



## **Apostila**

### **Desenvolvimento de Aplicações com HTML**

Este documento é propriedade intelectual do Núcleo de Educação a distância da NRsystem e distribuído sob os seguintes termos:

1. As apostilas publicadas pelo do Núcleo de Educação a distância da NRsystem podem ser reproduzidas e distribuídas no todo ou em parte, em qualquer meio físico ou eletrônico, desde que os termos desta licença sejam obedecidos, e que esta licença ou referência a ela seja exibida na reprodução.
2. Qualquer publicação na forma impressa deve obrigatoriamente citar, nas páginas externas, sua origem e atribuições de direito autoral (o Núcleo de Educação a distância da NRsystem e seu(s) autor(es)).
3. Todas as traduções e trabalhos derivados ou agregados incorporando qualquer informação contida neste documento devem ser regidas por estas mesmas normas de distribuição e direitos autorais. Ou seja, não é permitido produzir um trabalho derivado desta obra e impor restrições à sua distribuição. O Núcleo de Educação a distância da NRsystem deve obrigatoriamente ser notificado ([nrsystem@nrsystem.br](mailto:nrsystem@nrsystem.br)) de tais trabalhos com vista ao aperfeiçoamento e incorporação de melhorias aos originais.

Adicionalmente, devem ser observadas as seguintes restrições:

- A versão modificada deve ser identificada como tal;
- O responsável pelas modificações deve ser identificado e as modificações datadas;
- Reconhecimento da fonte original do documento;
- A localização do documento original deve ser citada;
- Versões modificadas não contam com o endosso dos autores originais a menos que autorização para tal seja fornecida por escrito.

A licença de uso e redistribuição deste material é oferecida sem nenhuma garantia de qualquer tipo, expressa ou implícita, quanto a sua adequação a qualquer finalidade. O Núcleo de Educação a distância da NRsystem não assume qualquer responsabilidade sobre o uso das informações contidas neste material.



## Sumário

INTRODUÇÃO .....	3
CAPÍTULO 1 – ENTENDO A INFRAESTRUTURA WEB.....	3
1.1 Domínios de IP .....	4
1.2 DNS - Sistema de Nomes de Domínios .....	4
1.3 Homepage.....	4
CAPÍTULO 2 – HTML E OS DOCUMENTOS DE HIPERTEXTO.....	5
2.1 Características da Linguagem HTML .....	5
2.2 Sintaxe HTML .....	6
CAPÍTULO 3 – EXEMPLOS DE UTILIZAÇÃO DE TAGS HTML.....	6
3.1 Tag título (Cabeçalho) .....	6
3.2 Tag parágrafo.....	7
3.3 Tag quebra de linha e linha na horizontal .....	7
3.4 Tag negrito.....	8
CAPÍTULO 4 – CRIANDO SUA PRIMEIRA PÁGINA WEB .....	8
4.1 Estrutura de um arquivo de hipertexto.....	10
CAPÍTULO 5 – VALIDANDO CONHECIMENTOS .....	13
CAPÍTULO 6 – FORMATAÇÃO BÁSICA.....	15
CAPÍTULO 7 – INDEXAÇÃO DOS SITES NOS MECANISMOS DE PESQUISA.....	16
CAPÍTULO 8 – COMENTÁRIOS E CARACTERES ESPECIAIS .....	17
8.1 Character Entities.....	18
CAPÍTULO 9 – INSERINDO LINKS E IMAGENS .....	18
CAPÍTULO 10 – CRIAÇÃO DE LISTAS EM HTML.....	19
CAPÍTULO 11 – CRIAÇÃO DE FORMULÁRIOS .....	19
CAPÍTULO 12 - PADRONIZAÇÃO NO DESENVOLVIMENTO DE INTERFACES.....	21
CAPÍTULO 13 – HOMOLOGAÇÃO DO SITE/APLICAÇÃO.....	21
SUGESTÕES DE LEITURA.....	22

## INTRODUÇÃO

HTML (Hypertext Markup Language) ou linguagem de marcação de hipertexto, foi inventada em 1990 por Tim Berners-Lee, trata-se de uma linguagem interpretada<sup>1</sup> no lado cliente, tendo como principal função, definir a estrutura do site/aplicação web.

Inicialmente, a linguagem HTML era usada para acesso e troca de informações e documentos de pesquisas entre cientistas de diferentes universidades. O projeto inicial tornou-se um sucesso jamais imaginado. Ao inventar a linguagem HTML, Tim Berners-Lee lançou as fundações da Internet tal como a conhecemos atualmente.

Para visualizar o código HTML de uma página use o menu “Exibir” no topo do seu navegador e escolha a opção “Código fonte”.

A linguagem HTML encontra-se na versão 5 (beta), o desenvolvimento de um projeto em HTML 5 para os navegadores atuais, deve levar em consideração que ainda existem algumas funcionalidades não suportadas, nestes casos, existem mecanismos em JavaScript que podem oferecer suporte para estas funcionalidades.

## CAPITULO 1 – ENTENDO A INFRAESTRUTURA WEB

Alguns softwares especializados são utilizados para administrar os sites e as aplicações hospedados na web. Entre os principais, destaque para o Servidor HTTP Apache (Apache Software Foundation) e o Internet Information Services (IIS) da Microsoft.

