



·rede
e-Tec
Brasil

Introdução à Programação para *Web*

Otilio Paulo da Silva Neto

Nádia Mendes dos Santos

Sandra Eliza Veloso Aguiar



·rede
e-Tec
Brasil

Teresina – PI
2013

Presidência da República Federativa do Brasil
Ministério da Educação
Secretaria de Educação Profissional e Tecnológica

© Instituto Federal de Educação, Ciência e Tecnologia do Piauí
Este Caderno foi elaborado em parceria entre o Instituto Federal de Educação, Ciência e Tecnologia do Piauí e a Universidade Federal de Santa Catarina para a Rede e-Tec Brasil.

Equipe de Elaboração

Instituto Federal de Educação, Ciência e Tecnologia do Piauí – IFPI

Coordenação Institucional

Francieric Alves de Araujo/IFPI

Coordenação do Curso

Thiago Alves Elias da Silva/IFPI

Professores-autores

Otilio Paulo da Silva Neto/IFPI
Nádia Mendes dos Santos/IFPI
Sandra Eliza Veloso Aguiar/IFPI

Comissão de Acompanhamento e Validação

Universidade Federal de Santa Catarina – UFSC

Coordenação Institucional

Araci Hack Catapan/UFSC

Coordenação do Projeto

Silvia Modesto Nassar/UFSC

Coordenação de Design Instrucional

Beatriz Helena Dal Molin/UNIOESTE e UFSC

Coordenação de Design Gráfico

Juliana Tonietto/UFSC

Design Instrucional

Eleonora Schlemper Mendonça/UFSC
Gustavo Pereira Mateus/UFSC

Web Master

Rafaela Lunardi Comarella/UFSC

Web Design

Beatriz Wilges/UFSC
Mônica Nassar Machuca/UFSC

Diagramação

Barbara Zardo /UFSC
Breno Takamine /UFSC
Roberto Colombo /UFSC

Revisão

Júlio César Ramos/UFSC

Projeto Gráfico

e-Tec/MEC

Catálogo na fonte pela Biblioteca Universitária da
Universidade Federal de Santa Catarina

S586i Silva Neto, Otilio Paulo da
Introdução à programação para Web / Otilio
Paulo da Silva Neto, Nádia Mendes dos Santos,
Sandra Eliza Veloso Aguiar. – Teresina : Instituto Federal de educação,
Ciência e Tecnologia do Piauí, 2013.
102 p. : il., tabs.

Inclui bibliografia

ISBN: 978-85-67082-03-5

1. Internet (Redes de computadores). 2. Linguagem de
programação (Computadores). 3. Sites
da Web – Desenvolvimento. I. Santos, Nádia Mendes dos. II.
Aguiar, Sandra Eliza Veloso. III. Título.

CDU: 681.31.066.1

Apresentação e-Tec Brasil

Bem-vindo a Rede e-Tec Brasil!

Você faz parte de uma rede nacional de ensino, que por sua vez constitui uma das ações do Pronatec - Programa Nacional de Acesso ao Ensino Técnico e Emprego. O Pronatec, instituído pela Lei nº 12.513/2011, tem como objetivo principal expandir, interiorizar e democratizar a oferta de cursos de Educação Profissional e Tecnológica (EPT) para a população brasileira propiciando caminho de o acesso mais rápido ao emprego.

É neste âmbito que as ações da Rede e-Tec Brasil promovem a parceria entre a Secretaria de Educação Profissional e Tecnológica (SETEC) e as instâncias promotoras de ensino técnico como os Institutos Federais, as Secretarias de Educação dos Estados, as Universidades, as Escolas e Colégios Tecnológicos e o Sistema S.

A educação a distância no nosso país, de dimensões continentais e grande diversidade regional e cultural, longe de distanciar, aproxima as pessoas ao garantir acesso à educação de qualidade, e promover o fortalecimento da formação de jovens moradores de regiões distantes, geograficamente ou economicamente, dos grandes centros.

A Rede e-Tec Brasil leva diversos cursos técnicos a todas as regiões do país, incentivando os estudantes a concluir o ensino médio e realizar uma formação e atualização contínuas. Os cursos são ofertados pelas instituições de educação profissional e o atendimento ao estudante é realizado tanto nas sedes das instituições quanto em suas unidades remotas, os polos.

Os parceiros da Rede e-Tec Brasil acreditam em uma educação profissional qualificada – integradora do ensino médio e educação técnica, - é capaz de promover o cidadão com capacidades para produzir, mas também com autonomia diante das diferentes dimensões da realidade: cultural, social, familiar, esportiva, política e ética.

Nós acreditamos em você!

Desejamos sucesso na sua formação profissional!

Ministério da Educação
Fevereiro de 2013

Nosso contato
etecbrasil@mec.gov.br



Indicação de ícones

Os ícones são elementos gráficos utilizados para ampliar as formas de linguagem e facilitar a organização e a leitura hipertextual.



Atenção: indica pontos de maior relevância no texto.



Saiba mais: oferece novas informações que enriquecem o assunto ou “curiosidades” e notícias recentes relacionadas ao tema estudado.



Glossário: indica a definição de um termo, palavra ou expressão utilizada no texto.



Mídias integradas: sempre que se desejar que os estudantes desenvolvam atividades empregando diferentes mídias: vídeos, filmes, jornais, ambiente AVEA e outras.



Atividades de aprendizagem: apresenta atividades em diferentes níveis de aprendizagem para que o estudante possa realizá-las e conferir o seu domínio do tema estudado.



Sumário

Palavra do professor-autor	9
Apresentação da disciplina	11
Projeto instrucional	13
Aula 1 – Internet	15
1.1 Introdução à internet.....	15
1.2 Internet em datas	18
1.3 Internet no Brasil.....	22
1.4 Internet: o que é, quem controla,	22
como funciona, quais os serviços e como acessar.....	22
1.5 Domínio na internet.....	30
Aula 2 – Elaborando um <i>site</i>	33
2.1 Introdução ao <i>site</i>	33
2.2 Planejamento do <i>site</i>	35
2.3 Desenvolvendo um <i>site</i>	37
2.4 Projeto do <i>site</i>	41
2.5 Fases na construção do <i>site</i>	44
2.6 Administração do <i>site</i>	44
2.7 W3C.....	46
Aula 3 – Introdução à HTML	49
3.1 Introdução á linguagem de marcação de hipertexto – HTML.....	49
3.2 Primeira página em HTML.....	50
3.3 Estudo de <i>tags</i> em HTML.....	54
Aula 4 – Criando imagens, <i>links</i> e tabelas em HTML	63
4.1 Imagens.....	63
4.2 URLs e <i>links</i>	64
4.3 Tabelas.....	68

Aula 5 – Frames, multimídia e folhas de estilos – CSS	73
5.1 <i>Frames</i>	73
5.1.1 Como funcionam os <i>frames</i>	73
5.2 Multimídia.....	78
5.3 Folhas de estilo.....	85
5.4 Posicionamentos.....	94
Referências	98
Currículo dos professores-autores	100

Palavra do professor-autor

Caro estudante!

Bem-vindo à disciplina Introdução à Programação para *Web*.

Este é o nosso caderno elaborado com o objetivo de contribuir para o desenvolvimento de seus estudos e para a ampliação de seus conhecimentos acerca da citada disciplina.

Este texto é destinado aos estudantes aprendizes que participam da Rede e-Tec Brasil, vinculado à Escola Técnica Aberta do Piauí (ETAPI) do Instituto Federal de Educação, Ciências e Tecnologia do Piauí (IFPI), com apoio da Prefeitura Municipal dos respectivos polos: Batalha e Valença do Piauí.

Um grande abraço!

Otilio Paulo
Nádia Mendes
Sandra Eliza



Apresentação da disciplina

Apresentação da disciplina

O texto é composto de cinco aulas distribuídas conforme a seguir.

Na Aula 1 faremos uma introdução à internet. Vamos entender como a internet surgiu e como evoluiu até os dias atuais. Saberemos, ao final, o que é a internet, o que controla, como funciona, quais seus principais serviços e como acessá-la.

Na Aula 2 apresentaremos as etapas na elaboração de um *site*. Veremos como fazer o planejamento, desenvolver e projetar um *site*. Saberemos quais as fases na sua construção e como administrar um *site* disponível na web. Teremos ainda algumas informações sobre o W3C.

Na Aula 3 apresentaremos uma introdução à *Hyper Text Markup Language* (HTML), uma linguagem de marcação de hipertexto para web, e a escolhida para este caderno. Aprenderemos a estrutura básica de um documento HTML. Quais as *tags* introdutórias da linguagem, ou seja, começaremos, literalmente, a programar para web.

Na Aula 4 apresentaremos o uso e a aplicação das *tags* de inserção de imagens, criação de *hyperlinks* internos e externos e o uso de tabelas com a linguagem de marcadores HTML.

Na Aula 5 Implementaremos *layouts* de páginas utilizando novos paradigmas em documentos HTML como: *frames*, multimídia e folha de estilo.



Projeto instrucional

Disciplina: Introdução à Programação para *Web* (carga horária: 75 h).

Ementa: Introdução à Internet: Internet em Datas; Internet: o que é, quem controla, como funciona, quais os serviços e como acessar; domínio na internet. Elaborando um *site*: introdução; planejamento; desenvolvimento e projeto de *sites*. Fases na construção e administração de *site*. W3C. Introdução à linguagem HTML: primeira página e estudo de *tags*. HTML Avançada: criando imagens, URL, *links* e tabelas, *frames*, *multimídia*. E Introdução às folhas de estilos – CSS.

AULA	OBJETIVOS DE APRENDIZAGEM	MATERIAIS	CARGA HORÁRIA (horas)
1. Internet	Conceituar a internet. Conhecer a trajetória da internet. Saber quem controla a internet. Entender como funciona a internet. Conhecer os principais serviços da internet. Saber como acessar a internet. Saber o que é um <i>browser</i> .	<i>Web</i> conferência; postagem de caderno; postagem de exercícios. AVEA: http://cefetpi.nucleoad.net/etapi ; Videoaula.	15
2. Elaborando um <i>site</i>	Planejar um <i>site</i> . Conhecer o desenvolvimento de um <i>site</i> . Projetar um <i>site</i> . Saber administrar um <i>site</i> disponível na <i>web</i> . Saber o que é o W3C.	<i>Web</i> conferência; postagem de caderno; postagem de exercícios. AVEA: http://cefetpi.nucleoad.net/etapi ; Videoaula.	15
3. Introdução à HTML	Conhecer conceitos básicos da linguagem HTML. Compreender a estrutura básica de um documento HTML. Saber o que é uma <i>tag</i> . Conhecer as <i>tags</i> introdutórias da linguagem HTML. Começar a programar para <i>web</i> utilizando HTML.	<i>Web</i> conferência; postagem de caderno; postagem de exercícios. AVEA: http://cefetpi.nucleoad.net/etapi ; Videoaula.	15
4. Criando imagem, <i>links</i> e tabelas em HTML	Implementar páginas HTML usando imagens. Criar documentos HTML com estruturas de <i>hyperlinks</i> internos, externos e <i>links para terceiros</i> . Construir páginas HTML com as estruturas de tabelas.	<i>Web</i> conferência; postagem de caderno; postagem de exercícios. AVEA: http://cefetpi.nucleoad.net/etapi ; Videoaula.	15
			Continua

AULA	OBJETIVOS DE APRENDIZAGEM	MATERIAIS	CARGA HORÁRIA (horas)
5. <i>Frames</i> , multimídia e folhas de estilos – CSS	<p>Criar estruturas de <i>frames</i>.</p> <p>Implementar documentos com elementos multimídia.</p> <p>Estruturar páginas HTML com folhas de estilo, criando-se uma padronização.</p>	<p>Web conferência; postagem de caderno; postagem de exercícios.</p> <p>AVEA: http://cefetpi.nucleoad.net/etapi;</p> <p>Videoaula.</p>	15
			Conclusão

Aula 1 – Internet

Objetivos

- Conceituar a internet.
- Conhecer a trajetória da internet.
- Saber quem controla a internet.
- Entender como funciona a internet.
- Conhecer os principais serviços da internet.
- Saber como acessar a internet.
- Saber o que é um *browser*.

1.1 Introdução à internet

Para se programar para *web*, é preciso conhecer primeiro o que é essa *web*; então, vamos lá!

Eu convido você para participar de uma viagem pelo mundo da internet. Ao final dessa jornada, você estará apto a navegar pela internet e ficará maravilhado com o mundo de novas possibilidades que irá se abrir para você.

Antes de iniciarmos nossos estudos sobre internet, é necessário uma explanação sobre a importância do computador pessoal e do surgimento das redes de computadores para a disseminação da internet.

As tecnologias de comunicação como o rádio, telefone, televisão contribuíram de forma significativa para tornar a transmissão da informação mais rápida e abrangente; porém, nada se compara ao computador que, consegue fazer-se passar ora por telefone, ora por secretária eletrônica, ora por televisão, ou até por DVD, ou todos ao mesmo tempo, demonstrando um dinamismo sem precedentes na história da tecnologia.

Por causa do preço cada vez menor, o computador tornou-se um item de consumo acessível à população em geral, deixando de ser um artigo de luxo, para, inclusive, marcar presença regularmente em prateleiras de supermercados e em grandes liquidações de lojas do varejo.

As redes de computadores proporcionaram o surgimento da internet. Elas surgiram durante a década de 1970 como facilitadoras dos acessos a recursos de *hardware* (impressoras, dispositivos de armazenamento e unidade de processamento) entre os diversos usuários. Isso porque os equipamentos dessa época apresentavam altos preços.

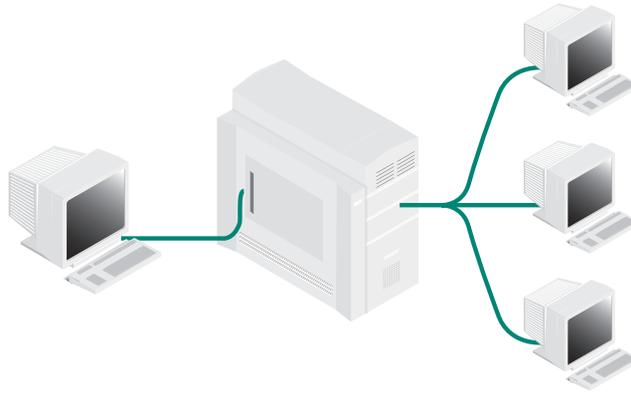


Figura 1.1: Rede de computadores

Fonte: Banco de Imagens do Google, 2011

A década de 1980 trouxe como evolução tecnológica o advento do PC (*Personal Computer* ou Computador Pessoal) e a conseqüente diminuição dos preços do *hardware* em geral, possibilitando dessa forma a criação de estações de trabalho dedicadas, para que se pudesse eliminar o compartilhamento de recursos.



Figura 1.2: Computador pessoal

Fonte: Site da IBM

No início da década de 1990, as redes de computadores tiveram seu modelo consolidado, possibilitando o seu uso na informatização das empresas. O que podemos constatar é que as redes de computadores são elementos básicos e muitas vezes fundamentais dos mais diferentes tipos de organizações, em qualquer parte do mundo.

Então, com a evolução dos computadores, o advento das redes de computadores, surgiu a “famosa” internet?

Sim, ela chegou como uma imensa onda que atingiu milhões de pessoas no mundo todo e que continua se alastrando pelo planeta com um ritmo avassalador. Com a internet, o mundo parece ter “encolhido”. Hoje se pode ficar sabendo, quase que instantaneamente, o que acontece em qualquer lugar do planeta, acompanhar em tempo real acontecimentos a milhares de quilômetros de distância, tudo isso sem sair de frente do computador.

Mas, ela já surgiu assim, dinâmica, ou seja, com tantos recursos e acessível a todos? Não; portanto, vamos agora entender como aconteceu seu surgimento.

Tudo começou no final dos anos 1960. Uma iniciativa do governo americano, especificamente do Departamento de Defesa, fez com que pesquisadores de várias instituições dos Estados Unidos fossem incumbidos de projetar um sistema de defesa, cuja grande diferença em relação a outros sistemas já projetados era que esse seria informatizado, capaz de resistir a ataques do inimigo que fossem realizados com armas nucleares.

Como solução, surgiu um sistema em que os dados eram divididos em pacotes, os quais seriam encaminhados, instantaneamente, por várias rotas que estivessem disponíveis, ou seja, o sistema era baseado em redes de computadores. Assim, o sistema poderia continuar em operação mesmo quando um ou mais computadores dessa rede fossem destruídos, ou seja, a comunicação poderia fluir para outros computadores que ainda estivessem funcionando.

Na prática, os pesquisadores americanos inventaram uma tecnologia voltada para a interligação de computadores que funcionava em condições precárias de comunicação, que de maneira simplória poderia denominar esse conjunto de simples tecnologias, porém extremamente confiável de internet.

O nome internet começou a ser usado em 1973, derivado do conceito de “*internetworking*” ou interligação de redes, que vinha sendo estudado pelos pesquisadores americanos desde o ano anterior. No entanto, a rede que conhecemos hoje como internet só surgiu na década de 1980.



Para saber mais sobre internetworking, consulte o link http://www.cisco.com/en/US/tech/tk331/tk336/technologies_white_paper09186a0080114a38.shtml#intro.

1.2 Internet em datas

Para entendermos melhor como surgiu a internet, nada como acompanhar a sua cronologia desde o nascimento das primeiras ideias relacionadas à redes de computadores até os dias atuais.

- **1957** – A União Soviética lança o Sputnik, primeiro satélite artificial da Terra. Em resposta, os Estados Unidos criam a *Advanced Research Projects Agency* (ARPA) dentro do Departamento de Defesa (DoD) para estabelecer a liderança dos Estados Unidos na ciência e na tecnologia militar.
- **1969** – No dia 1º de maio é montado o primeiro equipamento da rede na Universidade da Califórnia, de Los Angeles, (UCLA). Fica, então, criado o primeiro ponto de conexão da rede, que se chamaria ARPANET. Ainda em 1969 foram montados os três pontos restantes de conexão. O segundo foi o do Instituto de Pesquisas de Stanford, seguindo-se o Centro de Pesquisas da Universidade da Califórnia em Santa Bárbara (UCSB) e o da Universidade de Utah.
- **1970** – Primeira publicação do protocolo Host-Host original da ARPA. Primeiro artigo da AFIPS sobre a ARPANET. ALOHAnet, a primeira rede de comutação de dados via rádio. Os servidores da ARPANET começam a usar o Protocolo de Controle de Rede (*Network Control Protocol*, NCP), primeiro protocolo **host-host**.
- **1971** – É inventado o programa de *e-mail* para enviar mensagens através de rede distribuída. O programa original foi derivado de dois outros: um programa de *e-mail* intramáquina (SENDMSG) e um programa experimental de transferência de arquivo (CPYNET).
- **1976** – É inventado o programa de *e-mail* para enviar mensagens através de rede distribuída. O programa original foi derivado de dois outros: um programa de *e-mail* intramáquina (SENDMSG) e um programa experimental de transferência de arquivo (CPYNET).
- **1978** – Os protocolos TCP-IP foram divididos em TCP e IP.
- **1979** – Inauguração da primeira rede dedicada a Grupos de Discussão – USENET. É, também, lançado o primeiro *chat* – canal de conversa em tempo real acrescido da possibilidade de Role Playing Game (RPG).
- **1982** – Bob Kahn e Vint Cerf são os mentores e chefes da equipe que

acaba de inventar o TCP/IP, a linguagem comum a todos os computadores da internet. Pela primeira vez a coleção de redes espalhadas que constituíam a ARPANET é encarada como uma única rede interligada. As grandes empresas começam a usar a internet para se comunicarem entre elas e com os seus clientes.

- **1983** – Mudança do NCP para TCP/IP. *Movement Information Net* – MINET (Rede de Movimento de Informação) tem início no começo do ano na Europa, conectado à internet em setembro Conselho das Atividades da Internet (*Internet Activities Board*, IAB) é estabelecido, substituindo o ICCB, ARPANET se divide em ARPANET e MILNET; este último se integrou à Rede de Dados de Defesa (*Defense Data Network*), criada no ano anterior. Foram para a MILNET 68 dos 113 nós existentes.
- **1984** – É introduzido o Sistema de Nome de Domínio (*Domain Name System*, DNS).
- **1988** – A rede atinge os 60.000 servidores. É concluída a instalação do primeiro cabo de fibra óptica transatlântico ligando a Europa e a América do Norte. Esse cabo pode suportar até 40.000 chamadas telefônicas simultâneas. Surge o primeiro vírus na internet que tomou a designação de Internet Worm e afeta 6.000 servidores. Para proteger a rede e aumentar a sua segurança, é criado o CERT (Computer Emergency Response Team). Novos termos como “cracker” e “hacker” entram para o vocabulário da internet. Pelas mãos de Jarkko Oikarinen nasce o IRC – Internet Relay Chat. Canadá, Finlândia, França, Islândia, Noruega, Dinamarca e Suécia ligam-se à internet.
- **1990** – Surge nos EUA o primeiro *Internet Service Provider* comercial. Após ter sido absorvida pela internet, a ARPANET deixa, definitivamente, de existir. Peter Deutsh, Alan Emtage e Bill Heelan da McGill lançam o Archie. O primeiro *software World Wide web* é criado por Tim Berners-Lee. Surgem os primeiros ISP (Internet Service Providers) comerciais. O número de servidores na internet excede os 300.000. Argentina, Áustria, Bélgica, Brasil, Chile, Grécia, Índia, Irlanda, Coreia do Sul, Espanha e Suíça entram na internet.
- **1991** – Gopher é lançado na Universidade de Minnesota.
- **1992** – É inventada a WWW (*World Wide Web*) por Tim Berners-Lee no CERN (Laboratório Europeu de Física de Partículas), em Gênova. A ferramenta de procura no espaço Gopher, VERONICA, é lançada pela Univer-

sidade de Nevada. O número de servidores é agora de um milhão. Camarões, Chipre, Equador, Estónia, Kuwait, Luxemburgo, Eslováquia, Eslovénia, Venezuela, Antártida, Letónia, Malásia e Tailândia ligam-se à internet.

