

DESENVOLVIMENTO DE UM TUTORIAL HIPERTEXTO EM HTML¹

PAULO CÉSAR DA COSTA PINHEIRO

Departamento de Engenharia Mecânica da UFMG
Av. Antônio Carlos 6627; 31270-901 Belo Horizonte, MG
Fone: (31) 3499-5140; Fax: (31) 3443-3783

e-mail: pinheiro@dedalus.lcc.ufmg.br pinheiro@demec.ufmg.br

RESUMO

O hipertexto é a forma mais comum de multimídia, e tem sido amplamente utilizado na construção de tutoriais para o ensino de engenharia auxiliado por computador (CAE). O hipertexto cria uma árvore de conhecimento, possibilitando que o usuário navegue dentro de um conjunto de informações, de modo a encontrar as informações que deseja. Uma ferramenta muito interessante de se criar documentos hipertextos é a linguagem HTML (Hyper Text Markup Language), utilizada para desenvolver home-pages na Internet.

O HTML é uma linguagem interpretada de descrição de páginas que possui uma série completa de padrões para estruturação de dados em multimídia e hipertexto, incluindo gráficos, navegação hipertexto, som, vídeo etc. A principal característica do HTML é sua simplicidade e facilidade de implementação, o que ajudou muito na popularização das páginas WWW.

Os tutoriais desenvolvidos em HTML podem ser manipulados com qualquer *browser* para a Internet, funcionando off-line. A grande vantagem de desenvolver tutoriais em HTML é que podem ser utilizados em qualquer computador e sistema operacional (multiplataforma), e podem ser distribuídos diretamente pela Internet. Devido a estas características, o HTML deve ser tornar o padrão mundial de multimídia.

Este trabalho apresenta a metodologia do desenvolvimento de um tutorial hipertexto para o ensino de engenharia, usando como ferramenta o HTML. A implementação de um tutorial em HTML é de fácil, podendo-se, assim, desenvolver tutoriais de alto nível para ensino de engenharia.

INTRODUÇÃO

A multimídia tornou-se popular a partir de 1987 graças ao Macintosh e ao Software HyperCard. A multimídia combina uma interface amigável de hipermídia, com a habilidade de controlar periféricos, aumentando a interação usuário/software. Esta fusão foi uma dos maiores avanços em educação desde a introdução dos computadores. Pelo uso de lições, páginas e alvos (*targets, stack, cards* ou botões), os educadores podem criar e usar cursos que são facilmente assimilados pelos estudantes.

O hipertexto é a forma mais comum de multimídia, e tem sido amplamente utilizado na construção de tutoriais para o ensino de engenharia auxiliado por computador (CAE). O hipertexto cria

¹PINHEIRO Paulo César da Costa. Desenvolvimento de um Tutorial Hipertexto em HTML. In: XXV CONGRESSO BRASILEIRO DE ENSINO DE ENGENHARIA (COBENGE-97), 12-15 Outubro 1997, Salvador, BA, Anais... Salvador: ABENGE, Associação Brasileira de Ensino de Engenharia, 1997, v.3/4, p.1171-1184.

uma árvore de conhecimento, possibilitando a navegação amigável dentro de um banco de dados, de modo a encontrar as informações que se deseja. Uma ferramenta muito interessante de se criar documentos hipertextos é a linguagem HTML (Hyper Text Markup Language), utilizada para desenvolver *home-pages* na Internet.

HTML é uma linguagem de *scripts* (instruções) utilizada para compartilhar informações, publicar e transferir trabalhos científicos através da Internet, de uma forma simples e transparente para o usuário, interligando documentos através de vínculos hipertexto. Ela consiste basicamente de um texto normal com *tags* (funções, atributos) delimitando porções de texto selecionadas, que ensinam ao *browser* (software gráfico para a manipulação destas informações) formatá-lo de forma particular, inserir figuras, desenhar linhas, abrir outras páginas etc.

