informação e envia para o navegador a resposta. Exemplo: PHP, Java e Ruby.

Essa área de atuação destina-se a programação que é interpretada pelo servidor, por isso os profissionais que trabalham com essa área são chamados de programadores *back-end*.

### 1.3. Tipos de Páginas Web

Com o uso destas tecnologias podemos realizar diversas implementações, como:

- **Sistema de Gerenciamento de Conteúdo:** Trata-se dos sistemas que possuem como finalidade a divulgação de um conteúdo único, sem foco em campanhas para venda ou qualquer publicidade, pode-se citar como exemplo o WebSite do Wikipédia, o qual “tem como propósito fornecer um conteúdo livre, objetivo e verificável, que todos possam editar e melhorar”. Ainda, sistema de gerenciamento de conteúdo visam criar, visualizar, atualizar e remover conteúdos de forma sistemática e organizada, permitindo que todas as operações sejam realizadas com facilidade.

Existem alguns frameworks que possibilitam o uso destes tipos de informações, pode-se citar o Joomla e WordPress.

- **WebSite:** Conjunto de páginas Web relacionadas entre si. Usualmente são utilizadas para divulgação de uma empresa ou marca, e pode tratar todas as informações desta, como: informações, história, contato, entre outros dados.
- **HotSite:** Página web ou conjunto de páginas relacionadas publicadas temporariamente com intuito de divulgar alguma novidade.

Atualmente a internet encontra-se como um ambiente virtual interconectado com o mundo real, o qual deve refletir as necessidades dos usuários, atendendo as expectativas dos empresários. Assim, o termo marketing digital está cada dia mais relacionado com o nosso cotidiano.

“O marketing digital é o conjunto de atividades que uma empresa (ou pessoa) executa online com o objetivo de atrair novos negócios, criar relacionamentos e desenvolver uma identidade de marca. ”

PEÇANHA, 2018.

## 2. HTML

Para desenvolver nossas soluções em HTML vamos utilizar a IDE Notepad++.

### 2.1. Estrutura básica do HTML

Um documento HTML é demarcado com o uso de Tags, e todas as tags devem abrir e fechar. Todo documento é composto pela seguinte estrutura básica, contendo:

- HTML: identificação do documento
- HEAD: cabeçalho, onde conterà as configurações básicas da página, como título, estilização, entre outras.
- BODY: corpo do documento, ou seja, todas as informações que devem ser mostradas na página Web, devem estar dentro desta TAG. Exemplo: tabelas, listas, imagens, textos, vídeo, link, formulários, entre outros.

Quadro 1- Primeiro teste em HTML

```
<HTML>
  <HEAD>
    <TITLE> Bom dia! </TITLE>
  </HEAD>
  <BODY> Teste HTML </BODY>
</HTML>
```

No Quadro 1 pode-se perceber a estrutura do HTML, disposta em uma página teste. Assim percebe-se que as tags são delimitam o espaço de cada componente ou container. A sintaxe de uma tag é:

- Abrindo uma tag: "<" + nome da tag + ">"
- Fechando uma tag: "</" + nome da tag + ">"

Assim, toda tag deve ser aberta e fechada, respeitando, ainda, a estrutura da página, ou seja, se uma tag <p> foi aberta e após a tag de negrito foi aberta <b>, a tag </b> deve ser fechada antes da tag </p> .



Existem algumas tags conhecidas como tag autocontida, a qual não necessita abrir e fechar, porém estas devem abrir e fechar ao mesmo tempo, como é o caso da tags: `img` (para inserção de imagem), `hr` (traço horizontal), `br` (quebra de linha), entre outras.

### 2.1.1. Atributos

Os atributos de uma tag são as propriedades específicas de cada elemento, a partir do qual é possível realizar alguma configuração específica neste elemento. Para isso todo atributo é colocado dentro da tag, com seu valor entre aspas duplas, exemplo: `<body bgcolor="green">`. Neste exemplo na tag `body` é configurado a cor de fundo da página web, por meio do atributo `bgcolor`, para a cor verde.

Obs: As cores podem ser definidas pelo padrão R[ed]G[reen]B[lue], no qual cada pigmento varia de 0 a 255 (ou 00 a FF hexa), pode ser definida a cor em HTML pelo nome da cor (em inglês), ou no padrão RGB.