- **1993** – O Mosaic, de Marc Andreessen, é lançado pelo Centro Nacional de Aplicações de Supercomputação dos Estados Unidos (NCSA) e torna-se a primeira implementação de um *browser* para a internet correr em ambientes Windows, Unix e Macintosh. É criado o InterNIC pela Fundação Nacional de Ciência Americana. A Casa Branca e as Nações Unidas passam a estar *on line*. Bulgária, Costa Rica, Egito, Fiji, Gana, Guam, Indonésia, Cazaquistão, Quênia, Liechtenstein, Peru, Romênia, Federação Russa, Turquia, Ucrânia, Ilhas Virgens e Emirados Árabes Unidos ligam-se à rede.
- **1994** – Começa a funcionar o First Virtual, o primeiro banco exclusivamente acessível pela internet. Jim Clark e Marc Andreessen fundam a Mosaic Communications Corp., que viria originar a Netscape Communications Corp. Surge a primeira estação de rádio a transmitir via internet. É lançado o Netscape Navigator e respectivo *software* de servidor. O número de servidores na Net ultrapassa os três milhões. Algéria, Arménia, Bermuda, Burkina Faso, China, Colômbia, Polinésia Francesa, Líbano, Lituânia, Macau, Marrocos, Nova Caledónia, Nicarágua, Nigéria, Panamá, Filipinas, Senegal, Sri Lanka, Suazilândia, Uruguai, Uzbequistão, Jamaica e Jordânia entram para a rede. ARPANET/Internet celebra seu 25º aniversário. Aparecimento das primeiras páginas de comércio eletrônico. Localidades começam a conectar-se diretamente à internet. WWW supera a telnet para se tornar o segundo serviço mais popular da rede (atrás da FTP) baseado em porcentagem da distribuição do tráfego de pacotes e *bytes* na NSFNET. Os dez principais domínios por servidor #: com, edu, uk, gov, de, ca, mil, au, org, net.
- **1995** – Os dez principais domínios por servidor #: com, edu, net, gov, mil, org, de, uk, ca, au. Tecnologias do Ano: WWW, mecanismos de procura. Tecnologias Emergentes: Código móvel (JAVA, JAVAscript), ambientes virtuais (VRML), ferramentas de colaboração (*Collaborative tools*). A NSFNET volta a assumir um carácter exclusivamente acadêmico depois do tráfego de backbone da internet passar para as mãos dos ISP comerciais. Serviços *on-line* tradicionais começam a oferecer acesso à internet. As ações do Netscape atingem valores recordes. A WWW ultrapassa o FTP em volume de tráfego na internet. Lançamento do MS Internet Explorer 2.0 para Windows 95. James Gosling e uma equipe de programadores

da SUN Microsystems lançam o Java – linguagem de programação orientada à internet. Entram na rede a Etiópia, Costa do Marfim, Cook Islands, Cayman Islands, Anguilla, Gibraltar, Vaticano, Kiribati, Kyrgyzstan, Madagascar, Maurícias, Micronésia, Mónaco, Mongólia, Nepal, Nigéria, Samoa Oriental, San Marino, Tanzânia, Tonga, Uganda e Vanuatu.

- **1996** – É lançado o Netscape Navigator 2.0, que se torna o primeiro *web browser* a suportar JavaScripts. Três empresas de serviço de procura na rede lançam as suas ações no mercado – são elas a Yahoo, a Excite e Lycos. Bill Clinton aprova leis de telecomunicações no sentido de penalizar a distribuição de conteúdos “inadequados” na internet. A internet pela TV – *web TV* – é uma realidade. Os motores de pesquisa, o Internet Phone e a linguagem Java são as tecnologias do ano. A internet fica com quase dez milhões de servidores *on-line* e aproximadamente quarenta milhões de utilizadores em 150 países. Qatar, República Centro-Africana, Omã, Norfolk Island, Tuvalu, Polinésia Francesa, Síria, Aruba, Camboja, Guiana Francesa, Eritreia, Cabo Verde, Burundi, Benin, Bósnia-Herzegovina, Andorra, Guadalupe, Guernsey, Isle of Man, Jersey, Laus, Maldivas, Ilhas Marshall, Maurítânia, Ilhas Marianas do Norte, Ruanda, Togo, Líbano e Zaire ligam-se à rede.
- **1997** – Tecnologias emergentes: Push, Streaming Media. A America *on-line* Inc. adquire a Netscape Communications Corporation por um valor cifrado em 4,2 bilhões de dólares. Os selos postais eletrônicos tornam-se uma realidade com o US Postal Service, que permite a compra e *download* para impressão a partir da *web*. A Microsoft é processada pela Justiça americana, com base na Lei Antitruste, sob a alegação de prejudicar a concorrência, pois o Windows98 distribui gratuitamente o *browser* Internet Explorer. As tecnologias mais relevantes desse ano foram o e-Commerce, e-Auctions, Portais de Entrada e o XML.
- **1998** – Tecnologias do ano: Comércio Eletrônico, Leilões *on-line*, Portais. A Abilene, a rede Internet 2, atravessa o Atlântico e conecta-se com a NORDUnet e a SURFnet. Surge a ISTF – *Internet Societal Task Force*, liderada por Vint Cerf, com o objetivo de promover questões sociais e preocupações relacionadas com a internet. As tecnologias do ano foram o e-Trade, *on-line* Banking e o MP3. A Palestina entra na rede.
- **2000** – Avanços na internet, aumento do número de usuários, internet para dispositivos móveis.
- **Dias atuais** – A internet continua em constante expansão no planeta e número de serviços aumenta também sem precedentes.



A ARPANET introduziu a internet, servindo no início para fins militares.



Pesquise quem regulamenta a internet no Brasil e qual a função desse órgão no âmbito da internet.

1.3 Internet no Brasil

O Brasil e outros países perceberam a importância da internet e criaram suas redes acadêmicas próprias, estendendo-as para muito além das fronteiras dos EUA.

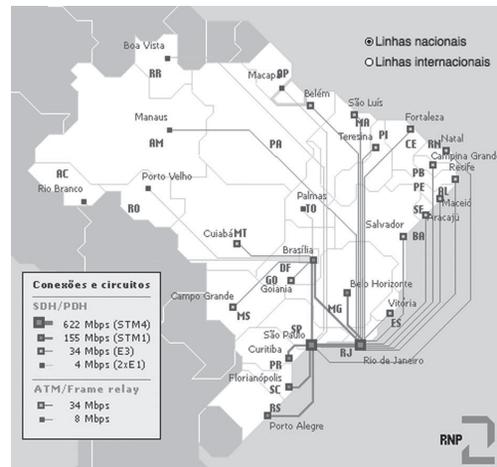


Figura 1.3: RNP no Brasil

Fonte: RNP

A Rede Nacional de Pesquisa (RNP), criada em 1989 pelo Ministério da Ciência e Tecnologia, com apoio de instituições governamentais de vários estados, serviu como a introdução do Brasil na internet. A rede brasileira deixou de ser acadêmica em 1995, sendo os serviços da internet utilizados por empresas e indivíduos de um modo geral.

1.4 Internet: o que é, quem controla, como funciona, quais os serviços e como acessar

Até o início da década de 1990, a internet era considerada uma “rede de redes de computadores”, expressão que evidenciava o aspecto físico da comunicação e integração de computadores pertencentes a várias instituições. Depois disso, a internet também foi considerada “o maior acervo de informações disponíveis publicamente”. Essa expressão enfatizava o aspecto informativo da rede, consequência do uso de novos serviços de rede,

complementares aos serviços ditos básicos e voltados para a disseminação de informações. Recentemente, a internet foi considerada “o protótipo da Supervia da Informação”, expressão usada para reforçar o aspecto universal, bidirecional e de multimídias da comunicação propiciada pela rede. O emprego do termo “protótipo” também revela as limitações a que a internet ainda está sujeita, entre as quais a precariedade dos meios de comunicação utilizados por ela. Portanto, pode-se perceber que a internet é um projeto inacabado, em contínua mutação, que apresenta resultados surpreendentes, problemas desafiadores, e que não se sabe dizer ao certo onde vai parar.

Podemos caracterizá-la da seguinte forma:

1. por ser uma rede pública ou privada,
2. não tem início nem fim, em constante mutação,
3. representa uma nova forma de comércio,
4. tem na usabilidade um dos seus focos,
5. evidencia uma nova publicidade ou divulgação de um produto,
6. une milhares de pessoas com interesses e desejos comuns, ligando o mundo inteiro,
7. é o produto de uma geração,
8. praticamente, uma rede pública e aberta,
9. possui uma taxa de crescimento sempre ascendente,
10. considerada uma fonte de informações, opiniões e entretenimento, dentre outros.

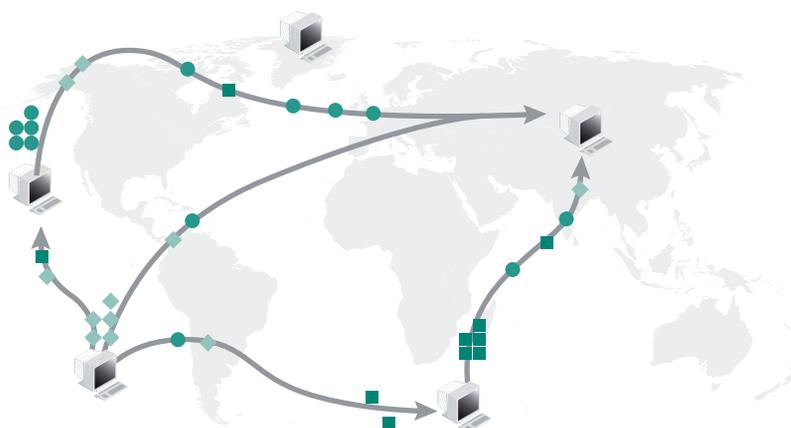


Figura 1.4: Internet

Fonte: Banco de Imagens Google, 2011

A Figura 1.4 evidencia a internet formando uma rede mundial de redes, que permite a ligação entre os cinco continentes. Essa nova tecnologia, chamada de internet, entrou em nossas vidas em ritmo alucinante, fazendo com que

possamos nos comunicarmos, trocar informações de um modo mais rápido e sem condicionamento físicos. Hoje em dia, podemos acessar as mais variadas informações, desde temas sobre concertos até banda de *rock* dos anos 1960, por exemplo. Podemos fazer compras sem sair de casa, conversar com pessoas em outros países, assistir a eventos internacionais em tempo real, saber notícias de tudo o que ocorre no mundo, tudo isso através da internet.

Após entender o que é a internet e como ela chegou ao ponto em que está hoje, pode-se também perguntar: quem é o dono da internet? Vamos melhorar nossa pergunta: quem controla a internet?

A internet como o modelo que existe hoje, não tem um dono ou mesmo quem a controle. Cada governo, companhia ou organização é responsável por manter a sua própria rede. Essas organizações/companhias/governos são independentes do ponto de vista administrativo e não existe um órgão central de controle da internet. Ao trabalhar em conjunto, essas organizações/companhias/governos criaram o que aparenta ser uma rede virtual em escala global. Por outro lado, existem organismos supervisores que cuidam dos vários aspectos do desenvolvimento, operação e uso da rede. No nosso país, eventuais situações de conflito de interesses ou que sejam consideradas abusivas podem ser relatadas ao Comitê Gestor da Internet, formado por representantes do governo e dos diversos setores que compõem a sociedade, pela parte que se considerar prejudicada.

A internet, por se tratar de uma rede de computadores, precisa de seu sistema básico de comunicação para funcionar, chamado de protocolo. Então, o que esse protocolo? O protocolo nada mais que um conjunto de especificações objetivas que os computadores entendem. No jargão tecnológico, é um conjunto de regras que caracterizam o formato, a sincronização, a sequência e detecção de erros e falhas na comutação de pacotes, ou seja, na transmissão de informação entre computadores. Portanto, para que dois ou mais computadores possam se comunicar numa rede, eles têm que usar o mesmo protocolo, ou seja, têm que falar a mesma linguagem.

Assim como as pessoas, os computadores podem falar várias linguagens, o que nos leva a compreender que eles podem usar vários protocolos. Veremos, a seguir, os protocolos mais populares na internet:

- **FTP** – *File Transfer Protocol* ou Protocolo de Transferência de Arquivos, protocolo responsável pela transferência de arquivos. O FTP permite-nos copiar arquivos de um computador remoto para o nosso PC – realizar *download* – bem como transferir arquivos do nosso PC para um computador remoto – realizar *upload*.
- **HTTP** – *Hypertext Transfer Protocol* ou Protocolo de Transferência de Hipertexto é um protocolo que serve de base à *web* – *World Wide web* (WWW), permitindo-nos visualizar as páginas *web*.
- **IP** – *Internet Protocol* ou Protocolo de Internet, é um dos mais importantes entre os protocolos presentes na internet. Ele tem por missão identificar as máquinas e redes e fazer o encaminhamento (roteamento) correto das transmissões entre elas.
- **SMTP** – *Simple Mail Transfer Protocol* ou Protocolo de Transferência de Correio Simples, é o protocolo que fornece os meios que possibilitam utilizar os mecanismos para transferir mensagens entre computadores remotos, ou seja, o serviço de correio eletrônico.
- **TCP** – *Transmission Control Protocol* ou Protocolo de Controle de Transmissão, é um dos protocolos pertencentes ao conjunto TCP/IP. De maneira geral, podemos afirmar que esse **protocolo** é o responsável pelo controle das transmissões de pacotes de informação na internet. É o protocolo TCP o responsável pelo reagrupamento dos pacotes de informações, reorganizando-os, se necessário, de acordo com a sua ordem original, de forma a garantir que a mensagem que chega ao receptor seja a mesma que foi enviada pelo emissor.
- **TCP/IP** – *Transmission Control Protocol/Internet Protocol* – Protocolo de Controle de Transmissão e o Protocolo de Internet.

O TCP/IP é considerado o protocolo padrão da internet.

Já sabemos o que é um protocolo e quais são os mais utilizados na internet; veremos a seguir dois serviços que ela pode nos disponibilizar.

- a) Serviço: correio eletrônico.** O correio eletrônico ou *e-mail* (*electronic mail*) permite às pessoas se comunicarem por meio de mensagens escritas.

A-Z

Protocolo

É um software que permite a comunicação entre computadores, funcionando como se fossem tradutores.



Como ocorre com a carta do correio postal, a mensagem do correio eletrônico também possui remetente e destinatário. Assim como é preciso conhecer o endereço postal do destinatário para enviar uma carta, sem se conhecer o endereço eletrônico dele não se pode enviar uma mensagem.

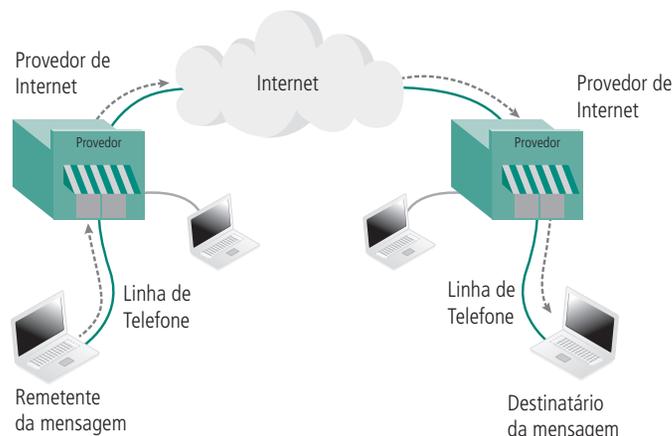


Figura 1.5: Serviço de e-mail

Fonte: Adaptada da Apostila RNP/REF / 0186 (2007)

Esse serviço surgiu nos primórdios da internet, sendo um dos três primeiros serviços pela rede. Para trocar *e-mails*, os computadores do remetente e do destinatário não precisam estar conectados à internet. Devidamente endereçada, a mensagem circula por ela e chega ao seu alvo. E o destinatário também não precisa estar conectado à internet no momento do envio, pois a mensagem fica armazenada na sua caixa postal eletrônica (ou caixa de mensagens eletrônica); quando se conecta à internet, ele pode ler a sua mensagem, respondê-la de imediato, enviar cópias para outros destinatários, armazená-la ou imprimi-la. O custo de envio de uma mensagem é independente da distância entre o remetente e o destinatário. Para enviar uma mensagem para sua mãe, que viajou para outra cidade, ou para o seu filho que está na escola, paga-se o mesmo preço. A falta de garantia do prazo de entrega da mensagem é considerada sua principal desvantagem. Portanto, uma mensagem enviada do remetente até o destinatário geograficamente distante, pode levar de 10 minutos a 10 horas; isso tudo depende das condições da rede no momento do envio. Para que um usuário possa trocar mensagens eletrônicas, tal como se sucede no correio tradicional, ele precisa de um endereço eletrônico (*e-mail*). Esse endereço é único, não podendo existir dois usuários com endereços iguais. O formato padrão dos endereços eletrônicos na internet é:

```
nome do usuário>@<provedor>.<natureza>.<país>  
Exemplo: renato@provedorX.com.br
```

E o que significa isso? A primeira sequência identifica o nome da caixa postal eletrônica do usuário, no caso, Renato. A segunda identifica o símbolo de arroba (@), que significa em, o provedor onde essa caixa postal eletrônica está localizada, provedorX, de natureza comercial (com) e o país onde ele atua (br); isso indica tratar-se de uma instituição comercial instalada no Brasil. Através do correio eletrônico, podemos realizar a troca de mensagens utilizando texto, imagens e arquivos de som.

b) Serviço: www. Outro serviço bastante difundido na internet é o *World Wide web* (WWW). O termo www ou Teia de Alcance Mundial vem do fato de que, se mapearmos as rotas de tráfego de dados e a sua distribuição, teremos uma espécie de Teia de Aranha que cobre todo o planeta.

A www, ou simplesmente *web*, foi criada por Tim Berners-Lee em 1989, nos laboratórios do CERN, como uma solução para os problemas de intercâmbio de informações entre os pesquisadores. Desde então, a tecnologia para construção de interfaces *web* vem sendo progressivamente incrementada, permitindo o desenvolvimento de aplicações cada vez mais complexas. O serviço www ou popularmente conhecida como *web*, marcou uma nova fase da internet, quando empresas passaram a se interessar e fazer uso mais intensivo dos recursos da rede. Foi a verdadeira popularização da rede, pois no princípio ela era utilizada apenas como instrumento de divulgação institucional e pouco a pouco passou a ser um canal de prestação de serviços. O uso da *web* pelas empresas transformou a internet de uma rede essencialmente acadêmica e experimental para uma rede comercial de serviços.

A www ou *web*, como é conhecida, é considerada a parte visual da internet.



O objetivo desta disciplina é entender o que é a *web*, para que se possa programar para ela. Através da *web* pode-se acessar hipertextos (documentos que podem conter textos, gráficos, imagens, áudio e vídeos). É também possível acessar grandes bases de dados e fazer transferências de arquivos. Atualmente, a *web* é um universo de informação onde se pode encontrar praticamente tudo. Nela podemos:

- fazer compras (comércio eletrônico);
- acessar a bolsa de valores;
- ler jornais e revistas;
- consultar e movimentar contas bancárias (*internet banking*);
- pagar serviços, água, luz, telefone, etc.;

- fazer reserva de hotel;
- visitar museus e galerias de artes;
- assistir a filmes;
- pesquisar assuntos diversos;
- ler livros;
- outras atividades *on-line*.

Documentos da *web* possuem a inovadora característica de ter “*links*” (ligações) que permitem o salto de uma parte para outra ou para outro documento. Os *links* (ligações) ou *hyperlinks* (ligações de hipertexto) nos possibilitam que as informações do hipertexto sejam lidas de forma não sequencial, ao contrário de um texto convencional, que sempre induz o leitor a fazer uma leitura linear e estejam localizadas em quaisquer computadores conectados à rede. O hipertexto nada mais é que um conjunto de informações compostas por textos, gráficos, fotografias, áudio e vídeo, associações entre essas informações, os *links* ou *hyperlinks*. Portanto, é mais fácil usar o hipertexto do que defini-lo.

Como se faz para acessar a *web* ou outro serviço qualquer da internet? Para isso precisamos de um *software* específico para essa tarefa, que é o *browser* (comumente conhecido como navegador) instalado na sua máquina. Tecnicamente o *browser* é um programa cliente que, geralmente, é utilizado para pesquisar e consultar informação disponibilizada em páginas *web* alojadas em servidores remotos também do tipo *web*. Os navegadores mais utilizados são o Microsoft Internet Explorer e o Mozilla Firefox.

Para acessar um *Site* temos que introduzir seu endereço. Assim como na vida real, quando queremos localizar uma pessoa ou encontrar um lugar, temos de saber onde os encontrar, ou seja, temos que saber seu endereço. A internet usa essa mesma filosofia. Todos os recursos disponíveis na *web* têm um endereço, que é único para cada *site* que se quer acessar. Esse endereço é chamado de URL (*Uniform Resource Locator*).

O formato padrão de URL é:

<protocolo><WWW.nome do domínio><diretório/página><recurso/página>
Exemplo: <http://www.ifpi.edu.br/index.html>

Além dos serviços de *e-mail* e *web*, temos possibilidade de realizar *download* e *upload*, manter a comunicação em tempo real através de *chat* ou vídeo conferência, entretenimento, entre outros.

Então, se você já tem um computador com o *browser* instalado, sabe o endereço que você quer acessar, só está falando a conexão com a internet. E como faço pra conectar meu computador na internet?

Para isso, é necessário fazer uma assinatura em um provedor de serviços da internet (ISP ou *Internet Service Provider*). Os ISPs são as instituições ou empresas que disponibilizam o acesso da internet a terceiros a partir de suas instalações. O acesso à internet pode ser do tipo discado (em desuso nos últimos tempos) ou por banda larga. O acesso discado é efetuado através de uma linha telefônica, e a visualização dos dados é realizada de forma lenta. Já o acesso por meio de banda larga é efetuado através de um provedor, cuja comunicação é efetuada via cabo de dados, satélite, rádio ou mesmo por linha telefônica com sinal banda larga, mais veloz que a discada. A figura a seguir demonstra o funcionamento de ISP.