Você já deve ter percebido que na barra de navegação do seu browser<sup>2</sup> onde você digita o endereço do site (URL) normalmente irá constar HTTP (Hypertext Transfer Protocol), este fato se dá devido a mensagens de requisição e resposta trocadas entre o navegador e o Servidor Web, ao tentarmos acessar uma página web, fazemos uma solicitação ao Servidor Web onde essa página está armazenada.

---

<sup>1</sup> Linguagens de programação são frequentemente classificadas como compiladas ou interpretadas. Nas compiladas, o texto (ou código-fonte) do programa é lido por um programa chamado compilador, que cria um arquivo binário, executável diretamente pelo hardware da plataforma alvo, por outro lado, programas escritos em linguagens interpretadas não são convertidos em arquivos executáveis, no caso do HTML, o código é interpretado diretamente pelo navegador.

<sup>2</sup> Programa computacional que possibilita a interação entre usuários e documentos virtuais da Internet.

## **1.1 Domínios de IP**

Os dispositivos conectados a uma rede são identificados através de endereços formados por sequências de números. Esses endereços são chamados de endereços IP (Internet Protocol). Em uma Intranet, quem controla os endereços IP dos dispositivos conectados é a própria organização que administra essa Intranet. Por outro lado, os endereços IP dos dispositivos conectados à Internet são gerenciados pelos provedores de acesso (ISP - Internet service provider).

O acesso a uma página de um site ou de uma aplicação web depende do seu número de IP. Tomando como exemplo, atualmente, o endereço IP do Servidor Web onde o site da NRSYSTEM está hospedado é 187.45.195.19.

## **1.2 DNS - Sistema de Nomes de Domínios**

É inviável “tentar” memorizar todos os números de IP dos sites que acessamos, o gerenciamento de nomes de domínio surgiu como uma solução para este problema. O sistema de gerenciamento de nomes é realizado através do protocolo DNS (Domain Name System), ele é utilizado para “traduzir” o domínio desejado para o endereço IP correspondente. Basicamente, a tarefa dos servidores DNS é informar o endereço IP associado a um domínio, desta forma, o domínio nrsystem.com.br está associado ao número de IP 187.45.195.19, a utilização de domínios facilita uma eventual troca do endereço IP de um Servidor Web.

## **1.3 Homepage**

A homepage é a página principal de um site, tem com principal função ser a página de entrada, em sua estrutura, além de encontrarmos os links para as demais páginas do site, normalmente são disponibilizados o logotipo da empresa, cores, imagens e conteúdos, dessa forma, fornecemos os elementos necessários para identificação das temáticas e assuntos tratados.

## CAPÍTULO 2 – HTML E OS DOCUMENTOS DE HIPERTEXTO

Embora o conceito de hipertexto exija várias considerações, para não fugir do foco desta apostila, levaremos em consideração apenas o fato de que se você pretende desenvolver websites terá que conhecer HTML. Mesmo que você use um programa para criação, tal como o Dreamweaver, um conhecimento básico de HTML será necessário para tornar as coisas mais simples e para criar um website de melhor qualidade. A boa notícia é que HTML é fácil de aprender e de usar.

### 2.1 Características da Linguagem HTML

Os comandos usados em HTML são denominados tag (rótulo), servem para informar ao navegador como o documento de hipertexto deve ser apresentado. Todas as tags têm o mesmo formato: começam com um sinal de menor "<" e acabam com um sinal de maior ">".

De uma forma geral, existem dois tipos de tags:

Tags de abertura: <rótulo>

Tags de fechamento: </rótulo>

(*Normal Elements*)

A diferença entre elas é que na tag de fechamento, existe uma barra "/" antes da instrução. Tudo que estiver contido entre uma tag de abertura e uma tag de fechamento será processado segundo o comando contido na mesma, mas, como toda regra tem sua exceção, em HTML a exceção é que para algumas tags a abertura e o fechamento se dá na mesma tag. As chamadas "*tags solitárias*" (Void Elements) contêm comandos que não necessitam de um conteúdo para serem processados, isto é, são tags usadas para execução de comandos isolados, por exemplo, um pulo de linha é conseguido com a tag `<br>` (quebra de linha).

Estudar HTML consiste em aprender a utilizar diferentes tags !!!

Observação: Na versão 5 do HTML os *Void Elements* não possuem obrigatoriedade de serem fechados, sendo assim, em HTML5 as tags `br` e `hr` podem seguir a notação simples: `<br>` e `<hr>`, por outro lado, em XHTML, a notação adotada é a forma simplificada `<br />` e `<hr />` que substitui `<br></br>` e `<hr></hr>`.

## 2.2 Sintaxe HTML

Um website é constituído por um ou mais documentos HTML. Quando “navegamos” na Web, estamos abrindo diferentes documentos HTML, sendo assim, as tags HTML escritas por você devem estar contidas em um documento HTML.

Embora HTML não seja Case Sensitive<sup>3</sup>, de acordo com os padrões W3C<sup>4</sup>, a maneira correta de declarar a sintaxe<sup>5</sup> HTML é usando letras minúsculas (verificar exceções como a tag `DOCTYPE`, por exemplo).