A HTML foi criada em 1990 por Tim Berners-Lee do CERN (Centre Européen pour la Recherche Nucleaire, Suíça). A popularização do HTML aconteceu com a criação do *browser* Mosaic, em 1992, pelo NCSA (Centro Nacional de Aplicações em Supercomputação). Estava assim criada a WWW, serviço mais utilizado na Internet e, sem dúvida, o responsável pela difusão comercial da Internet. A simplicidade e facilidade de implementação das *home-pages* em HTML, ajudou muito na popularização da WWW.

VANTAGENS DA HTML

O HTML é uma linguagem interpretada de descrição de páginas que possui um conjunto de padrões para estruturação de dados em multimídia, incluindo gráficos, navegação hipertexto, som, vídeo etc. O HTML possui muitas características úteis e sua habilidade de controlar e integrar um grande número de diferentes dispositivos nos *stacks* utilizando uma linguagem de *scripts* (tal como o HyperCard) simples. O HTML é capaz de apresentar os recursos básicos dos aplicativos GUI: botões de radio, listas de opções, caixas de texto multilinhas etc. Os *browsers* são capazes de exibir tais recursos. Um formulário HTML é tanto um documento ativo quanto um aplicativo centrado em documento.

Diferentemente de outras linguagens que possuem muitos recursos, mas que são de difícil e demorado aprendizado, tutoriais complexos podem ser facilmente desenvolvidos em HTML, através de softwares de edição Wysiwyg (What you see is what you get). Como a implementação de um tutorial em HTML é fácil, pode-se assim, desenvolver tutoriais de alto nível para ensino de engenharia. Estes tutoriais podem ser manipulados com qualquer *browser* para a Internet, funcionando *off-line*. O *browser* mais utilizado no mundo (80% EUA; 89% Brasil) é o Netscape Navigator.

A grande vantagem de desenvolver tutoriais em HTML é que podem ser utilizados em qualquer computador e sistema operacional (multiplataforma), e podem ser distribuídos diretamente pela Internet. Um *browser* para PC, Unix ou Macintosh pode receber a mesma página igualmente de um servidor PC, Unix ou Macintosh, em qualquer lugar do mundo. Para visualizar um aplicativo HTML é somente necessário o *browser*, nenhum outro software. Como o aplicativo fica disponível na WWW, os usuários podem acessar a versão mais atualizada deste aplicativo. Devido a estas características, o HTML deve ser tornar o padrão mundial de multimídia.

A PROGRAMAÇÃO HTML

É muito fácil programar diretamente na linguagem HTML, a partir de qualquer editor de textos (ASCII) [1, 2, 3, 4, 5, 9, 10, 12]. Os mais de sessenta comandos (*tags*) disponíveis na HTML, podem ser classificados em 2 tipos: os que se destinam a formatar a aparência da página, com seus elementos (textos e imagens), e os responsáveis pela criação das ligações (*links*) com outras páginas da rede. A

sintaxe de uma *tag* é a seguinte: <TAG>texto</TAG>

Onde <TAG> inicia o comando, </TAG> termina o comando e "texto" é o que será exibido pelo *browser*. Os comandos podem ser escritos tanto em maiúsculo quanto em minúsculo. Alguns desses comandos são do tipo liga-desliga: a primeira ocorrência do comando ativa o seu efeito e a segunda </TAG> o fecha. Se esquecida, o comando continuará ativo. Em outros comandos, somente a primeira parte é necessária, como no caso do comando <P>, usado para terminar um parágrafo.

A estrutura mínima de um programa HTML é composta dos seguintes comandos:

Comando HTML	Operação
<html>	Inicializa o programa
<head>	Início dos comandos do título da página
<title> Texto para Título </title>	Título da página
</head>	Fim dos comandos do título da página
<body>	Início da formatação da página
{ Corpo do Programa em HTML }	
</body>	Fim da formatação da página
</html>	Finaliza o programa

Os Comandos de Estilo permitem formatar a página. Entretanto, as limitações do HTML não permitem a especificação dos tipos de letras ou estilos. Os únicos atributos de texto que se pode utilizar são tamanho da fonte, negrito, itálico etc, sendo possível sobrepor dois ou mais comandos.