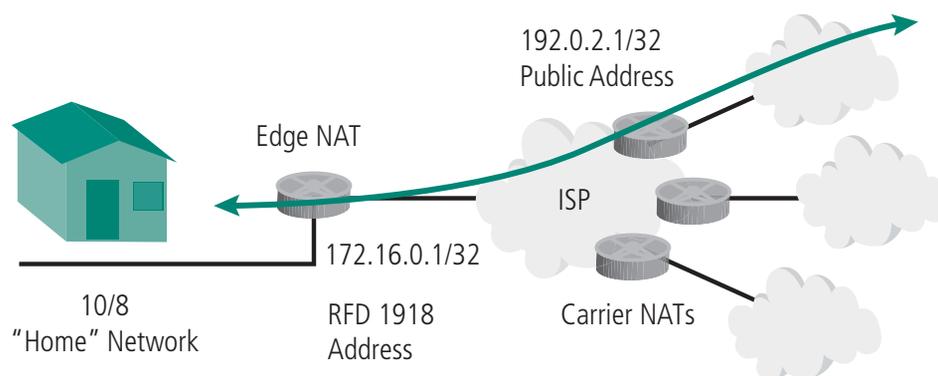


Figura 1.6: ISP – Provedor de Serviço da Internet

Fonte: Banco de imagens Google, 2011

Um conceito, que surgiu direcionada às empresas, com o advento da internet, foi o de **intranet**, que, tal como o próprio nome sugere, é uma rede interna, implantada por muitas empresas, cujos serviços estão disponíveis somente aos seus funcionários. Essas redes locais, de fácil implementação, assentam a sua base na tecnologia e soluções TCP/IP (*Transmission Control Protocol / Internet Protocol*). Outro conceito curioso é o de **extranet**, que pode ser considerada uma intranet que abriu as portas ao meio externo. A extranet possibilita o acesso à internet, da empresa ou instituição, por utilizadores remotos. Geralmente, são funcionários de outros escritórios, colaboradores, clientes, fornecedores ou outros utilizadores que mantêm relações comerciais ou sociais com essa empresa ou instituição.

1.5 Domínio na internet

Toda a informação que trafega na internet está armazenada em computadores. Todos esses computadores têm um endereço que, na terminologia técnica, se designa por endereço IP (*Internet Protocol*) e que indica sua localização na *web*. Para ter acesso à informação de um determinado computador, o internauta deverá, portanto, dirigir-se ao endereço específico. O endereço IP (*Internet Protocol*) é o responsável pelo endereçamento na internet; cada computador na internet tem que ter um endereço IP que deve ser único no Mundo. Ele é constituído por algarismos agrupados em quatro números separados por pontos, todos eles inferiores a 256; por exemplo, um endereço IP 200.224.124.66, sendo o protocolo de endereçamento o IPv4, ou seja, IP versão 4. Por ser difícil de memorizar cada número de IP, utiliza-se na internet o serviço de DNS (*Domain Name System*), que faz a tradução do endereço IP em nomes. Esse nome fica ligado ao um determinado endereço IP, que o internauta normalmente desconhece. Quando alguém digita um nome de domínio para acessar a *web*, entra em funcionamento um processo que faz a correspondência entre o nome e o endereço IP do computador no qual está hospedado o *site*.

Quem coordena o funcionamento de domínios na internet? A ICANN (*The Internet Corporation for Assigned Names and Numbers* ou Instituição para Designação de Nomes e Números da Internet), desde sua criação em outubro de 1998. É uma organização internacional sem fins lucrativos, responsável pelo controle das operações utilizadas pela internet; é ela quem coordena todos os registros de *sites*, além de também credenciar todas as empresas que prestam o serviço de registro, para que, assim, não existam dois domínios iguais no mundo. Então, para registrar um domínio, é necessário realizar uma busca prévia, verificando se tal domínio está ou não disponível.

No Brasil, a entidade que cuida desse procedimento é o Comitê Gestor da Internet (CGI.br). Já o órgão responsável por registros de domínios nacionais (.br) é o Registro.br. Essa organização determinou que os domínios nacionais (.br) devem conter no mínimo 2 (dois) e no máximo 26 (vinte e seis) caracteres. Os domínios não podem conter apenas números e não levam acentuação ou cedilha; já hifens são permitidos. A proteção do domínio é mundial e sua função básica de nome de domínio, bem como de uma marca, é identificar um produto, serviço ou atividade, distinguindo-o de seus concorrentes. Na hora de registrar um domínio, podemos determinar a natureza da organização, conforme exemplos do Quadro 1.1 a seguir.

Quadro 1.1: Natureza das organizações

.com	Comercial
.edu	Educacional
.gov	Governamental
.net	Rede
.info	Informação
.biz	Empresas, comércios, negócios
.org	Organizações
.eti	especialista em tecnologia da informação
.coop	Cooperativas
.ind	Indústria
.mil	Forças armadas

Fonte: Elaborado pelos autores

Outro cuidado que se deve ter na hora de registrar o domínio é com a região geográfica. O Quadro 1.2 demonstra a sigla que deve ser usada em cada país.

Quadro 1.2: Região geográfica

.br	Brasil
.ar	Argentina
.fr	França
.us	USA
.de	Alemanha
.uk	Reino Unido
.eu	União Europeia
.nl	Holanda
.cn	China
.it	Itália
.ch	Suíça

Fonte: Elaborado pelos autores

Após entender o que é o domínio, veremos a seguir alguns exemplos de domínios brasileiros na *web*. O acesso à *web* obedece a certas regras predefinidas, conforme o formato do padrão URL, já visto anteriormente.

Quadro 1.3: Endereços de sites

ENDEREÇO	DESCRIÇÃO
http://www.bol.com.br	Serviço de e-mail gratuito
http://www.uol.com.br	Portal de informações – Universo Online
http://www.americanas.com.br	Portal de comércio eletrônico
http://www.ifpi.edu.br	Instituto Federal de Educação, Ciência e Tecnologia do Piauí
http://cefetpi.nucleoad.net/etapi/	Escola Técnica Aberta do Piauí
http://www.mec.gov.br	Ministério da Educação
http://www.mct.gov.br	MCT – Ministério da Ciência e Tecnologia
http://www.mar.mil.br	Marinha do Brasil
http://www.cielo.ind.br	Indústria de Mecatrônica
http://colivre.coop.br	Cooperativa de Tecnologias Livres

Fonte: Elaborado pelos autores

Resumo

Nesta aula abordamos conceitos introdutórios sobre a *web*. Vimos o que é a internet e, por meio de datas, visualizamos a rápida evolução dessa rede de computadores, do surgimento até os dias atuais. Pudemos ter o conhecimento sobre essa supervia da informação, quem a controla, como ela funciona, quais são os serviços que podemos acessar por ela. Vimos também a importância do domínio na internet. Uma vez que já temos um embasamento teórico sobre a internet, estamos nos preparando para desenvolver para *web*.

Atividades de aprendizagem

1. O que é a internet e quem tem o seu controle?
2. Qual a importância dos protocolos na internet?
3. Cite duas datas importantes desde o surgimento da internet.
4. O que é um ISP ou Provedor de Serviços da Internet?
5. O que é um *browser*? Exemplifique.
6. Pode-se enviar um *e-mail* para um destinatário cujo computador esteja desligado? Explique essa situação.
7. Cite alguns serviços que estão disponíveis na internet.
8. Como fazer para acessar a internet?
9. Dê exemplo de cinco atividades que podem ser desenvolvidas na *web*.
10. Qual é o órgão responsável pelo registro de domínios no Brasil?

Aula 2 – Elaborando um *site*

Objetivos

Planejar um *site*.

Conhecer o desenvolvimento de um *site*.

Projetar um *site*.

Conhecer as fases na construção de *site*.

Saber administrar um *site* disponível na *web*.

Saber o que é o W3C.

2.1 Introdução ao *site*

Iremos agora entender o que é um *site*. Se vamos aprender a construir *site*, então precisamos saber o que é, como projetá-lo, conhecer cada etapa de sua construção. Detalharemos informações substanciais a respeito de *site*, no decorrer desta aula.

A palavra *site* em inglês tem exatamente o mesmo significado de sítio em português, pois ambas derivam do latim *situs* (lugar demarcado, local, posição). Entretanto, surgiu o termo *website* (às vezes *web site*) para designar um sítio virtual, um conjunto de páginas virtualmente localizado em algum ponto da *web*. Um *site*, na verdade, é um conjunto de páginas *web*, isto é, de hipertextos acessíveis geralmente pelo protocolo HTTP na internet.



Figura 2.1: Ilustração de sites

Fonte: Banco de Imagens Google, 2011

Outro conceito é o de *homepage*. E o que significa *homepage*? Muitas vezes, confundido com *site*, o termo *homepage* é designado para nomear a primeira página ou a página principal de um *site*. A *homepage* corresponderia à capa de um livro. Especificamente, na *web*, esse termo funciona como a página introdutória de um *site*, com explicações sobre o que será encontrado nas demais páginas. Ela tem o papel de índice de um *site* a ser visitado, ajudando os visitantes a ter uma visão geral dos conteúdos e serviços que o *site* contém e auxiliando na navegação, sendo, portanto, a primeira página visualizada quando o *browser* é aberto.

Todo *website* (*web site* ou simplesmente *site*) deve ser inicialmente planejado. Antes de criá-lo, deve-se identificar recursos financeiros, técnicos e humanos para o acompanhamento do projeto, o seu desenvolvimento, atualização e promoção. Portanto, o planejamento pode até representar mais trabalho no início, mas tornará o andamento do *site* bem mais fácil. Poderíamos comparar o planejamento de um *site* com a construção de uma casa, por exemplo, uma vez que há necessidade de uma planta para a sua construção. É praticamente impossível construir uma casa sem uma planta, pois o resultado pode não ser satisfatório e ela pode chegar a desmoronar. Assim também é o planejamento de um *site*; se for dada a atenção necessária à etapa de planejamento, evitar-se-á uma série de erros na sua elaboração.

2.2 Planejamento do *site*

No planejamento do *site*, deve ser bem definido o seu propósito, quem vai fazer uso dele e o que as pessoas irão fazer quando o visitarem.

Para dar início às essas etapas, primeiramente precisa-se definir o tipo de conteúdo que será apresentado no *site*. Existem diversos *sites* na *web*, com diversos tipos de conteúdos diferentes, nos quais são comumente encontram-se:

- informações pessoais;
- filmes;
- músicas,
- livros;
- veículos;
- jornais em versão digital;
- revistas em versão digital;
- empresas;
- instituições federais, estaduais, municipais;
- instituições filantrópicas;
- manuais, dicionários, legislações, em versão digital;
- comércio eletrônico (*e-commerce*) ou lojas virtuais;
- transações bancárias (*e-banking*), em versão digital;
- universidades, escolas técnicas, empresas particulares – modalidade, ensino a distância.

É necessário analisar a informação que será oferecida, ou seja, o conteúdo do *site*, como esse conteúdo pode ser acessado e como será exibido.

A figura abaixo apresenta um *site* com conteúdo não estruturado (com ambiente de informação não planejado) e outro com conteúdo estruturado (com ambiente de informação planejado).

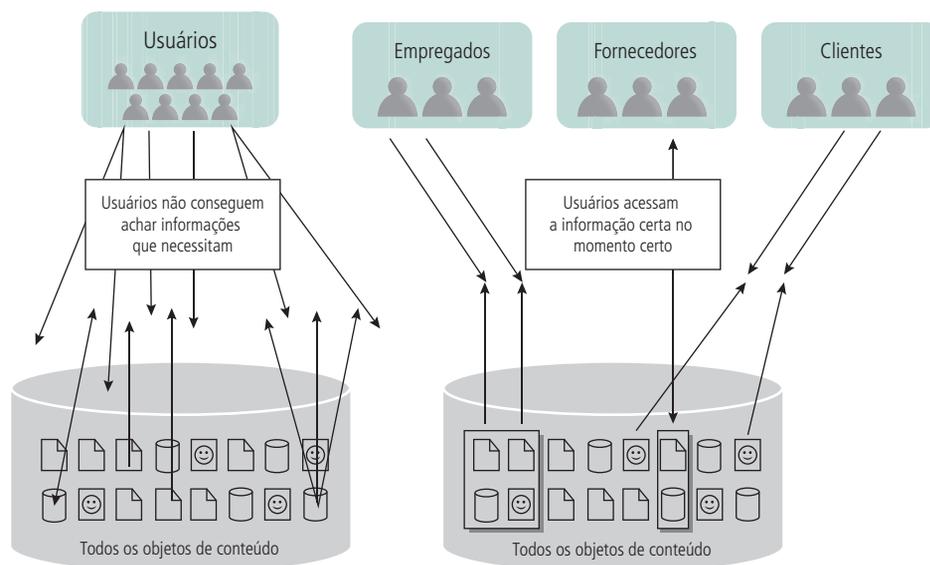


Figura 2.2: Ambientes de informação

Fonte: Rosenfeld e Morville (2002)

É visível o contraste mostrado pela figura. Percebe-se claramente que os usuários da esquerda têm dificuldade em acessar as informações, o que acaba gerando perda de tempo que acarretará na provável saída do *site*, pelo elevado grau de insatisfação por parte do usuário. Já na figura da direita, os usuários acessam o que procuram sem problema e sem demora. De que maneira o usuário pode ser afetado pelo mau planejamento e pela falta de estrutura no *site*, como visto no exemplo? Em *sites* sem planejamento, a desorganização acaba por gerar confusão e frustra o usuário, uma vez que ele não consegue encontrar o que procura, além de perder tempo. Os problemas causados pelo mau planejamento de *site*, em geral são:

- homepage* desorganizada;
- carregamento lento;
- inconsistência na navegação;
- arquitetura de informações pobre;
- outros problemas.

Em relação ao modo como o conteúdo deve ser exibido, as recomendações descritas a seguir são respeitadas pelos *sites* mais comumente visitados na atualidade:

- Uma boa navegabilidade permite que qualquer visitante possa se movimentar no *site* e que consiga identificar os conteúdos ou serviços pretendidos, sem dificuldade. É necessário, para isso, que todas as páginas

possuam *links* para seus diferentes conteúdos, que suas páginas sejam facilmente carregáveis pelo *browser*.

- Deve-se ter muita atenção na hora da escrita dos conteúdos; o ideal é procurar manter um padrão nas fontes, tamanhos e estilos nos textos do *site*. Outro ponto que merece cuidado é no contraste do texto e o fundo: normalmente evitam-se fundos escuros, porque isso dificulta a leitura e a impressão dos conteúdos, podendo representar um custo elevado para o usuário. O estilo de fontes sublinhadas não deve ser utilizado para que não se possa confundir com um *link*. Fontes em letras maiúsculas devem ser evitadas, pois utilizá-las na internet é o mesmo que “gritar”.
- É prática na internet, fazer *download* de algum *software* para os conteúdos que estão disponíveis no formato de arquivos PDF ou de áudio/vídeo. O *link* para o *download* do *software*, seja ele gratuito ou pago, deve ser disponibilizado pelo site.
- Testes sobre o *site*, especificamente de suas páginas, em diferentes navegadores, deverão ser realizados para que se possa assegurar ao usuário uma boa visualização. Os testes deverão ser realizados em navegadores diferentes, em versões diferentes, e deve-se utilizar os navegadores com maior popularidade, em suas versões mais atualizadas.
- O tempo para visualizar (carregar) um *site* deve ser reduzido. Ele está diretamente relacionado com o tamanho do *site* e com o tipo de conexão que o usuário *web* utiliza. Portanto, deve-se elaborá-lo prevendo sua visualização em conexões razoavelmente lentas, evitando-se inserir grandes gráficos e muitas imagens nas páginas *web*.
- Deve-se atrair um grande número de visitantes ao *site*. Para isso, inclui-se algo que o público-alvo gosta de ouvir, ver, comentar.
- Periodicamente, os conteúdos devem ser atualizados.

2.3 Desenvolvendo um *site*

Para realização do desenvolvimento de *sites* para uma empresa ou pessoas, é necessário estabelecer, junto com os clientes, suas finalidades, ideias, forma do *layout*, logomarca e escopo da diagramação da página (navegabilidade). Deve-se responder à seguinte pergunta: qual o objetivo do *site*? O principal objetivo

de toda a apresentação *web*, seja ela uma loja virtual ou um curso *on-line*, é a informação. O *site* deve ter informações claras e consistentes, que consigam fazer o usuário querer voltar a ele, que consigam prender sua atenção.

E como conseguir atrair esse usuário e ainda fazê-lo voltar ao *site*?

O *site* deve ter informações relacionadas à área, enquanto faz propaganda sutil no *layout*, sempre apresentando a logomarca. Um exemplo poderia ser a criação de uma revista eletrônica sobre informática, na qual o conteúdo do *site* não traria apenas o que a revista faz, ou como contatá-la, ou reportagens, mas também notícias sobre cursos na área, vagas no mercado de trabalho, fórum sobre temas variados, em que os usuários pudessem interagir entre si ou mesmo com autores de artigos da revista, um *blog* para informar sobre tendências, envio de *newsletters* dentre outras opções. Mas o que é *newsletter*? É um boletim informativo enviado por *e-mail* que reúne as notícias e informações diversas produzidas pelo *site*. Para receber o *newsletter* é preciso submeter o registro no próprio *site*. No final, o usuário acaba voltando a ele, seja para saber sobre um curso, uma vaga de trabalho, ou mesmo para ler o *blog*, e acaba fixando a marca do *site*, e até passando essa informação para outros usuários da internet. Assim, o *site* pode ser considerado bem-sucedido.



Blog ou *weblog* é o espaço virtual em que é feito o registro frequente de informações. Veja a seguir alguns *blogs* mais usados:

Blog Terra Brasil –
<http://blog.terra.com.br/>;

BliG – <http://blig.ig.com.br/>;

Pop Blog – <http://www.pop.com.br/popblog/>;

Blog-se – <http://www.blog-se.com.br/blog/>;

Blogger.com.br – <http://blogger.globo.com/index.jsp>;

My 1st blog - <http://my1blog.com>

Definidos os objetivos do *site*, é hora de pensar na forma de apresentação das informações. O ideal é agrupá-las separando-as em blocos organizados por tópicos ou assuntos principais, manter todos os conteúdos listados em tópicos e, posteriormente, definir o seu modo de navegação.

É no modo de navegação que se define o processo de navegabilidade do *site*. O usuário *web* precisa se movimentar nas páginas sem dificuldade, localizar as informações que quiser e ter facilidade em voltar a alguma posição já conhecida do *site*. Os modos de navegação podem ser classificados em: hierárquica, linear, linear com alternativas, combinação da linear com hierárquica, e estrutura de rede ou *web pure*.

- **Navegação hierárquica:** é um tipo de navegação em que os documentos *web* podem ser estruturados de maneira mais fácil e lógica. Na navegação lógica, o usuário descobre sem dificuldade nenhuma em que posição se encontra na hierarquia do *site*. Comumente, a página inicial fornece uma visão geral do conteúdo que está subordinado a ela, e ainda define quais os vínculos das páginas dos níveis inferiores na hierarquia do *site*. Procura-se manter de dois a três níveis no máximo na hierarquia *web*.

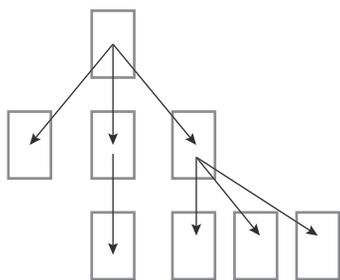


Figura 2.3: Navegação por hierarquia

Fonte: Elaborada pelos autores

- **Navegação linear:** é um tipo de navegação considerada rígida e limita quanto à liberdade de consultar informações do *site*. Nesse tipo de estrutura, a página principal do *site* é o local de vínculo com todas as outras páginas, ou seja, que levam de uma página a outra.

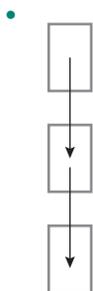


Figura 2.4: Navegação linear

Fonte: Elaborada pelos autores

- **Navegação linear com alternativas:** é uma navegação linear menos rígida, pois permite que o usuário *web* se desvie do caminho principal. A navegação linear com alternativas parte da página principal, mas as ramificações podem surgir em um nível mais baixo da estrutura, ou continuar se ramificando em níveis inferiores seguindo caminhos próprios até chegar a um fim.

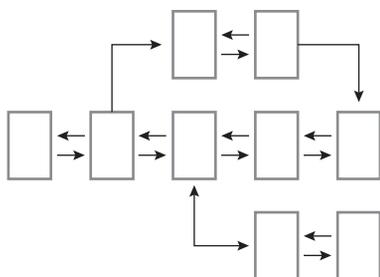


Figura 2.5: Navegação linear com alternativas

Fonte: Elaborada pelos autores

- **Navegação linear com hierarquia:** é fruto da combinação da estrutura linear e hierárquica. Esse tipo de página da *web* deve oferecer vínculos para que o usuário possa avançar, retroceder, retornar ao início e subir ou descer um nível na hierarquia.

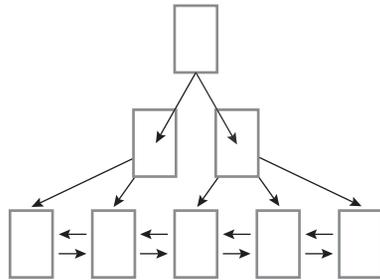


Figura 2.6: Navegação linear com hierarquia

Fonte: Elaborada pelos autores

- **Navegação de rede ou *web pure*:** esse tipo de navegação representa a própria navegação da *web*, pois todas as páginas possuem vínculos entre si, a navegação é aleatória. O usuário, a partir da *homepage*, pode ir e vir a qualquer lugar do *site*, sem dificuldade; por isso a denominação *web pure*.

