

**CENTRO PAULA SOUZA**

GOVERNO DO ESTADO DE  
**SÃO PAULO**

**Faculdade de Tecnologia de Americana  
Curso de Processamento de Dados**

# **PADRÕES PARA DESENVOLVIMENTO WEB**

**RODRIGO TURATTI CASSIANO**

Americana, SP  
2010

**CENTRO PAULA SOUZA**

**GOVERNO DO ESTADO DE  
SÃO PAULO**

**Faculdade de Tecnologia de Americana  
Curso de Processamento de dados**

## **PADRÕES PARA DESENVOLVIMENTO WEB**

**RODRIGO TURATTI CASSIANO**

**rodturatti@gmail.com**

**Estágio em Análise e Projetos de sistemas – ESTANAL, desenvolvido em cumprimento à exigência curricular do Curso de Processamento de Dados da Faculdade de Tecnologia de Americana, sob orientação da Prof. Humberto Celeste Innarelli.**

**Área: Desenvolvimento Web**

**Americana, SP  
2010**

**BANCA EXAMINADORA**

**Prof. MSc. Humberto Innarelli (Orientador)**

**Profa. Maria Elizete Luz Saes (Convidada)**

**Prof. Antônio Alfredo Lacerda (Presidente da banca)**

## **AGRADECIMENTOS**

Em primeiro lugar à minha família, principalmente aos meus pais, por nunca deixarem faltar nada e por darem total apoio e incentivo aos meus estudos, sempre se sacrificando para me oferecerem o melhor;

Aos professores da FATEC Americana, por transmitirem seus conhecimentos durante toda a minha vida universitária, e principalmente aos professores Humberto, por todo apoio, orientação e paciência para que este trabalho fosse possível, e Elizete, por fazer mais do que o necessário para ajudar seus alunos;

Aos funcionários da FATEC Americana, por toda a ajuda e dedicação no nosso dia-a-dia;

E finalmente, aos meus amigos de classe, que por tanto tempo dividiram momentos que levarei comigo para a vida toda.

## DEDICATÓRIA

À minha noiva Marina, uma pessoa muito especial que entrou em minha vida para me ensinar o que é o amor, estando ao meu lado em todos os momentos, seja dividindo os de alegria e felicidade ou me ajudando e estando ao meu lado nos de dificuldade, dando todo carinho, apoio e incentivo para minha vida profissional;

À minha tia Nair, que sempre sonhou (mas infelizmente não deu tempo) em me ver ingressar em uma faculdade.

## RESUMO

A internet tem se mostrado uma ferramenta de vital importância na vida do ser humano, nessas duas últimas décadas. Desde sua invenção, restrita apenas aos meios acadêmicos, tornou-se o principal método de pesquisa existente, além de conectar pessoas de diferentes partes do mundo. O que era apenas um dispositivo de troca de informações se transformou em um dos instrumentos mais utilizados de todos os tempos, o que trouxe para o usuário a possibilidade de estabelecer interações profissionais, culturais, sociais e de entretenimento como: ouvir músicas, assistir a filmes, fazer compras e ler livros em formato digital, o que representou, sem dúvida, um enorme avanço para a humanidade. Entretanto, a revolução das mídias informacionais e comunicacionais associada às constantes inovações tecnológicas podem desencadear também a exclusão digital de potenciais usuários dessas tecnologias. Por essa razão, surgiu a necessidade de uma padronização da rede mundial de computadores, para que as informações pudessem ser acessíveis para todas as pessoas, principalmente aos portadores de necessidades especiais, independentemente de sistema operacional, navegadores ou dispositivos utilizados, trazendo assim a ideia de acessibilidade para todos.

**Palavras Chave:** internet, padronização, acessibilidade.

## ABSTRACT

The internet has shown itself to be a tool of vital importance in the life of the human being in the last two decades. Since its invention, limited to academic use, has become the existing main research method, and also has connected people from different parts of the world. What was supposed to be just an information exchange device has turned into one of the most used instruments of all time, bringing to the user the possibility of establish professional, social, cultural and entertainment interactions such as listening to music, watching movies, shopping online and reading e-books, that represented, without any doubt, a huge advance for humanity. However, the revolution of information and communication media associated with the technological innovations can also bring about the digital exclusion of potential users of these technologies. Therefore the necessity of a standardization of this computer world-wide web become an important factor, in order that the information could be accessible to everyone, particularly to those with special needs, regardless of the operational system, web browser or devices used, bringing out, in this way, the idea of accessibility for everyone.

**Keywords:** internet, standardization, accessibility.

## SUMÁRIO

<b>LISTA DE ABREVIATURAS E SIGLAS</b> .....	<b>13</b>
<b>1 INTERNET</b> .....	<b>16</b>
1.1 O INÍCIO .....	16
1.2 A LINGUAGEM HTML .....	17
<b>2 BROWSERS</b> .....	<b>18</b>
<b>3 METODOLOGIAS DE DESENVOLVIMENTO</b> .....	<b>21</b>
3.1 MÉTODO DE DESENVOLVIMENTO TRADICIONAL .....	21
3.2 POR UMA INTERNET MELHOR .....	24
3.3 MÉTODO DE DESENVOLVIMENTO SEGUINDO OS PADRÕES WEB .....	25
<b>4 ACESSIBILIDADE E USABILIDADE</b> .....	<b>30</b>
4.1 DEFINIÇÃO .....	30
4.2 DECLARAR UM TIPO DE DOCUMENTO .....	30
<b>4.2.1 TRANSITIONAL</b> .....	<b>31</b>
<b>4.2.2 STRICT</b> .....	<b>31</b>
<b>4.2.3 FRAMESET</b> .....	<b>31</b>
4.3 ESTRUTURA DA PÁGINA .....	32
4.4 IMAGENS .....	34
<b>4.4.1 BENEFICIANDO OS SITES DE BUSCA</b> .....	<b>34</b>
<b>4.4.2 BENEFICIANDO OS DEFICIENTES VISUAIS</b> .....	<b>36</b>
4.5 TEXTOS .....	36
<b>4.5.1 Cabeçalhos (Títulos)</b> .....	<b>36</b>
<b>4.5.2 PARÁGRAFOS</b> .....	<b>37</b>
<b>4.5.3 QUEBRA DE LINHA</b> .....	<b>37</b>
<b>4.5.4 NEGRITO E ITÁLICO</b> .....	<b>38</b>
<b>4.5.5 CORES</b> .....	<b>38</b>
<b>4.5.6 FORMULÁRIOS</b> .....	<b>39</b>
<b>5 CRIANDO UM SITE NOS PADRÕES WEB</b> .....	<b>42</b>



5.1	XHTML .....	42
5.2	CABEÇALHO E DECLARAÇÕES DO SITE.....	42
5.3	CRIANDO A ESTRUTURA DO SITE.....	43
5.4	DIV'S .....	44
5.5	FORMATAÇÃO DO SITE .....	47
<b>5.5.1</b>	<b>REFERENCIANDO UM ARQUIVO CSS .....</b>	<b>47</b>
<b>5.5.2</b>	<b>APLICANDO FORMATAÇÕES .....</b>	<b>47</b>
<b>5.5.3</b>	<b>LARGURA, ALTURA, MARGEM E POSICIONAMENTO .....</b>	<b>48</b>
<b>5.5.4</b>	<b>CRIAÇÃO DO MENU.....</b>	<b>50</b>
5.5.4.1	MENU VERTICAL .....	51
5.5.4.2	MENU HORIZONTAL.....	54
<b>5.5.5</b>	<b>CRIANDO AS DIV'S DO CONTEÚDO .....</b>	<b>55</b>
<b>5.5.6</b>	<b>TOPO DO SITE.....</b>	<b>57</b>
<b>5.5.7</b>	<b>RODAPÉ.....</b>	<b>59</b>
<b>5.5.8</b>	<b>CONFIGURANDO PÁGINAS INTERNAS .....</b>	<b>60</b>
<b>5.5.9</b>	<b>CRIANDO A PÁGINA DE FALE CONOSCO .....</b>	<b>61</b>
<b>6</b>	<b>TESTES E VALIDAÇÕES .....</b>	<b>67</b>
6.1	FIREFOX, CHROME, OPERA E SAFARI .....	68
6.2	INTERNET EXPLORER .....	69
6.3	OPERA MOBILE .....	71
<b>6.3.1</b>	<b>OPÇÕES DE VISUALIZAÇÃO .....</b>	<b>72</b>
6.4	VALIDANDO O SITE .....	74
<b>6.4.1</b>	<b>VALIDANDO O XHTML.....</b>	<b>75</b>
<b>6.4.2</b>	<b>VALIDANDO O CSS.....</b>	<b>76</b>
<b>7</b>	<b>CONSIDERAÇÕES FINAIS.....</b>	<b>78</b>
<b>8</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>80</b>
	<b>ANEXO .....</b>	<b>82</b>

## LISTA DE FIGURAS E DE TABELAS

Figura 1 - Lynx, um exemplo de navegador em modo texto .....	18
Figura 2 - Etapas desenvolvimento tradicional de um <i>website</i> .....	23
Figura 3 - O criador da Web e do W3C, TIM Berners-Lee .....	25
Figura 4 - Etapas do desenvolvimento de um website utilizando padrões.....	27
Figura 5 - Exemplo de site usando CSS.....	28
Figura 6 - Exemplo do mesmo site, mas com CSS diferente .....	28
Figura 7 - Exemplo de uma página com frames .....	32
Figura 8 - Código-fonte da página exibida na figura anterior.....	33
Figura 9 - Imagem sendo exibida no navegador .....	35
Figura 10 - Propriedade “Alt” quando a imagem não é carregada .....	35
Figura 11 - Propriedade “Alt” em imagens .....	36
Figura 12 - Resultado no browser do código anterior .....	37
Figura 13 - Exemplo da tag de parágrafo <p>.....	37
Figura 14 - Uso do recurso de quebra de linha ( ) .....	37
Figura 15 - Formatação em negrito e itálico .....	38
Figura 16 - Duas formas de inserir cores por HTML .....	38
Figura 17 - Código fonte de um formulário completo .....	39
Figura 18 - Visualização no browser do código da figura anterior .....	40