### 2.1.2. Tags textuais

Como tags textuais podemos utilizar algumas TAGs de formatação de texto ou mesmo de delimitação. Dentre as TAGs de formatação, utiliza-se:

#### Tags de Estilos Físicos:

- `<B></B>` - Negrito
- `<I></I>` - Itálico
- `<U></U>` - Sublinhado

#### Headings (níveis de títulos)

- `<Hn></Hn>` - n variando de 1 até 6

#### Quebra de Linhas

- `<br />`

#### Parágrafo

- `<p>...</p>`

## **Linhas Horizontais**

- `<hr/>` - insere uma linha horizontal no documento

## **Subscrito e Sobrescrito**

Em algumas situações pode ser necessário a inserção de subscrito e sobrescrito, por exemplo para inserção de fórmulas, para isso se utiliza as tags:

- `<sub></sub>` - insere informações em subscrito
- `<sup></sup>` - insere informações em sobrescrito

## **Alinhamento**

Para tratar o alinhamento do texto podemos utilizar três formatos:

- Algumas Tags permitem o atributo align.

```
<p align="center"> teste </p>
```

- Pela Tag `<center>` é possível ajustar o alinhamento de determinado texto delimitado como centralizado

```
<p><center> teste </center></p>
```

- Uso do CSS, usando o atributo style.

```
<p style="text-align:center;"> teste </p>
```

Obs: O alinhamento definido pelas Tags style e align podem ser definidos como: justify, center, right e left.

### **2.1.3. Formatação de caracteres especiais:**

Nem sempre os caracteres acentuados são mostrados como deveriam, por isso podemos inserir a codificação de cada caractere ou especificar a codificação que os caracteres especiais e acentuações devem seguir.

#### **Sintaxe:**

```
<meta charset="character_set">
```

O valor do charset pode ser definido, dentre outras opções, como:

- UTF-8 – Codificação de caracteres para Unicode
- ISO-8859-1 - Codificação de caracteres para o alfabeto latino.

#### **2.1.4. Comentário**

Trata-se de um bloco do código que será rejeitado pelo navegador, ou seja, não será lido ou interpretado pelo navegador. Sendo assim, pode ser utilizado para isolar um bloco de códigos para teste ou mesmo para documentação do código, possibilitando que outro desenvolvedor possa visualizar seu código e rapidamente compreendê-lo.

##### **Sintaxe:**

```
<!-- Bloco de código -->
```

### **2.2. Listas**

Existem, em HTML, três tipos de listas: Ordenadas, não ordenadas e de Definição.

#### **2.2.1. Listas Ordenadas**

Trata-se de listas que possuem alguma necessidade de ordenação, ou seja, quando se faz necessário uma sequência de informações de forma ordenada (numeradas uma a uma).

##### **Sintaxe:**

```
<OL START=n TYPE= A|a|I|i|1>  
  <LI> Item 1 </LI>  
  <LI> Item 2 </LI>  
  (...)  
</OL>
```

Por padrão ao inserir uma lista ordenada esta será ordenada em forma numérica, iniciando a contagem em 1. Assim, utilizando a sintaxe para criação

de listas ordenadas podemos modificar esses padrões, como a sequência de início da lista, por meio do atributo `start=n`, sendo `n` o valor numérico que representa a ordem de início da lista. Podemos também alterar o padrão do tipo de lista, o qual pode ser ordem alfabética maiúscula (`A`) ou minúscula (`a`), ou em ordem de números romanos maiúsculos (`I`) ou minúsculos (`i`), como exemplo podemos ver a Figura 1, que representa a inserção de uma lista ordenada em números romanos minúsculos.

Figura 1 – Exemplo de código HTML para Lista Ordenada

**Exemplo:**  
**Meus animais de estimação:**  
`<OL TYPE=i>`  
    `<LI> Ajudante de Papai Noel </LI>`  
    `<LI> Coelho Pernalonga </LI>`  
`</OL>`

Percebe-se que o padrão da sequência inicial não foi alterado, iniciando em *i* (conforme Figura 2), esta poderia ser alterado com a inserção do atributo `start`. Por exemplo, para nossa lista começar em *iv* (quarta ordem numeral dos números romanos), bastaria inserir `start=4` na tag `OL`.