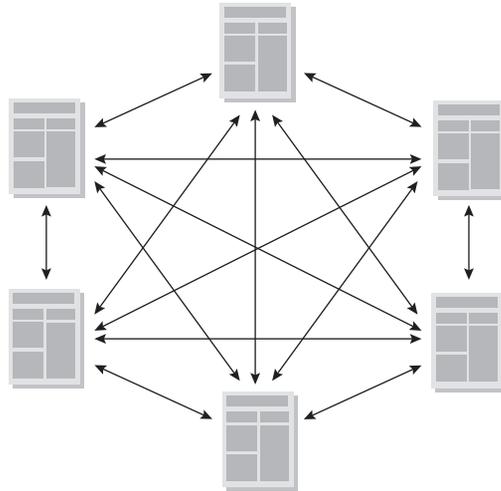


Figura 2.7: Navegação de rede ou *web pure*

Fonte: Elaborada pelos autores

Na continuação do desenvolvimento *web*, precisamos determinar o conteúdo que será apresentado em cada uma das páginas e criar os vínculos que possibilitem a navegação por essas páginas. Para isso, podemos usar o *storyboard*, o qual tem seu conceito emprestado do cinema, em que cada cena e cada tomada de câmera é esboçada na ordem em que ocorre no filme. O uso de *storyboard* ajuda a visualizar uma apresentação como um todo e a prever sua forma final, antes de efetivamente pôr em prática as ideias.

O *storyboard* é uma técnica de baixo custo, pois mudanças no papel ou em protótipos são fáceis de realizar, sendo mais difícil e custoso fazer essas mudanças diretamente no *site*, principalmente se ele já estiver implementado. Com o *storyboard* pode-se definir quais tópicos estarão disponíveis em cada página, os vínculos básicos, as imagens gráficas que serão usadas e onde elas deverão ser incluídas na apresentação *web*.

Na fase de desenvolvimento *web*, podemos também elaborar o organograma do *site*. Deve-se esboçar os *storyboards* e os respectivos organogramas em que eles estarão referenciados. Veja a seguir um modelo de organograma de *site*.

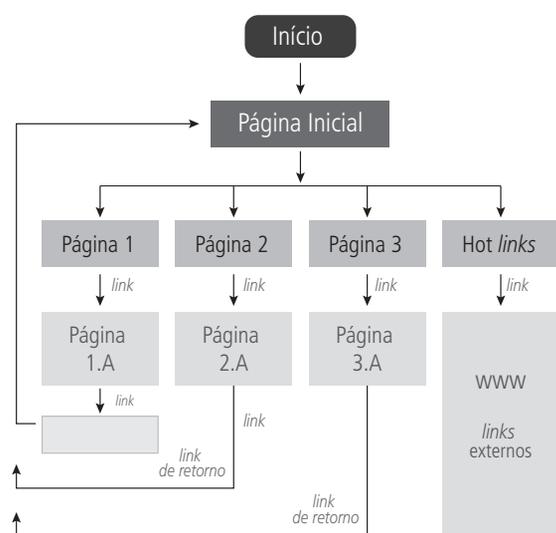


Figura 2.8: Organograma de um *site*

Fonte: Elaborada pelos autores

O modelo da Figura 2.8 demonstra a página principal, que em geral é uma apresentação e deve ser desenvolvida para que os usuários *web* conheçam o conteúdo do *site*. As páginas 1, 2 e 3 representam a continuidade da página principal. Cada uma delas, além do conteúdo, devem proporcionar ligações com outras páginas, dependendo do tipo de navegação definida. A quantidade de níveis de cada página depende do assunto que se quer tratar. Nesse caso, a página 1 possui um subnível a mais que as páginas 2 e 3.

2.4 Projeto do *site*

Na fase de projeto do *site*, as ideias saem do papel e ganham forma, cor e, se necessário, áudio e vídeo. Nessa etapa é feita a diagramação *web*, em que são definidos os conteúdos e formas do *site* e é definida a equipe de desenvolvimento. É nessa fase também que é feito o *web design* do *site*.

A diagramação *web* é o momento da disposição de elementos que compõem a página do *site*. E que elementos são esses? São as imagens, textos, vídeos. Eles devem ser dispostos no *site* de forma que sua visualização seja agradável ao usuário *web*. Nessa diagramação também são definidos: o tamanho das fontes, a localização das imagens, dos vídeos, a localização do conteúdo, *links* para a navegação, etc.

Ainda na etapa de diagramação, deve-se ter cuidado no uso de imagens em um *site*, elas em geral deverão estar nos formatos .GIF (*Graphics Interchange Format*) e .JPG ou JPEG (*Joint Photographics Experts Group*). A diferença básica entre esses formatos de imagens é que os arquivos JPG ou JPEG são mais usados em fotos ou gráficos grandes, e apresentam pequenas perdas na qualidade na resolução dessas imagens. Esse formato suporta até 16,7 milhões de cores. Já o GIF é usado em figuras com transparências e pequenos gráficos. Esse formato, de no máximo 256 cores, apresenta maior perda na qualidade de imagens. Quanto à visualização das imagens no vídeo, não se necessita de alta resolução, sendo aconselhável a densidade de 72 dpi (abreviatura para '*dots per inch*' – pontos por polegada, o que resulta em imagens mais reduzidas em *bytes* que facilitam o tráfego pela rede.

É nessa fase do projeto, em que deve se estabelecer qual a melhor linguagem de programação para sua execução. Poderíamos citar como linguagens utilizadas para desenvolvimento *web*: HTML, XHTML, VBScript, JavaScript, Perl, ASP, PHP, XML, JSP, ASP.NET. AJAX. A HTML (*Hipertext Markup Language*), que significa Linguagem de Marcação de Hipertexto, é uma linguagem dedicada à construção de páginas *web*, tem como finalidade básica formatar o texto exibido e criar ligações entre as páginas do *site*, permitindo dessa forma o surgimento do hipertexto, que são documentos com *links* (ligações) entre si. O HTML será a linguagem utilizada nesta disciplina de Programação para *Web*.

Há também nessa fase de projeto do *site* a preocupação com o conteúdo e forma. Deve-se seguir as recomendações sobre conteúdo dos *sites* mais comumente visitados da atualidade, descritas anteriormente no início desta aula. Quanto à forma, o *design* do *site* (estrutura de navegação, projeto gráfico) tem que ser definido nesse momento. E um detalhe no projeto *web* que não pode passar despercebido é a constante atualização do *site*. Uma vez publicado, ele precisa ser atualizado periodicamente, pois se sabe que aqueles que ficam muito tempo sem ser atualizados são fadados ao fracasso. Deve-se estabelecer qual a melhor linguagem de programação para execução do projeto

Para finalizar a fase do projeto *web*, é necessário definir a equipe de desenvolvimento. porque o *site*, por mais simples que seja, precisa de alguém para efetuar essa tarefa, mesmo que seja uma única pessoa, a qual exercerá todas as funções necessários para o desenvolvimento. As equipes de desenvolvimento para *web* são normalmente menores que as de desenvolvimento de *software* tradicional. No desenvolvimento para a *web*, elas são divididas em grupos multidisciplinares, que construirão diferentes seções do aplicativo para a *web*, mas em geral irão trabalhar em problemas similares. A construção de um *site* exige uma equipe composta por:

- **Provedor de conteúdo:** é o responsável pela coleta, geração, correção e atualização de conteúdos.
- **Web publisher:** faz a conexão entre o pessoal técnico que constrói o aplicativo e o pessoal não técnico que desenvolve o conteúdo.
- **Web designer:** é um profissional que tende à multidisciplinaridade, agregando áreas como a programação, usabilidade, acessibilidade.
- **Especialista de suporte:** é aquele que dá continuidade no suporte ao aplicativo para a *web*, responsável pelas correções, adaptações e melhorias no *site*.
- **Webmaster:** é um profissional que tem por finalidade a tarefa de instalar, dar manutenção, criar e realizar a gestão de um determinado *site*. É a pessoa que gerencia e cuida do funcionamento diário de um *site*, da implementação de procedimentos de segurança e direitos de acesso, medição e análise de tráfego.
- **Web developer:** é o profissional que cuida da lógica de um *site*; é ele quem faz as ligações com banco de dados e a interação com usuário via programação. Ele desenvolve as aplicações que serão executadas sob a interface dos *sites* por meio de tecnologias e ferramentas de programação.
- **Analistas de sistemas:** é o responsável pela organização dos dados e do banco de dados que o *site* pode usar.

2.5 Fases na construção do *site*

A construção de um *site* requer, por parte de sua equipe de desenvolvimento, uma série de fases e procedimentos que devem ser seguidos para que ele seja construído no tempo hábil, de acordo com as solicitações do cliente. Pode-se resumir a construção de um *site* em quatro fases, descritas a seguir:

1ª fase – Organização da informação: é a fase de organizar as informações que irão compor o *site* e adequar documentos e material para serem disponibilizados na *web* no formato de hipertexto. Os dados precisam ser acessados e atualizados com facilidade. As imagens de alta qualidade não são recomendadas nessa fase.

2ª fase – Modelo do *site*: é a fase de criação do *site*, quando acontecem as reuniões com toda a equipe de desenvolvimento. Nessa fase é definida e criada a estrutura de navegação e o projeto gráfico do *site*.

3ª fase – Montagem: durante a montagem do *site*, cada membro da equipe de desenvolvimento trabalhará em sua área específica, ou seja, irá montar sua parte.

4ª fase – Testes: são responsáveis pelos últimos ajustes do *site*; é quando se testa a navegação de toda a aplicação *web*, verificando os *links* dos textos e das imagens. Testes são feitos em diferentes plataformas e conexões, em diferentes navegadores.

Com o *site* devidamente construído e aprovado pelo cliente, será colocado no ar em seu próprio domínio.

2.6 Administração do *site*

Um *site* funcionando a pleno vapor, ou seja, disponível na *web* com vários acessos diários, precisa de uma administração. Essa função de administrador cabe ao *webmaster*; são suas atribuições verificar com frequência o funcionamento do *site*, a manutenção, o gerenciamento da infraestrutura necessária para viabilizar os acessos a ele, a divulgação, estatística, checagem dos *links*; portanto, ele é o responsável absoluto pelo *site*. Seu contato via endereço eletrônico deve estar disponível na *homepage* e em outras páginas do *site*, para que se possa contatá-lo sempre que for necessário.

É na administração do *site* que se faz a sua divulgação. Porque não adianta ter um *site* bem elaborado e disponível na *web*, se não for feita a sua divul-

gação. O *webmaster* deve utilizar todos os recursos disponíveis na *web* para conseguir alcançar um número significativo de acessos. As principais formas de divulgação *on-line* são:

- Incluir o *site* no cadastro dos motores de buscas mais populares da internet, sejam eles nacionais ou internacionais (Google, Cadê, Yahoo e outros). Dessa forma, qualquer usuário poderá encontrar as informações sobre ele sem dificuldade.
- Utilizar *banners* (tarja com informações promocionais sobre um *site*) estrategicamente localizados em páginas de *sites* que vendem espaço publicitário.

O envio de *e-mail* com divulgação do *site* deve ser evitado, pois é uma prática não recomendada na *web*.



O *webmaster* deve dispor de ferramentas para recolher informações estatísticas, para que assim possa definir o perfil do usuário e atualizar o *site* visando ao aumento do número de acesso. Ele pode, por exemplo, procurar saber quais as páginas mais acessadas, as menos acessadas, o número de visitas diárias, horários mais e menos acessados, dentre outras informações.

Na administração do *site*, a checagem de *links* ocupa um importante papel; portanto, é importante estabelecer uma rotina periódica para checagem de *links* internos e externos. Já existe software específico para essa tarefa. O *webmaster* deve diferenciar os *links* e torná-los fáceis de visualizar; não devendo usar instruções como “**Clique aqui**”, como um nome de *link*; *links* coloridos devem ser usados para indicar os que já foram visitados e não visitados; a palavra “*link*” não deve ser usada para indicar um *link* existente na página; indicar a presença do *link* com fontes com sublinhado e cor azul. Quando um *link* estiver vinculado a um arquivo PDF, acionar um equipamento de áudio ou vídeo, acionar um aplicativo de mensagens de *e-mail* ou acionar um outro aplicativo qualquer, ele deve explicar sucintamente o que acontecerá quando for clicado.

A organização de arquivos e diretórios também é uma atribuição do administrador do *site*. A estruturação da informação do *site* pode seguir o formato de módulos e submódulos e, conseqüentemente, irá gerar uma estrutura de diretórios e subdiretórios a serem administrados pelo *webmaster*.

2.7 W3C

O *World Wide Web Consortium* ou Consórcio *World Wide Web* (W3C), comunidade internacional que desenvolve padrões com o objetivo de garantir o crescimento da *web*, foi fundado por Tim Berners-Lee em 1994.

O W3C atualmente conta com a participação ativa de cerca de 500 membros e desenvolve tecnologias que irão se tornar padrões da *web* para a criação e a interpretação dos conteúdos, ou seja, faz uma padronização de tecnologias para a *web*. Apresentações *web* desenvolvidas seguindo os padrões W3C podem ser acessadas e visualizadas por qualquer pessoa ou tecnologia, independentemente de *hardware*, *software* ou plataforma utilizada, isso tudo de forma rápida e compatível com os novos padrões e tecnologias que possam surgir com a evolução da internet.

Seguindo as recomendações do W3C, o *site* passa a ter um código limpo e bem estruturado. Portanto, toda equipe de desenvolvimento *web* deve respeitar e seguir os padrões de acessibilidade do W3C, pois de outro modo poderá impor barreiras tecnológicas a diversas pessoas, desestimulando e até mesmo impedindo o acesso a suas páginas. O W3C defende o uso de tecnologias abertas, criando dessa forma aplicações intercomunicáveis e interoperáveis.

O W3C iniciou suas operações no Brasil em 1º de novembro de 2007. Antes de chegar ao país, a entidade já possuía escritórios na Austrália, China, Alemanha, Áustria, Finlândia, Grécia, Hungria, Índia, Israel, Itália, Coreia, Marrocos, África do Sul, Espanha, Suécia, Reino Unido e Irlanda. A presença do W3C no Brasil visa estimular o mercado brasileiro a adotar padrões internacionais e abertos no desenvolvimento de aplicações voltadas para a *web*. A publicação em português das principais recomendações do W3C e a criação de um fórum amplo para a discussão de padrões para *web* estão entre as atividades do consórcio no Brasil.

Resumo

Ao final desta aula, aprendemos muito sobre *site*, desde sua elaboração, surgimento da ideia, até a sua administração. Conhecemos as instruções sobre como dispor as informações no *site*, como montar seus conteúdos. Detalhamos seu desenvolvimento, chegando à sua construção. Aprendemos ainda alguns detalhes sobre o W3C, o consórcio internacional que desenvolve os padrões para *web*. Conhecemos o HTML (*Hipertext Markup Language*), que significa Linguagem de Marcação de Hipertexto, linguagem escolhida para construção de páginas *web*, o qual será visto nas próximas aulas.

Atividades de aprendizagem

1. O que é um *site*?
2. O que é *homepage*?
3. Cite três recomendações seguidas pelos *sites* mais comumente visitados, no que se refere ao modo de exibição de seus conteúdos.
4. O que é um *blog*?
5. Quais as modalidades de navegação de *site* disponíveis para *web*?
6. Qual a finalidade de um *storyboard*?
7. O que significa diagramar um *site*?
8. Quais profissionais englobam a equipe de desenvolvimento para *web*?
9. Qual membro da equipe de desenvolvimento *web* é o responsável pela administração do *site*?
10. O que é o W3C?

Aula 3 – Introdução à HTML

Objetivos

Conhecer conceitos básicos da linguagem HTML

Compreender a estrutura básica de um documento HTML

Saber o que é uma *tag*.

Conhecer as *tags* introdutórias da linguagem HTML.

Começar a programar para *web* utilizando HTML.

3.1 Introdução à linguagem de marcação de hipertexto – HTML

Vamos, a partir deste ponto, aprender a programação para *web* e elaborar nossa apresentação utilizando uma linguagem de marcação de hipertexto (*Hypertext Markup Language*), ou seja, o HTML. Mas o que é HTML? Quando e como posso utilizá-lo?

O HTML é uma linguagem, como o próprio nome diz, de marcação de texto; ela é simples, composta por marcações de formatação e diagramação de informações como textos, imagens, sons e também possibilita a inclusão de hipertexto. Hipertexto? Os hipertextos são referências que uma página pode fazer para si mesma ou para outros documentos. É esse diferencial que torna o HTML tão popular na *web*. Atualmente, existem muitas linguagens de programação para *web*, mas o HTML continua entre as mais populares; encontra-se na sua provável última versão, a 4.01, segundo o W3C, datada de 24 de dezembro de 1999.

Para que se possa visualizar um documento HTML, é necessário entender uma série de passos, conforme descritos a seguir:

Passo 1: Com um computador conectado à internet, é necessário ter, entre seus aplicativos instalados, um *browser* (navegador) para visualizar as páginas HTML.

Passo 2: O navegador é direcionado a um servidor da *web*; então ele solicita uma página.

Passo 3: O servidor da internet, responde à solicitação do *browser* (navegador) e envia a página de volta a ele.

Passo 4: A página HTML, então, é visualizada no *browser* do computador conectado à internet.

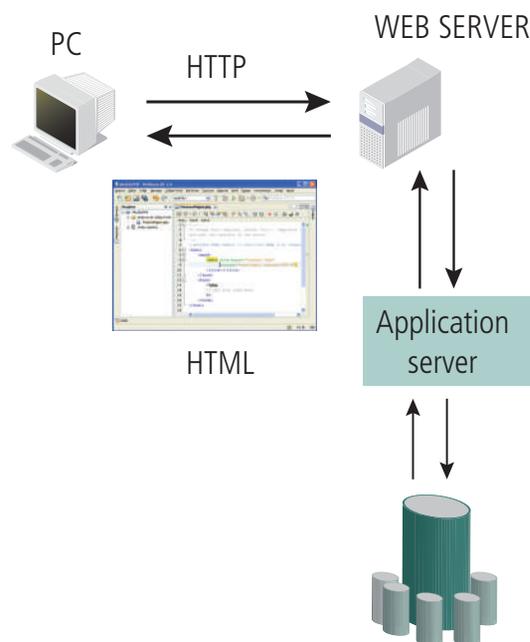


Figura 3.1: Imagem ilustrativa do funcionamento do protocolo HTTP

Fonte: Banco de Imagens Google, 2011

Um documento HTML é um arquivo de texto comum (ASCII), que tem como extensão .htm ou .html. Para que ele seja visualizado, basta que seja aberto em qualquer *browser*, pois se trata de uma linguagem interpretada. Então, para criarmos uma página para *web*, precisamos ter conhecimentos básicos da linguagem HTML de um editor de textos para gerar os códigos-fonte, e um *browser* que possa interpretar o código e visualizar as páginas.

3.2 Primeira página em HTML

A linguagem HTML possui seus comandos de feitos com *tags* (etiquetas) que equivalem aos comandos de formatação da linguagem. Uma *tag*, em HTML, não é mais do que marcas padrões utilizadas para fazer as indicações necessárias ao *browser*. Todo documento HTML apresenta as *tags* cercadas por um sinal de menor ('<') e maior ('>').

Existem dois tipos de *tags*. As compostas por uma *tag* de abertura e outra de fechamento, como por exemplo:

<etiqueta> </etiqueta>

ou

<tag> </tag>

```
<INICIO – Estrutura Principal>
  <CABEÇALHO>
    <TÍTULO> Título do Documento </TÍTULO>
  </CABEÇALHO>
  <CORPO DO DOCUMENTO>
    Texto, imagem, links, outras tags...
  </CORPO DO DOCUMENTO>
</FIM – Estrutura Principal>
```

E as *tags* mais simples, em que se usa apenas uma única *tag*:

</etiqueta> ou </tag>

```
1 <HTML>
2 <HEAD> <TITLE> Primeira Página em HTML </TITLE>
3 </HEAD>
4 <BODY>
5   Nosso texto, imagens, link, tags
6 </BODY>
7 </HTML>
```

As *tags* HTML não têm diferença entre maiúsculas e minúsculas, ou seja, tanto faz escrever <HTML>, <html>, <Html>, <HTml>, <HTMLI>, etc.

Para elaborar um programa em HTML é preciso saber que ele é dividido em três partes: a estrutura principal, o cabeçalho e o corpo do documento. A estrutura básica em HTML é:

```
<INICIO – Estrutura Principal>
  <CABEÇALHO>
    <TÍTULO> Título do Documento </TÍTULO>
  </CABEÇALHO>
  <CORPO DO DOCUMENTO>
    Texto, imagem, links, outras tags...
  </CORPO DO DOCUMENTO>
</FIM – Estrutura Principal>
```

Figura 3.2: Estrutura básica em HTML

Fonte: Elaborada pelos autores

A seguir, um exemplo que ilustra a utilização das *tags* estruturais de um documento HTML:

```
1 <HTML>
2 <HEAD> <TITLE> Primeira Página em HTML </TITLE>
3 </HEAD>
4 <BODY>
5     Nosso texto, imagens, link, tags
6 </BODY>
7 </HTML>
```

Figura 3.3: Exemplo de tags estruturais em HTML

Fonte: Elaborada pelos autores

A primeira *tag* de um documento HTML, como se observou na Figura 3.3, é `<HTML>`; é ela que informa ao *browser* que teve início a partir daquele ponto a página *web*, e sua correspondente `</HTML>` finalizando o mesmo documento.

Dentro das *tags* iniciais temos duas seções, compostas pelas *tags* `<HEAD>... </HEAD>` e `<BODY>... </BODY>`.

A `<HEAD>... </HEAD>` indicam o cabeçalho do documento e contêm as informações de configuração da página. Entre as informações importantes sobre a página está o título, que corresponde às *tags* `<TITLE>... </TITLE >`. O texto colocado como título da página será apresentado na barra de título da janela do *browser*. O título tem um papel importante na elaboração do *site*, porque todo documento *web* deve ter um título, que é referenciado em buscas pela internet, criando assim uma identidade para o documento.