<b>Figura 19 – Wireframe do site a ser criado .....</b>	<b>44</b>
<b>Figura 20 - Divs topo_logo e topo_texto visualizadas no browser .....</b>	<b>50</b>
<b>Figura 21 - Propriedade "height" em todos os elementos.....</b>	<b>50</b>
<b>Figura 22 - Resultado da lista não-ordenada .....</b>	<b>51</b>
<b>Figura 23 - Resultado final do menu vertical usando apenas CSS.....</b>	<b>54</b>
<b>Figura 24 - Ilustração dos elementos border, padding, height e margin.....</b>	<b>54</b>
<b>Figura 25 - Resultado final da página inicial do nosso site.....</b>	<b>60</b>
<b>Figura 26 - Formulário de contato formatado com CSS .....</b>	<b>63</b>
<b>Figura 27 - Formulário de contato após a aplicação de classes nos botões.....</b>	<b>64</b>
<b>Figura 28 - Site nos navegadores Firefox, Chrome, Opera e Safari.....</b>	<b>69</b>
<b>Figura 29 - Problema com a margem no Internet Explorer 6 e 7.....</b>	<b>70</b>
<b>Figura 30 - Visualização do menu corrigido no Internet Explorer 6 e 7.....</b>	<b>71</b>
<b>Figura 31 - Nosso site sendo visualizado no Opera Mobile .....</b>	<b>72</b>
<b>Figura 32 - Ampliação de tela no Opera Mobile.....</b>	<b>73</b>
<b>Figura 33 - Visualização no modo "Mobile View" .....</b>	<b>74</b>
<b>Figura 34 – Validação com sucesso .....</b>	<b>75</b>
<b>Figura 35 – Validação com erro .....</b>	<b>76</b>
<b>Figura 36 - Correta validação da Folha de Estilo.....</b>	<b>76</b>
<b>Figura 37 - Validação incorreta da Folha de Estilo.....</b>	<b>77</b>

<b>Tabela 1 - Browsers mais utilizados até o mês de setembro de 2010 .....</b>	<b>19</b>
<b>Tabela 2 - Exemplo de uso correto de tabelas.....</b>	<b>34</b>
<b>Tabela 3 - Explicação dos elementos do formulário de contato .....</b>	<b>40</b>

## LISTA DE ABREVIATURAS E SIGLAS

<b>ARPA</b>	Advanced Research and Projects Agency - Agência de Pesquisas em Projetos Avançados
<b>ASP</b>	Active Server Pages
<b>CERN</b>	Centre European pour la Recherche Nucleaire - Organização Européia para a Investigação Nuclear
<b>CSS</b>	Cascading Style Sheets – Folhas de Estilo em Cascata
<b>DOM</b>	Document Object Model
<b>DTD</b>	Document Type Definition
<b>HTML</b>	HyperText Markup Language
<b>IBGE</b>	Instituto Brasileiro de Geografia e Estatística
<b>IE</b>	internet Explorer
<b>JSP</b>	Java Server Pages
<b>MIT</b>	Massachussets Institute of Technology
<b>MS-DOS</b>	MicroSoft Disk Operational System
<b>PHP</b>	PHP: Hypertext PreProcessor
<b>W3C</b>	World Wide Web Consortium
<b>WWW</b>	World Wide Web - Teia de Alcance Mundial
<b>XHTML</b>	eXtensible HTML
<b>XML</b>	eXtensible Markup Language

## INTRODUÇÃO

A *World Wide Web*, ou Teia de Alcance Mundial, é a parte da Internet que nos possibilita a navegação através de sites, envio de e-mails e trocas de informações com pessoas de diferentes lugares do mundo. Ela foi criada no início da década de 1990 pelo engenheiro britânico Timothy John Berners-Lee, nos laboratórios do CERN, na Suíça, com o objetivo único de facilitar a comunicação interna entre os funcionários e cientistas.

Com o passar de duas décadas, a invenção de Berners-Lee tem se tornado uma das ferramentas mais utilizadas de todos os tempos, trazendo para o usuário a possibilidade de realizar tarefas rotineiras através da tela de seu computador ou dispositivo móvel, como o envio de currículos para as empresas, ouvir músicas, assistir a filmes, realizar compras e fazer leitura de livros em formato digital.

O que de um lado torna essa revolução um enorme avanço para a humanidade, por outro torna as informações cada vez menos acessíveis para algumas pessoas, principalmente para as que possuem necessidades especiais, pois cada vez mais as empresas têm o objetivo de fazer sempre o melhor e focar na “maioria”, e se esquecem de que todos têm direito à acessibilidade, independentemente de localização e características pessoais, pois como diria o “pai” da web:

"O poder da Web está em sua universalidade. Acesso por todos, independentemente da deficiência é um aspecto essencial."  
(Tim Berners-Lee, Diretor do W3C e criador da World Wide Web)

O objetivo principal desta monografia é abordar todas as vantagens que um site proporciona ao se adaptar aos padrões impostos pela W3C, com a utilização de técnicas teóricas, exemplos práticos e um estudo de caso real da criação de um site acessível.

O trabalho foi estruturado em seis capítulos, seguindo a seguinte estrutura:

**Capítulo 1:** traz todo o conceito de internet, de como era no início e a criação da linguagem por trás de todos os *websites*: o *HyperText Markup Language* (HTML);

**Capítulo 2:** conceitua os *browsers*, como eram os primeiros navegadores e quais os principais do mercado;

**Capítulo 3:** aponta qual a diferença entre os dois métodos de desenvolvimento: o método tradicional e o novo método seguindo os padrões da W3C;

**Capítulo 4:** conceitua Acessibilidade e Usabilidade, trazendo os tipos de documentos utilizados para a declaração do site, a sua estrutura e os principais elementos da linguagem HTML;

**Capítulo 5:** realizado um estudo de caso onde é explicado, passo a passo, a criação de um site seguindo os Padrões Web;

**Capítulo 6:** é feito todo o processo de testes nos principais navegadores e de validação do site produzido no capítulo 5, utilizando ferramentas da W3C que verificam se o site criado está de acordo com os padrões impostos por ela para um site acessível;

**Capítulo 7:** se reserva às considerações finais com base no conteúdo teórico e prático através do estudo realizado.

## 1 INTERNET

### 1.1 O INÍCIO

"Não pensávamos em negócio, mas em internet como uma forma de comunicação global."

(Jerry Yang – co-fundador do Yahoo!)

Na década de 1960, nos períodos da Guerra Fria, durante a qual Estados Unidos e União Soviética realizavam uma disputa ideológica pela hegemonia mundial, o governo americano temia a realização de um ataque russo a suas bases militares, tornando-os muito vulneráveis. Foi então que se chegou à conclusão de que era preciso um método para manter a comunicação das bases militares americanas, mesmo que houvesse um ataque nuclear inesperado.

Os primeiros registros realizados através de redes foram feitos em 1962 no *Massachusetts Institute of Technology* (MIT), e nada mais eram do que memorandos que discutiam o conceito de uma “Rede Galáctica”, através da qual computadores seriam interconectados globalmente e todos poderiam acessar dados e programas de qualquer local rapidamente.

Em 1969, a empresa norte-americana ARPA<sup>1</sup> criou a ARPANET, por meio da qual os departamentos de pesquisas poderiam se conectar com as bases militares.

A ARPANET cresceu e se tornou a Internet, que foi baseada na ideia de uma rede de arquitetura aberta, podendo conectar dispositivos dos mais variados (computadores, celulares, satélites, etc.) de todas as partes do mundo.

No início da década de 1990, Tim Berners-Lee, com a ajuda de Robert Cailliau (ambos do CERN<sup>2</sup>, na Suíça) criaram a *World Wide Web* – Teia de Alcance Mundial – na qual foi criado o primeiro *Website*, tornando possível o envio de e-mails, navegação através de *browsers*<sup>3</sup> e liberando a internet para uso público.

(A História da Internet - <http://www.isoc.org/internet/history/brief.shtml>)

---

<sup>1</sup> Advanced Research and Projects Agency - Agência de Pesquisas em Projetos Avançados

<sup>2</sup> Centre European pour la Recherche Nucleaire - Organização Européia para a Investigação Nuclear

<sup>3</sup> Programa necessário para a navegação em websites; será visto com detalhes no capítulo 2



## 1.2 A LINGUAGEM HTML

Para que fosse possível a navegação através de um *browser*, seria necessária a utilização de uma linguagem, e foi então que Berners-Lee criou o HTML (*HyperText Markup Language* - Linguagem de Marcação de Hipertexto), uma linguagem exclusivamente para conteúdo, sem se preocupar com a formatação dos documentos. Segundo o criador da *Web*, o seu objetivo foi o de "dar suporte e de melhorar a nossa existência no mundo como uma rede." (Berners Lee, 1999, p. 123)

Com o gigantesco crescimento da internet, o comércio começou a voltar sua atenção para essa nova tecnologia, e o interesse por sites foram crescendo cada vez mais. Como o HTML não foi criado com a finalidade de um apelo visual, os desenvolvedores tiveram que criar novas técnicas para, a partir do HTML, melhorar o aspecto visual de seus *websites*.

Foi nesse momento que surgiu a ideia de realizar a diagramação de um site a partir de tabelas, fazendo assim com que os elementos ficassem melhores posicionados na página. O problema é que, agindo dessa maneira, o significado de "tabelas" estava sendo usado equivocadamente, pois seu uso deveria ser pura e exclusivamente para tabular dados, e não para diagramar sites.

A partir disso, a situação foi ficando cada vez pior, pois cada empresa estava criando seu próprio método de criação de páginas, cada navegador tinha uma forma para interpretar o HTML da página, e cada vez mais a internet estava ficando desorganizada.

Para resolver esse problema, em 1994 foi criado o *World Wide Web Consortium* (W3C), por Tim Berners-Lee, com o intuito de tentar organizar e "padronizar" o método de criação de páginas, fazendo assim com que uma página possa ser interpretada da mesma forma por todos os tipos de dispositivos.

O objetivo dessa monografia é exatamente mostrar quais são esses padrões, que fazem com que os *browsers* (tanto dos microcomputadores como de dispositivos móveis) possam exibir a mesma página de maneira correta. Para isso, o próximo capítulo explicará sobre os responsáveis pela exibição das páginas, os navegadores ou *Browsers*.