## CAPÍTULO 3 – EXEMPLOS DE UTILIZAÇÃO DE TAGS HTML

A lista completa das tags html pode ser acessada neste link <http://www.w3schools.com/tags/>, apresentarei a seguir, exemplos de utilização de algumas dessas tags.

### 3.1 Tag título (Cabeçalho)

Os cabeçalhos servem para estruturar os conteúdos na página de forma semântica<sup>6</sup> e são designados pelas tags `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` e `<h6>`. A tag `<h1>` é o cabeçalho de primeiro nível e apresenta o maior tamanho de texto, `<h2>` o cabeçalho de segundo nível e aquele apresentado com tamanho de texto um pouco menor e `<h6>` o cabeçalho de sexto nível e apresenta o menor tamanho de texto.

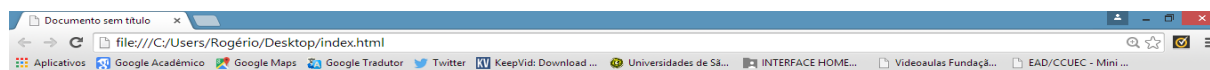
**Exemplo:**

`<h1>`Este é um título de primeiro nível`</h1>`

`<h2>`Este é um título de segundo nível`</h2>`

`<h3>`Este é um título de sexto nível`</h3>`

Vai ser exibido no navegador, como mostrado a seguir:



**Este é um título de primeiro nível**

**Este é um título de segundo nível**

**Este é um título de terceiro nível**

<sup>3</sup> HTML não faz diferenciação entre caracteres maiúsculos e minúsculos

<sup>4</sup> W3C - World Wide Web Consortium: Consórcio internacional responsável pela padronização web

<sup>5</sup> Sintaxe - Maneira como declaramos os comandos

<sup>6</sup> Semântica - dá sentido ao texto

### 3.2 Tag parágrafo

A tag `<p>` é utilizada para delimitar parágrafos.

O HTML adiciona automaticamente uma linha em branco extra antes e após um parágrafo.

Exemplo:

`<p>` Este é um exemplo de parágrafo de texto `</p>`

`<p>` Este é outro parágrafo `</p>`

### 3.3 Tag quebra de linha e linha na horizontal

A tag `<br>` é usada para terminar uma linha sem a necessidade de se começar um novo parágrafo. A tag `<br>` força a quebra de linha onde quer que você a coloque.

Exemplo:

`<p>` Este texto será exibida em uma linha. `<br>` Este outro texto será exibido em outra linha. `<p>`

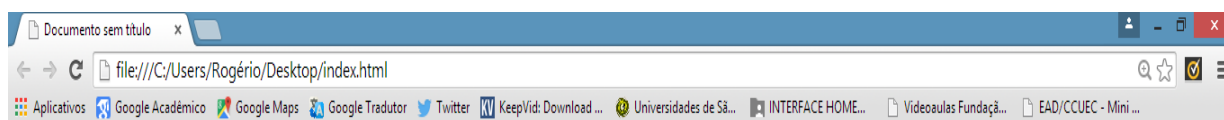
Se por um lado a tag `<br>` “pula uma linha” sem inserir nenhum elemento visual, por outro lado, a tag `<hr>` funciona como um elemento de bloco, ou seja, haverá uma quebra de linha acima e abaixo de uma linha horizontal.

Exemplo:

`<p>` Este texto é dividido por uma linha horizontal `<hr>`

Esta é a segunda parte do texto. `<p>`

Neste caso, o conteúdo será exibido conforme exibido a seguir.



Este texto é dividido por uma linha horizontal

Esta é a segunda parte do texto.

### 3.4 Tag negrito

As tags `<b>` ou `<strong>` são utilizadas na formatação para aplicação de negrito em textos.

#### Exemplo 1:

`<b>`Este texto deve ser em negrito.`</b>`

O mesmo efeito é conseguido através da tag `<strong>`. O uso da tag `<strong>` é mais indicado devido a questões de acessibilidade, em leitores de tela utilizados por pessoas com deficiência visual, o conteúdo desta tag pode ser identificado.

#### Exemplo 2:

`<strong>`Este texto deve ser em negrito.`</strong>`

Para efeito visual, em ambos os casos, `<b>` ou `<strong>` o texto será apresentado no navegador da mesma forma.

## CAPÍTULO 4 – CRIANDO SUA PRIMEIRA PÁGINA WEB

Com o que você aprendeu anteriormente está a alguns minutos de criar sua primeira página web.

A princípio, precisamos apenas de um navegador e o Notepad (Bloco de notas ou um editor de texto similar).

Abra o Notepad (Iniciar » Programas » Acessórios):

Agora, estamos prontos para começar!

Observação: Evite usar o Wordpad ou o Microsoft Word para digitar seu código, pois eles podem inserir formatações indesejadas.

Primeiramente, devemos “declarar” ao navegador que tipo de documento estamos utilizando. Isto é feito com a tag `<!DOCTYPE html>` que deve ser escrita na primeira linha do documento.