As *tags* `<BODY>... </BODY>` contêm todo o conteúdo do *site*. Além de textos, imagens, *links*, essas *tags* ainda suportam a inserção de outras *tags*, que serão vistas nas próximas seções. A importância dessas *tags* é tão ampla que, a partir da versão 3.0 da HTML, elas passaram a admitir atributos para definir as cores para o texto, para os *links*, para o fundo das páginas e até para a imagem de fundo (marca d'água) da página. Veja a seguir os atributos da *tag* `<BODY>`:

- **BACKGROUND:** imagem de fundo da página, ou seja, uma marca d'água. Para inserir uma imagem como pano de fundo de uma página HTML, teríamos que usar o comando `< BODY BACKGROUND= "imagem_fundo.jpg">... </BODY>`;
- **TEXT:** cor do texto da página *web*; a cor padrão é preta. Para definir a cor do texto, usaríamos o seguinte comando HTML, `<BODY TEXT="#CC0000">... </BODY>`;

- **LINK:** cor dos *links*, a cor padrão é azul. Por exemplo: `< BODY LINK="#0000FF">... </BODY>;`
- **ALINK:** cor dos *links* clicados; a cor padrão é vermelha. Por exemplo: `<BODY ALINK="#000000">... </BODY>;`
- **VLINK:** cor dos *links* depois de visitados; a cor padrão é azul escuro ou roxo. Por exemplo: `<BODY VLINK="#FFDB03">... </BODY>;`
- **BGCOLOR:** cor de fundo da página; caso não venha especificado, esse atributo geralmente irá mostrar a cor padrão, branca ou cinza. Por exemplo: `<BODY BGCOLOR="#000000">... </BODY>;`

As cores na linguagem HTML, merecem um destaque especial. Elas podem ser definidas de duas formas:

1. Pelo nome, são 16 cores disponíveis: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow.
2. Pela numeração RGB. A sigla RGB vem de Red Green Blue, a numeração RGB possui seis dígitos; os primeiros dois dígitos correspondem à parcela de cor vermelha que será utilizada para compor a cor, os dois centrais correspondem à parcela de verde e os últimos a azul. Essa quantidade de cores vem definida em hexadecimal e tem sua variação entre 0 e 255 (00a FF). Por exemplo: `bgcolor= "#00FFFF"`. A seguir temos os códigos RGB que podem ser utilizado na linguagem HTML.

000000	000033	000066	000099	0000CC	0000FF
003300	003333	003366	003399	0033CC	0033FF
006600	006633	006666	006699	0066CC	0066FF
009900	009933	009966	009999	0099CC	0099FF
00CC00	00CC33	00CC66	00CC99	00CCCC	00CCFF
00FF00	00FF33	00FF66	00FF99	00FFCC	00FFFF
330000	330033	330066	330099	3300CC	3300FF
333300	333333	333366	333399	3333CC	3333FF
336600	336633	336666	336699	3366CC	3366FF
339900	339933	339966	339999	3399CC	3399FF
33CC00	33CC33	33CC66	33CC99	33CCCC	33CCFF
33FF00	33FF33	33FF66	33FF99	33FFCC	33FFFF

Figura 3.4: Codificação RGB

Fonte: Elaborada pelos autores

A seguir, temos um exemplo que ilustra a utilização do atributo de cor de fundo (`BGCOLOR = "numero ou nome da cor"`) e a numeração RGB, em HTML.

```
<HTML>
<HEAD> <TITLE> Este documento tem a cor de fundo preta </TITLE> </HEAD>
<BODY BGCOLOR="#000000">
.....
</BODY>
</HTML>
```

Figura 3.5: Exemplo BGCOLOR em HTML

Fonte: Elaborada pelos autores



As *tags* do HTML não são suportadas por todos os *browsers*, pois esses *softwares* não as interpretam de maneira idêntica. Por isso mesmo, deve-se testar um documento HTML em mais de um navegador para conferir o resultado do *site*.

Utilizamos como *browser* padrão das nossas aulas o Mozilla Firefox, por se tratar de um software livre e de grande aceitação pelos usuários *web*. Outro ponto importante a ser observado é, quando se for salvar os documentos HTML, procurar salvá-los utilizando a opção **salvar como**; isso independe do seu editor de texto, e nomeie o mesmo com a extensão **.htm** ou **.html**. Posteriormente, é só visualizar o resultado no navegador.

3.3 Estudo de *tags* em HTML

A *tag* **
** corresponde a uma das mais simples, em que se usa apenas uma única *tag*, quando queremos mudar de linha, ou fazer uma quebra de linha em algum ponto do documento HTML.

```
<HTML>
  <HEAD> <TITLE> Aprendendo a tag BR </TITLE> </HEAD>
  <BODY>
    Aprendendo a tag BR – LINHA 1 <BR>
    Aprendendo a tag BR – LINHA 2 <BR>
    Aprendendo a tag BR – LINHA 3 <BR>
  </BODY>
</HTML>
```

**Figura 3.6: Exemplo de *tag*
 em HTML**

Fonte: Elaborada pelos autores

Resultado do código anterior, usando a *tag* **
**:

Aprendendo a tag BR – LINHA 1
Aprendendo a tag BR – LINHA 2
Aprendendo a tag BR – LINHA 3

**Figura 3.7: Imagem do resultado da *tag*
**

Fonte: Elaborada pelos autores

As tags `<P> ... </P>` são usadas para dividir o texto em parágrafos, conforme mostra o exemplo a seguir:

```
<HTML>
  <HEAD> <TITLE> Aprendendo a tag BR </TITLE> </HEAD>
  <BODY>
    <P> Aprendendo a tag P – PARÁGRAFO 1 </P>
    <P> Aprendendo a tag P – PARÁGRAFO 2 </P>
    <P> Aprendendo a tag P – PARÁGRAFO 3 </P>
  </BODY>
</HTML>
```

Figura 3.8: Exemplo tag `<P>` em HTML

Fonte: Elaborada pelos autores

Resultado do código anterior, usando a tag `<P>`:

Aprendendo a tag P – PARÁGRAFO 1

Aprendendo a tag P – PARÁGRAFO 2

Aprendendo a tag P – PARÁGRAFO 3

Figura 3.9: Imagem do resultado da tag `<P>`

Fonte: Elaborada pelos autores

A tag `<P>` possui o atributo de alinhamento representado pela propriedade `<ALIGN>`. O texto, então, pode vir alinhado ao centro (**CENTER**), alinhado à direita (**RIGHT**) ou à esquerda ou padrão (**LEFT**). As tags `<P ALIGN=CENTER> ... </P>`, `<P ALIGN=RIGHT> ... </P>` e `<P ALIGN=LEFT> ... </P>` são usadas para alinhar os parágrafos do texto, conforme mostra o exemplo a seguir:

```
<HTML>
  <HEAD> <TITLE> Aprendendo a tag BR </TITLE> </HEAD>
  <BODY>
    <P ALIGN=CENTER> Aprendendo a tag P – PARÁGRAFO 1 </P>
    <P ALIGN=RIGHT> Aprendendo a tag P – PARÁGRAFO 2 </P>
    <P ALIGN=LEFT> Aprendendo a tag P – PARÁGRAFO 3 </P>
  </BODY>
</HTML>
```

Figura 3.10: Exemplo de tag `<P>` com a propriedade `ALIGN` em HTML

Fonte: Elaborada pelos autores

Resultado do código anterior, usando as tags `<P ALIGN=CENTER>... </P>`, `<P ALIGN=RIGHT> ... </P>` e `<P ALIGN=LEFT> ... </P>`:

Aprendendo a tag P – PARÁGRAFO 1

Aprendendo a tag P – PARÁGRAFO 2

Aprendendo a tag P – PARÁGRAFO 3

Figura 3.11: Imagem do resultado da tag `<P>` com ALIGN

Fonte: Elaborada pelos autores

Pode-se usar as tags HTML para formatar frases, palavras ou letras da página web. As formatações permitidas num documento HTML são: negrito, itálico, sublinhado, sobrescrito, subscrito; também, é permitido que o texto deslize sobre a página web. As tags `.. ` para formatação do texto em negrito, as tags `<I>.. </I>` para formatação do texto em itálico, as tags `<U>.. </U>` para formatação do texto em sublinhado, as tags `^{..}` para formatação do texto em sobrescrito, as tags `_{..}` para formatação do texto em subscrito. E as tags `<MARQUEE>.. </MARQUEE>` para fazer com que o texto deslize sobre a tela. A seguir temos um exemplo do uso das tags de formatação de texto:

```
<HTML>
  <HEAD> <TITLE> Aprendendo a tag BR </TITLE> </HEAD>
  <BODY>
    <B> Texto em Negrito </B>
    <I> Texto em Itálico </I>
    <U> Texto Sublinhado </U>
    <SUP> Texto Sobrescrito </SUP>
    <SUB> Texto Subscrito </SUB>
    <MARQUEE> Texto Deslizante </MARQUEE>
  </HTML>
```

Figura 3.12: Exemplo de tags de formatação de texto em HTML

Fonte: Elaborada pelos autores

Resultado do código anterior, usando as tags de formatação de texto:

Texto em Negrito	<i>Texto em Itálico</i>	<u>Texto Sublinhado</u>	Texto Sobrescrito	Texto Subscrito
				Texto Deslizante

Figura 3.13: Imagem do documento HTML usando formatação de texto

Fonte: Elaborada pelos autores

Em HTML, para organizar o texto, pode-se usar as listas, especialmente quando se quer dividi-lo em tópicos numerados ou não, ou num menu. O HTML nos permite criar listas de diversos tipos e com diferentes formatações para apresentar dados. Temos, portanto, a possibilidade de criar em HTML as listas ordenadas e as não ordenadas.

A tag **** (de *unorderedlist*) deve delimitar do primeiro até o último item da lista não ordenada. E cada item da lista não ordenada deve ser precedido da tag **** (de *line item*), conforme nos mostra o exemplo a seguir:

```
<HTML>
  <HEAD> <TITLE> Aprendendo sobre LISTAS </TITLE> </HEAD>
  <BODY>
    <UL>
      <LI> LISTA - ITEM 1
      <LI> LISTA - ITEM 2
      <LI> LISTA - ITEM 3
      <LI> LISTA - ITEM 4
    </UL>
  </BODY>
</HTML>
```

Figura 3.14: Exemplo de tags (UL) em HTML

Fonte: Elaborada pelos autores

Resultado do código anterior, usando as tags para elaborar as listas não ordenadas:

- LISTA - ITEM 1
- LISTA - ITEM 2
- LISTA - ITEM 3
- LISTA - ITEM 4

**Figura 3.15: Imagem do resultado da tag **

Fonte: Elaborada pelos autores

Nas listas não ordenadas em HTML, pode-se utilizar a tag **** para gerar sublistas, ou seja, listas que são elaboradas a partir de um item de uma lista primeira. Analise o código a seguir.

```

<HTML>
  <HEAD> <TITLE> Aprendendo sobre LISTAS </TITLE> </HEAD>
  <BODY>
    <UL>
      <LI> LISTA1 – ITEM 1, SUBLISTA 1
        <UL>
          <LI> LISTA1 – ITEM 1, SUBLISTA 1
          <LI> LISTA1 – ITEM 2, SUBLISTA 1
        </UL>
      <LI> LISTA2 – ITEM 2, SUBLISTA 2
        <UL>
          <LI> LISTA2 – ITEM 1, SUBLISTA 2
          <LI> LISTA2 – ITEM 2, SUBLISTA 2
        </UL>
    </UL>
  </BODY>
</HTML>

```

Figura 3.16: Outro exemplo de tag em HTML

Fonte: Elaborada pelos autores

Resultado do código anterior, usando as *tags* para elaborar as listas não ordenadas e sublists não ordenadas:

- LISTA1 – ITEM 1, SUBLISTA 1
 - LISTA1 – ITEM 1, SUBLISTA 1
 - LISTA1 – ITEM 2, SUBLISTA 1
- LISTA2 – ITEM 2, SUBLISTA 2
 - LISTA2 – ITEM 1, SUBLISTA 2
 - LISTA2 – ITEM 2, SUBLISTA 2

Figura 3.17: Imagem do resultado da tag

Fonte: Elaborada pelos autores

Para gerar listas ordenadas usando a tag **** (de *ordered list*), ele deve delimitar do primeiro até o último item da lista ordenada. E cada item da lista ordenada deve ser precedido da tag **** (de *line item*). A lista ordenada pode ser personalizada com os atributos TYPE, START ou VALUE. O atributo TYPE indica o tipo de lista ordenada que se quer utilizar; ele pode assumir cinco valores diferentes, sejam eles:

- **Type= "1"**: pode-se numerar a lista com (1, 2, 3, 4..., etc.); esse é o modo padrão;
- **Type= "a"**: pode-se numerar a lista com letras minúsculas (a, b, c, d... etc.);
- **Type= "A"**: pode-se numerar a lista com letras maiúsculas (A, B, C, D... etc.);
- **Type= "i"**: pode-se numerar a lista com numeração romana minúscula (i, ii, iii, iv... etc.);

- **Type= "I"**: pode-se numerar a lista com numeração romana maiúscula (I, II, III, IV... etc.).

Analise a seguir o código de uma lista ordenada com o uso de atributo Type para gerar a lista com letras maiúsculas:

```
<HTML>
  <HEAD> <TITLE> Aprendendo sobre LISTAS </TITLE> </HEAD>
  <BODY>
    <OL TYPE="A">
      <LI> LISTA - letra A
      <LI> LISTA - letra B
      <LI> LISTA - letra C
      <LI> LISTA - letra D
    </OL>
  </BODY>
</HTML>
```

Figura 3.18: Exemplo de tag e TYPE em HTML

Fonte: Elaborada pelos autores

Resultado do código anterior, usando as tags para elaborar as listas ordenadas com a utilização de letras maiúsculas:

- A. LISTA - letra A
- B. LISTA - letra B
- C. LISTA - letra C
- D. LISTA - letra D

Figura 3.19: Imagem do resultado da tag usando TYPE

Fonte: Elaborada pelos autores

O atributo START é usado numa lista ordenada para determinar o número ou a letra com que a lista deve ser iniciada. Comumente, a lista se inicia com o número 1; para mudar essa ordem, pode-se utilizar o atributo START, conforme exemplo a seguir:

```
<HTML>
  <HEAD> <TITLE> Aprendendo sobre LISTAS </TITLE> </HEAD>
  <BODY>
    <OL TYPE="A" start=4>
      <LI> LISTA - letra A
      <LI> LISTA - letra B
      <LI> LISTA - letra C
      <LI> LISTA - letra D
    </OL>
  </BODY>
</HTML>
```

Figura 3.20: Exemplo de tag e START em HTML

Fonte: Elaborada pelos autores

Resultado do código anterior, usando as *tags* para elaborar as listas ordenadas com a utilização de letras maiúsculas e o atributo *START* iniciando com 4:

- D. LISTA - letra A
- E. LISTA - letra B
- F. LISTA - letra C
- G. LISTA - letra D

Figura 3.21: Imagem do resultado da tag usando START

Fonte: Elaborada pelos autores

```
<HTML>
  <HEAD> <TITLE> Aprendendo sobre LISTAS </TITLE> </HEAD>
  <BODY>
    <OL TYPE="A">
      <LI> LISTA - letra A
      <LI VALUE="5"> LISTA - letra B
      <LI> LISTA - letra C
      <LI> LISTA - letra D
    </OL>
  </HTML>
```

Figura 3.22: Exemplo de tag e VALUE em HTML

Fonte: Elaborada pelos autores

Resultado do código anterior, usando as *tags* para elaborar as listas ordenadas com a utilização de letras maiúsculas e o atributo *VALUE* iniciando com 5 na segunda *tag* :

1. LISTA - letra A
2. LISTA - letra B
3. LISTA - letra C
4. LISTA - letra D

Figura 3.23: Imagem do resultado da tag e VALUE em HTML

Fonte: Elaborada pelos autores

Resumo

Nesta aula foi feita uma introdução à Linguagem de Marcação de Hipertexto (HTML). Tivemos o primeiro contato com a programação para *web*, aprendemos qual é a estrutura de um documento HTML, o que é um título, o cabeçalho, as *tags* de um documento HTML. Aprendemos também como usar quebra de linha, parágrafos, alinhar um texto e usar as cores na elaboração de uma apresentação *web*. Vimos ainda como utilizar listas ordenadas e não ordenadas, e a formatação de texto através do negrito, itálico, sublinhado, sobrescrito e subscrito na montagem do *site*. Nas próximas aulas, iremos aprofundar nossos conhecimentos em HTML.

Atividades de aprendizagem

1. Qual a estrutura de um documento HTML?
2. O que é uma *tag*?
3. Para que serve a *tag*
 no comando HTML?
4. Qual *tag* é responsável pelo parágrafo no HTML?
5. Quais os tipos de alinhamentos possíveis num parágrafo do documento HTML? Demonstre.
6. Qual o código HTML para elaborar o texto abaixo:

A *Disciplina* **Introdução à Programação para Web** tem na HTML, sua ^{principal} linguagem de _{programação}.

7. O que faz o seguinte código em HTML?
8. Escreva o código em HTML que gera o seguinte texto:

a) Estes são meus itens escolhidos para supermercado do mês de Junho/2010

- ARROZ
- FARINHA
- AÇÚCAR
- SAL

b) Estes são meus itens escolhidos para supermercado do mês de Junho/2010

1. ARROZ
2. FARINHA
3. AÇÚCAR
4. SAL

9. Explique qual tipo de lista é usada em cada alternativa da questão anterior:

a) _____

b) _____

Aula 4 – Criando imagens, links e tabelas em HTML

Objetivos

Implementar páginas HTML usando imagens.

Criar documentos HTML com estruturas de *hyperlinks* internos, externos e *links* para terceiros.

Construir páginas HTML com as estruturas de tabelas.

4.1 Imagens

Você pode obter uma imagem através de uma cópia feita a partir de uma página da internet, através de *CD-ROMs*, *scanners* e outros. Pode também criá-las a partir de um programa de imagens.

4.1.1 A tag `img`

Permite colocar uma imagem no corpo do documento, o que proporciona dar uma melhor roupagem nas páginas HTML.

SINTAXE:

```
<imgsrc="url" align="..." width="..." height="..." border="..."  
alt="texto" vspace="..." hspace="...">
```

Onde:

ALIGN: refere-se ao alinhamento da imagem em relação ao texto, que pode ser **bottom**, **top** e **middle**.

SRC: é o nome da figura, se estiver no mesmo diretório do programa ou seu caminho completo.

ALT: exibe o texto quando o **browser** não encontra a imagem, ou uma legenda quando o cursor do **mouse** passa sobre ela mostrando o texto especificado.

WIDTH: especifica a largura de exibição da imagem, em **percentual** ou em **pixel**. Se for informado o sinal de porcentagem, ele é relativo à largura da janela.

HEIGHT: especifica a altura de exibição da imagem, que pode ser em **percentual** ou em **pixel**. Se for informado o sinal de porcentagem, ele é relativo à altura da janela.

BORDER: especifica em **pixels** a largura da borda da imagem. O valor zero remove a borda da imagem.

VSPACE: determina em **pixels** o espaço que deve ser deixado em branco na parte de cima e na parte de baixo da imagem.

HSPACE: determina em **pixels** o espaço que deve ser deixado em branco nas laterais da imagem.



Com exceção do parâmetro SRC, todos os demais são opcionais.

Agora veremos como exibir uma imagem chamada **foto.gif** utilizando a tag **IMG**:

```
<html>
  <head>
    <title>Imagens</title>
  </head>
  <body>
    <h1> Minha Foto </h1>
    
  </body>
</html>
```

Figura 4.1: Exemplo do uso da tag IMG

Fonte: Elaborada pelos autores

4.2 URLs e links

Até agora só mostramos como criar documentos simples; veremos outra forma de deixar os documentos mais sofisticados através de **links**.

Na internet há criação de documentos com o conceito de **hipertexto**, ou seja, um documento que se liga a outros por meio de referências especiais chamados **hyperlinks**. Com esse conceito você pode criar documentos que façam referência a um endereço qualquer do computador ou da *web* e permitam ao usuário acessar essas referências, não levando em conta se elas estão em outra página *web*, no seu computador ou em qualquer servidor na internet.

4.2.1 URLs

Na internet usa-se uma nomenclatura específica para indicar o endereço de um determinado documento. Chama-se *Uniform Resource Locator* (Localizador Universal de Recursos) ou simplesmente *URL*. A *URL* deve conter o protocolo de comunicação do documento (como por exemplo http, Gopher ou WAIS), o endereço servidor e a localização do arquivo, usando a seguinte sintaxe: *Protocolo://servidor.NomeDoArquivo*

<http://www.hardcore.com.br/index.htm>

Onde:

http: é o protocolo

www.hardcore.com.br: é o nome do servidor

index.htm: é o nome do documento que será acessado.

4.2.2 Criando um *link*<A>

Podemos entender que existem três tipos de *links*: o **interno**, que faz referência à navegação dentro do documento, conhecido também como *âncora*, o **externo local**, que faz referência a outros documentos dentro do mesmo *site* e os **externos para terceiros**, que faz referência aos documentos espalhados na *web*, fora do seu *site*.

SINTAXE: <ahref="..."name="...">Texto

Onde:

HREF: especifica o endereço do URL ao qual o *link* está associado. Pode ser usado dentro e fora do documento.

NAME: especifica o nome da seção de um documento à qual um *link* de hipertexto faz referência.

4.2.3 Criando um *link* interno (âncora)

Para navegar dentro de um documento HTML, você precisa criar uma âncora (conhecida também como indicador) no início de cada seção do documento. Essa âncora recebe um nome que será mencionado pelo *link* que acessará.