Figura 2 - Saída no navegador, conforme código da Figura 1

**Exemplo:**  
**Meus animas de estimação:**  
**i. Ajudante de Papal Noel**  
**ii. Coelho Pernalonga**

### 2.2.2. Listas Não Ordenadas:

Trata-se da inserção de listas, na qual a ordem dos itens não interfere na lista em si.

**Sintaxe:**

```
<UL TYPE= disk|square|circle >  
  <LI> Item 1 </LI>  
  <LI> Item 2 </LI>  
  ...  
</UL>
```

Ao inserir uma lista não ordenada cada item da lista será marcada por meio de um caractere simbólico (disco, quadrado ou círculo), na qual o caractere padrão é o disco. Assim, utilizando a sintaxe para criação destas listas podemos modificar esse caractere padrão, por meio do atributo type em que o valor pode ser *circle* (caractere círculo), *square* (caractere quadrado) ou *disk* (caractere disco), como exemplo podemos ver a Figura 3, que representa a inserção de uma lista não ordenada com o caractere disco para marcação dos itens.

Figura 3 - Exemplo de código HTML para lista não ordenada

**Exemplo:**

**Meus animais de estimação:**

```
<UL TYPE="disk">  
  <LI> Ajudante de Papai Noel </LI>  
  <LI> Coelho Pernalonga </LI>  
</UL>
```

A Figura 4 representa a visualização em um navegador do código da Figura 3. Percebe-se que o padrão de caractere foi ajustado, porém mantendo o padrão (disco).

Figura 4 - Saída no navegador, conforme código da Figura 3

**Meus animais de estimação:**

- Ajudante de Papai Noel
- Coelho Pernalonga

### 2.2.3. Listas de Definição:

Trata-se de uma lista utilizada para definição de termos, conhecida também como glossário.

#### Sintaxe:

```
<DL>  
    <DT> Termo </DT>  
    <DD> Definição </DD>  
</DL>
```

Ao inserir uma lista de definição, necessita-se inserir o termo (tag DT) e a definição deste termo (tag DD), possibilitando assim a inserção de um glossário na página Web, como exemplo podemos ver a Figura 5, que representa a inserção de uma lista de definição.

Figura 5 -Exemplo de código HTML para lista de definição.

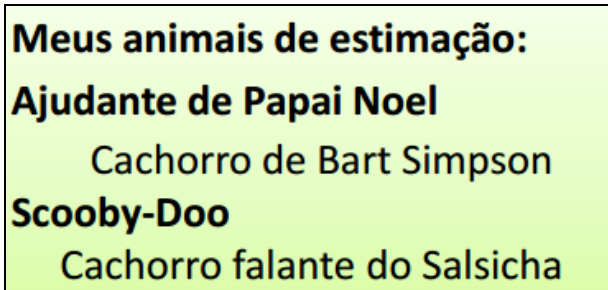
**Exemplo:**

**Meus animas de estimação:**

```
<DL>  
    <DT> Ajudante de Papai Noel </DT>  
    <DD> Cachorro do Bart Simpson </DD>  
    <DT> Scooby-Doo </DT>  
    <DD> Cachorro falante do Salsicha </DD>  
</DL>
```

A Figura 6 representa a visualização em um navegador do código da Figura 5. Percebe-se que o foram inseridos dois termos nesse exemplo (Ajudante de Papai Noel e Scoob-Doo) e a respectiva definição para cada um.

Figura 6 -Saída no navegador, conforme código da Figura 5



## 2.3.Link

Os Links são facilmente encontrados em todas as páginas Web, pois eles permitem a navegação do usuário entre diferentes páginas, para isso a tag `<a>` é usada. Em HTML os links são conhecidos como hyperlink.

### Sintaxe:

```
<a href="url">texto para o link</a>
```

### Exemplo de uso:

```
<a href="http://www.ifsul.edu.br/">IFSUL</a>
```

Por padrão, os links serão exibidos da seguinte forma em todos os navegadores:

- Um link não visitado é sublinhado e azul
- Um link visitado é sublinhado e roxo
- Um link ativo está sublinhado e vermelho

### 2.3.1. Atributos

Dentre os atributos que permitem configurar um hyperlink, encontram-se:

- `href`: indica o destino do link.
- `target`: especifica onde a página vinculada será exibida.