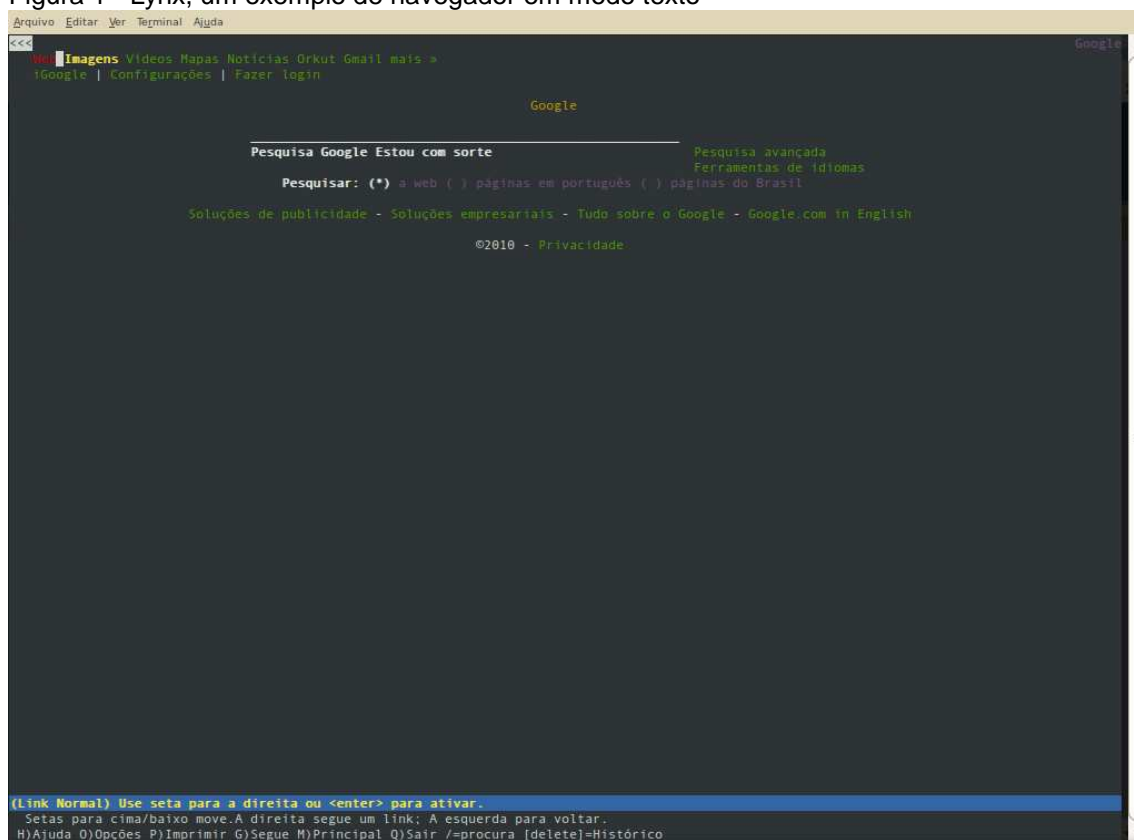
## 2 BROWSERS

Os primeiros navegadores eram todos em modo texto, com a aparência da tela parecida com a do sistema operacional MS-DOS, da Microsoft. Àquela altura, a internet era uma ferramenta mais empregada por acadêmicos, pesquisadores e cientistas, ficando de difícil acesso à “pessoas comuns”, principalmente em consequência da sua dificuldade de utilização. Mas, em abril de 1993, o primeiro grande navegador gráfico foi criado, o Mosaic, criado por Marc Andreessen, e foi a partir dessa data que a internet, particularmente a *World Wide Web*, começa a cair no encanto dos usuários leigos.

Segundo Andreessen:

“Naquela época, a internet era apenas para acadêmicos, navegadores e cientistas, e não havia nada que uma pessoa comum pudesse fazer nela, mas era tão útil para quem usava que eu achava que todos deveriam poder usar. (Andreessen, 1993)”

Figura 1 - Lynx, um exemplo de navegador em modo texto



Fonte: <http://www.dicasbrowser.com/2010/03/lynx-um-navegador-em-modo-texto.html> (17/09/2010)

No ano de 1995, é criado o primeiro grande navegador comercial, o Netscape, desenvolvido pela empresa *Netscape Communications*, criada por Marc Andreessen, um dos criadores do *Mosaic*. Instantaneamente, o *Netscape* torna-se um sucesso e adquire rapidamente a liderança dos *browsers* mais utilizados.

Em agosto do mesmo ano, a Microsoft, que ainda não tinha um navegador padrão, adquire o código fonte do *Mosaic* e desenvolve o seu navegador oficial, o *Internet Explorer* (IE), e deste ponto em diante, este *browser* vem incluído em todas as versões do Windows, o Sistema Operacional da empresa. O IE veio para o mercado com o objetivo de concorrer diretamente com a *Netscape* pela liderança dos *browsers* mais usados, mas apesar de toda a publicidade, o navegador de Andreessen manteve a liderança a princípio. Anos mais tarde, o *Netscape* foi ficando cada vez mais obsoleto, e o navegador da gigante Microsoft assumiu a liderança dos *browsers* mais usados. (Discovery Brasil, 2009)

A liderança da Microsoft seguiu firme sem nenhum adversário a altura, até que, no ano de 2004, a empresa Mozilla lançou o navegador Mozilla Firefox, que em pouco mais de três meses atingiu a incrível marca de 25 milhões de *downloads*, dando início à chamada “Segunda Guerra dos Navegadores”, devido à concorrência com o *Internet Explorer* e com outros *browsers*, como o Opera (da Opera Software) e o Safari (da empresa *Apple*).

O site da W3schools traz as estatísticas dos *browsers* mais usados mês a mês, como mostra a tabela a seguir:

Tabela 1 - Browsers mais utilizados até o mês de setembro de 2010

2010	<u>Internet Explorer</u>	<u>Firefox</u>	<u>Chrome</u>	<u>Safari</u>	<u>Opera</u>
September	31.1 %	45.1%	17.3%	3.7%	2.2%
August	30.7 %	45.8%	17.0%	3.5%	2.3%
July	30.4 %	46.4%	16.7%	3.4%	2.3%
June	31.0 %	46.6%	15.9%	3.6%	2.1%
May	32.2 %	46.9%	14.5%	3.5%	2.2%
April	33.4 %	46.4%	13.6%	3.7%	2.2%
March	34.9 %	46.2%	12.3%	3.7%	2.2%
February	35.3 %	46.5%	11.6%	3.8%	2.1%
January	36.2 %	46.3%	10.8%	3.7%	2.2%

Fonte: [http://www.w3schools.com/browsers/browsers\\_stats.asp](http://www.w3schools.com/browsers/browsers_stats.asp) - 08/10/2010

Conhecendo o responsável por fazer as páginas serem exibidas, é chegada a hora saber o porquê da mudança proposta, e qual a diferença entre os métodos de desenvolvimento.

### 3 METODOLOGIAS DE DESENVOLVIMENTO

#### 3.1 MÉTODO DE DESENVOLVIMENTO TRADICIONAL

No início, a Internet utilizava o HTML como uma linguagem visual, usado para desenvolver tanto o layout como o conteúdo da página. Para criar uma maior interação com o usuário, é empregada a linguagem de comportamento JavaScript, muitas vezes até como sendo essencial para poder acessar determinado site, como os sites de agências bancárias, que em quase sua totalidade fazem uso desse recurso. O fato é que alguns *browsers* não tinham o suporte necessário para essa tecnologia, acontecendo de o usuário não conseguir acessar o site devido a essa carência em seu navegador.

Além disso, diversos problemas começaram a aparecer nas etapas de desenvolvimento de um site, como os descritos a seguir:

Desenvolvimento: O método de desenvolvimento era visual e utilizava ferramentas como o *Dreamweaver* (que na época era da empresa Macromedia, hoje adquirida pela Adobe) ou o FrontPage da Microsoft, ambos softwares proprietários. Esses softwares geravam todo o código HTML e JavaScript automaticamente, dificultando assim o entendimento do código por um usuário que não dispunha de uma ferramenta visual e tornando o desenvolvimento lento e complexo.

Criação de Layouts: Os *layouts* eram todos baseados em tabelas e a partir delas era desenvolvida toda a estrutura da página.

Incompatibilidade: os sites eram criados apenas para funcionarem em determinados browsers e até em determinados Sistemas Operacionais. Por exemplo, muitas vezes um usuário que utiliza o sistema operacional Linux já passou pela árdua experiência de, ao tentar acessar um site, se deparar com a seguinte mensagem:

“O seu navegador não é suportado. É necessário ter o Internet Explorer para uma correta visualização da página”
---

O grande problema do usuário, nesse caso, é que o Linux não tinha suporte ao *browser* da Microsoft, fazendo com que o usuário não conseguisse acessar essa página.

Dispositivos móveis: Celulares, *Smartphones* e *Palmtops* tornavam-se cada vez mais populares, porém os usuários não conseguiam acessar corretamente algumas páginas devido a problemas de incompatibilidade e tamanho das mesmas.

Resolução: a página era feita apenas para um tipo de resolução de tela, sendo que se tivesse sido projetada para 1024 X 768 pixels e o usuário utilizasse 800 X 600 pixels, certamente teria que usar a barra de rolagem horizontal para ver todo o conteúdo da mesma.

Inclusão digital: as páginas eram todas feitas com animações e imagens, usando textos apenas nos conteúdos. O grande problema disso é que os leitores de tela, usados pelos deficientes visuais, só lêem texto, não sendo capazes de captar outros recursos. Um exemplo do quanto isso era prejudicial consiste no fato de que a maioria dos sites antigos utilizava imagens para fazer os itens do menu, sem empregarem alguma indicação de texto nelas, fazendo com que um deficiente visual usando um leitor de tela simplesmente não tivesse acesso às sessões do site.

Ruim para os buscadores: o excesso de imagens e animações também é ruim para os navegadores, pois estes fazem a indexação dos sites exatamente pelos textos que eles contêm.