SINTAXE: `Texto Opcional`

Iremos criar agora uma página web com um exemplo de *link* âncora:

```
<html>
  <head>
    <title>Link interno âncora</title>
  </head>
  <body>
    <h1> Criando âncoras </h1>
    <a href="#inicio"> Inicio </a>
    <a href="#fim"> Vá par o Fim </a>
    Link âncora <br> Link âncora <br> Link âncora <br> Lin
âncora <br> Link âncora <br> Link âncora <br> Link âncora <br>
âncora <br> Link âncora <br> Link âncora <br> Link âncora <br>
âncora <br> Link âncora <br> Link âncora <br> Link âncora <br>
âncora <br> Link âncora <br> Link âncora <br> Link âncora <br>
    <a href="#Fim"> Fim </a>
    <a href="#inicio"> Vá par o início </a>
  </body>
</html>
```

Figura 4.2: Exemplo de *links* âncoras

Fonte: Elaborada pelos autores

4.2.4 Usando uma imagem como *hyperlink*

Podemos usar uma imagem como *hyperlink*, bastando para isso apenas incluir a *tag* IMG dentro da *tag* <A>, ou seja, substituindo o texto no exemplo anterior por uma imagem. Vejamos como fica:

```
<html>
  <head>
    <title>Link interno Âncora</title>
  </head>
  <body>
    <h1> Criando âncoras com imagens</h1>
    <a href="#inicio"> Inicio </a>
    <a href="#fim">  </a>
    Link âncora <br> Link âncora <br> Link âncora <br> Link âncora <br>
âncora <br> Link âncora <br> Link âncora <br> Link âncora <br> Link âncora
âncora <br> Link âncora <br> Link âncora <br> Link âncora <br> Link âncora
âncora <br> Link âncora <br> Link âncora <br> Link âncora <br> Link âncora
âncora <br> Link âncora <br> Link âncora <br> Link âncora <br> Link âncora
    <a href="#Fim"> Fim </a>
    <a href="#inicio">  </a>
  </body>
</html>
```

Figura 4.3: Exemplo de imagem como *hyper link*

Fonte: Elaborada pelos autores

4.2.5 Criando um *link* externo local

Os *links* externos locais, como já foi dito anteriormente, referenciam um documento fora do documento atual, mas dentro do mesmo *site*.

SINTAXE: `Texto opcional`

Imaginemos que uma página, o arquivo **index.html**, queira fazer referência para outro documento, o *pagina1.html*; então, dentro do arquivo **index.html**, devemos referenciá-lo por meio da *tag* <A>; vejamos o exemplo a seguir:

Cuidado! Quando vamos especificar ou referenciar o endereço de outros documentos internos ao nosso *site* para podermos mostrá-los na *web* usando um esquema de caminhos de pastas, devemos utilizar a barra (/) para separarmos as pastas dos arquivos.



```
<html>
  <head>
    <title>Link Externo</title>
  </head>
  <body>
    <h1> Criando Links Externos </h1>
    <a href="pagina1.html"> Acesse a o arquivo página1.html </a>
  </body>
</html>
```

Figura 4.4: Exemplo de *link* externo local

Fonte: Elaborada pelos autores

4.2.6 Criando um *link* externo para terceiros

Os *links* externos para terceiros referenciam um documento fora do seu *site*, mas que esteja na *web*, hospedado em algum servidor.

SINTAXE: Texto Opcional

Onde:

HREF: referencia o endereço do *site* ou documento que esteja na *web*.

TARGET: especifica se você quer que essa página seja aberta em outra janela do navegador. Caso queira, especifique TARGET="BLANK".

Imaginemos que uma página, o arquivo **index.html**, queira fazer referência ao *site* do Google; então, devemos adicionar a *tag* o **pagina1.htm**; então, dentro do arquivo **index.html** devemos referenciá-lo por meio da *tag* <A>. Vejamos a seguir o exemplo:

```

<html>
  <head>
    <title>Link Externo</title>
  </head>
  <body>
    <h1> Criando Links Externos </h1>
    <a href="HTTP://www.google.com.br"> Acesse o site do Google</a>
  </body>
</html>

```

Figura 4.5: Exemplo de *link* externo para terceiros

Fonte: Elaborada pelos autores

4.2.7 Enviando e-mail diretamente da página HTML

Você pode acionar o programa padrão de e-mail diretamente de uma página HTML usando uma variação do comando **<A>**, que usa a opção **MAILTO** no lugar da URL.

**Tire as suas dúvidas **



Podemos também definir destinos para os nossos *links*, (recurso muito utilizado em *frames*) através do parâmetro *target*. E podem ser *_self*: Mesmo quadro, *_top*: página inteira, *_blank*: nova janela e *_parent*: quadro pai.

```

<html>
  <head>
    <title>Links</title>
  </head>
  <body>
    <p align="center">Links</p>
    <p><a href="http://www.microsoft.com.br">Microsoft</a>
    <p><a href="http://www.macromedia.com.br">Macromedia</a>
    <p><a href="http://www.adobe.com.br" target="_blank">Adobe</a>
    <p><a href="mailto:softwaresuporte@hardcoreinformatica.zzn.com">e-mail</a>
    <hr>
    <p align="center">Criando as Âncoras</p>
    <a href="#textos">Textos</a> &nbsp;&nbsp;&nbsp;<a href="#imagens">Imagens</a>
    <p>
      <a name="textos">TEXTOS</a>
    <p>&nbsp;&nbsp;&nbsp;</p><p>&nbsp;&nbsp;&nbsp;</p><p>&nbsp;&nbsp;&nbsp;</p><p>&nbsp;&nbsp;&nbsp;</p><p>&nbsp;&nbsp;&nbsp;</p>
    <a name="imagens">IMAGENS</a>
    <p align="center"></p>
  </body>
</html>

```

Figura 4.6: Exemplo de *links* externo e para e-mail

Fonte: Elaborada pelos autores

4.3 Tabelas

Tabelas em HTML são criadas através de elementos específicos com certa semelhança aos arquivos feitos em Excel. O recurso de tabelas é muito interessante e muito usado nas páginas *web*.

Podemos definir tabelas como um conjunto de linhas e colunas; na interseção delas estão as células, que por sua vez recebem os dados que serão mostrados ao usuário.

Na linguagem HTML, você pode inserir nas células tudo o que normalmente faz parte do corpo de um documento, como textos, *links*, imagens, listas e até outras tabelas.

4.3.1 Elementos básicos de uma tabela

<TABLE>...</TABLE>

São as *tags* que englobam a definição de uma tabela. Todas as demais marcas referentes a tabelas, linhas e células somente serão consideradas se incluídas entre **<TABLE>** e **</TABLE>**.

<CAPTION>...</CAPTION>

Elemento opcional que define o título da tabela e que deve constar juntamente com outras marcas, mas separado das que definem linhas e colunas. Sem parâmetros, o título é apresentado acima da tabela e centralizado.

<TR>...</TR> *Table row* ou *linha de tabela*

Este elemento é utilizado na definição de linhas de tabelas. Tabelas são definidas em linhas, sendo estas compostas de células. O número de linhas de uma tabela corresponde ao número de **<TR></TR>**.

<TD>...</TD> *Table data* ou *dados de tabela*

Marcadores que delimitam as células que compõem as linhas, que por sua vez devem estar inseridas entre os marcadores de linhas. Numa célula podemos colocar tudo o que normalmente se coloca no corpo de um documento HTML, como *links*, referências a imagens, textos, inclusive tabelas.

O alinhamento padrão de uma célula é à esquerda horizontalmente e centralizado verticalmente. No caso de o número de células ser diferente de uma linha para outra, as linhas com menos células terão o tamanho redimensionado à direita das células em branco, igualando-as.

<TH>...</TH> *Table header* ou *cabeçalho de tabela*

Elemento que define cabeçalho nas células. Células de cabeçalho têm características idênticas a células de dados definidas por **<TD>**; a diferença é que seu alinhamento horizontal é centralizado e a sua fonte, em negrito.

4.3.2 Parâmetros

São elementos básicos, que permitem um maior controle sobre alguns detalhes da apresentação da tabela.

a) Parâmetros da tag <TABLE>

BORDER: A apresentação padrão de uma tabela é sem bordas. A presença do parâmetro BORDER indica justamente que deve ser desenhada uma borda em torno de cada célula da tabela. Pode ser usado para indicar a espessura da borda (em *pixels*), com BORDER=X.

Exemplo: <TABLE BORDER=2>

BORDER COLOR: Permite que se coloque cores nas bordas da tabela e possui duas variações: *border color light* e *border color dark*.

WIDTH: Especifica a largura da tabela, que pode ser definida em percentual ou em *pixels*, referentes à largura da janela. Caso isso não seja especificado, o próprio navegador se encarrega de determinar a largura mais adequada, baseado nos textos inseridos nas células.

Exemplo: <TABLE WIDTH=75%>

CELLSPACING: Define o espaço entre as células, ou seja, a largura das linhas de grade (as bordas que envolvem as células). É especificado em *pixels*.

Exemplo: <TABLE CELLSPACING=3>

CELLPADDING: Determina, em *pixels*, o espaço entre o conteúdo e as bordas da célula da tabela.

Exemplo: <TABLE CELLPADDING=6>

ALIGN: Configura o alinhamento horizontal da tabela em relação aos outros elementos da página. Pode conter os valores LEFT (esquerda), CENTER (centro) ou RIGHT (direita).

BGCOLOR: Define a cor de fundo da tabela.

b) Parâmetros do elemento <TR>

ALIGN: Configura o alinhamento horizontal dos elementos contidos nas

células de uma linha. Os valores podem ser LEFT, CENTER OU RIGHT. Caso não seja informado nenhum valor, o alinhamento será à esquerda para as células de dados (<TD>), e centralizado para células de cabeçalho (<TH>).

VALIGN: Define o alinhamento vertical dos elementos contidos nas células de uma linha. Os valores podem ser **TOP** (topo), **MIDDLE** (meio) ou **BOTTOM** (fundo). Caso não seja informado nenhum valor, o alinhamento é ao meio.

BGCOLOR: Define a cor de fundo da linha.

c) Parâmetros das tags <TD> e <TH>

```
<html>
<head>
  <title>Tabelas</title>
</head>
<body>
  <table border="10" width="100%" bordercolorlight="#000080" bordercolordark="#FF00FF">
    <tr>
      <th width="100%" colspan="3">
        <p align="center">Titulo da Tabela</p>
      </tr>
    <tr>
      <td width="30%">Texto1</td>
      <td width="30%">Texto 2</td>
      <td width="40%">Texto 3</td>
    </tr>
  </table>
</body>
</html>
```

Figura 4.7: Exemplo da tag <TABLE>

Fonte: Elaborada pelos autores

As tabelas servem para exibir dados, como numa planilha feita em Excel; os elementos das tabelas são usados para formatar páginas de uma forma geral. Utilizamos ainda tabelas para delimitar áreas específicas de uma página das mais diversas maneiras possíveis.



Resumo

Nesta aula aprendemos o uso de imagens em uma página HTML e suas propriedades. Conhecemos as estruturas de *hyperlinks* internos, externos e *links* para terceiros e as suas aplicações. Por fim, aprendemos a trabalhar com tabelas para melhor padronizar nossas estruturas de visibilidade das páginas HTML. Nas próximas aulas, iremos aprofundar nossos conhecimentos em HTML estudando as estruturas de *layout* de páginas.



Podemos também, usar tabelas para garantir que um determinado texto fique na posição exata em que desejamos mostrá-lo, de forma que os elementos fiquem alinhados na página.

Atividades de aprendizagem

1. Analisando o conteúdo estudado, defina para que serve a utilização de imagens em um documento HTML.
2. Construa uma *homepage* com a mensagem “Olá Mundo!”.
3. Cite três formas de *links* utilizados em um documento HTML.
4. Crie uma página em HTML para representar os tipos de *links* descritos na questão anterior, sendo nela “linkado” o *site* do Google.
5. Construa uma tabela com as seguintes características:
 - a) cabeçalho: NOME, ENDEREÇO E FONE;
 - b) 1ª linha: Otilio, Rua X, 86 3333-3333
 - c) 2ª linha: Nádia, Rua Y, 86 4444-4444
 - d) 3ª linha: Sandra, Rua Z, 86 5555-5555

Aula 5 – Frames, multimídia e folhas de estilos – CSS

Objetivos

Criar estruturas de *frames*.

Implementar documentos com elementos multimídia.

Estruturar páginas HTML com folhas de estilo, criando uma padronização.

5.1 Frames

Os *frames* ou janelas são utilizados para separar em pedaços a tela do monitor, na sua página, de forma a dividir todos os elementos nela contida; permitem ainda criar páginas HTML que podem ser visualizadas simultaneamente na janela do *browser*. Com esse recurso, a janela do *browser* é dividida em *frames* que compartilham o espaço disponível.

Usando esse recurso, podemos exibir mais de um *frame* simultaneamente.

5.1.1 Como funcionam os *frames*

Ao contrário da maioria dos recursos da linguagem HTML, o uso de *frames* requer um planejamento prévio; inicialmente devemos imaginar quantas linhas e quantas colunas iremos utilizar. O planejamento consiste na criação dos *frames*, quando especificamos a quantidade deles, a disposição que eles terão na tela, a largura e a altura de cada um, assim como o conteúdo que será exibido em cada um deles. Trabalhar com *frames* exige duas etapas. Na primeira, você cria o *layout* dos *frames*, na segunda, define seu conteúdo.

Na criação de *frames*, o documento HTML não possui as *tags* **<BODY>** e **</BODY>**. Elas são substituídas por **<FRAMESET></FRAMESET>**. Dentro dessas *tags* são especificados todos os atributos dos *frames* que serão criados. Cada *frame* especificado precisará de um outro comando chamado **<FRAME>** para definir suas características individuais.

5.1.2 A tag FRAMESET

Esta *tag* é do tipo contêiner, ou seja, deve ser aberta e fechada. Veja a sintaxe deste comando:

Sintaxe:

```
<FRAMESET TIPO=COLS/ROWS FRAMEBORDER=1/0  
FRAMEPACING=spacing ROWS=tamanho></FRAMESET>
```

- a) **Cols:** É usado para criar *frames* em colunas. Nele deverá ser especificado o tamanho de cada coluna do documento. Pode ser especificado um valor padrão para todas as colunas ou tamanhos individuais. Os valores podem ser especificados em *pixels*, porcentagem ou tamanho relativo representado pelo caractere asterisco (*).

Usando valores em *pixels*:

```
<FRAMESET COLS="100,300,200">
```

Usando valores percentuais de forma a determinar o tamanho de uma coluna é especificar os valores em forma de percentual, totalizando 100 por cento, que é o tamanho total da tela.

```
<FRAMESET COLS=25%,50%,25%>
```

Usando valores relativos para dividir os *frames* igualmente na janela do *browser*, a melhor opção é usar um asterisco (*) para cada coluna.

```
<FRAMESET COLS=*,*,*>
```

Os diferentes valores podem ser combinados entre si. Por exemplo, para criar um **FRAMESET** com três colunas de modo que a primeira ocupe 25%, a terceira tenha 100 *pixels* e a outra ocupe o espaço restante, pode-se usar o seguinte comando:

```
<FRAMESET COLS=25%,*,100>
```

- b) **Rows:** funciona da mesma forma que o atributo COLS, só que cria os *frames* horizontalmente. Tudo o que foi dito para COLS, aplica-se a ROWS.

- c) **Combinando ROWS e COLS:** Pode-se combinar a criação de *frames* usando os dois atributos. Você pode criar uma janela dividida em dois *frames* horizontais, em que o primeiro deles é dividido em duas colunas. Para que isso seja possível, basta aninhar conjuntos de *tags* **<FRAMESET>**

O atributo **<FRAMESPACING=valor>**

Esse atributo tem como finalidade alterar o espaço entre os *frames*, dando a impressão de que a borda dos *frames* foi aumentada. Contudo, ele apenas distancia um *frame* de outro pela quantidade de *pixels* especificada.

O código pode ser o mesmo do exemplo anterior. Apenas a primeira linha deve ser alterada para:

```
<FRAMESET ROWS=50%,50% FRAMESPACING=10>
```

O atributo **<FRAMEBORDER=1/0>**

Esse atributo tem a finalidade de eliminar as bordas dos *frames*. Como padrão, ele usa o valor 1, que exibe a borda. Se for especificado o valor 0, as bordas serão omitidas. A omissão de bordas é interessante para criar a ilusão de uma única janela, em que as partes dessa janela são independentes das outras. Outro efeito interessante é a mudança da cor de fundo de cada janela, permitindo dividir a tela em áreas coloridas distintas.

5.1.3 A tag **FRAME SRC**

Sintaxe:

```
<FRAME ALIGN=posição frameBORDER=1/0 MARGINHEIGHT=altura  
MARGINWIDTH=largura NAME=nome SCROLLING=yes/no  
SRC=endereço
```

ALIGN: Ajusta o alinhamento do *frame* ou do texto. Os valores permitidos são:

TOP: O texto ao redor do *frame* é alinhado pela parte superior.

MIDDLE: O texto ao redor do *frame* é alinhado pelo meio.

BOTTOM: O texto ao redor do *frame* é alinhado pela parte inferior.

LEFT: O *frame* é alinhado à esquerda, deixando o texto posicionado no seu lado direito.

RIGHT: O *frame* é alinhado à direita, deixando o texto posicionado no seu lado esquerdo.

Veja outros atributos que podem ser utilizados nos *frames*:

FRAMEBORDER: Idêntico ao atributo de mesmo nome do comando `<FRAMESET>`, ele ativa ou desativa a exibição de borda para o *frame* atual. Pode assumir 0 ou 1.

MARGINHEIGHT: Especifica a altura da margem superior e inferior do *frame* em *pixels*.

MARGINWIDHT: Especifica a largura das margens esquerda e direita do *frame* em *pixels*.

Name: Especifica o nome para o *frame*, de maneira que possa ser identificado e localizado para carregar documentos.

Noresize: Evita que o *frame* seja redimensionado pelo usuário.

Scrolling: Quando ajustado para o valor **No**, não permite que o *frame* tenha barras de rolagem quando a janela for redimensionada. Pode assumir **Yes** ou **No**.

SRC: Define o endereço ou URL do documento que será exibido no *frame*.

Ajustando as margens internas do *frame*

O desenvolvedor pode criar uma margem interna para o *frame*, de forma a distanciar o conteúdo da janela da sua borda. Para criar uma margem nas laterais da janela, deve ser usado o comando **MARGINWIDTH**. Para criar uma margem na parte superior e outra no rodapé do *frame*, deve ser usado o atributo **MARGINHEIGHT**. O tamanho da margem deve ser especificado em *pixels*.

5.1.4 Criando a estrutura dos *frames*

Criaremos agora uma estrutura básica de um *frame*, disposto em colunas através da *tag* COLS. Esse arquivo contém a definição de dois *frames*, definidos em colunas. Veja o código a seguir.

```

<HTML>
  <HEAD>
    <TITLE> Criando Frames </TITLE>
  </HEAD>
  <FRAMESET COLS=20,30>
    <FRAME SRC="#">
    <FRAME SRC="#">
  </FRAMESET>
</HTML>

```

Figura 5.1: Exemplo da tag FRAME usando COLS

Fonte: Elaborada pelos autores

Vejamos agora um outro exemplo de *frame* em forma de linhas, utilizando o parâmetro **ROWS** para especificar as linhas.

```

<HTML>
  <HEAD>
    <TITLE> Criando Frames </TITLE>
  </HEAD>
  <FRAMESET ROWS=20,30>
    <FRAME SRC="#">
    <FRAME SRC="#">
  </FRAMESET>

```

Figura 5.2: Exemplo da tag FRAME usando ROWS

Fonte: Elaborada pelos autores

No código acima pode-se ver também que foi utilizada a tag `<NOframes></NOframes>`; essa tag é utilizada para avisar ao usuário que o navegador dele não suporta *frames*; então, pode-se colocar qualquer texto ou figura HTML para o usuário.

```

<html>
  <head>
    <title>IFRAME</title>
  </head>
  <body bgcolor="#000000" text="#FFFFFF">
    <p>&nbsp;</p>
    <p align="center">Utilizando IFRAME</p>
    <div align="center">
      <center>
        <table border="0" width="80%" height="193">
          <caption>&nbsp;</caption>
          <tr>
            <td width="100%" height="187" align="center">
              <p align="center"><IFRAME SRC="textos.htm"></iframe></td>
            </tr>
          </table>
        </center>
      </div>
    </body>
  </html>

```

Figura 5.3: Exemplo da tag Iframe

Fonte: Elaborada pelos autores

5.2 Multimídia

O principal elemento de sucesso da internet é o *hyperlink*, o qual permite a criação de documentos com ligações para outros conteúdos em qualquer computador que esteja conectado. A maioria dos usuários que acessa a internet faz isso a partir de ambientes gráficos, como o Windows, cuja capacidade em trabalhar com multimídia é reconhecida. Nesta seção vamos aprender a trabalhar com os elementos multimídia em nossas páginas.

Principais elementos multimídia para uma aplicação:

- Fotos ou imagens estáticas.
- Animação.
- Áudio.
- Vídeo.

5.2.1 Consideração sobre o uso de multimídia

O grande inimigo da utilização de multimídia é o relacionado com as imagens, a velocidade de transferência de informações via internet. Se uma imagem de 70Kb pode levar vários segundos, ou até mesmo minutos para ser carregada, imagine a execução de um clipe de vídeo com mais de 1MB.

Os vídeos vão se popularizando à medida que a internet a cabo vai se tornando mais acessível.

5.2.2 Inserindo um vídeo

Existem várias maneiras de inserir um vídeo em sua página. O vídeo pode aparecer sob a forma de um *link*, que ao ser clicado executa o programa responsável pela sua exibição, ou aparecer diretamente na página.

A maneira mais simples de incluir um clipe de vídeo em sua página é utilizar as opções do comando IMG. O principal parâmetro usado para inserir vídeo é a *DYNSRC*, abreviatura de *Dynamic Source*.

a) Atributos utilizados em vídeos

O vídeo, como qualquer imagem em um documento HTML, também possui uma infinidade de atributos, os quais estão descritos no Quadro 5.1 a seguir.