Para termos certeza que não iremos esquecer-nos de fechar a tag HTML (faça isso agora mesmo), digite `</html >` localizada a algumas linhas abaixo, assim você irá escrever seu documento entre as tags digitadas:

`<!DOCTYPE html>`

`</html>`

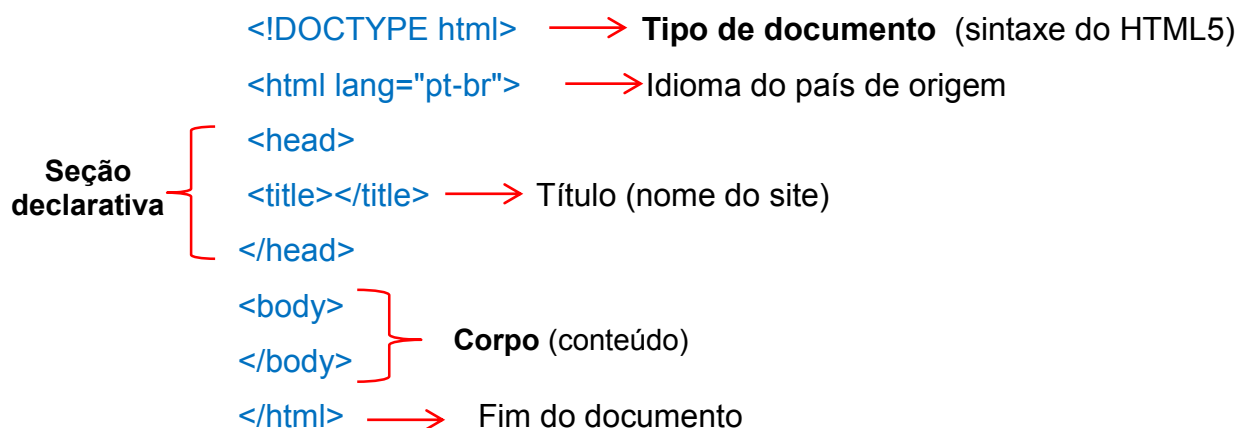


Seu código deverá ser  
inserido entre essa duas tags



Após declararmos o tipo de documento, devemos iniciar o "head" (seção declarativa) que é utilizada para fornecer informações adicionais sobre o documento, como por exemplo, o título do site `<title>`, isto facilita a localização do documento pelos mecanismos de busca. Na sequência criamos a seção `<body>` "corpo"; o conteúdo do documento deve estar contido nesta seção.

Abaixo podemos constatar o *template*<sup>7</sup> básico e sua estruturação em seções:



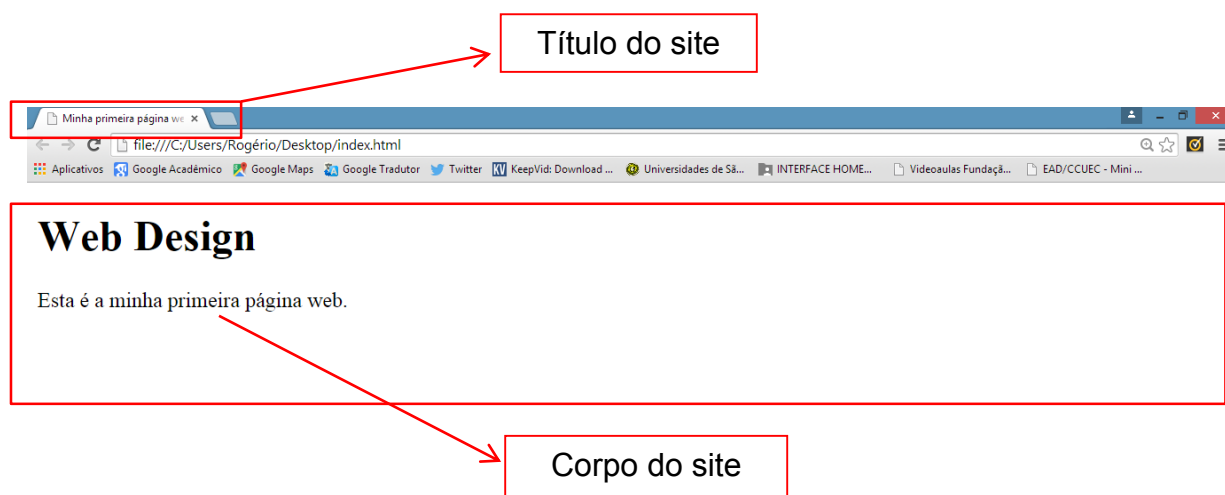
Note como o código acima foi estruturado em linhas diferentes, a princípio é indiferente a formatação referente aos recuos e espaçamentos do seu código, mas é altamente recomendado que você adote uma estrutura clara e nítida para seu HTML. O uso de linhas, como mostrado no exemplo acima, facilita a manutenção do código.

Se o seu documento está como o exemplo exibido na página anterior, você construiu sua primeira página web - uma página particularmente chata e provavelmente nada parecido com o que você sonhou fazer quando começou a ler esta apostila, mas de qualquer forma, um tipo de website. Isto que você fez é um *template* básico que poderá ser utilizado para seus futuros documentos HTML.