Guerra entre designers e programadores (Acoplamento): no desenvolvimento tradicional, não havia separação entre o layout e a programação. Sendo assim, para alterar algum componente em um site, era necessário fazer a alteração em todos eles, tornando o trabalho “seqüencial”, transformando o que seria uma simples manutenção no design em um “redesign”, tendo que começar praticamente do zero. Como exemplo disso, pode-se explicar quais as etapas de desenvolvimento de um website: em primeiro lugar, é preciso um levantamento junto ao cliente de todas as informações que ele julga importantes para aparecerem no seu site, bem como pesquisa de mercado e de concorrentes. A esse processo dá-se o nome de Briefing. Passada essa etapa de Arquitetura de Informação, é o momento de elaborar o Layout, no qual o designer faz todo o desenho do layout do site em uma ferramenta

gráfica (como o PhotoShop ou o Fireworks, ambos da Adobe), e manda para aprovação. Uma vez aprovada pelo cliente (este, inclusive, assinava um documento dizendo que aprovou o layout e que este não seria modificado), esse layout é encaminhado para o programador, que começa a transformar o desenho em site, introduzindo outras linguagens de programação (como o PHP, JSP e ASP.NET) para tornar o site dinâmico. Após essa etapa, ele é apresentado ao cliente para aprovação final. Entretanto, frequentemente, o cliente muda de ideia e pede uma pequena alteração no layout, gerando o problema de que, como o site é todo sequencial, qualquer modificação no layout levaria a uma modificação na estrutura completa da página, desde seu desenho até a programação, de tal forma que até mesmo as “pequenas” mudanças demorariam dias. Todo esse processo pode ser visualizado na ilustração abaixo.

Figura 2 - Etapas desenvolvimento tradicional de um *website*



Fonte: Palestra da Atípico de 2007 – [www.tableless.com.br](http://www.tableless.com.br) – 01/10/2010

Código descentralizado: como tudo era feito no HTML, desde a estruturação do site até a sua programação, qualquer mudança em um bloco deveria ser feita em todas as páginas. Por exemplo, ao querer modificar a cor de todos os títulos das páginas de um site, o processo deveria ser feito individualmente; sendo assim, se fossem 20 páginas, seriam 20 títulos a serem alterados de maneira individual.

O resultado de tudo isso é que os sites estavam se tornando cada vez mais pesados, ficando com um tamanho total muito grande e um considerável tempo de carregamento das páginas.

### 3.2 POR UMA INTERNET MELHOR

Devido a todos os problemas citados anteriormente, a *Web* sentiu a necessidade de uma padronização. Foi então que, em 1994, o engenheiro britânico Timothy John Berners-Lee fundou o *World Wide Web Consortium* (W3C).

O W3C é um consórcio internacional no qual organizações filiadas, uma equipe em tempo integral e o público trabalham juntos para desenvolver padrões e diretrizes para a *Web*. A missão do W3C é:

“Conduzir a *World Wide Web* para que atinja todo seu potencial, desenvolvendo protocolos e diretrizes que garantam seu crescimento de longo prazo.” (W3C Brasil - <http://www.w3c.br/sobre>).

Basicamente, o W3C cumpre sua missão com a criação de padrões e diretrizes para a *Web*. Desde 1994, o W3C publicou mais de 110 desses padrões, denominados “Recomendações do W3C”. O W3C também se envolve em educação e divulgação, desenvolve softwares e atua como fórum aberto para discussões sobre a *Web*. Para que a *Web* atinja todo o seu potencial, suas tecnologias mais fundamentais precisam ser compatíveis entre si e permitir que todos os equipamentos e softwares utilizados para acessar a *Web* funcionem juntos. (W3C 2010).

Tim Berners-Lee e outros criaram o W3C como um consórcio do setor dedicado a obter consenso em torno das tecnologias da *Web*. Berners-Lee, que criou a *World Wide Web* em 1989 enquanto trabalhava na Organização Européia para Pesquisa Nuclear (CERN), tem servido como Diretor do W3C desde sua fundação, em 1994.



Figura 3 - O criador da Web e do W3C, TIM Berners-Lee



Fonte: <http://www.w3.org/People/Berners-Lee> (23/09/2010)

A proposta do W3C vem sendo amplamente seguida, tanto que os novos métodos de desenvolvimento vêm ganhando cada vez mais mercado e caindo no agrado dos desenvolvedores. Esses métodos serão discutidos no próximo item desse capítulo.

### **3.3 MÉTODO DE DESENVOLVIMENTO SEGUINDO OS PADRÕES WEB**

No item 3.1 foi apresentado como é o método de desenvolvimento tradicional, ainda usado por algumas empresas e agências Web. O objetivo desse capítulo é apresentar a solução proposta para combater todos os problemas apontados no antigo método de desenvolvimento.

Desenvolvimento: O método de desenvolvimento deixou de ser visual passou a ser através de softwares de edição de códigos XHTML (versão atual do HTML em junção com o XML, que será estudado no capítulo 5) e CSS (linguagem utilizada para formatação, também será mais aprofundada no capítulo 5). Apesar de causar “medo” em alguns designers, são duas linguagens que possuem um código muito simples e rápido de se aprender e têm como maior benefício o fato de não ficarem presas a editores visuais, de tal forma que a manutenção de um site como esse pode ser facilmente feita em outro software.

Criação de layouts: a diagramação do layout deixa de ser organizada em tabelas, passando a ter toda a sua estrutura em XHTML e sua formatação em CSS, separando assim o conteúdo da formatação.

Acesso em qualquer dispositivo: Fim da incompatibilidade. Os sites podem ser acessados de qualquer dispositivo que esteja conectado à rede, independentemente do *browser* que o usuário utilize e até mesmo do Sistema Operacional. Um estudo de caso sobre o assunto será feito no capítulo 5 e os testes em diversos *browsers* no capítulo 6.

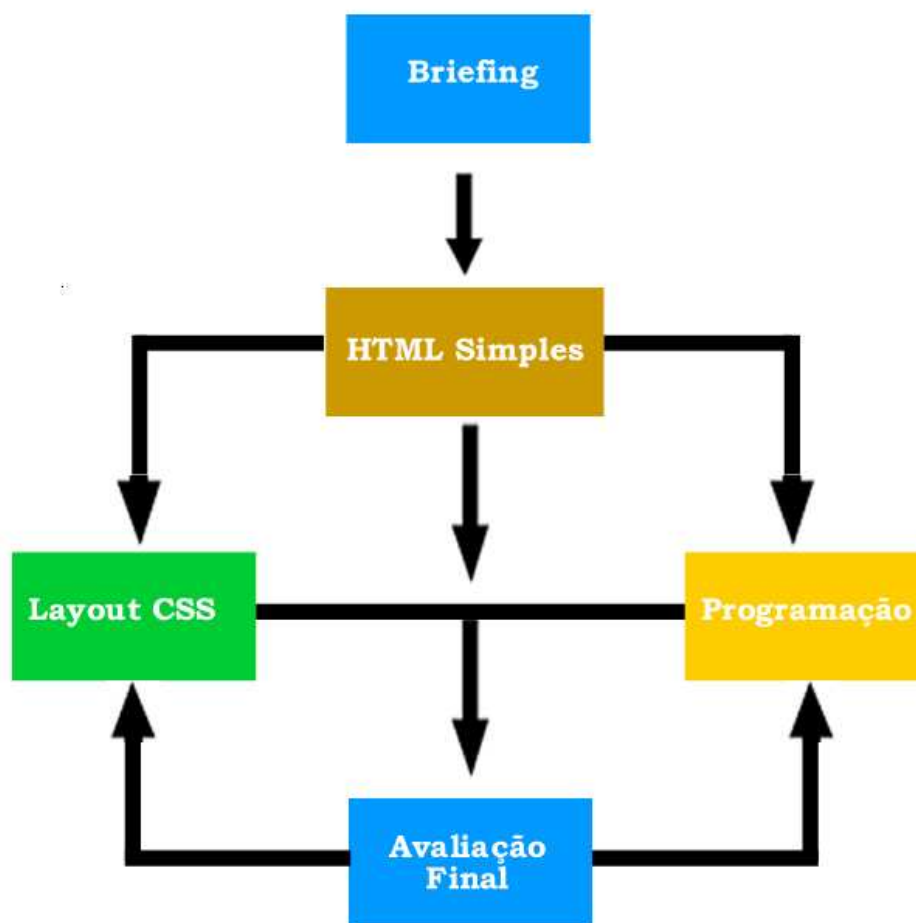
Ter o controle do seu site: o usuário vai conseguir acessar o seu site independentemente da resolução de tela, tipo de fonte, *browser* ou sistema operacional que ele utilizar.

Inclusão Digital: Devido ao uso de CSS para a formatação, os principais elementos da página, entre eles links e menus, são criados por simples textos, e não mais por imagens, fazendo assim com que os leitores de telas consigam reproduzir o conteúdo e que os deficientes visuais possam acessar a página sem terem maiores problemas. Mais sobre Acessibilidade e Inclusão digital será tratado no capítulo 4.

Bom para os buscadores: fazendo bom uso dos padrões, os sites passam a ser indexados muito mais rapidamente pelos mecanismos de busca.

Fim da guerra entre designers e programadores (construção em camadas): ocorre uma separação entre o layout e o conteúdo: o HTML é a linguagem utilizada para inserir conteúdo no site, enquanto o XHTML fica responsável por estruturá-lo. O layout é reproduzido por uma linguagem própria para formatar o HTML, chamada CSS (*Cascading Style Sheets* – Folhas de Estilo em Cascata). Pode-se alterar o layout todo de um site modificando-se apenas o seu código CSS, sem mexer na sua estrutura XHTML. Mantém-se a utilização de JavaScript através de DOM (*Document Object Model* - Modelo de Objetos de Documentos,) que é um padrão da W3C que torna o JavaScript fácil de se escrever, para que o mesmo possa funcionar em qualquer navegador, tornando-o não essencial, de tal forma que mesmo que o usuário use um *browser* que não tenha acesso a JavaScript, ele consiga acessar a informação naturalmente. Ocorre então o fim do acoplamento, pois os elementos se tornam independentes.