Obs: uma página vinculada é normalmente exibida na janela atual do navegador, a menos que você especifique outro destino, ou seja, ao clicar em

um link, por padrão, a página do link será aberta na mesma página e aba que encontra-se a atual. Para modificar isso é necessário alterar o atributo *target*=" \_blank "

_blank	Abre o link em uma nova janela ou aba do navegador
_self	Abre o link no mesmo local onde foi clicado (padrão)

### 2.3.2. Voltar e Avançar:

Botões de voltar e avançar são comuns e úteis na navegação de páginas Web. Possivelmente você já navegou em uma página que continha um botão “voltar”, “retornar”...

Para isso é possível utilizar código javascript, útil pois não será preciso editar o código em todas as páginas, atualizando o nome da página anterior, já que ele será capturado automaticamente do histórico do navegador.

#### Sintaxe:

Voltar:

```
<a href="javascript:window.history.go(-1)"> Voltar </a>
```

Avançar:

```
<a href="javascript:window.history.go(1)"> Avançar </a>
```

### 2.3.3. Âncora:

Trata-se de uma marcação dentro do documento (página), possibilitando a navegação no documento para estas marcações.

Em outras palavras uma âncora é um link que irá pular para alguma parte da mesma página onde se localiza o link.

Como exemplo, pesquise alguma informação no site da Wikipedia, nele todas as informações referentes ao item buscado localizam-se na mesma página, contudo há um índice superior que permite a navegação entre as informações.

#### Sintaxe:

Criando a marcação:



```
<a name="NOME">Nome</a>
```

Link para direcionar a esta marcação:

```
<a href="#NOME">meu link</a>
```

Lembre-se que ao referenciar um link para o local da âncora o caractere “#” na frente do nome da âncora é obrigatório, caso contrário o navegador tentará buscar uma outra página HTML com este nome.

## 2.4. Imagem:

Para inserção de imagem em HTML utiliza-se a tag <IMG>

### Sintaxe:

```

```

### 2.4.1. Atributos:

Os principais atributos da tag IMG encontram-se descritos na Figura 7.

Figura 7 - Atributos Tag IMG

Atributo	Função
src	Especifica a URL da imagem
alt	Texto alternativo para a imagem
Height	Altura da imagem (em pixel)
Widht	Largura da imagem (em pixel)

#### ➤ O atributo src:

Corresponde ao caminho da imagem, o qual pode ser:

- **URL:**

Trata-se da forma de identificar um recurso no domínio da internet, ou seja, neste caso corresponde a um link para uma imagem na Web.

Obs: uma URL é representada pela seguinte sintaxe:

Protocolo://servidor/caminho do arquivo

Por exemplo:

`http://painel.passofundo.ifsul.edu.br/uploads/img/201702141917561913785765_p.jpg`

Onde:

`http` → protocolo

`painel.passofundo.ifsul.edu.br` → servidor

`uploads/img/201702141917561913785765_p.jpg` → Caminho do arquivo

- **Caminho relativo:**

Quando o arquivo encontra-se relacionado ao diretório da página em questão.

Exemplo: Se a imagem se encontra no mesmo diretório do arquivo HTML, pode-se referenciar o caminho da imagem apenas referenciando o nome do arquivo.

```

```

Neste caso podemos referenciar uma imagem navegando a partir do diretório do arquivo HTML, ou seja, se o arquivo HTML encontra-se em um diretório e a imagem encontra-se em um diretório nomeado *Figuras* dentro do diretório raíz, basta referenciar esse caminho.

```

```

- **Caminho absoluto:**

Quando informamos o caminho completo da imagem referente ao Computador.

```

```

Estes formatos de referenciar arquivos seguem o mesmo padrão em todos os tipos de arquivos que desejar referenciar no documento HTML, por exemplo: links, vídeos, imagens, entre outros.

Ainda, qual formato optar deve-se levar em conta a necessidade de cada desenvolvimento, no qual o modo menos indicado é o caminho absoluto, pois este só servirá em seu computador.

### ➤ **Atributo Alt**

Este atributo serve para vincular um texto a uma imagem, ou seja, um valor alternativo que será lido quando a imagem não for carregada. Ainda, este é também lido por leitores de tela destinados a deficiências visuais, por isso devemos prestar atenção e descrever a imagem de forma que pessoas com dificuldade visual possam navegar em nossas páginas.