Quadro 5.1: Relação de atributos utilizados em vídeos

Atributo	Finalidade	Exemplo
ALIGN=TOP, MIDDLE, ou BOTTOM	O texto que margeia a imagem é alinhado pelo topo, meio ou base do vídeo.	 Este texto é alinhado pelo meio da figura.
ALIGN=LEFT ou RIGHT	A imagem, ou vídeo, é alinhada à esquerda ou direita da página.	 Este texto aparece do lado esquerdo da imagem.
ALT=texto	Especifica um texto alternativo para ser exibido no local do vídeo.	IMG SRC="vídeo.avi" ALT="figura exibida".
BORDER=n	Especifica a largura da borda em <i>pixels</i> do vídeo.	Esta imagem possui borda de cinco <i>pixels</i> de largura.
CONTROLS	Exibe botões de controle do tipo videocassete.	
DYN SRC=URL	Especifica o URL do vídeo que será exibido.	
HEIGHT=n	Especifica a altura do vídeo. Se este possuir outro tamanho, será ajustado para o tamanho especificado.	
HSPACE=n	Especifica uma margem horizontal em volta da imagem para afastá-la do texto que a envolve.	
LOOP=n LOOP=INFINITE	Especifica quantas vezes o vídeo será executado. Se n for igual a -1 ou INFINITE, o vídeo será executado continuamente. Caso contrário, é executado o número de vezes especificado.	 o vídeo é executado três vezes.
SRC	Especifica o endereço do vídeo.	<IMG SRC http://www.hardcore.com.br/vídeo.avi >
START=FILEOPEN/ MOUSEOVER	Para clipes de vídeo, especifica quando o vídeo deve ser executado. Como padrão, o vídeo assume o valor FILEOPEN e é executado imediatamente após seu carregamento. O valor MOUSEOVER faz com que o vídeo seja executado quando o cursor passa sobre a área da imagem do vídeo.	DYN SRC="vídeo.avi" START= FILEOPEN> O vídeo é executado ao ser carregado.
VSPACE=n	Especifica uma margem vertical para o vídeo, em <i>pixels</i> .	
WIDTH=n	Especifica a largura da imagem.	

Fonte: Elaborado pelos autores

b) Usando EMBED para exibir vídeo

O comando EMBED permite a exibição não somente de vídeo, mas também de arquivos de diferentes tipos em uma página HTML, que são executados por algum *plug-in* previamente instalado. Ele funciona tanto no Internet Explorer quanto no Netscape Navigator.

Sintaxe:

```
<EMBED ALIGN="LEFT" | "RIGHT" | "TOP" | "BOTTOM"  
BORDER="pixels"  
frameBORDER="NO"  
HEIGHT="pixels"  
HIDDEN="TRUE" | "FALSE"  
HSPACE="pixels"  
NAME="appletName"  
PALLETE="FOREGROUND" | "BACKGROUND"  
PLUGINSOURCE="instrURL"  
SRC="endereço"  
TYPE="MIMEtype"  
VSPACE="pixels"  
WHIDTH="pixels"  
  
</EMBED>
```

Para manter a compatibilidade com *browsers* antigos, usa-se o comando <NOEMBED> para exibir uma mensagem indicando que o *browser* não tem capacidade para exibir o objeto.

```
<html>  
  <head>  
    <title>Multimídia</title>  
  </head>  
  <body>  
    <IMG DYNSSRC="Cyclers.avi" START="fileopen" Loop="Infinite">  
    <HR>  
    <embed src="admiradora.asf"></embed>  
    <HR>  
    <embed src="ambulancia.mpeg"></embed>  
  </body>  
</html>
```

Figura 5.4: Exemplo da tag ENBED

Fonte: Elaborada pelos autores

5.2.3 Comamndo MARQUEE

O Internet Explorer introduziu um comando muito simpático chamado MARQUEE, que faz com que um texto especificado fique rolando em uma determinada área da página.

Veja no Quadro 5.2 a sintaxe do comando MARQUEE.

Quadro 5.2: Relação de atributos da tag MARQUEE

ALIGN=posição	Especifica como o texto que envolve o MARQUEE será alinhado. Pode conter os seguintes valores:	
	TOP	Alinha o texto pela sua parte superior
	MIDDLE	Alinha o texto pelo meio
	BOTTOM	Alinha o texto pela sua parte inferior
BEHAVIOR=tipo	Especifica o comportamento do texto, ou seja, como será a sua rolagem pela janela. Pode conter os seguintes valores.	
	SCROLL	Inicia a rolagem introduzindo o texto de um lado e rola o texto até que ele saia completamente da janela no lado oposto ao que começou. Após a última letra ter saído da tela, o texto reaparece no canto inicial, repetindo continuamente o processo.
	SLIDE	Inicia a rolagem introduzindo o texto de um lado e rola o texto até que ele atinja a borda oposta. Quando a primeira letra do texto bater na borda, o texto para de rolar e permanece naquela posição.
ALTERNATE	Cria um efeito de movimento para o texto, que inicia de um lado, entrando pela borda da janela e, ao bater na outra borda, inverte o seu deslocamento voltando para a borda inicial.	
BGCOLOR=cor	Especifica a cor do texto do segundo plano. Caso não seja especificada uma cor, é assumida a cor de segundo plano atual.	
DIRECTION=direção	Especifica a direção que o texto vai usar para o deslocamento. O padrão é LEFT, que faz o texto deslocar-se do canto direito da janela para o canto esquerdo. Pode ser especificado também o valor RIGHT, que inverte o sentido de deslocamento.	
HEIGHT=n	Especifica a largura do painel, ou a área do MARQUEE. Se for especificado um número, ele é considerado como quantidade de <i>pixels</i> . Se for seguido do sinal %, indica a largura com relação à janela por meio da porcentagem.	
HSPACE=n	Especifica a largura em <i>pixels</i> das margens esquerda e direita, para afastá-lo do texto ou de outros objetos que o cercam.	
LOOP=n	Especifica quantas vezes será executado. Se o valor de n for -1 ou INFINITE ele será executado continuamente.	
SCROLLAMOUNT=n	Indica a quantidade de <i>pixels</i> que será usada para deslocar o MARQUEE a cada movimentação.	
SCROLLDELAY=n	Especifica em milissegundos o tempo entre cada deslocamento do texto.	
VSPACE=n	Especifica em <i>pixels</i> a margem superior e inferior do MARQUEE.	
WIDTH=n	Especifica a altura do painel, ou área do MARQUEE. Se for especificado um número, ele é considerado como quantidade de <i>pixels</i> . Se for seguido do sinal %, indica a altura em relação à janela por meio de porcentagem.	

Fonte: Elaborado pelos autores

Temos certeza de que você irá gostar de usar esse comando para chamar a atenção dos seus usuários. Antes, porém, teste a visualização da página com um *browser* que não seja compatível com esse comando, para verificar como ficará a tela. Infelizmente, a Netscape deixou de incorporar esse comando à versão 4 do seu *browser*.

a) Combinando a utilização de vários MARQUEES

Você pode criar alguns efeitos interessantes combinando o uso de vários MARQUEES. Eles podem ser colocados lado a lado ou um sobre o outro para a criação de várias linhas. Os primeiros dois MARQUEES são colocados lado a lado com o texto rolando em posição contrária. Eles partem das bordas e colidem no centro da página. O segundo exemplo é uma variação do primeiro. Porém, foi adicionada uma quebra de linha com o comando
 entre os dois e delimitada uma largura.

No terceiro exemplo, três MARQUEES são colocados um abaixo do outro, criando um painel de três linhas.

```
<html>
  <head>
    <title> Combinando vários Marquees</title>
  </head>
  <body bgcolor="#FFFFFF">
    <marquee height="15" width="300" behavior="Scroll" direction="right"
      bgcolor="#FFFF00" loop=infinite>
      É fácil juntar dois Marquees</marquee>
    <marquee height="15" width="300" behavior="Scroll" direction="Left"
      bgcolor="#00FF00" loop=infinite>
      É fácil juntar dois Marquees</marquee>
    <hr>
    <body bgcolor="#FFFFFF">
    <marquee height="15" width="200" behavior="Scroll" direction="right"
      bgcolor="#FFFF00" loop=infinite>
      É fácil juntar dois Marquees</marquee>
    <br>
    <marquee height="15" width="200" behavior="Scroll" direction="Left"
      bgcolor="#00FF00" loop=infinite>
      É fácil juntar dois Marquees</marquee>
    <hr>
    <body bgcolor="#FFFFFF">
    <marquee height="15" width="200" behavior="Scroll" direction="left"
      bgcolor="#FFFF00" loop=infinite>
      Para criar um Marque com</marquee>
    <br>
    <marquee height="15" width="200" behavior="Scroll" direction="Left"
      bgcolor="#FFFF00" loop=infinite>
      diversas linhas bastam especificar</marquee><br>
    <marquee height="15" width="200" behavior="Scroll" direction="Left"
      bgcolor="#FFFF00" loop=infinite>
      vários Marquees em seqüência </marquee>
    </body>
  </html>
```

Figura 5.5: Exemplo da tag MARQUEE

Fonte: Elaborada pelos autores

5.2.4 SOM

A inclusão de som em uma *homepage* ou aplicação internet pode ser um recurso bastante interessante para chamar a atenção do usuário. Menos problemático do que o uso do vídeo, devido ao tamanho dos seus arquivos, o uso de som também enfrenta problemas de velocidade de transmissão por causa das limitações da rede ou do *browser*.

Existem vários tipos de formatos para arquivos de áudio, porém quatro ou cinco deles dominam o mercado e são compatíveis com a maioria dos *browsers* atuais. Veja quais são eles:

AU: Esse formato é dominante na plataforma Unix e é compatível com quase todas as demais plataformas. É adequado para instrumentos e voz.

WAV: Formato padrão de som da plataforma Windows – oferece boa qualidade de som, porém gera arquivos de tamanho muito grande, o que o torna problemático para ser executado quando não existe boa velocidade de transmissão entre o *browser* e o servidor.

MIDI: Esse formato é um padrão para a representação de instrumentos musicais produzidos eletronicamente, como um sintetizador. Ao contrário dos formatos anteriores, ele não é compatível com voz, pois na verdade o arquivo MIDI não é uma gravação digital de som e sim um arquivo com comandos para a execução de sons. Sua vantagem é que ele possui um tamanho extremamente reduzido e oferece ótima qualidade para música orquestrada.

MP3: Esse formato ou todos os formatos de MPA são arquivos de áudio compactados, equivalentes aos arquivos de vídeo MPEG. Seu tamanho é menor do que os arquivos WAV e com qualidade superior. Existe hoje na *web* uma procura enorme por arquivos MP3.

RAM ou **RA:** É o formato Real Audio e está se tornando muito popular, pois permite a execução sob demanda, ou seja, permite transmissões ao vivo e, quando executado a partir de um arquivo, reproduz o som à medida que ele vai sendo lido.

a) Inserindo áudio

Assim como no caso dos clipes de vídeo ou imagens, podemos usar duas técnicas para incluir áudio na página. A primeira é através de *hyperlinks*; a segunda, através do comando META e, dependendo do *browser*, através de comandos específicos.

b) Inserindo áudio através de *hyperlinks*

Veja o código abaixo no qual os arquivos de áudio estão vinculadas às páginas através de *hyperlinks*.

```

<html>
  <head>
  </head>
  <body bgcolor="#FFFFFF">
    <a href="mission2.mid">Missão impossível</a>
    <a href="solar2.wav"> do tipo .WAV</a>,
    <a href="game1.au">no formato .AU</a> ou
    <a href="eg-follow.mp2"> no formato .mpeg</a>
  </body>
</html>

```

Figura 5.6: Exemplo de uso de áudio em HTML

Fonte: Elaborada pelos autores

c) Inserindo música de fundo

O BGSOUND (Internet Explorer), novo comando introduzido pelo *browser* da Microsoft, é extremamente simpático. Ele carrega e executa imediatamente um arquivo de som, assim que a página é acessada. Uma vez iniciada a execução do som, pode-se interrompê-la pressionando a tecla ESC ou mudando de página. A sintaxe está mostrada no QUADRO 5.3 a seguir.

Quadro 5.3: Relação de atributos para inserir músicas em HTML

Atributo	Finalidade	Exemplo
SRC= URL	Especifica o endereço do arquivo de áudio que será executado.	<BGSOUND SRC="ring.au"
LOOP=n	Especifica quantas vezes o arquivo será reproduzido.	<BGSOUND SRC="ring.au" LOOP=3>
LOOP=INFINITE	O valor INFINITE reproduz o arquivo continuamente	Executa o arquivo infinitamente.

Fonte: Elaborado pelos autores

Veja um exemplo abaixo:

```

<html>
  <head>
  </head>
  <bgsound src="africa.mid">
  <body bgcolor="#FFFFFF">
    <font color=Red>
      <h1> O comando BGSOUND </h1>
    </font>
  </body>
</html>

```

Figura 5.7: Exemplo de uso de música na abertura da página

Fonte: Elaborada pelos autores

5.3 Folhas de estilo

A *Cascading Style Sheets* (CSS) traz para a *web* a mesma conveniência de “um só lugar” para definir os estilos que estão disponíveis na maioria dos editores de texto. Você pode definir uma CSS em uma localização central para afetar a aparência das *tags* HTML em uma única página da *web* ou em todo um *site* da *web*.

Embora a metodologia da CSS funcione com a HTML, não é HTML. Em vez disso, a CSS é um código separado que amplia as capacidades da HTML, permitindo que você redefina o modo como as *tags* HTML funcionam.

5.3.1 Versões da CSS

A CSS evoluiu nos últimos anos sob a orientação do W3C (www.w3.org/Style/CSS/) até a sua versão atual, a 2 (dois). Embora a maioria dos *browsers* modernos suporte a versão mais recente, os *browsers* mais antigos suportam as combinações das versões antigas da CSS. Elas são:

CSS-1: O W3C lançou a primeira versão oficial da CSS em 1996. Essa primeira versão incluía a capacidade de núcleo associada às CSS, tais como a capacidade de formatar texto, definir fontes e margens.

CSS – Positioning (CSS-P): Os criadores da *web* precisavam de uma forma de posicionar os elementos na tela de forma precisa. A CSS 1 já havia sido lançada e a CSS-2 ainda estava distante, de modo que a W3C lançou uma solução intermediária: A CSS-Positioning. Esse padrão destinava-se a ser uma proposta que seria discutida pelas várias partes envolvidas antes de ele ser oficializado. A Netscape e a Microsoft assumiram essas propostas, porém, e incluíram as ideias preliminares nas versões 4 de seus navegadores. Embora a maioria dos recursos básicos sejam suportados em ambos os *browsers* com o nome de “marca” , diversos recursos foram deixados de fora.

CSS-2: A versão mais recente da CSS surgiu em 1998. O nível 2 inclui todos os atributos das duas versões anteriores, mais uma ênfase maior na facilidade de acesso e na capacidade de CSS específicas de mídia. A partir do IE e Netscape 6, suportam CSS-2.

CSS-3: Esse padrão ainda está em desenvolvimento e, mesmo após o lançamento, geralmente leva alguns anos para que os navegadores o suportem. Sem dúvida, a nova adição será o *Scalable Vector Graphics* (SVG). Esse é um formato que permite incluir formas (linhas, círculos, curvas e outras), como vetores e não *bitmaps*, levando o poder dos gráficos baseados no vetor e na tipografia à *web*.

5.3.2 Regras da CSS

O melhor da folha de estilo em cascata é que ela é incrivelmente fácil de configurar. Não exige *plug-ins* ou *softwares* diferentes – apenas regras.

Podemos definir regras que dizem a uma *tag* HTML especificar o seu conteúdo, ou poder criar regras genéricas e, em seguida, aplicá-las às *tags* como quiser. Existem três etapas nas regras da CSS.

Seletor HTML: A parte de texto de uma TAG HTML se chama seletor. Exemplo:

p {font: bold 12pt times;}

Classe: Uma classe é uma regra de “agente livre” que pode ser aplicada a qualquer tag HTML de acordo com a sua vontade. Você pode dar à classe o nome que quiser. Uma classe é o tipo de seletor mais versátil. Exemplo:

.minhaclasse{fontbold 12pt times;}

ID: As regras de ID funcionam como os seletores de classe, porque podem ser aplicadas a qualquer *tag* HTML. Os seletores de ID, porém, geralmente são aplicados somente uma vez na página a determinada TAG HTML para criar um objeto para ser usado com uma função JavaScript. Exemplo:

#objeto{fontbold 12pt times;}

5.3.3 Incluindo CSS no documento

Embora a CSS signifique nunca ter que definir a aparência de cada *tag* individualmente, você ainda tem a liberdade de definir os estilos dentro das *tags* individuais. Isso é particularmente útil para substituir cada um dos outros estilos que estão definidos para a página.

Veja o exemplo abaixo:

```
<html>
  <head>
    <title>CSS</title>
  </head>
  <body style="background-color:black;">
    <br>
    <h1 style="color:red">A Bruxa de Blair</h1>
    <h2 style="color:yellow">O Filme</h2>
    <BR>
    <p style="color:White">
```

```

o filme o filme o filme o filme o filme o filme
o filme o filme o filme <br>
o filme o filme o filme o filme o filme o filme
o filme o filme o filme o filme <br>
o filme o filme o filme o filme o filme o filme
o filme o filme o filme
</p>
</body>
</html>

```

Figura 5.8: Exemplo de inclusão de CSS no documento HTML

Fonte: Elaborada pelos autores

5.3.4 Incluindo a CSS em uma página web

Utiliza-se a CSS principalmente para definir as regras de um documento. Para fazer isso, você deve incluir as regras de estilo no título do documento aninhado dentro de um contêiner de estilo.

Embora os resultados do acréscimo de estilo dessa forma possam parecer idênticos ao acréscimo dos estilos diretamente em uma tag HTML, a colocação em uma localização comum permite alterar os estilos de um documento a partir de um único lugar.

Veja o mesmo exemplo acima como ficaria:

```

<html>
  <head>
    <title>CSS</title>
    <style type="text/css">
      <!--
        body{background-color:black;}
        h1{color:red;}
        h2{color:yellow;}
        p{color:white;}
      -->
    </style>
  </head>
  <body>
    <br>
    <h1>A Bruxa de Blair</h1>
    <h2>O Filme</h2>
    <BR>
    <P>
      o filme o filme o filme o filme o filme
o filme o filme o filme <br>
o filme o filme o filme o filme o filme
o filme o filme o filme o filme <br>
o filme o filme o filme o filme o filme
o filme o filme o filme
    </P>
  </body>
</html>

```

Figura 5.9: Exemplo de inclusão de CSS em uma página HTML

Fonte: Elaborada pelos autores

5.3.5 Incluindo a CSS em um *site da web*

Um dos benefícios da CSS é a possibilidade de criar uma folha de estilo para ser usada não apenas em um único documento HTML, mas em todo um *site da web*. Você pode aplicar essa folha de estilo externa a uma centena de documentos HTML.

O estabelecimento de um arquivo de CSS externo é um processo de duas etapas. Em primeiro lugar, configuram-se as regras em um arquivo de texto. Em seguida, configuram-se as regras em um arquivo e vincula-se ao documento HTML; o arquivo que contém as formatações deve possuir a extensão CSS.

Veja como codificar o arquivo CSS:

```
body{background-color:black;}
```

```
h1{color:red;}
```

```
h2{color:yellow;}
```

```
p{color:white;}
```

5.3.6 Principais atributos de uma folha de estilo

Apresentaremos os principais atributos da CSS para facilitar o seu uso em documento HTML.

Quadro 5.4: Relação dos atributos de posição

Elemento	Definição	Exemplo de Atributo
Position	Tipo de posicionamento.	<i>Absolute</i> ou <i>relative</i>
Top	Distância vertical para o canto superior esquerdo do quadro.	10in, 150px, 30 cm
Left	Distância horizontal para o canto superior esquerdo do quadro.	10in, 150px, 30 cm
Width	Largura do quadro.	10in, 150px, 30 cm
Height	Altura do quadro.	10in, 150px, 30 cm
Z-index	Camada usada quando se sobrepõem quadros.	1,2,3,4
Padding	Margem entre as bordas do quadro e o texto interno ao quadro.	2,5,10,17
Margin	Margem entre as bordas do quadro e o texto externo ao quadro.	1,2,3,4
Border-width	Largura da borda do quadro.	Thin, medium, thick
Border-color	Cor da borda do quadro.	Yellow,#00FFAA
Border-style	Estilo da borda.	None, dotted, dashed, solid, double
Border-width	Largura da borda do quadro.	Thin, medium, thick

Fonte: Elaborado pelos autores

Quadro 5.5: Relação de atributos de fontes

Elemento	Definição	Exemplo de atributo
Font-family	Fonte que será usada para mostrar o texto.	Helvetica, Arial, Courier
Font-size	Tamanho da fonte.	12pt, 10cm, 5in
Font-style	Estilo.	Normal, italic
Font-weight	Peso da fonte.	Bold, lighter, 100, 300

Fonte: Elaborado pelos autores

Quadro 5.6: Relação de atributos e cores de fundo da página

Elemento	Definição	Exemplo de atributo
Color	Cor do texto.	Marrom, #ffffcc
Background-color	Cor de fundo.	Marrom, #ffffcc
Background-image	Imagem de fundo.	Imagem.gif

Fonte: Elaborado pelos autores

Quadro 5.7: Relação de atributos de texto

Elemento	Definição	Exemplo de atributo
Word-spacing	Espaçamento entre as palavras.	1 em, 5pt
Letter-spacing	Espaçamento entre as letras.	0,1 em, 0,1cm, 2 pt
Text-decoration	Decoração de texto.	None, underline, blink
Vertical-align	Alinhamento vertical.	Middle, top, sub, super
Text-align	Alinhamento horizontal.	Left, center, right
Text-height	Altura da linha.	200%, 1.2, 20pt

Fonte: Elaborado pelos autores

5.3.7 Definindo classes

O uso de seletor de classes permite configurar um estilo independente que você pode aplicar em seguida a qualquer *tag* HTML

Para definir um seletor de classe, digita-se um ponto (.) e um nome de classe; em seguida, abre-se a sua definição com uma chave ({}).