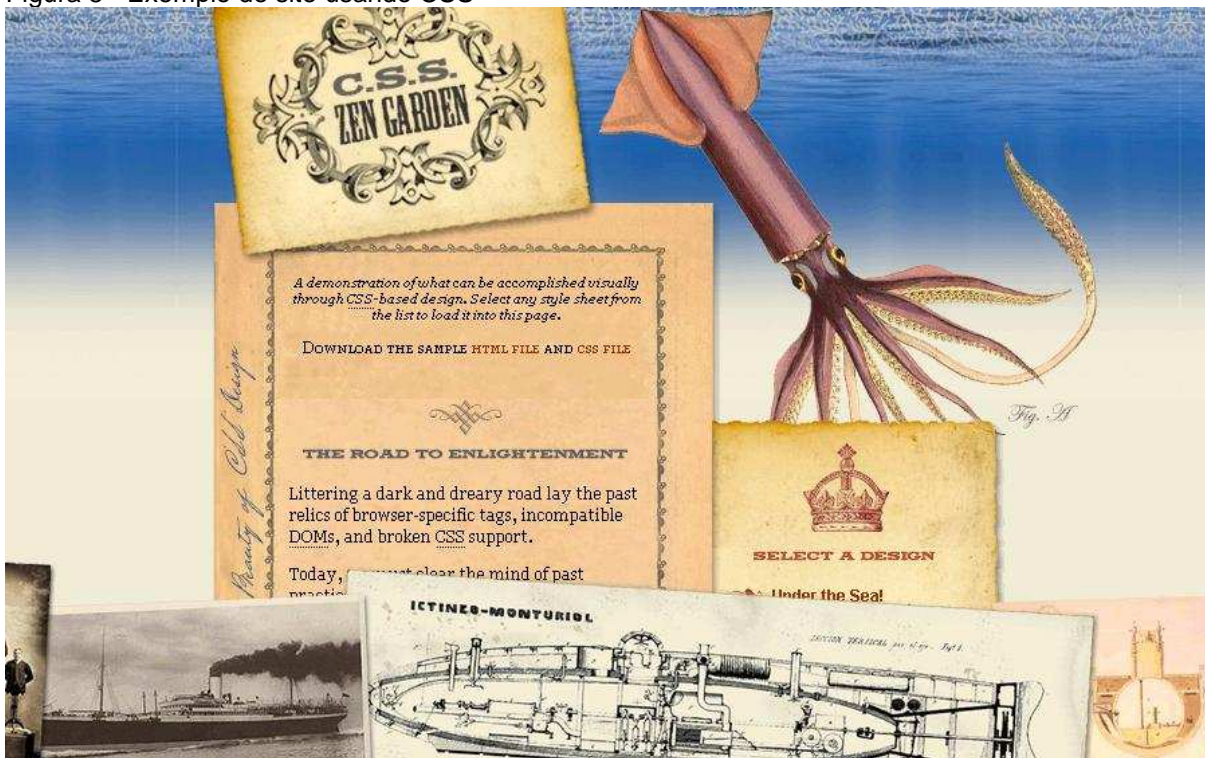
Figura 4 - Etapas do desenvolvimento de um website utilizando padrões



Fonte: Palestra da Atípico de 2007 – [www.tableless.com.br](http://www.tableless.com.br) – 01/10/2010

O exemplo poderia ser o mesmo utilizado no item 3.1. Imagine agora que o mesmo cliente peça uma alteração no layout do site. Como agora ocorre uma separação entre a formatação do site e a sua estrutura, pode-se fazer um layout completamente diferente sem mexer nesta. Para demonstrar isso, serão apresentadas duas figuras de dois *layouts* de sites.

Figura 5 - Exemplo de site usando CSS



Fonte: CSS Zen Garden - <http://www.csszengarden.com/?cssfile=/213/213.css&page=0> – 22/09/2010)

Figura 6 - Exemplo do mesmo site, mas com CSS diferente



Fonte: CSS Zen Garden - <http://www.csszengarden.com/?cssfile=/207/207.css&page=0> – 22/09/2010)

Os dois *layouts* apresentados contêm exatamente o mesmo código XHTML e a mesma estrutura; o que mudou foi apenas o CSS, a formatação da página.

O resultado obtido com isso é que sites que tradicionalmente demoravam semanas ou até mesmo meses para serem produzidos agora são criados em poucos dias, acessíveis a todos e aparecendo nas primeiras páginas dos buscadores.

## **4 ACESSIBILIDADE E USABILIDADE**

### **4.1 DEFINIÇÃO**

No capítulo anterior, foram apresentados os métodos de desenvolvimento, tanto o tradicional como o usado pelos padrões. Nesse capítulo, serão tratados na prática todos os termos técnicos para que um site fique acessível a todos, incluindo pessoas com deficiência visual.

Segundo Fabio Gameleira, acessibilidade na Internet pode ser definida como:

“Permitir que uma maior quantidade de pessoas tenha acesso à informação, através da utilização de recursos que naturais da própria linguagem HTML permitam e facilitem que os sites e páginas da Internet possam ser lidas por todos, inclusive as pessoas cegas. Existem diversos tipos de deficiências, além da deficiência visual, que também se beneficiam quando um site utiliza as técnicas de acessibilidade, como por exemplo: Daltonismo, Dislexia, Cognitivas, Baixa Visão entre outras.” (Cartilha da Acessibilidade - <http://www.lupadigital.info> – 24/09/2010)

Se fosse perguntado para vários Desenvolvedores Web e Programadores se eles se preocupam com as pessoas com deficiência ao desenvolverem seus sites, a quase totalidade deles diria que não, e acrescentaria: “Hoje em dia, temos que nos preocupar com a maioria; não dá para pensarmos em todo mundo”. O que esse profissional está esquecendo é que no Brasil, segundo o Censo do IBGE do ano 2000, existem cerca de 150 mil cegos e cerca de 16,6 milhões de pessoas com dificuldade para enxergar. Por isso, conclui-se que é uma quantia muito significativa para ser negligenciada, ainda mais sendo a Internet a tecnologia que mais cresce atualmente. (IBGE – Censo 2000)

A partir do próximo item, serão apresentadas todas as etapas para deixar o site acessível.

### **4.2 DECLARAR UM TIPO DE DOCUMENTO**

Em HTML, todo documento criado é de um determinado tipo. E o mais importante para que o site possa ser corretamente validado pelas ferramentas de

avaliação (que serão vistas mais adiante) é ter esse documento declarado logo no início do documento.

Um desses elementos a serem declarados é a declaração DOCTYPE, que também é conhecida como *Document Type Definition* (Definição de Tipo de Documento), ou simplesmente DTD.

As declarações possíveis de DOCTYPE oferecidas pelo XHTML, segundo o W3C, são: *Transitional*, *Script* e *Frameset*.

#### 4.2.1 TRANSITIONAL

Usado por desenvolvedores que utilizavam o método tradicional de programação e estão migrando aos poucos para os métodos com padrões.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

#### 4.2.2 STRICT

Tipo de Documento que obriga o desenvolvedor a usar os métodos segundo os padrões Web, pois não terá suporte a elementos que a W3C remove à medida que o CSS evolui.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

#### 4.2.3 FRAMESET

É o menos utilizado de todos. Essa DTD permite a utilização de *frames*, método que já está em desuso.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

Declarando corretamente as DTD's das páginas, os validadores conseguirão validar corretamente os sites. A seguir, a explicação de uma linha de DTD:

- DOCTYPE = diz ao *browser* que está sendo especificado um tipo de documento para essa página
- HTML = significa que <HTML> é o elemento raiz na página
- W3C//DTD XHTML 1.0 Frameset//EN = especifica que será usada a versão 4.01 do HTML e escrito em inglês
- transitional.dtd = aponta um arquivo que identifica este padrão em particular

### 4.3 ESTRUTURA DA PÁGINA

Desde o começo, os *websites* eram todos estruturados com o uso de tabelas, método que ainda persiste em alguns sites da Web. A partir disso, toda a estrutura de site, desde o topo, o menu, o corpo do documento e o rodapé, como demonstra o exemplo abaixo:

Figura 7 - Exemplo de uma página com frames

Titulo da Página	
MENU	Desenvolvimento
Menu 1	Para desenvolver um site, você não precisa adquirir ferramentas de edição sofisticadas e dispendiosas. Particularmente, os editores do tipo WYSIWYG, que são tão bem-aceitos, procurados e usados, são uma ferramenta de desenvolvimento não-recomendada para o início do aprendizado. Tais ferramentas devem ser usadas apenas quando o desenvolvedor tiver total conhecimento e controle da maneira como geram o código.
Menu 2	
Menu 3	
© Todos os Direitos Reservados	

Ao analisar o código dessa página, será encontrado o seguinte:



Figura 8 - Código-fonte da página exibida na figura anterior

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Monografia sobre Padrões Web - Estruturando com Tabelas</title>
6 </head>
7
8 <body>
9 <table width="877" height="440" border="1">
10 <tr>
11 <td height="63" colspan="3">Titulo da Página</td>
12 </tr>
13 <tr>
14 <td width="105" valign="top"><p>MENU</p>
15 <p>Menu 1<br />
16 <p>Menu 2<br />
17 <p>Menu 3</p></td>
18 <td width="482" colspan="2" valign="top"><p>Desenvolvimento</p>
19 <p>Para desenvolver um site, você não precisa adquirir ferramentas de edição sofisticadas e dispendiosas. Particularmente, os editores do tipo WYSIWYG, que são
20 tão bem-aceitos, procurados e usados, são uma ferramenta de desenvolvimento
21 não-recomendada para o início do aprendizado. Tais ferramentas devem ser usadas
22 apenas quando o desenvolvedor tiver total conhecimento e controle da maneira
23 como geram o código.</p></td>
24 </tr>
25 <tr>
26 <td colspan="3" align="center" valign="top"><div align="center">&copy; Todos os Direitos Reservados</div></td>
27 </tr>
28 </table>
29 </body>
30 </html>
31

```

Através do código, pode ser observada a utilização das tags `<table>` e `</table>`, nas linhas 9 e 29, respectivamente, empregadas para fazer a abertura e o fechamento de uma tabela. Nota-se que, como essa tabela foi utilizada para fazer toda a estrutura da página, ela abrange todos os elementos do corpo desta (`<body></body>`). O efeito negativo disso é exatamente o que foi dito anteriormente: não só a tabela perde a sua verdadeira funcionalidade, que é a de tabular dados, como também essa página demorará mais para ser carregada. Um exemplo disso é que algumas versões do browser Internet Explorer da Microsoft fazem a leitura do primeiro `<table>` e não mostram nada na tela até que encontrem o fechamento da tag (`</table>`), sendo que se a página fosse grande, o navegador primeiro faria a leitura da página toda e apenas depois mostraria na tela do navegador.

Um exemplo do uso correto de tabelas seria a criação de uma tabela de preços.

Tabela 2 - Exemplo de uso correto de tabelas

Produtos	Preços
CoreDRAW Graphics Suite X4	R\$ 1.899,00
MAC OS X 10.6 Snow Leopard - Apple	R\$ 79,00
Kaspersky Anti-Virus Personal 2010	R\$ 79,90
Office 2007 Home & Student	R\$ 129,00

#### 4.4 IMAGENS

O uso correto de inserção de imagens nas páginas pode beneficiar tanto os sites de busca como os deficientes visuais, apenas com o uso dos atributos “alt” e “title”.

##### 4.4.1 BENEFICIANDO OS SITES DE BUSCA

Com o uso do atributo “alt”, os navegadores conseguirão fazer a indexação dessa imagem de forma correta.

O código ficaria da seguinte forma:

```

```

No navegador, a imagem seria exibida da seguinte maneira:

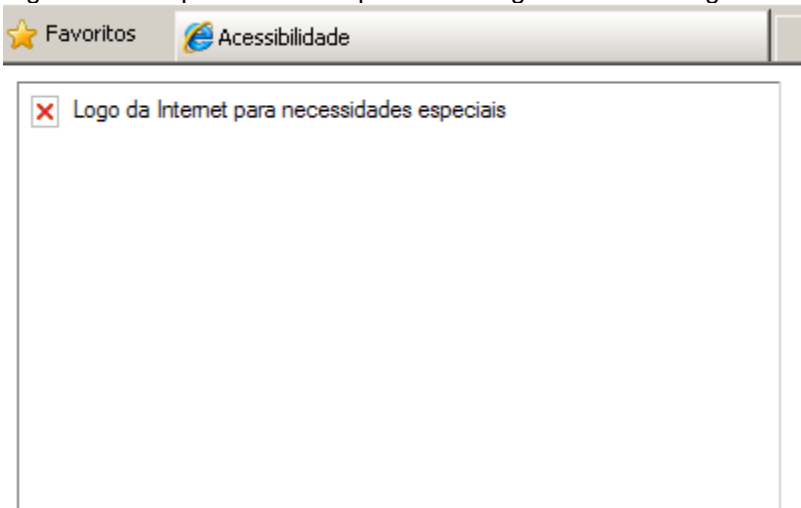
Figura 9 - Imagem sendo exibida no navegador



Fonte: <http://www.acessibilidade.net/web/ine/livro.html> - 25/09/2010

A importância disso é que, se por algum motivo o *browser* não conseguir exibir essa imagem, com o uso da propriedade “Alt” seria exibido o conteúdo desse atributo na tela do navegador, ficando da seguinte maneira:

Figura 10 - Propriedade “Alt” quando a imagem não é carregada



Observa-se que, ainda assim, seria completamente compreensível qual é o conteúdo dessa imagem.

## 4.4.2 BENEFICIANDO OS DEFICIENTES VISUAIS

Os leitores de tela também conseguirão fazer a leitura das imagens. Assim uma pessoa com deficiência visual, mesmo que não consiga visualizar a imagem, terá uma descrição sobre essa imagem. Isso pode ser conseguido colocando um link com imagem invisível após a inserção da imagem principal, como mostra o código a seguir:

Figura 11 - Propriedade "Alt" em imagens

```

8 <body>
9 <a href="http://www.acessobrasil.org.br/">
10 
12 </a>
13 <a href="http://www.acessobrasil.org.br/">
14 
17 </a>

```

Quando o usuário chegar sobre essa imagem, a partir de toques na tecla TAB, o leitor de tela conseguirá realizar a leitura dessa imagem de forma correta.

## 4.5 TEXTOS

As tags aplicadas nos textos também têm um papel muito importante no código e, quando bem aplicadas, pode-se diferenciar cada conteúdo de uma página apenas pelo seu código. As principais tags de textos são:

### 4.5.1 Cabeçalhos (Títulos)

Para formatar os títulos dos textos, são utilizadas as tags <h1>, <h2>, <h3>, <h4>, <h5> e <h6>. Cada uma delas produz um tamanho de texto diferente.

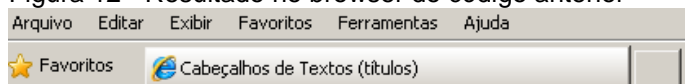
Ex:

```

<h1>Padrões Web</h1>
<h2>Padrões Web</h2>
<h3>Padrões Web</h3>
<h4>Padrões Web</h4>
<h5>Padrões Web</h5>
<h6>Padrões Web</h6>

```

Figura 12 - Resultado no browser do código anterior



# Padrões Web

## Padrões Web

### Padrões Web

#### Padrões Web

##### Padrões Web

###### Padrões Web

## 4.5.2 PARÁGRAFOS

Para identificar um parágrafo em um texto, é utilizada a tag <p>.

Figura 13 - Exemplo da tag de parágrafo &lt;p&gt;

```

10 <p>A expressão "acessibilidade", presente em diversas áreas de atividade, tem também na informática um importante
11 significado.
12 </p>
13
14 <p>Representa para o nosso usuário não só o direito de acessar a rede de informações, mas também o direito de
15 eliminação de barreiras arquitetônicas, de disponibilidade de comunicação, de acesso físico, de
16 equipamentos e programas adequados, de conteúdo e apresentação da informação em formatos alternativos.
17 (Acessibilidade Brasil - http://www.acessobrasil.org.br)
18 </p>

```

## 4.5.3 QUEBRA DE LINHA

Para realizar o término de uma linha, porém sem a criação de um parágrafo, usa-se o recurso de quebra de linha, utilizando a tag <br />.

Figura 14 - Uso do recurso de quebra de linha (&lt;br /&gt;)

```

10 <p>Representa para o nosso usuário não só o direito de acessar a rede de informações, mas também o direito de
11 eliminação de barreiras arquitetônicas, de disponibilidade de comunicação, de acesso físico, de
12 equipamentos e programas adequados, de conteúdo e apresentação da informação em formatos alternativos.
13 <br />(Acessibilidade Brasil - http://www.acessobrasil.org.br)
14 </p>

```

#### 4.5.4 NEGRITO E ITÁLICO

Essas formatações podem ser utilizadas de duas formas: `<b>` e `<i>`, indicando respectivamente negrito e itálico, ou também podem ser representadas com `<strong>` e `<em>`. Embora ambos os casos funcionem perfeitamente bem em navegadores diferentes, as tags `<b>` e `<i>` não têm efeito em leitores de tela; por isso, é recomendado o uso das tags `<strong>` e `<em>` para fazer esse tipo de formatação.

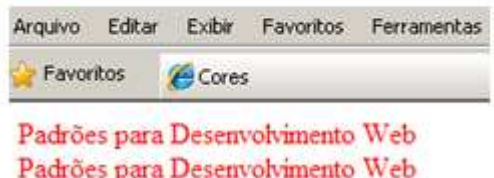
Figura 15 - Formatação em negrito e itálico

<pre> 10 &lt;strong&gt;Negrito&lt;/strong&gt; &lt;br /&gt; 11 &lt;em&gt;Itálico&lt;/em&gt; 12 13 &lt;br /&gt; 14 &lt;br /&gt; 15 16 &lt;b&gt;Negrito&lt;/b&gt; &lt;br /&gt; 17 &lt;i&gt;Itálico&lt;/i&gt; </pre>	<pre> <b>Negrito</b> <i>Itálico</i>  <b>Negrito</b> <i>Itálico</i> </pre>
--	---

#### 4.5.5 CORES

As cores num texto HTML podem ser representadas de duas formas: digitando o nome da cor em inglês ou o seu valor em Hexadecimal.

Figura 16 - Duas formas de inserir cores por HTML

<pre> 9 10 &lt;font color="#FF0000"&gt;Padrões para Desenvolvimento Web&lt;/font&gt; &lt;br /&gt; 11 &lt;font color="red"&gt;Padrões para Desenvolvimento Web&lt;/font&gt; 12 </pre>	
--	---

Lembrando que as cores também podem ser definidas diretamente no CSS, como será explicado no próximo capítulo.

## 4.5.6 FORMULÁRIOS

Os formulários constituem possivelmente o recurso mais utilizado dentro de um website. Com eles, pode-se fazer tanto a interação entre o usuário e o proprietário do site através de um formulário de contato (o famoso “Fale Conosco”) quanto o cadastramento do usuário no site, através de um formulário de cadastro. Todo o formulário deve estar entre as *tags* `<form>` e `</form>`.

O grande problema em relação à acessibilidade de um formulário é que, quando um campo de texto recebe o foco, alguns leitores de tela deixam de ler o título desse campo, deixando o usuário sem saber com o que preencher.

A solução desse problema é simples: basta envolver todo o campo com o elemento `<LABEL>`, responsável por dar um título a esse campo, que será lido corretamente pelos leitores de tela.

Abaixo um exemplo de um formulário completo.

Figura 17 - Código fonte de um formulário completo