Exemplo:

```

```

### ➤ **Atributos Width e Height**

Estes atributos referem-se a largura e altura da imagem, respectivamente, as quais são referenciadas em pixel. Assim, quando não referenciamos estes atributos a imagem é mostrada de forma real (com suas dimensões reais). Assim, quando informamos somente um destes valores a imagem é ajustada para que a dimensão seja atribuída de forma relativa.

Exemplo:

```

```

Na imagem acima foi alterado a largura para 200px, logo a altura será ajustada para que a imagem não se distorça. Porém, algumas vezes precisamos enquadrar a imagem em um determinado bloco pré-definido, ou seja, onde a largura e altura possuem um valor definido. Neste caso a imagem poderá se distorcer para se adaptar a dimensão informada.

Exemplo:

```

```

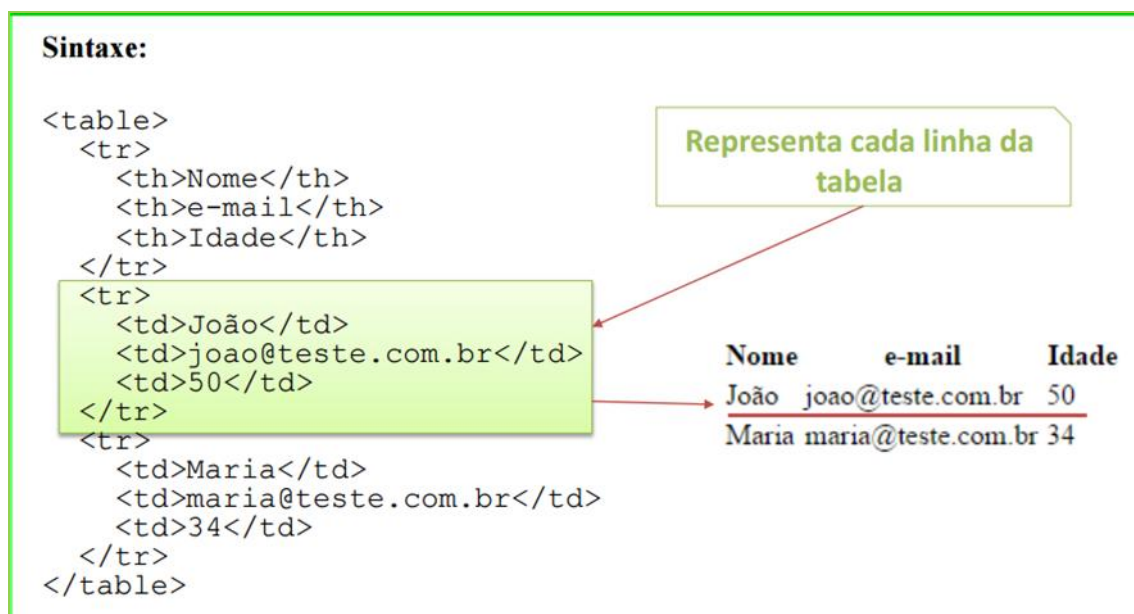
## 2.5. Tabela:

Tabelas podem ser utilizadas para criar tabela ou para Layout da Página (não indicado). Para inserir uma tabela em um documento HTML utilizamos a tag <TABLE>, na qual cada linha da tabela é definida pela tag <TR> e cada célula de dados da linha é definido com a tag <TD>.

O valor de uma célula de dados, ou seja, um dado pode ser: texto, imagem, listas, formulários, outras tabelas, etc.

Ainda, o cabeçalho da tabela é definido pela tag <TH> para cada célula do cabeçalho. Por padrão o cabeçalho é formatado com a estilização em negrito e centralizado.

Figura 8 - Sintaxe de Tabela em HTML



Podemos segmentar a tabela entre as partes: cabeçalho, corpo e rodapé. O que facilita a estilização desta, porém não é obrigatória.

Para isso usamos a tag `<thead>` para identificação do cabeçalho, `<tbody>` para identificação do corpo da tabela e `<tfoot>` para o rodapé. O interessante de utilizar essa sintaxe é que independente da ordem das três informações, o navegador interpreta cada elemento correto da tabela e monta seguindo a estrutura: cabeçalho + corpo + rodapé.