Agora que já conhecemos a estrutura básica de um documento de hipertexto, vamos começar com algo simples. Que tal uma página que exiba o nome do site, um título de primeiro nível e um parágrafo de texto? Na figura a seguir é apresentado um modelo.

---

<sup>7</sup> Template - Documento sem conteúdo, possui apenas a estrutura que será utilizada para a apresentação



Você verá que “programar em HTML” é uma tarefa simples de se fazer, a linguagem HTML é lógica e muito intuitiva. Primeiramente você deve levar em consideração que o navegador interpreta os códigos HTML de forma idêntica como você lê um texto qualquer, de cima para baixo e da esquerda para a direita, desta forma um documento HTML deve iniciar com aquilo que deve ser a primeira coisa a aparecer na página e terminar com a última coisa a ser exibida.

#### 4.1 Estrutura de um arquivo de hipertexto

No Capítulo 4, aprendemos que a estrutura básica de um documento HTML é composta por três partes:

**DOCTYPE:** primeira linha do documento - informa ao navegador à linguagem que está sendo utilizada (tipo de documento);

**Seção declarativa:** head - contém informações sobre a página;

**Corpo do site:** body - conteúdo que constitui a página.

A delimitação de um documento de hipertexto é utilizada para definirmos onde serão inseridas as informações sobre o site, isso se faz necessário, pois precisamos definir o que é importante para os mecanismos de busca ou para os internautas e vice-versa, por exemplo, enquanto o título (nome do site) deve ser inserido na seção declarativa facilitando sua busca em mecanismos de busca como o Google, por exemplo. O conteúdo da página deve ser inserido no corpo `<body>`.

O código a seguir exemplifica a utilização da seção “head”.

```
<head>  
<title>Minha primeira página web</title>  
</head>
```

Note que o título não aparece na página propriamente dita. Tudo que você quer que apareça na página é conteúdo, e com tal, deverá ser colocado entre as tags `<body>` e `</body>`.

Conforme combinamos no início, queremos uma página que possua sua identificação, um título de primeiro nível e um parágrafo de texto.

Como texto a ser disponibilizado, foi sugerido algo do tipo:

**Nome do site:** Minha primeira página web

**Título:** Web Design

**Parágrafo:** Esta é a minha primeira página web.

A partir dessas informações iniciais, já sabemos o que deve ser inserido no body.

Seguindo o enunciado, digite os seguintes comandos na seção body:

```
<h1> Web Design </h1>  
<p> Esta é a minha primeira página web.</p>
```

- A letra h na tag `<h1>` vem de "heading" (cabeçalho), neste caso, h1 é um cabeçalho de 1º nível.

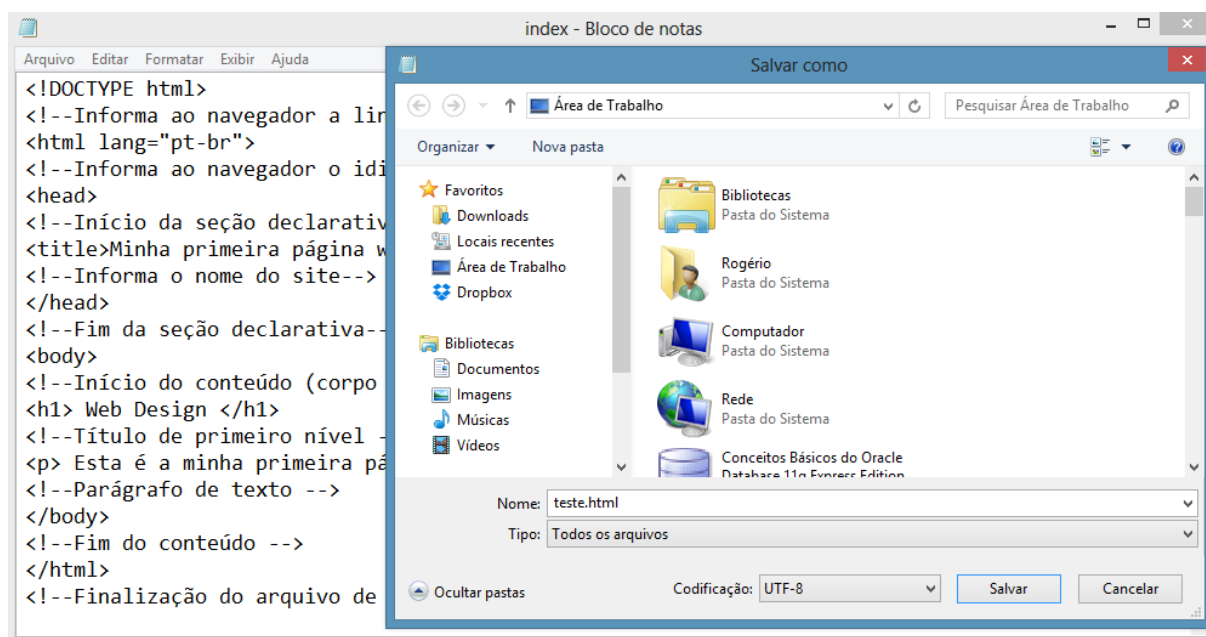
- A letra p na tag `<p>` é a abreviatura para "paragraph" (parágrafo) que é exatamente o que o texto é - um parágrafo de texto.