 NEGRITO 	 	 1 a 7;default = 3
<I> <i>ITÁLICO</i> </I>	<i> </i>	<H1> TAMANHO MÁXIMO </H1>
<U> <u>SUBLINHADO</u> </U>		<H2> TAMANHO GRANDE </H2>
<I><U> <i><u>TUDO JUNTO</u></i> </I></U>		<H3> TAMANHO MÉDIO </H3>
 Parágrafo (sem pular linha)		<H5> TAMANHO PEQUENO </H5>
<P> Parágrafo (pula 1 linha)		<H6> TAMANHO MÍNIMO </H6>

Para criar o recurso de hipertexto (palavras-chaves ou imagens que, ao ser clicadas, abrem outros documentos ou arquivos da rede), existem dois comandos associados aos *hyperlinks*: os externos (que ligam a uma outra página), e os internos (que ligam a um determinado ponto dentro da mesma página).

O comando para *Links* externos é: Texto do link onde a expressão "endereço" é o local onde se encontra o arquivo, ou a página, a ser acessado. Esse endereço pode ser o nome do arquivo, se ele se encontra no mesmo diretório da página, ou uma URL, que é a denominação da Internet para o endereço único da página em toda a rede. No caso de uma URL, deve ser definido o protocolo HTTP. O "Texto do Link" é o hipertexto exibido pelo *link*, a ser clicado, e que aparecerá com cor diferente e sublinhado na tela, para identificá-lo como um hipertexto.

Os *Links* Internos, quando ativados, levam o *browser* a outro ponto do documento, acessando rapidamente partes de documentos extensos. Os *links* internos são muito úteis na criação de índices ou sumários. Para tanto, é necessário definir o *label* (alvo, âncora), que são marcas pelas quais o *browser* identifica a região do texto.

O comando para *Links* internos é: Texto onde "#" indica ao *browser* que aquele endereço é um *label*, e não um *link* externo. O *label* deve ser posicionado no início da seção do documento a ser exibida, para onde foi direcionado o *browser*, através do comando:

 Texto opcional

Para apresentar uma figura utiliza-se o seguinte comando:

```
<IMG ALT="texto_alternativo" SRC="nome_figura.GIF">
```

onde "texto_alternativo" é um texto alternativo para o caso da figura não ser apresentada; "nome_figura", o nome e localização (caso não esteja no mesmo diretório) da figura, e "GIF" a extensão da figura (JPG, GIF, BMP). Como a Internet Brasileira é lenta, recomenda-se declarar o tamanho de todas as imagens, de modo a evitar que a tela fique em branco por muito tempo. Assim, é mostrado o texto formatado, antes de se carregar as imagens:

```
<IMAGE SRC="nome_figura.GIF WIDTH=xxx HEIGHT=yyy >
```

Vários outros comandos que permitem a apresentação de cores, linhas, tabelas, formulários, gráficos sensíveis, applets, sons, vídeo etc, fogem do escopo deste trabalho. Entretanto, somente com esses comandos aqui apresentados, é possível desenvolver um aplicativo em HTML. O conhecimento mais profundo dos recursos do HTML podem ser obtidos em vários tutoriais disponíveis *online*, com todos os comandos da linguagem [1, 3, 5, 6, 12].

Devido a suas limitações, o HTML não permite desenvolver tutoriais como preencher o espaço em branco, múltipla escolha, falso/verdadeiro, pergunta/resposta, questões abertas e planilhas de cálculo, sendo necessário um software que processe questões aplicadas durante o tutorial. Este processamento pode ser realizado em linguagem C ou Java, utilizando-se a CGI (Common Gateway Interface). A página de questões é aberta como um FORM.

O FORM submete os dados de entrada do usuário ao programa CGI, que acessam um banco de dados com as questões, e se encarrega do processamento desses dados. Desta forma pode-se efetuar a avaliação do aprendizado do aluno. O aluno preenche um formulário em na home-page, e as respostas são submetidas a um programa que as avaliará, retornando ao aluno, a nota de sua avaliação online [7].