Exemplo:

Figura 9 - Sintaxe da tabela delimitada

```
<table>
  <caption>...</caption>
  <thead>
    <tr>
      <th> ... </th> ...
    </tr>
  </thead>
  <tfoot>
    <tr>
      <th> ... </th>...
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td> ... </td> ...
    </tr>
  </tbody>
</table>
```

Opcionalmente podemos utilizar o elemento `<caption>`, o qual permite inserir um título para a tabela, no formato legenda, como pode ser visto na Figura 9.

### 2.5.1. Mesclando Células

Um recurso semelhante ao mesclar do Excel é possível em HTML por meio do atributo `colspan` ou `rowspan`, dependendo do que deseja-se mesclar, células em colunas ou células em linhas, respectivamente.

Assim, na Figura 10 pode-se verificar a sintaxe para mesclar células de dados em colunas, em que o atributo `colspan` refere-se à quantidade de colunas que esta célula deve ocupar.

Figura 10 - Sintaxe de mesclar células por colunas

```
<h4>Celula ocupando duas colunas:</h4>

<table>
  <tr>
    <th>Nome</th>
    <th colspan="2">Telefone</th>
  </tr>
  <tr>
    <td>Bill Gates</td>
    <td>555 77 854</td>
    <td>555 77 855</td>
  </tr>
</table>
```

O mesmo pode ser visto na Figura 11, a qual trata da sintaxe para mesclar células de dados em linhas, em que o atributo *rowspan* refere-se à quantidade de linhas que esta célula deve ocupar.

Figura 11 - Sintaxe de mesclar células por linhas

```
<h4>Celula ocupando duas linhas:</h4>
<table>
  <tr>
    <th>Nome</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telefone:</th>
    <td>555 77 854</td>
  </tr>
  <tr>
    <td>555 77 855</td>
  </tr>
</table>
```

Como exemplo, na Figura 12 pode-se verificar o código à esquerda e o resultado gerado a direita.

Figura 12 - Exemplo de uso de células mescladas

```
<table border="1">
  <tr>
    <th>Frutas</th>
    <th>Verduras</th>
    <th>Grãos</th>
    <th>...</th>
  </tr>
  <tr>
    <td>Maçã</td>
    <td>Alface</td>
    <td>Arroz</td>
    <td rowspan="4">3 linhas</td>
  </tr>
  <tr>
    <td>Laranja</td>
    <td>Beterraba</td>
    <td>Trigo</td>
  </tr>
  <tr>
    <td colspan="3">3 colunas</td>
  </tr>
</table>
```

Frutas	Verduras	Grãos	...
Maçã	Alface	Arroz	3 linhas
Laranja	Beterraba	Trigo	
3 colunas			

## 2.6. Formulários

Por meio de formulários Web é possível o envio de dados de uma página para o servidor. A partir deste recurso que permite que o usuário interaja com a página, seja em um formulário de contato, em um campo de busca ou ao fechar uma compra em um e-commerce.

Com o uso do formulário podemos enviar dados a um servidor, ou mesmo validar informações antes que essas sejam enviadas, por exemplo: se o endereço de e-mail é um endereço válido.

Para inserção de um formulário em um documento HTML utiliza-se a tag `<form>` e internamente no formulário podemos ter diversos elementos de envio de dados, por meio da tag `<input>`. Os elementos de entrada de dados pode ser um campo de texto, caixas de seleção, seleção de informação em radio button, botões, entre outros dados.

### 2.6.1. Interação com servidor

Ao enviar dados a um servidor web, por meio de um formulário, esse retorna apenas conteúdo estático. Exemplo: um website externo ou um arquivo, referenciando uma página web. Isso é possível por meio do atributo *action*. Contudo, muitas vezes precisamos de uma resposta dinâmica do servidor, como: validar informações, obter informações de um banco de dados ou mesmo modificar em tempo de execução páginas web. Para isso, é necessária a intervenção por meio de uma linguagem de programação.

#### Método de envio das informações:

O atributo “*method*” da tag *form* é utilizado para informar o método de envio das informações para o servidor, as quais podem ser *get* ou *post*.