```

10 <form action="envia_email.php" name="frmEnvia" method="POST">
11   <fieldset>
12     <legend>Dados Pessoais </legend>
13     <label for="Nome">Nome: <input type="text" name="nome" /> </label> <br />
14     <label for="Sobrenome">Sobrenome: <input type="text" name="sobrenome" /> </label><br />
15     <label for="Sexo">
16     Sexo:
17     <select name="sexo">
18       <option value="" selected="selected"></option>
19       <option value="m">Masculino</option>
20       <option value="f">Feminino</option>
21     </select>
22     </label>
23 </fieldset>
24 <br />
25 <br />
26
27 <fieldset>
28   <legend>Dados de acesso </legend>
29   <label for="Login">Login: <input type="text" name="login" /> </label><br />
30   <label for="Senha">Senha: <input type="text" name="senha" /> </label><br />
31 </fieldset>
32 <br />
33 <input type="submit" name="enviar" value="Enviar!" />
34 <input type="reset" name="limpar" value="Limpar!" />
35
36 </form>

```

### Explicação sobre os elementos contidos no código:

Tabela 3 - Explicação dos elementos do formulário de contato

Action	Aponta qual o arquivo que receberá os dados desse formulário após o envio. Geralmente, os dados são tratados por uma tecnologia de servidor, como PHP, JSP ou ASP.
Method	Indica qual método será utilizado para enviar o formulário, podendo ser POST (utilizado quando os dados do formulário são sigilosos) ou GET.
Fieldset	Usado para separar em grupos cada parte do formulário, por exemplo para discriminar “Dados Pessoais” de “Dados de Acesso”.
Label	Utilizado para dar um título ao campo, tornando assim possível a leitura por softwares leitores de tela.
Select	Utilizado para criar um campo de seleção de lista.
Option	Valores da lista.

O formulário será visualizado no navegador da seguinte forma:

Figura 18 - Visualização no browser do código da figura anterior

The screenshot shows a web browser window with a menu bar (Arquivo, Editar, Exibir, Favoritos, Ferramentas, Ajuda) and a toolbar (Favoritos, Formulários, Home, Back, Forward, Print, Page, Security, Tools). The main content area displays a contact form with two fieldsets:

- Dados Pessoais:** Contains three input fields: 'Nome:', 'Sobrenome:', and 'Sexo:' (a dropdown menu).
- Dados de acesso:** Contains two input fields: 'Login:' and 'Senha:'.

At the bottom of the form, there are two buttons: 'Enviar!' and 'Limpar!'.

Nesse capítulo foram apresentadas as principais formas de colocar os elementos corretamente para manter seu site acessível. A preocupação com a aparência não foi levada em conta, por se tratar apenas de conteúdo HTML, que como foi visto no item 1.2, deve ser usado apenas para estruturar o site, deixando a parte de formatação com o CSS.



No próximo capítulo, será criado um estudo de caso onde será estruturado um pequeno site, explicando como formatar os elementos vistos nesse capítulo e colocar em prática todos os conceitos vistos até o momento.

## 5 CRIANDO UM SITE NOS PADRÕES WEB

No capítulo 3, foi comentado o método de desenvolvimento tradicional e apontados os erros cometidos nesse método de desenvolvimento. Nesse capítulo será realizado um estudo de caso, no qual será construído passo a passo um site seguindo os padrões da W3C discutidos nos capítulos anteriores.

### 5.1 XHTML

Como foi estudado no capítulo anterior, o HTML serve para fazer a estrutura do site. Mas para aperfeiçoar ainda mais o HTML, após a sua versão 4, foi criado o *eXtensible* HTML (XHTML), que nada mais é do que a mistura do HTML com o XML.

Mas o que seria XML?

A sigla XML significa *eXtensible Markup Language*, ou Linguagem de Marcação Extensível. Pode-se dizer que o XHTML é uma extensão do HTML, tornando o desenvolvimento mais flexível e nos dando a opção de fazer coisas que não seriam possíveis apenas com o HTML. A declaração DOCTYPE(DTD) vista no item 4.3 do capítulo anterior nada mais é do que a linguagem para definir a estrutura de um documento XML

Para um melhor entendimento sobre essa linguagem, pode ser usado o HTML para comparação. Como foi visto, o HTML basicamente define os elementos que se pode usar com as suas marcações pré-estabelecidas, como `<body></body>`, ou `<head></head>`, etc, obrigando a utilizar essas marcações já existentes para escrevermos o código. Mas e se quiséssemos fazer uma marcação com nossas próprias tags, por exemplo, `<descrição></descrição>`, isso seria possível? Com XML sim, e foi com essa junção do HTML com o XML que nasceu o XHTML.

### 5.2 CABEÇALHO E DECLARAÇÕES DO SITE

A partir de agora, será realizada a criação do nosso site. Como foi dito no capítulo 3, adotando os padrões Web tem-se a opção de separar a estrutura da formatação. Sendo assim, a primeira etapa será criar a estrutura da página.

Para fazer a indicação que o site será construído usando XHTML, essa indicação deve ser feita logo no início da página, em seu Doctype, fazendo assim com que os navegadores saibam a forma que eles deverão interpretar o site.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Como pode ser visto, foi informada a versão do XHTML que será utilizada e o Doctype (Transitional).

Após isso, deve ser feita a indicação da versão e do idioma do XML que será utilizado no documento.

```
<html xmlns="http://www.w3.org/1999/xhtml xml:lang="pt-br" lang="pt-br">
```

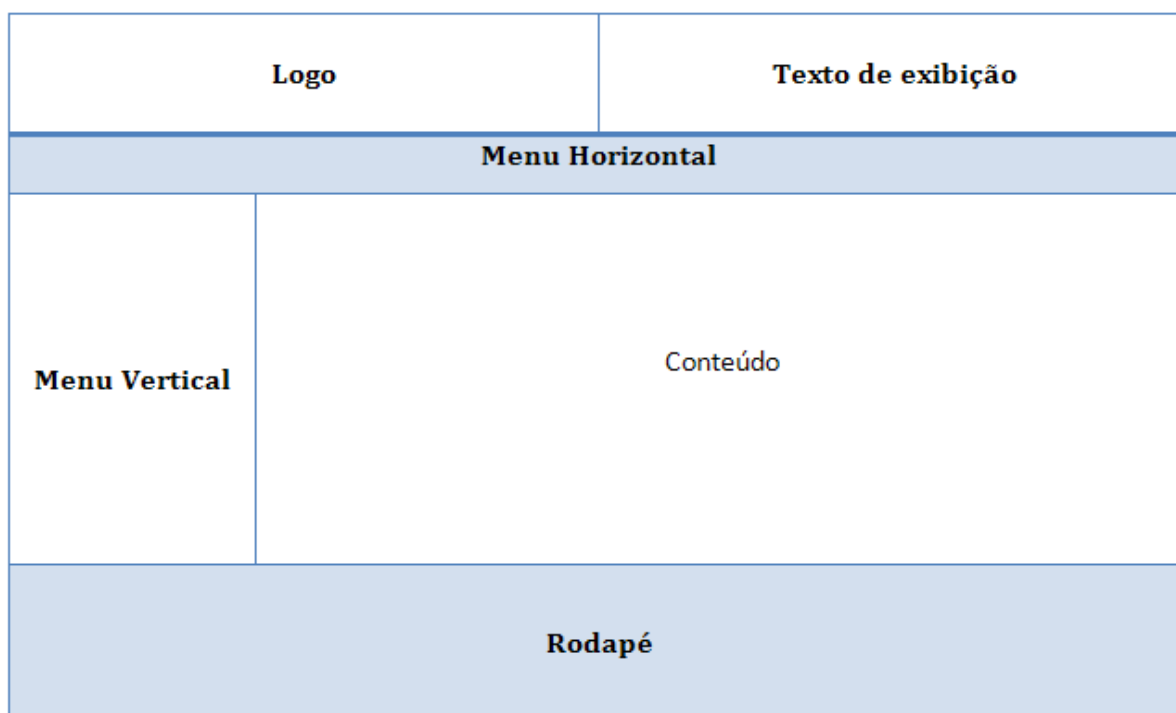
Já no cabeçalho da página, serão utilizadas as *meta-tags* para fazer a descrição da página, tanto para os usuários como para os buscadores. A tag meta é usada para trazer alguma informação sobre a página.

```
<meta name="keywords" content="web, padrões, W3C" />

<meta name="description" content="Criando um site nos padrões Web" />
```

### 5.3 CRIANDO A ESTRUTURA DO SITE

A partir de agora, será iniciada a estruturação do site. O projeto terá a seguinte estrutura:

Figura 19 – Wireframe<sup>4</sup> do site a ser criado

Toda estrutura do site será criada em XHTML, deixando de lado o método tradicional no qual se utilizavam tabelas para realizar esse recurso. Esse método recebe o nome de Tableless, que significa exatamente “Sem tabela”, no qual toda a parte da formatação será feita toda por CSS. Para separar a estrutura da formatação, toda a parte estrutural ficará em um arquivo chamado **index.html**, e a parte da formatação em um arquivo chamado **estilos.css**.

#### 5.4 DIV'S

Em um site estruturado por tabelas, utilizavam-se as linhas e colunas para o posicionamento dos elementos. Em tableless, esse posicionamento é feito através do elemento Div, que nada mais é do que camadas onde são colocadas cada parte do site. Para entender melhor, pode-se analisar o *Wireframe* acima: cada parte do site ficaria em uma div, sendo uma para o logo, uma para cada menu, uma para o conteúdo principal, uma para o rodapé, e assim por diante.

<sup>4</sup> Wireframe é um “rascunho” que organiza a estrutura de um site a ser criado

Para organizar ainda melhor, será utilizado o método de “Div sobre Div”, criando Divs “mãe” para separar ainda mais cada estrutura do site, ficando da seguinte forma:

- a div Logo e a div Texto ficarão alocadas dentro de uma div mãe chamada **Topo**.
- o menu Horizontal em uma div chamada **Menu**
- o menu vertical e o conteúdo ficarão em uma div chamada **Principal**.
- o rodapé ficará em uma div chamada **Rodapé**.
- por fim, todas as divs ficarão em uma div “mestra” chamada **Box**. Essa div é importante para definir a largura do site e para centralizá-lo na tela.

Para fazer a identificação de uma div, cada uma delas terá o seu “ID”, que nada mais é do que o nome da Div, servindo para identificação não só no XHTML, mas também para ser formatada pelo CSS posteriormente.

Sendo assim, a criação da DIV mestra Box seria da seguinte forma:

```
<div id="box"
  (conteúdo)
</div> <!--BOX-->
```

Observe que, como no HTML, cada div deve ser aberta e posteriormente fechada. Tudo que aparecer entre os sinais de <!-- --> são comentários HTML, e não serão interpretados pelo navegador, servindo apenas para uma boa prática de documentação do site. Os comentários são importantes, pois facilitarão o reconhecimento dos fechamentos das divs à medida que o código se estende.

Como comentado anteriormente, a div Logo e a div Texto ficarão dentro de uma div chamada **Topo**, e todas dentro da div **Box**, da seguinte forma:

```
<div id="box"
  <div id="topo">
    <div id="topo_logo">
    </div> <!--LOGO-->

    <div id="topo_texto">
```

```

        <div> <!--TEXTO-->
    </div>

</div>

```

À primeira vista pode parecer complicado, mas entendendo o conceito de div mãe e div filha, tudo se torna mais simples. Até o momento tem-se a div **Box**, que englobará todas as divs do site, e a div **Topo**, que tem as divs filhas **topo\_logo** e **topo\_texto**. Para uma melhor identificação das divs, cada div filha terá antecedendo o seu nome o nome da sua div mãe; por isso foi concatenado como “topo\_” que é o nome da div mãe seguinte por “logo”, e nunca se esquecendo dos comentários.

Feito isso, serão construídas todas as divs do conteúdo principal, sem a preocupação, por enquanto, com o conteúdo.