O nome da classe pode ser qualquer um, desde que se usem letras e números. Por exemplo, podemos criar uma classe chamada **.copy{**.

Ela é uma classe independente, que você pode usá-la com qualquer *tag* HTML, com uma condição: as propriedades definidas para a classe devem funcionar com o tipo de *tag* que você usa.

Para aplicar a sua classe a uma *tag* HTML, inclua `class="nome"` na *tag* na qual você deseja aplicar a classe.

Exemplo:

```
<p class="copy">...</p>
```

5.3.8 Definindo IDs

Assim como o seletor de classe, o seletor de ID pode ser usado para criar estilos exclusivos que são independentes de qualquer *tag* HTML em particular. Assim sendo, eles podem ser designados a qualquer *tag* HTML que se aplique.

As regras de ID sempre começam com (#) e, em seguida, o nome do ID. O nome pode ser uma palavra ou qualquer conjunto de letras ou números que você quiser. Digite as suas definições para essa classe, separando-as com um ponto e vírgula. Você pode usar um ID com qualquer tipo de propriedade, mas o melhor uso para os seletores de ID é para definir os objetos exclusivos na tela.

Um ID não funcionará até ser especificada uma *tag* HTML individual dentro de um documento.

Exemplo:

```
#area{color:red;margin-left:9em;position:relative;}
```

```
<P id="area">...</p>
```

5.3.9 Definindo tags dentro do contexto

Quando uma *tag* é cercada por outra, ficando uma dentro da outra, são chamadas de *tags* aninhadas. Em um conjunto aninhado, a *tag* externa é chamada de pai, e a interna, de filha.

Digite o seletor HTML da *tag* pai seguida por um espaço. Você pode digitar quantos seletores HTML quiser, para quantos pais e *tags* aninhadas diferentes tiver, mas o último seletor da lista é aquele que recebe todos os estilos da regra. Veja o exemplo abaixo:

```
<html>
  <head>
    <title>Tags Aninhadas</title>
    <style type="text/css">
      <!--
        p a:link{color:red;
          text-decoration:underline;}
        div.menu{color:#900;
          font-weight:bold;
          text-decoration:none;
          font-size:20;}
        p{font:12px;}
      -->
    </style>
  </head>
  <body>
    <div class="menu">
      <a href="anterior.htm">&lt; Capitulo Anterior</a>
      <a href="proximo.htm">Próximo Capítulo &gt;</a>
    </div>
```

```

        <h3>Novo Capítulo</h3>
        <p>Teste de parágrafo, Teste de parágrafo, Teste de pa
parágrafo, Teste de parágrafo,
        <a href="curioso.htm">Efeito curiosos</a></p>
    </body>
</html>

```

Figura 5.10: Exemplo de CSS dentro do contexto

Fonte: Elaborada pelos autores

5.3.10 Criando uma definição “!important”

O valor **!important** pode ser incluído em uma definição para fornecer o máximo de peso na determinação da ordem em cascata.

Para forçar que uma definição seja sempre usada, abra uma regra da CSS com um seletor e uma chave ({}), digite uma definição de estilo, um espaço, **!important** e um ponto e vírgula para fechar a definição.

Atenção: O navegador Netscape 4 não suporta **!important**.



Exemplo:

```
p{ font-size:12px !important;}
```

5.3.11 Definindo a CSS para impressão

Quando a maioria das pessoas pensa em páginas *web*, pensa nessas páginas exibidas na tela. Entretanto, mais cedo ou mais tarde, a maioria das pessoas quer imprimir pelo menos algumas páginas da *web*. O que aparece bem na tela nem sempre é bom quando impresso.

A CSS permite dizer ao *browser* para usar folhas de estilo diferentes, dependendo se a página da *web* se destina ao monitor do computador ou à impressora.

Crie duas folhas de estilo (arquivos.css), uma otimizada para o uso em uma tela de computador e outra adaptada para a impressão. Veja os códigos abaixo:

```

body{
color:white;
font-family:Arial;
background:black url(images/backarvore.jpg) no-repeat;}
h1,h2{
font-weight:bold;}
.casseea{
color:#999999;}
CSS para Impressão:
body{
color:black;
font-family:Arial;
h1,h2{
font-weight:bold;}
.casseea{
color:#999999;}
Código no HTML:

```

```

<html>
  <head>
    <title>Mídias Diferentes</title>
    <link rel="stylesheet" href="tela.css" media="screen">
    <link rel="stylesheet" href="impressao.css" media="print">
  </head>
  <body>
    <h1> Mídia de tela </h1>
    <br>
    <span class="classea">O que pode ser visto na tela será diferente
    na impressão</span>
    <hr>
    <h2>Teste</h2>
  </body>
</html>

```

Figura 5.11: Exemplo de CSS para impressão

Fonte: Elaborada pelos autores

5.3.12 Definindo segundo plano

Podemos usar a propriedade *background* para definir a imagem e a cor do segundo plano de toda a página ou a imagem e cor de segundo plano imediatamente atrás de cada elemento individual da página.

Para definir o segundo plano, iniciamos a digitação com a propriedade **background** seguida de dois pontos e, em seguida, dos valores de segundo plano:

White → Digite um valor para a cor que você deseja para o segundo plano seguido de um espaço. Esse valor pode ser o nome da cor, um valor hexadecimal de cor ou um valor RGB.

url(imagem/figura.jpg) → Para utilizar uma figura de fundo, podemos definir a localização do arquivo. Alternativamente pode-se usar *none* em vez de url; ele instrui o *browser* a não utilizar uma imagem de segundo plano.

Repeat → Define qual será o modo de repetição a ser utilizado:

Repeat-x: Instrui o *browser* para repetir o gráfico de segundo plano na horizontal.

Repeat-y: Instrui o *browser* para repetir o gráfico de segundo plano na vertical.

No-repeat: Faz com que o gráfico de segundo plano apareça apenas uma vez.

Fixed: Instrui para que a figura fique fixa, ou seja, não rola com o *site*.

Scroll: Instrui para que a imagem role junto com a tela.

Right top: Digitam-se dois valores separados por um espaço para especificar o lugar onde o segundo plano deve aparecer em relação ao canto superior esquerdo do elemento.

Exemplo:

```
<html>
  <head>
    <title>Cores e planos de fundo</title>
  </head>
  <style type="text/css">
    <!--
      body{background:black url(images/astalavistagrouplogo3.jpg)
        no-repeat fixed center;
        color:white;}
      h2{background-color:yellow;color:red;}
      .copia{background-color:orange;color:brown;}
    -->
  </style>
  <body>
    <h2> testando cores de preenchimento</h2>
    <hr>
    <span class="copia">texto </span>
  </body>
</html>
```

Figura 5.12: Exemplo de CSS para imagem de plano de fundo

Fonte: Elaborada pelos autores

5.3.13 Bordas

Para definir um atributo de borda para uma caixa simultaneamente, a CSS fornece a propriedade `border`; você pode usar `border` para definir a largura, o estilo e/ou a cor. Também é possível definir a borda do elemento em cada lado da caixa individualmente.

As principais propriedades para bordas são: **`border-width`**, **`border-style`**, **`border-color`**.

Quadro 5.8: Relação de propriedades das bordas em CSS

Boderstyle	Borderwidth
None	Thin
Dotted	Medium
Dashed	Tick
Solid	Lenght
Double	
Groove	
Ridge	
Inset	
Outset	

Fonte: Elaborada pelos autores

Exemplo:

```
<html>
  <head>
    <title>Untitled</title>
    <style type="text/css">
      <!--
        .inset{border-style:inset;
              border-color:red;
              border-width:15px}
        .double{border-style:double;
              border-color:orange;
              border-width:10px;}
        td{text-align:center;}
      -->
    </style>
  </head>
  <body>
    <table class="inset">
      <tr>
        <td colspan=2>Célula 1</td></tr>
        <tr><td>abaixo</td><td>abaixo e à direita</td></tr>
      </table>
    <p class="double">Texto envolvido por bordas<br>com duas linhas</p>
  </body>
</html>
```

Figura 5.13: Exemplo de bordas em CSS

Fonte: Elaborada pelos autores

5.3.14 Envolvendo um elemento com texto

No início do desenvolvimento HTML, quando a capacidade de fazer o texto flutuar ao redor de um gráfico foi incluída, os *designers* de toda parte estavam maravilhados. A CSS levou esse passo um pouco adiante, permitindo que o texto não apenas flutue ao redor do texto, mas também que flutue ao redor dos outros blocos de texto e do texto ao redor das tabelas; para isso usa-se a propriedade *float*.

Inicie sua definição digitando a propriedade *float* seguida de dois pontos. A seguir digite uma palavra-chave para dizer ao *browser* o lado da tela no qual o elemento deve flutuar, as quais podem ser: *right*, *left* e *none*. Por exemplo:

```
Img{float:right;}
```

5.4 Posicionamentos

Quando você define os atributos de uma *tag* HTML, por meio de um seletor em uma CSS, na verdade, você separa daquela *tag* todo o conteúdo dentro de um contêiner, como um elemento exclusivo da janela. Em seguida é possível manipular esse elemento exclusivo por meio do posicionamento da CSS.

Um elemento pode ter um de quatro valores de posicionamento – estático, relativo, absoluto ou fixo – embora apenas os três primeiros normalmente estejam disponíveis na maioria dos *browsers*. O tipo de posicionamento diz ao *browser* como ele deve tratar o elemento ao posicioná-lo na janela.

5.4.1 Posicionamento estático

Como padrão, os elementos são posicionados como estáticos dentro da janela, a menos que você os defina com os outros posicionamentos. Entretanto, o posicionamento estático é diferente, porque um elemento estático não pode ser posicionado ou reposicionado explicitamente.

5.4.2 Posicionamento relativo

Um elemento definido como posicionado relativamente flui até o seu lugar dentro da janela ou dentro de seu elemento pai, assim como o comportamento padrão de qualquer outro elemento do HTML – ou seja, ele aparece após tudo que está antes dele no HTML e antes de tudo que está após ele no HTML.

Você pode mover um elemento posicionado relativamente a partir de sua posição natural na janela usando as propriedades *top* e *left*. Essa técnica é útil para controlar o modo como os elementos aparecem com relação aos outros elementos da janela.

5.4.3 Posicionamento absoluto

O posicionamento absoluto cria um elemento independente – um agente livre – separado do restante do documento, no qual você pode colocar qualquer tipo de conteúdo HTML que quiser. Os elementos que são definidos dessa forma são colocados em um ponto exato da janela por meio das coordenadas *x* e *y*. O canto superior esquerdo da janela ou de seu elemento contêiner é a origem.

5.4.4 Posicionamento 3D

Apesar de a área da tela ser bidimensional, os elementos que são posicionados podem receber uma terceira dimensão: uma ordem de empilhamento na qual um elemento se relaciona ao outro.

Os elementos posicionados recebem números automáticos de empilhamento, a partir do 0 (zero) incrementalmente com 1,2,3 e assim por diante. Esse sistema é chamado de índice Z. O número de índice Z de um elemento é um valor que mostra sua relação 3D com os outros elementos da janela.

Para definir o índice Z de um elemento, primeiro defina o posicionamento (ex.: `absolute`) depois `z-index:nº;`.

Veja u exemplo na Figura 5.14 a seguir.

```
<html>
<head>
  <title>Posicionamento 3D</title>
  <style type="text/css">
    <!--
      BODY{font-family:Arial;font-size:11pt;background-color:yellow;color:black;}
      .titulo1{position:absolute;top:20px;left:20px;z-index:1;font-family:Arial Black;
        color:navy;font-size:28pt;}
      .titulo2{position:absolute;top:17px;left:17px;z-index:2;font-family:Arial Black;
        color:blue;font-size:28pt;}
      .titulo3{position:absolute;top:14px;left:14px;z-index:3;font-family:Arial Black;
        color:cyan;font-size:28pt;}
      .box2{position:absolute;top:185px;left:60px;z-index:2;width:70px;font-family:Times;background-
color:blue;
        color:white;font-size:10pt;font-style:italic;font-weight:bold;padding:15}
      .box3{position:absolute;top:90px;left:150px;z-index:3;width:200px;font-family:Courier New;
background-color:navy;
        color:white;font-size:13pt;font-style:italic;font-weight:bold;padding:15;text-align:center
;}
      B{color:yellow;font-size:16pt;}
      a{color:yellow;text-decoration:none;}
      a:visited{color:white;background-color:black;}
    -->
  </style>
</head>
<body>
  <span class="titulo1">Posicionamento 3D</span>
  <span class="titulo2">Posicionamento 3D</span>
  <span class="titulo3">Posicionamento 3D</span>
</span>
<span class="box2">Nesta folha de estilo forma utilizados posicionamento,
IDS e classes e utilizamos o Span que chama as IDS e as classes</span>
<span class="box3">Para saber mais sobre folhas de estilo acesse o site
da <a href="http://www.w3.org/TR/REC-CSS2">W3</a> onde você poderá até mesmo conseguir alguns
exemplos.</span>
</body>
</html>
```

Figura 5.14: Exemplo de posicionamento em CSS

Fonte: Elaborada pelos autores

5.4.5 Definindo a visibilidade de um elemento

A propriedade *visibility* designa se um elemento é visível quando está inicialmente na janela. Se a visibilidade estiver definida como *hidden*, o elemento é invisível, mas ainda ocupa espaço no documento, e um grande retângulo vazio aparece no lugar do elemento.

Exemplo:

```
.hide{position:relative;
visibility:hidden;}
```

Resumo

Nesta aula conhecemos o uso de *frames* para melhor dividir a tela do computador e estruturar de forma mais elegante e organizada os documentos HTML. Aprendemos a utilizar as estruturas de folhas de estilos, mais conhecidas como CSS, recurso que nos permite criar padrões de telas e *layouts* mais sofisticados para atender de forma satisfatória às novas tendências dos *sites* dinâmicos.

Portanto, agora você já está apto a criar suas próprias páginas HTML, com base no conteúdo abordado nesta disciplina. Aproveite e ganhe o mundo.

Atividades de aprendizagem

1. Analisando o conteúdo estudado, defina para que serve a utilização de *frames*.
2. Demonstre a estrutura básica de um documento HTML utilizando *frame*.
3. Comente por que a *tag* <BODY> não é utilizada quando se trabalha com *frames*.
4. Existem três etapas na regra de criação dos seletores de um arquivo CSS; defina cada um deles.
5. Construa um arquivo CSS exemplificando cada um dos tipos de seletores.

Referências

ALMEIDA, Rafael Soares de. **Aprendendo HTML com o PlainHTML 7 para Iniciantes**. Rio de Janeiro: Ciência Moderna, 2010. 216 p.

CAMARGOS, Luiz Fernando Macedo; MENEZES, Marco Antônio Figueiredo. **Introdução à HTML e PHP**. Rio de Janeiro: Ciência Moderna, 2008.

CERN – European Laboratory for Particle Physics. **An overview of the world wide web: history and growth**. 1997. Disponível em: <<http://www.cern.ch/Public/CHIEVEMENTS/WEB/history.html>>. Acesso em: 21 dez. 2012.

DANNY GOODMAN. **Cookie recipes**: client side persistent data. Disponível em: <http://developer.netscape.com/viewsource/goodman_cookies.html>. Acesso em: 21 dez. 2012.

DANNY GOODMAN. **Java Script Object Road map and compatibility guide**. 1997. Disponível em: <<http://www.dannyg.com/update.html>>. Acesso em: 21 dez. 2012.

ECMA General Assembly. **ECMA-262**: ECMA script language specification (ISO/IEC16262). 1998. Disponível em: <<http://www.ecma.ch/stand/ecma-262.htm>>. Acesso em: 21 dez. 2012.

GORDON McComb. Beginner's JavaScript. **Java World Magazine**, march 1996. Disponível em: <<http://www.javaworld.com/javaworld/jw-03-1996/jw-03-javascript.intro.html>>. Acesso em: 21 dez. 2012.

GORDON McComb. New Java Script Features in Navigator 3.0. **Java World Magazine**, october 1996. Disponível em: <<http://www.javaworld.com/javaworld/jw-10-1996/jw-10-javascript.html>>. Acesso em: 21 dez. 2012.

GORDON McComb. Frequently sought solutions in JavaScript. **JavaWorld Magazine**, november 1996. Disponível em: <<http://www.javaworld.com/javaworld/jw-11-1996/jw-11-javascript.html>>. Acesso em: 21 dez. 2012.

HELDER L. S. da Rocha. **[9] Netscape Corporation**: Java Script Reference (1.3). 1998. Disponível em: <<http://developer.netscape.com/docs/manuals/communicator/jsref/>>. Acesso em: 21 dez. 2012.

MICROSOFT CORPORATION. **Jscript 4.0/5.0beta**: reference and tutorial. 1998. Disponível em: <<http://www.microsoft.com/scripting/>>. Acesso em 21 dez. 2012.

MICROSOFT CORPORATION. DHTML. **Document object model**. 1998. Disponível em: <<http://www.microsoft.com/workshop/author/dhtml/reference/objects.asp>>. Acesso em: 21 dez. 2012.

NETSCAPE CORPORATION. **JavaScript Guide (1.3)**. 1998. Desenvolvendo Web sites interativos com Java ScriptA-2 JS11-01-1999/01 A4. Disponível em: <<http://developer.netscape.com/library/documentation/communicator/jsguide4>>. Acesso em: 21 dez. 2012.

NETSCAPE CORPORATION. **JavaScript 1.0: Authoring Guide**. 1995. Disponível em: <<http://home.netscape.com/eng/mozilla/2.0/handbook/javascript/>>. Acesso em: 21 dez. 2012.

NETSCAPE CORPORATION. **JavaScript Sample Code**. 1998. Disponível em: <<http://developer.netscape.com/library/examples/javascript.html>>. Acesso em: 21 dez. 2012.

NETSCAPE CORPORATION. **JavaScript (1.1) Guide for Netscape 3.0**. 1996.

REAZ HOQUE. Getting Started with JavaScript's 13 Event Handlers. **Netscape World Magazine**, December, 1996. Disponível em: <<http://www.netscapeworld.com/netscapeworld/nw-12-1996/nw-12-javascript1.html>>. Acesso em: 12 dez. 2012.

RECHS, Matthew J.; SIRIGOS, Angelo; WILLIAMS, Nick. **DevEdge Newsgroup FAQ: JavaScript**. Netscape Corporation. 1998. Disponível em: <<http://developer.netscape.com/support/faqs/champions/javascript.html>>. Acesso em: 21 dez. 2012.

REIN, Lisa; SPELMAN, Jennifer. ECMA script in a nutshell: our guide to the new specification. **Netscape World Magazine**, July 1997. Disponível em: <<http://www.netscapeworld.com/nw-07-1997/nw-07-javascript.html>>. Acesso em: 21 dez. 2012.

ROSENFELD, L.; MORVILLE, P. **Information architecture for the word wide web**. 2 ed. Sebastopol: O'Reilly, 2002.

SPAINHOUR, Stephen; QUERCIA, Valerie. **WebMaster in a NutShell**: a desktop quick reference. O'Reilly and Associates, 1996.

STANEK, William R. Netscape **ONE (Open Network Environment)**: developer's guide. SamsNet/Macmillian Computer Publishing, 1997.

TIM BERNERS-LEE. Information management: a proposal. **CERN**: European Laboratory for Particle Physics, Mar. 1989- May 1990. Disponível em: <<http://www.w3.org/History/1989/proposal.html>>. Acesso em: 21 dez. 2012.

WORLD WIDE WEB CONSORTIUM – **W3C. Document Object Model Specification**. 1998. Disponível em: <http://www.w3.org/DOM/>>. Acesso em: 21 dez. 2012.

WORLD WIDE WEB CONSORTIUM – **W3C. HTML 3.2/4.0 Specifications**. 1998. Disponível em: <<http://www.w3.org/pub/WWW/MarkUp/Wilbur/>>. Acesso em: 21 dez. 2012.

Currículo dos professores-autores

Otilio Paulo da Silva Neto

Mestrando em Computação pela Universidade Federal Fluminense (UFF), é bacharel em Sistemas de Informação pela Faculdade Integral Diferencial, pós-graduado em Docência do Ensino Superior e atualmente é diretor da Netsoft Tecnologia Ltda, professor titular do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, professor da Faculdade R. Sá, professor substituto da Universidade Estadual do Piauí e analista de sistemas da Fundação José Elias Tajra. Tem experiência na área de Ciência da Computação, com ênfase em Banco de Dados e Programação para *Web*, atuando principalmente nos seguintes temas: Controle de versão semântica de modelos, netsac, patrimônio, procedimentos médicos, plano de saúde e NetProcess – Sistema de controle de processos totalmente via *web*, Aplicações para dispositivos móveis e Ekids – uma ferramenta no auxílio da aprendizagem de crianças.

Nádia Mendes dos Santos

Bacharel em Ciências Contábeis pela Universidade Federal do Piauí, Campus Ministro Reis Velloso, em Parnaíba, e tecnóloga em Processamento de Dados pela Universidade Estadual do Piauí, Campus de Parnaíba. Possui pós-graduação *lato Sensu* em Banco de Dados, pelo Centro Federal de Educação do Piauí. Atualmente cursa Mestrado em Engenharia Elétrica pela Universidade Federal do Ceará, linha de pesquisa em Inteligência Artificial. Desde 2006, é professora do quadro efetivo do Instituto Federal de Ciência e Tecnologia do Piauí, Campus Angical, na categoria de Dedicação Exclusiva, onde exerce a função de coordenadora do Eixo Informática e Comunicação. Seus principais interesses de pesquisa e atuação são inteligência artificial, estrutura de dados, eletrônica digital, microprocessadores digitais, banco de dados e linguagens de programação. Faz parte da Equipe de Educação a Distância do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, atuando como tutora a distância do polo de Valença. Atua também como professora conteudista.

Sandra Elisa Veloso Aguiar

Tecnóloga em Processamento de Dados, tem formação superior em Psicologia. É atualmente professora titular do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Teresina Central. Tem experiência na área de Ciência da Computação, com ênfase nas disciplinas de fundamentos em informática nos cursos técnicos, técnico integrado e nível superior.