- **GET:** as informações são codificadas na URL
  - Exemplo:  
**`http://www.ifsul.edu.br/contato.php?nome=maria&senha=12345`**
- **POST:** as informações são enviadas dentro de uma requisição HTTP, ou seja, não é visível ao usuário pela URL.
  - Exemplo:  
**`http://www.ifsul.edu.br/contato.php`**



### 2.6.2. Atributos da tag *input*

O atributo *type* delimita o tipo de informação do elemento, dentre os quais os mais utilizados encontram-se listados abaixo.

Valor do atributo <i>type</i>	Tipo de dados
text	Campo de texto
Password	Envio de dados sigilosos (os valores aparecem com um caractere, ocultando o valor informado)
checkbox	Seleção de mais de uma opção entre um conjunto de opções
radio	Escolha de um valor entre um conjunto de opções.
Submit	Botão para envio de dados
Reset	Botão para resetar as informações
Hidden	Informação oculta ao usuário para envio ao servidor

Atributo *name* é utilizado para nomear elementos. Atributo *id* é usado para dar informar um identificador único para um determinado elemento. O *id* deve ser único, permitindo obter as informações dos valores informados pelos usuários.

Atributo *maxlength* é usado para especificar a quantidade máxima de caracteres que o elemento permitirá que o usuário informe. Assim, podemos usar também o atributo *size*, o qual informa a largura do elemento, porém o campo poderá aceitar mais caracteres.

Ainda, podemos usar o atributo *readonly*, o qual permite modificar o elemento para somente leitura, ou seja, um campo visível em que o usuário não poderá manipulá-lo.

O atributo *value* refere-se ao valor do atributo, o qual pode ser informado um valor inicial para o elemento, ou ainda pode ser utilizado para ter acesso ao valor informado pelo usuário.

Ainda, para informar um texto ao lado do elemento utiliza-se o elemento `<label>`, o qual direciona o foco para o elemento quando é clicado.

### Sintaxe:

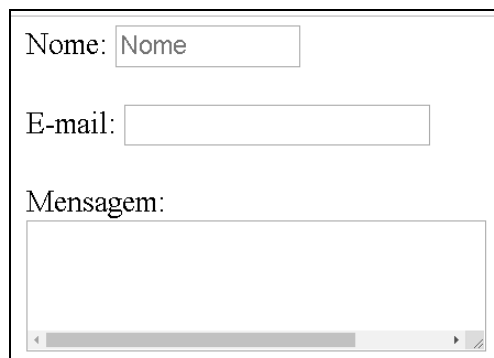
```
<head>
  <title> Tabelas </title>
</head>
<body>
<form>
  <p>
    <label>Nome
    <input name="nome" type="text" />
    </label>
  </p>
</form>
</body>
</html>
```

### 2.6.3. Outros campos válidos para um formulário:

#### Área de Texto

Utilizado para permitir que o usuário envie uma mensagem, em que um simples campo de texto pode ser insuficiente, não permitindo ao usuário ter acesso, visualização e confiança em seu envio. Esse tipo de campo é facilmente encontrado em formulários de contato.

Figura 13 - Área de Texto em um formulário de contato



Nome:

E-mail:

Mensagem:

### Sintaxe:

```
<textarea rows="x" cols="y">
</textarea>
```

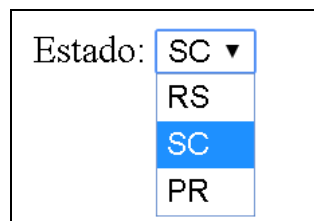
A tag para área de texto não é autocontida, assim, é necessário abrir e fechar a tag. Esse campo é criado informando seu número de linhas e colunas. Como exemplo é possível verificar o código usado para gerar a área de texto acima:

```
<textarea name="mensagem" cols="30" rows="4">  
</textarea>
```

## Caixa de Seleção

Utilizado para selecionar um elemento em uma lista de elementos pré-definidos. Como é o caso de um formulário de cadastro em que o usuário necessita selecionar o estado em que reside, dentre as UFs do sul do Brasil, para registro do seu endereço atual.

Figura 14 - Caixa de seleção do estado em que o usuário reside



Estado:	SC ▼
	RS
	SC
	PR

## Sintaxe:

```
<select>  
  <option>Opção 1</option>  
  <option>Opção 2</option>  
</select>
```

Para gerar a visão da Figura 14 foi utilizado o código a seguir.