Na sequência será apresentado o *template* básico acrescido das tags necessárias para disponibilização dos conteúdos propostos. Abaixo de cada tag é feito um comentário explicativo. No capítulo 8 são disponibilizadas maiores informações sobre a utilização de comentários.

## Estrutura com as tags utilizadas até o momento:

```
<!DOCTYPE html>
<!--Informa ao navegador a linguagem e versão utilizada-->
<html lang="pt-br">
<!--Informa ao navegador o idioma padrão (português do Brasil)-->
<head>
<!--Início da seção declarativa -->
<title>Minha primeira página web</title>
<!--Informa o nome do site-->
</head>
<!--Fim da seção declarativa-->
<body>
<!--Início do conteúdo (corpo do site) -->
<h1> Web Design </h1>
<!--Título de primeiro nível -->
<p> Esta é a minha primeira página web.</p>
<!--Parágrafo de texto -->
</body>
<!--Fim do conteúdo -->
</html>
<!--Finalização do arquivo de hipertexto -->
```

Se seu documento está como exibido acima, basta que você salve seu trabalho e depois visualize no navegador:



**Salvando o arquivo:** No Bloco de Notas vá ao menu "Arquivo" no topo da janela e escolha a opção "Salvar como...". No box "Salvar como tipo" Escolha "Todos os arquivos". Isto é muito importante caso você não faça isto, o arquivo poderá ser salvo como texto e não como documento HTML.

Salve seu documento com o nome "teste.html" (a extensão ".html" indica que se trata de um documento HTML. Você pode salvar o documento onde quiser no seu computador, esteja certo de salvar em um lugar que depois você possa achar com facilidade.

**Abrindo o arquivo:** Localize o seu documento HTML, clique com o botão direito e escolha a opção "Abrir com...".

Escolha o navegador disponível (Internet Explorer, Google Chrome, etc.).

Pronto, seu primeiro site está concluído!

## CAPÍTULO 5 – VALIDANDO CONHECIMENTOS

Até o momento, foram apresentadas as principais tags (comandos) e sua utilização, aprendemos também que o *template* ou estrutura básica é composto por três seções, sendo que o conteúdo da página deve estar contido na seção "body".

No exemplo a seguir, veremos que o título do site da NRSYSTEM é exibido no topo superior esquerdo do navegador Chrome (barra de título). O título é muito importante pois além de "nomear" a aplicação, é usado pelos mecanismos de pesquisa (Google, Yahoo, etc.) para indexar conteúdo\*.

**Observação:** A indexação não depende apenas do título, ver também metadados (Cap. 7).

No exemplo do site da NRSYSTEM vemos como a segmentação do documento em três seções pode influenciar sua renderização<sup>8</sup> pelo navegador. Vale ressaltar que o processo de renderização não é padronizado e pode mudar de navegador para navegador.

---

<sup>8</sup> Renderização - Processo pelo qual se pode obter o produto final de um processamento digital qualquer.

O nome do site (Título) fica visível na Barra de títulos do navegador



### Exercício:

Tente criar algumas páginas. Por exemplo, construa uma página com um título, um cabeçalho, algum texto (parágrafo), um subtítulo e mais algum texto (parágrafo), aplique uma formatação básica destacando palavras através de negrito ou itálico. No tópico a seguir alguns itens de formatação são abordados.

Dica: para ganhar tempo, comece sempre com o *template* básico, independente do tamanho ou tipo do site, a estrutura segmentada em três partes se mantém sempre a mesma.

Mãos à obra!

## CAPÍTULO 6 – FORMATAÇÃO BÁSICA

A partir de agora, aprenderemos como formatar textos, inserir imagens e links; por enquanto, vamos fazer uma personalização básica, nos aprofundaremos no tema com a [apostila de CSS](#). O HTML define uma série de elementos para formatar a saída, como texto em negrito ou itálico.

Você já sabe que pode obter negrito com a tag `<b>` ou `<strong>`, agora saiba que pode obter itálico com a tag `<em>`. Há algum tempo atrás, utilizávamos a tag `<i>` para itálico “i” vêm de “italic” enquanto “em” tem o sentido de ênfase, embora os navegadores ainda deem suporte, a tag `<i>`, assim como a tag `<b>`, ambas foram declaradas obsoletas pela W3C.

### Exemplo 1:

`<i>`Este texto deve ser itálico.`</i>`

### Exemplo 2:

`<em>`Este texto deve ser itálico.`</em>`

Para efeito visual, em ambos os casos, o resultado será o mesmo.

A tabela a seguir apresenta as **tags para formatação de texto**:

Tag	Descrição
<code>&lt;b&gt;</code>	Define texto em negrito
<code>&lt;big&gt;</code>	Define texto grande
<code>&lt;em&gt;</code>	Define texto enfatizado;
<code>&lt;i&gt;</code>	Define texto em itálico
<code>&lt;small&gt;</code>	Define texto pequeno
<code>&lt;strong&gt;</code>	Define texto forte
<code>&lt;sub&gt;</code>	Define texto subescrito
<code>&lt;sup&gt;</code>	Define texto superescrito
<code>&lt;ins&gt;</code>	Define texto inserido
<code>&lt;del&gt;</code>	Define texto cancelado

**Importante:**

**1º** Não esqueça que sempre que for salvar um documento de hipertexto, na frente do nome do mesmo (letras minúsculas, sem acentos, espaços ou caracteres especiais) deve constar a extensão (.html). **Exemplo:** `servicos.html`

**2º** Por uma questão de convenção, devemos levar em consideração que a primeira página de um site deve chamar-se de index, então seria legal já ir se acostumando com essa regra.