Apesar de ser muito fácil programar uma *home-page* diretamente em HTML, existem softwares especializados (Web Editors), que permitem a construção de *home-pages*, sem ser necessário o conhecimento da linguagem HTML. Com esses programas, procedimentos relativamente complexos da linguagem (tabelas, formulários e equações) são implementados facilmente com a ajuda de janelas e menus amigáveis. Uma vez gerado o código, a página final pode ser editada ou visualizada, sem deixar o programa. Esses programas também possuem tutoriais, exemplos de páginas e guias de referência de HTML. Entre os mais conhecidos estão o Hot Dog, Navigator Gold e HoTMetaL. Prestigiando o software nacional, recomendamos o editor W3E, grátis e poderoso. Uma relação dos principais softwares de edição HTML pode ser encontrado em <http://www.cwsapps.com/html.html>.

Como cada *browser* tem uma interpretação diferente das *tags*, não existe um verdadeiro editor HTML Wysiwyg. Entretanto, a maioria dos editores se aproximam bastante da interpretação das *tags* dos dois *browsers* mais populares: Netscape e Mosaic.

Netscape Navigator Gold	http://home.netscape.com
HotDog	http://www.sausage.com
HotMetal	http://www.sq.com
W3E	http://www.nce.ufrj.br/~cracky/w3e/

Para os usuários do programa *Word*, o software *Word Internet Assistant*, um Add-on do *Word*, disponível em *freeware* na Internet (<http://www.microsoft.com/msword/internet/ia>), permite a construção de uma página HTML dentro do *Word* [11]. Pronta a página, deve-se rodar o *browser* usual e abrir o arquivo para ver o resultado. Existem também conversores HTML para vários outros tratamentos de textos (latex, framemaker etc), disponíveis em:

HTML Converters: <http://union.ncsa.uiuc.edu/HyperNews/get/www/html/converters.html>

DESENVOLVIMENTO DE UM TUTORIAL HIPERTEXTO EM HTML

Os tutoriais são compostos de "páginas", de qualquer tamanho, interligadas por um mecanismo de hipertexto. As páginas do tutorial são interligadas por hiperlinks (botões), que podem ser manipulados pelas teclas de direção ou mouse. Cada página pode conter texto, figuras, fotos, sons etc. Como as páginas são ligadas umas às outras pelo hipertexto, o conhecimento é apresentado de forma não linear, o mesmo modo de aprender e de pensar da maioria das pessoas.

A criação de um *site* técnico na internet deve ser enxuta, com pouca firula e muita informação. Assim, estas *home-pages* são relativamente fáceis de montar. Para facilitar o trabalho, em vez de programar diretamente em HTML, recomenda-se utilizar um bom software de edição Wysiwyg.

O tutorial sobre Sistemas Térmicos (ainda em construção) foi desenvolvido a partir de uma apostila utilizada na disciplina Sistemas Térmicos I (EMA-003) do Curso de Engenharia Mecânica da UFMG. A apostila encontra-se editada no tratamento de textos *WordPerfect*. Como o tamanho ideal de uma página hipertexto é aproximadamente o da tela de computador, esta apostila foi dividida em dezenas de arquivos, arquivados em diferentes subdiretórios de acordo com a organização inicial da apostila. Esta organização visou não perder o controle total do projeto, num conjunto desordenado de arquivos. Todos os arquivos foram impressos em páginas separadas, arrumadas em ordem em um colecionador, a fim de anotar manualmente os *links* implementados.

Através do *WordPerfect Internet Publisher* (<http://www.novell.com>), um programa que permite criar páginas HTML dentro do *WordPerfect*, criamos ligações hipertexto entre as páginas, marcando os textos (palavras, frases) que serviriam de *links*. Além disto, foram introduzidos em todas as páginas os marcadores (<próxima página>, <página anterior>), que permitem a navegação linear de todo o texto, tal qual na apostila original, bem como o marcador <menu> que leva à página inicial do tutorial. Todas as páginas foram arquivadas no formato .HTM.