Quadro 2- Código HTML referente à Figura 14.

```
<select name="estado" id="iEstado">
  <option>RS</option>
  <option selected="selected">SC</option>
  <option>PR</option>
</select>
```

O atributo *selected*, na segunda linha de opção “<option selected="selected">SC</option>” informa que esta opção será a default, ou seja, estará selecionada ao carregar a página, podendo o usuário alterar essa opção.

#### 2.6.4. HTML5 em formulários

HTML 5 é a versão atual do HTML e uma de suas muitas vantagens é auxiliar na validação de campos em um formulário Web.

### Sintaxe HTML5

A estrutura básica de um documento HTML5 mantém-se o mesmo que a versão anterior (HTML4), incluindo somente uma anotação na primeira linha do código, conforme Quadro 3.

Quadro 3- Sintaxe HTML5

```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <META charset="UTF-8" />
    <TITLE> </TITLE>
  </HEAD>
  <BODY> Teste HTML5 </BODY>
</HTML>
```

A instrução “<!DOCTYPE HTML>” deve ser a primeira linha do código, e trata-se de uma anotação, ou instrução, que indica ao navegador qual a versão de marcação o código foi escrito.

## O que o HTML5 trouxe de novo em formulários?

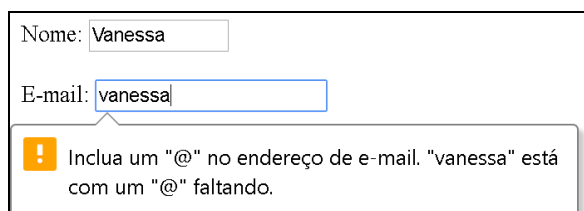
Possibilidade de validação sem o uso explícito de scripts. Isso é possível com o uso de novos valores para o atributo `type`, da tag `input`, tais como:

Valor do atributo <i>type</i>	Tipo de dados
email	Endereço de e-mail
number	Valor numérico, podendo ser especificado restrições e limites de valores aceitos.
color	seleção e envio de uma cor específica
date	Envio de uma data específica
range	Seleção de um valor entre um limite pré-estabelecido, em que o valor exato não é importante.
url	Um endereço de URL

Obs: Existem outros valores novos para o atributo `type`, da tag `input`, porém, alguns ainda não possuem suporte dos browsers, nesse caso esses irão se comporta como um campo de texto comum (`<input type="text"/>`).

A validação dos campos ocorre permitindo somente valores válidos para cada um dos campos, a exemplo pode-se verificar na Figura 15, em que ao submeter um formulário com um e-mail inválido o próprio HTML realiza a validação e impede o envio sem o devido ajuste no campo.

Figura 15 - Exemplo de Validação HTML5 para e-mail



Nome:

E-mail:

! Inclua um "@" no endereço de e-mail. "vanessa" está com um "@" faltando.

Além de novos valores para o atributo `type`, o HTML5 trouxe, como atributos, algumas restrições a tag `input`:

Atributos da tag <i>input</i>	Descrição
max	Especifica o valor máximo para um valor de entrada. Utilizado nos campos numéricos ( <i>number</i> e <i>range</i> ).
min	Especifica o valor mínimo para um valor de entrada. Utilizado nos campos numéricos ( <i>number</i> e <i>range</i> ).
pattern	Especifica uma expressão regular para checar o valor de entrada de um campo.
required	Especifica que o campo possui preenchimento obrigatório.
step	Especifica o intervalo de valor numérico aceitável.
placeholder	Breve dica referente ao valor esperado de um campo de entrada e é mostrada antes do usuário inserir um valor no campo. Utilizado nos seguintes tipos de entrada: <i>text</i> , <i>search</i> , <i>url</i> , <i>tel</i> , <i>email</i> e <i>password</i> .

## **Referências**

TANENBAUM, Andrew S.; WETHERALL, David. Redes de computadores. 5. ed. São Paulo: Pearson Prentice Hall, 2011.

GETTING STARTED. Um guia para iniciantes na área da Web. Disponível em: <<http://tableless.github.io/iniciantes/manual/obasico/o-que-front-back.html>>. Acesso em out. 2018.

PEÇANHA, Vitor. O que é Marketing Digital? Entenda o conceito e aprenda agora mesmo como fazer! Disponível em: <<https://marketingdeconteudo.com/marketing-digital/>>. Acesso em out. 2018.