## **CAPÍTULO 7 – INDEXAÇÃO DOS SITES NOS MECANISMOS DE PESQUISA**

Hoje em dia, a principal forma de encontrar um site é utilizar alguma ferramenta de busca, como o Google, por exemplo. As ferramentas de pesquisa analisam previamente os sites.

A técnica de SEO (Search Engine Optimization) é utilizada para potencializar a indexação dos sites, isto permite respostas mais rápidas às consultas realizadas. A tag `<title>` não é suficiente para que o site seja indexado nos mecanismos de pesquisa, devemos adotar estratégias para otimizar a indexação. Uma das maneiras normalmente adotadas para aumentar a eficiência nas buscas orgânicas é inserir metadados<sup>9</sup> na seção “head”, a seguir, alguns exemplos de metadados:

**Palavras-chaves:** são utilizadas para informar de forma mais precisa possível o assunto principal do site.

Em um site de faculdade/universidade, por exemplo, as palavras chaves podem conter o nome da instituição, tipos de cursos ofertados, slogan, etc. A seguir, exemplo de palavras chaves da empresa NRSYSTEM:

```
<meta name="keywords" content=" nrsystem, tecnologia da informação, soluções integradas, consultoria, treinamento, cursos, ead, desenvolvimento web, Santo André, know-how, maturidade, gestão de projetos, multimídia ">
```

---

<sup>9</sup> Metadados - informações adicionais sobre a página “dados sobre dados”.



**Descrição:** resumo do assunto da página. Texto utilizado para descrever a empresa/site. Exemplo:

```
<meta name="description" content=" Empresa especializada em multimidia e EAD, há mais de 5 anos provendo soluções integradas em Tecnologia da Informação. Nosso know-how permite oferecer a melhor relação custo benefício ">
```

Para maiores informações sobre Otimização nos mecanismos de pesquisa, acesse [Técnicas de SEO](#)

### **QUIZ - Teste seus conhecimentos**

A forma de exibição das páginas web é padrão em todos os navegadores?

Sites só podem ser implantados na Internet?

Domínios são associados a endereços IP?

Técnicas de SEO melhoram a velocidade de carregamento de um site?

Técnicas de SEO melhoram o posicionamento dos sites nas ferramentas de pesquisa?

Em quantas seções um website é dividido?

### **CAPÍTULO 8 – COMENTÁRIOS E CARACTERES ESPECIAIS**

Podemos adicionar comentários em um documento HTML dentro das tags `<!--` e `-->`. Os comentários são úteis para “entender” como o site foi desenvolvido. Comentários podem descrever a função de cada tag, demarcar início e fim de cada bloco de informação, etc.

Os comentários são ignorados pelos navegadores não tendo nenhum impacto na apresentação visual (renderização).

#### **Exemplo de comentário em HTML:**

```
<html lang= "pt-br">
```

```
<!-- Idioma do país de origem (pt-br português do Brasil) -->
```

## 8.1 Character Entities

Para inserir determinados caracteres em um documento HTML, devemos utilizar o código do caractere desejado. A utilização desses códigos evita problemas de *encoding* (codificação) na exibição das páginas web. Você pode conhecer esses caracteres e os seus respectivos códigos através do seguinte endereço <http://dev.w3.org/html5/html-author/charref>. Esses caracteres são conhecidos como *Character Entities* ou *HTML Entities*.

### Exemplo:

É comum no rodapé dos sites, disponibilizarmos os direitos autorais do proprietário ou desenvolvedor, nestes casos, o direito autoral é simbolizado através do ©, em HTML, o código que representa este símbolo é `&copy;`, como exemplificado a seguir:

```
<p> &copy; 2013 NRSYSTEM SOLUÇÕES </p>
```

## CAPÍTULO 9 – INSERINDO LINKS E IMAGENS

Link ou hiperlink permitam a ligação entre partes de um mesmo documento, ou outros documentos de hipertexto. Os links possuem duas extremidades e uma direção, denominadas de âncoras, consistem em uma “base que podemos clicar”, como por exemplo, uma palavra, texto ou imagem. O exemplo a seguir mostra um link para se conectar ao Google.

```
<a href="http://google.com.br"> Clique aqui para acessar o Google </a>
```

A inserção de imagens em documentos de hipertexto se dá através da tag `<img>`, desta forma, podemos também utilizar imagens no conteúdo do site, inclusive, no formato de link. O exemplo a seguir descreve um link para o Google utilizando uma imagem.

```
<a href="http://google.com.br">  </a>
```

Observação: vale ressaltar que no caso de utilização de imagens, o complemento `src` (source) da tag `img` indica a origem da imagem e sua extensão, neste caso, se no exemplo anterior a imagem utilizada fosse do tipo `jpg` ao invés de `png` e não estivesse na raiz e sim em uma subpasta com o nome `imagens`, o código seria o seguinte:

```
<a href="http://google.com.br">  </a>
```

## CAPÍTULO 10 – CRIAÇÃO DE LISTAS EM HTML

Podemos criar listas ordenadas ou não ordenadas em HTML. Usamos a tag `<ol>` (ordered lists), quando queremos uma lista ordenada, dentro desta tag inserimos os itens da lista através da tag `<li>` (list item).