O tutorial foi intensamente testado, visando encontrar algum furo nas ligações hipertexto. Em seguida, analisando manualmente o colecionador de páginas impressas, verificamos se existiam no mínimo dois *links* de entrada e dois de saída em cada página (além daqueles da navegação linear). Onde eles não existiam, verificamos a possibilidade de sua implantação, de modo que a página não ficasse perdida, e inútil, dentro do tutorial. Algumas páginas, que tinham somente a função de explicar ou definir um ponto, foram retirados os marcadores lineares, deixando somente um marcador de <voltar> à página de origem da questão.

Uma vez implementado o hipertexto, realizou-se a arte final em cada página, reformatando o texto e introduzindo título, cores, bordas, figuras, fotos etc. As imagens foram convertidas para os padrões GIF (gráficos, desenhos) e JPEG (fotos), formatos de compressão padrões da Internet. Apesar do HTML 3.0 permitir a edição de equações, para compatibilidade com o *browser* utilizado, as equações foram salvas na forma de um gráfico GIF. Não se deve esquecer, que cada *browser* visualiza a página criada de uma forma diferente. Pronta a página, recomenda-se visualizar a página no *browser* usual para ver o resultado final.

Todos os arquivos .htm, foram armazenados em um disquete, juntamente com o *browser off-line* *Likse* (<http://www.faico.net>), que funcionou bem e é pequeno (400kb). Assim, o disquete contém um tutorial portátil, que os alunos podem utilizar onde quiserem. Uma outra opção é utilizar um compilador HTML, como o InfoCourier (<http://www.smartcode.com>), que gera um aplicativo executável para o ambiente Windows®, dispensando a necessidade de um *browser*.

Devido ao fato deste tutorial estar desenvolvido em HTML, ele pode ser armazenado num servidor ligado à internet, podendo ser acessado de qualquer ponto do mundo e navegado por qualquer *browser*. Com a implantação da *home-Page* do Curso de Engenharia Mecânica da UFMG, desejamos

deixar este tutorial disponível *on-line*.

É FAZENDO QUE SE APRENDE

Como se pode notar, a criação de um tutorial em HTML é "simples, fácil e não requer prática nem habilidade". A HTML é a linguagem de programação mais simples existente, não possuindo muitos recursos de modo que as páginas construídas sejam compreendidas pelos vários *browsers*, em diversos sistemas operacionais. Assim, o desenvolvimento de *home-pages* exige boa dose de criatividade para compensar as limitações da linguagem.

Para que o tutorial hipertexto cumpra seus objetivos, ele deve possuir uma estrutura lógica bem organizada, de modo que o usuário possa dirigir a navegação de acordo com suas associações pessoais. Encher de *links* inúteis e desnecessários torna o tutorial confuso e chato. Por outro lado, se os *links* hipertexto são usados criteriosamente, eles aumentam a organização da informação, tornando-a mais acessível os usuários. Eles podem criar novas interrelações lógicas no conhecimento, sem apagar as interrelações que já possuíam.

Como o aprendizado se faz pela construção do conhecimento, o desenvolvimento de tutoriais pelos próprios alunos, é uma técnica poderosa de ensino. Usualmente, os professores solicitam aos alunos a realização de trabalhos de pesquisa sobre determinado assunto. Muitos alunos, fazem um apanhado do assunto, e apresentam ao professor, sem realizar uma análise crítica, isto quando não copiam direto de uma só fonte. Se em vez de pedir a pesquisa, for solicitado o desenvolvimento de uma *home-page* sobre o assunto, o aluno além da pesquisa bibliográfica vai necessitar realizar uma análise profunda do texto, de modo a criar os *links* entre seus diversos pontos. Só quem estudou e entendeu consegue. Isto força a análise crítica, e produz mais aprendizado do que o uso posterior do tutorial.