### Exemplo de Lista ordenada:

```
<ol>
<li> HTML </li>
<li> CSS </li>
</ol>
```

### Após renderização, será exibido:

1. HTML
2. CSS

### Exemplo de Listas não ordenada

Caso a tag `<ol>` seja substituída por `<ul>` (unordered lists), o resultado exibido será:

- HTML
- CSS

## CAPÍTULO 11 – CRIAÇÃO DE FORMULÁRIOS

Os formulários permitem a inserção de dados que poderão ser armazenados através da passagem de parâmetros, como por exemplo, uma conexão PHP/MySQL (método POST).

Um formulário pode conter campos de texto, campos de área de texto, menus *drop-down*, botões radiais, caixas de seleção, etc. A seguir, exemplo formulário simples com as principais tags e seus devidos comentários:

```
<form action="enviar.php" method="post">
```

<!--A tag form define o formulário, action é um atributo que indica o script que processará os dados do formulário e Method é um atributo que define como os dados são enviados do formulário para o programa que irá processá-los -->

```
<label>Nome: </label>    <br>
```

<!--A tag label serve para identificar um rótulo para o element input-->

```
<input type="text">    <br>
```

<!-- A tag de formulário mais usada é a tag "input". O tipo de input é especificado com o atributo type (neste caso, será exibido uma caixa texto)-->

```
<label>E-mail: </label>    <br>
```

```
<input type="email"> <br>
```

<!-- Campo que exige um E-mail com formato válido -->

```
<label>Mensagem: </label> <br>
```

```
<textarea cols="16" rows="4" name="mensagem"></textarea>    <br><br>
```

<!-- Área de texto. O atributo "cols" define a quantidade de colunas e "rows" a quantidade de linhas que serão ocupadas-->

```
<input type="submit">
```

<!-- Submit – submeter à apreciação "enviar", o atributo submit é utilizado para gerar um botão do tipo enviar -->

Nome:

E-mail:

Mensagem:

Enviar

Visualização do formulário renderizado

## **CAPÍTULO 12 - PADRONIZAÇÃO NO DESENVOLVIMENTO DE INTERFACES**

Durante a evolução web, houve mudanças no perfil dos usuários e dispositivos a ela conectados. Em um projeto de aplicação para web devemos oferecer alternativas de acesso ao conteúdo pelo maior número possível de pessoas, independente de limitação física ou tecnológica.

Os conceitos de interoperabilidade<sup>10</sup>, acessibilidade<sup>11</sup> e usabilidade<sup>12</sup> devem ser adotados no desenvolvimento de interfaces para internet visando atender:

1. Os diversos dispositivos conectados na web e suas diferentes resoluções;
2. Possibilitar acesso ao conteúdo por pessoas com necessidades específicas;
3. Reduzir o tempo para acessar uma informação.

Maiores informações sobre interoperabilidade na web podem ser obtidas em:

<http://www.w3c.br/Noticias/W3cConfirmaLastCallDoHtml5ParaMaioDe2011EPadiaoCompletoPara2014>

## **CAPÍTULO 13 – HOMOLOGAÇÃO DO SITE/APLICAÇÃO**

Antes de publicarmos o site ou aplicação no ambiente de produção, é importante certificarmos-nos de que tudo esteja funcionando corretamente de acordo com parâmetros pré-definidos.

A realização de testes pelo próprio desenvolvedor é o primeiro passo para a homologação de uma aplicação web. Existem diversas ferramentas online para validação de código e testes de acessibilidade, destaque para [Markup Validation Service da W3C](#) e [AccessMonitor](#).

---

<sup>10</sup> Interoperabilidade - Capacidade de uma aplicação funcionar em plataformas diferentes, independente de hardware ou software.

<sup>11</sup> Acessibilidade - Consiste em oferecer alternativas que facilitem o acesso ao conteúdo independente do dispositivo ou das necessidades especiais do usuário.

<sup>12</sup> Usabilidade - Qualidade de experiência do usuário ao interagir com um sistema (fácil de usar, fácil de lembrar, maximiza a produtividade e minimiza a quantidade de erros).

## **SUGESTÕES DE LEITURA**

COSTA, R.F. Fundamentos de Design de Interfaces para Internet. Disponível em: [http://nrssystem.com.br/Design\\_Interfaces\\_Internet.pdf](http://nrssystem.com.br/Design_Interfaces_Internet.pdf). Acessado em: 13 de dezembro de 2015.

CSS - Introdução à folha de estilos. Disponível em: [http://nrssystem.com.br/Introducao\\_CSS.pdf](http://nrssystem.com.br/Introducao_CSS.pdf). Acessado em: 13 de dezembro de 2015.