Como se pode notar o desenvolvimento de um tutorial em HTML é muito simples, e qualquer aluno pode desenvolver um. Não estamos propondo a criação de uma *home-page* profissional, cheio de firulas, sons etc, mas a estruturação do conhecimento de uma forma lógica e interativa. Além disto, devido à febre da Internet, os alunos são altamente motivados em realizar este tipo de trabalho. O custo final é baixo (1 disquete 3,5"), é fácil para o professor analisar os resultados, e permite a difusão (a outros alunos) do conhecimento.

A LINGUAGEM JAVA

A linguagem Java foi desenvolvida pela Sun Microsystems, para controlar aparelhos eletrônicos domésticos por rede. É uma linguagem de programação orientada a objetos, e simplificação da linguagem C⁺, que adiciona recursos gráficos, de animação, de sons e cálculos à HTML. Como é multiplataforma, transcende as particularidades de cada máquina, realizando a tarefa em qualquer ambiente operacional, independente da plataforma: PC, Mac, Unix, VM etc. É uma linguagem implementável sobre qualquer tipo de plataforma e que deverá em breve se tornar padrão. É a realização do sonho de unificar as linguagens de programação.

O entusiasmo com a invenção levou a SUN a procurar parceiros que estivessem interessados em usar esta tecnologia, mas fracassou. A salvação veio com a WWW. A idéia: utilizá-la para criar um programa executável que seria enviado juntamente com a página WWW, agregando animações, formulários interativos etc, algo impossível com o HTML.

A linguagem Java, em vez de interpretar a linguagem, um programa compilado (applet) é transferido para ser executado no cliente, conseguindo-se assim uma maior velocidade de execução. Isto permite o processamento no cliente. Anteriormente, o processamento tinha que acontecer no servidor

Web. Isto torna possível aplicações como planilhas de cálculo ou validação de dados, além de animações com qualidade muito superior à do HTML. A linguagem Java lembra muito o C++, é orientada a objeto, mas não é tão simples como o HTML. Para facilitar, foi desenvolvida uma versão interpretada da Java, o JavaScript.

O código da linguagem Java é transformado em uma estrutura chamada "bytecode", que vai para o computador cliente juntamente com a página, sem entretanto fazer parte do código HTML, onde é interpretada. O código Java pode ser compilado e gerar um software off-line. Existe uma outra linguagem voltada para a WWW, o JavaScript, de aprendizado mais fácil, porém menos poderosa, e cujo o código é escrito juntamente com o código HTML. Os aplicativos JavaScript só funcionam em conjunto com um *browser*.

Os JavaScripts são capazes de carregar imagens dinamicamente conforme necessário, dependendo de eventos específicos ou da interação com o usuário. Os scripts são capazes de detectar *plug-ins* instalados, *Arrays*, e interagir com os applets Java carregados no *browser*. Tanto o bytecode, o JavaScript e HTML são posteriormente interpretados pelo *browser*, o que garante que o programa será executado em qualquer plataforma.

Java home page: <http://java.sun.com>

Cursos, exemplos, notícias: <http://sunsite.unc.edu/java.faq>

Utilitários Java: <http://www.blackdown.org/~kbs/index.html>

Tutorial Java. André A. Cesta: <http://www.dcc.unicamp.br/~cmrubira/aacesta/>

Curso JavaScript, Sérgio Galino: <http://www.elogica.com.br/bbs/curso/FrameJava.htm>

CONCLUSÕES

O desenvolvimento de um tutorial hipertexto para o ensino auxiliado por computador é uma tarefa relativamente simples. A linguagem HTML é simples e de fácil aprendizado e implementação, diferentemente das linguagens de alto nível tais como C++ ou Pascal. Com um tratamento de texto usual e o conhecimento do HTML é fácil desenvolver um tutorial hipertexto. O tutorial resultante permite a apresentação de uma grande quantidade de informações, de uma forma mais atraente e amigável do que nos livros texto e apostilas, devido à interatividade e à facilidade com que se pode navegar pelo documento. Apesar da programação HTML não ser complicada, existem alguns softwares que facilitam a implementação das *home-pages* em HTML.

Assim como na televisão, os tutoriais hipertexto apresentam seu conteúdo de maneira lógica e direta, não sendo necessário grande esforço de interpretação e análise por parte do aluno. A visão construtivista fica assim prejudicada. Assim, é imprescindível a presença do professor no processo de aprendizagem. O computador não é uma panacéia para os problemas do ensino, ao contrário, pode acentuar os erros já existentes no modelo pedagógico atual, um vez que o professor perde o controle do processo de aprendizagem, e não pode acompanhar como se desenvolve o ensino. É preciso saber usá-lo para ajudar a desenvolver o conhecimento do aluno. Os professores possuem um papel fundamental no processo, mas é preciso mudar mentalidades. O professor deixa de ser aquele que só ensina, e passa a ser alguém que também aprende, inclusive com os alunos.

Apesar dos alunos ficarem motivados ao tomar o aprendizado em suas próprias mãos, como o controle do progresso de aprendizado é realizado pelo próprio computador, o feedback de incentivo e repreensão é mecânico, impessoal, não satisfazendo o ego e criando um conflito ético. Por mais completo que seja o aplicativo multimídia, ele é incapaz de inferir, aprender e se desenvolver e solucionar todas as dúvidas, erros e enganos dos alunos. O programa sempre sabe que o aluno errou, mas normalmente não sabe qual o erro, nem porque o aluno errou. Errar nunca foi tão humano quanto no tempo dos softwares educacionais.

O desenvolvimento de tutoriais hipertexto pelos próprios alunos, é uma técnica poderosa de ensino. Se em vez de pedir a pesquisa, for solicitado o desenvolvimento de uma *home-page* sobre o assunto, o aluno além da pesquisa bibliográfica vai necessitar realizar uma análise profunda do texto, de modo a criar os *links* entre seus diversos pontos. Só quem estudou e entendeu consegue. Isto força a análise crítica, e produz mais aprendizado do que o uso posterior do tutorial.

REFERÊNCIAS

- [1] CASTRO Maria Alice Soares. Tutorial HTML, <http://www.icmsc.sc.usp.br/manuals/>
- [2] CRIE sua Home-Page na Internet. Micro Sistemas, v.19, n.154, p.24-26, 28-30, Nov.1995
- [3] HTML Guide. <http://coolzine.com/html>
- [4] MOREIRA José de Albuquerque. Guia de Iniciação ao HTML.
<http://www.mat.unb.br/~moreira/cartilha.html>
- [5] NCSA. A Beginner's Guide to HTML.
<http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html>; <ftp://pub@ncsa.uiuc.edu>
- [6] <news:comp.infosystems.www.authoring.html>
- [7] PEREIRA Luiz Gustavo Anflor. Tutorial sobre a Construção de Questões. <http://www.pops.rs.rnp.br/ovni/questao/questao.htm>
- [8] PINHEIRO, P.C.C.; KOURY R.N.N.; HUEBNER R. Desenvolvimento de um Tutorial Hipertexto Com o WinHelp para o Ensino de Mecânica dos Fluidos. In: XXIV Congresso Brasileiro de Ensino de Engenharia (COBENGE-95), 13-16 Outubro 1996, Manaus, AM, *Anais...* Manaus: ABENGE, Associação Brasileira de Ensino de Engenharia, 1996, v.2, p.753-765.
- [9] RAMALHO José Antônio. Monte Sua Home Page e Caia na Rede. Informática Exame, v.11, n.122, p.94-96, 98, Maio 1996. <http://www2.uol.com.br/info/arquivo/ie122/tutorial.html>
- [10] ROCHA Helder L.S. Como Criar sua Home Page. <http://www.dsc.ufpb.br/~helder/html-ref/tutorial/>
- [11] SAIBA Como Criar Sites Usando o "Word". Folha de São Paulo, n.24766, Caderno Informática, p.5, 22 Janeiro 1997.
- [12] YALE. Yale Web Style Guide. <http://info.med.yale.edu/caim/manual>