```

<div id="box">
    <div id="topo">
        <div id="topo_logo">
            </div> <!--LOGO-->
        <div id="topo_texto">
            <div> <!--TEXTO-->
        </div>
    <div id="menu">
    </div> <!--MENU HORIZONTAL-->
    <div id="principal">
        <div id="menu_vert">
        </div> <!--MENU VERTICAL-->
        <div id="conteudo">
        </div> <!--CONTEÚDO PRINCIPAL-->
    </div> <!--PRINCIPAL-->
</div> <!--BOX-->
<div id="rodape">
</div>

```

Salvando o arquivo com o nome de **principal.html** e abrindo-o em um navegador, nota-se que não aparecerá nada na tela, pois o que foi feito até o momento foi estruturar a página, e não foi adicionado ainda nenhum conteúdo e nenhuma formatação em CSS.

## 5.5 FORMATAÇÃO DO SITE

Definida a estrutura do site, chega a hora de partir para a sua formatação, que será toda realizada através de um arquivo CSS. O CSS em uma página pode ser realizado tanto de modo **interno** (com os códigos CSS na própria página onde se encontra a estrutura do site), ou **externo**, através de um arquivo separado.

A segunda forma é a mais recomendada, pois assim consegue-se separar o que é estrutura e conteúdo do que é formatação. Essa será a forma utilizada.

### 5.5.1 REFERENCIANDO UM ARQUIVO CSS

Para fazer a referência do arquivo CSS externo na página onde é encontrada a estrutura, utiliza-se a seguinte chamada dentro do cabeçalho (*head*) da página html:

```
<link rel="stylesheet" href="estilos.css" type="text/css" />
```

- rel = indicação que o documento será uma folha de estilo
- href = atributo que faz a ligação do arquivo CSS externo ao arquivo XHTML
- type = tipo de documento que será utilizado, que no caso, é um texto CSS.

### 5.5.2 APLICANDO FORMATAÇÕES

Para visualizar o funcionamento das CSS, será criado um arquivo chamado **estilos.css** e serão aplicadas algumas formatações, referenciando o elemento HTML que receberá a formatação. Por exemplo, o corpo da página em HTML é referenciado pela tag **body**, e a cor de fundo pelo atributo **background**, sendo que

se houver a necessidade de mudar o fundo da página, deve-se prosseguir da seguinte maneira no arquivo CSS:

```
body{  
    background-color: #666;  
}
```

As cores sempre são referenciadas através de valores hexadecimais e ao executar o arquivo **index.html** no navegador, a página já será visualizada com o fundo cinza escuro.

### 5.5.3 LARGURA, ALTURA, MARGEM E POSICIONAMENTO

Através do CSS, serão definidas a largura, a altura e a margem dos elementos. É importante ressaltar que a altura será definida apenas para poder identificar os elementos na página, pois ao adicionar conteúdo, a altura é obtida automaticamente.

A primeira definição será a da div **box**, pois ela é responsável por englobar todos os elementos da página. O prosseguimento ocorrerá da seguinte forma, no arquivo CSS:

```
#box{  
    width: 780px;  
    height: 400px;  
    margin: 0 auto;  
    border: 1px solid #FFF;  
}
```

Perceba que foi definida a largura (width em 780 pixels, pois assim mesmo que o usuário esteja usando uma resolução tanto de 1024 X 768 pixels ou de 800 X 600 pixels, não precisará utilizar a barra de rolagem horizontal), a altura (height, que conforme dito, ao inserir conteúdo, será retirada do código), uma margem (no caso definido como “0”, para que o conteúdo fique sem nenhuma margem em relação à largura definida, e “auto”, para que fique automaticamente centralizada horizontalmente na tela) e uma borda (com a largura de 1px, tipo sólida



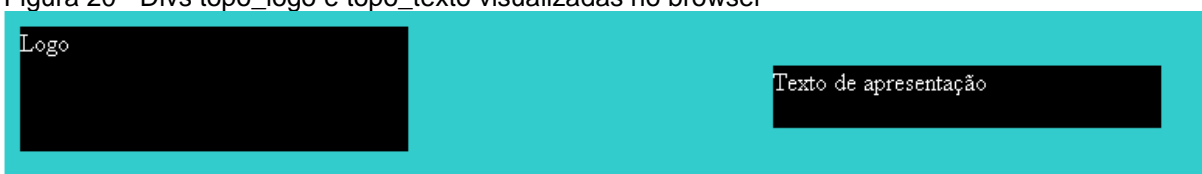
e cor branca, apenas para ilustração). Perceba também que antes da tag `box` foi utilizado o símbolo de “#”, pois ela não se trata de um elemento HTML, e sim de um criado por nós, e para identificar que estará sendo tratada uma div “id”.

Quando uma div possui div filhas, deve-se usar o elemento “float” para informar onde cada div filha será posicionada. Para entender melhor esse procedimento, será analisada a div **Topo**. Essa div possui duas divs filhas: **topo\_logo** e **topo\_texto**. Se houver a necessidade que a div `topo_logo` fique à esquerda da div mãe, e a `topo_texto` à direita, faz-se da seguinte forma:

```
#topo{
    height: 110px;
    background: #33CCCC;
}
#topo_logo{
    width: 250px;
    height: 80px;
    background: #000;
    margin: 10px;
    float: left;
}
#topo_texto{
    width: 250px;
    height: 40px;
    background: #000;
    float: right;
    margin: 35px;
}
```

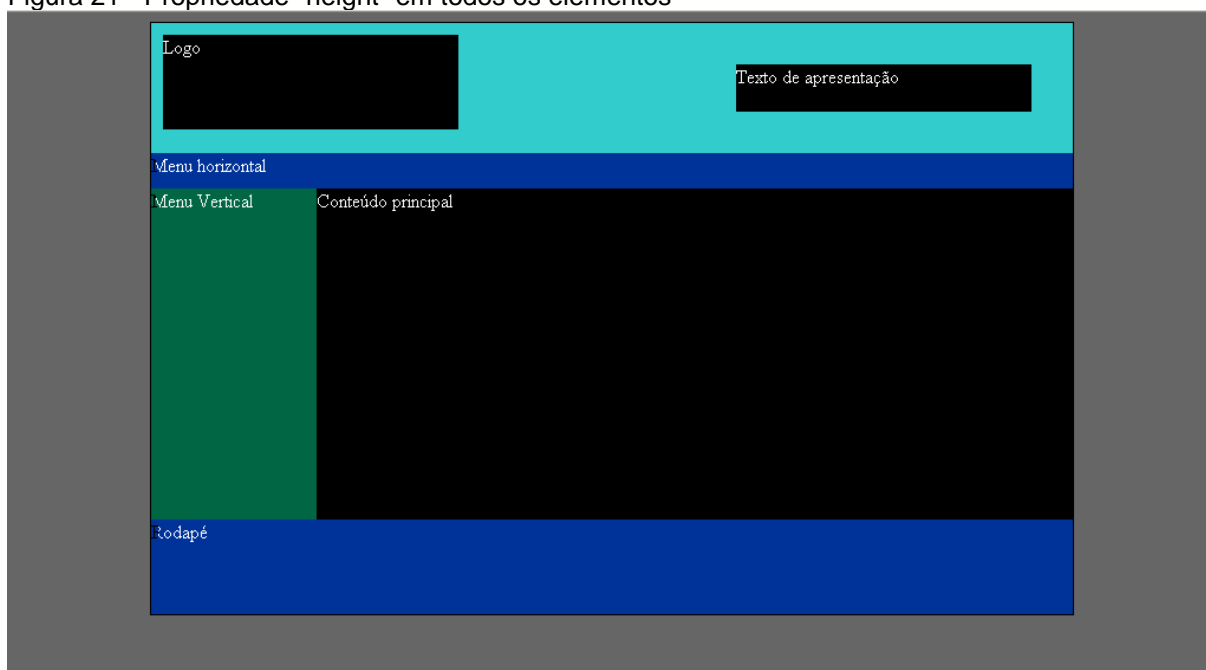
Dessa maneira, consegue-se um posicionamento das divs, como mostra a figura a seguir:

Figura 20 - Divs topo\_logo e topo\_texto visualizadas no browser



Para visualizar todos os elementos da tela, coloca-se uma altura e uma cor de fundo em cada um deles<sup>5</sup>. Fazendo isso, tem-se a seguinte estrutura:

Figura 21 - Propriedade "height" em todos os elementos



Perceba que, sem o uso de tabelas, é possível construir toda a estrutura do site. É de extrema importância fazer esses passos de estruturação do site antes de começar a inserir conteúdo, para já ter uma ideia de como o site ficará. Feito isso, o próximo passo é a inserção dos conteúdos para uma melhor formatação da página.

#### 5.5.4 CRIAÇÃO DO MENU

O site criado apresentará dois tipos de menu: vertical e horizontal. Como servirão apenas para efeito de aprendizado, eles terão as mesmas opções.

---

<sup>5</sup> Todo o código fonte do projeto estará disponível no Anexo no final deste trabalho.

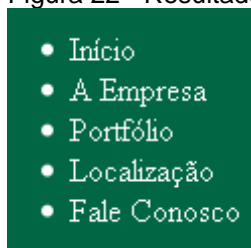
### 5.5.4.1 MENU VERTICAL

No método de desenvolvimento tradicional, a maioria dos sites apresentava menus em formato de flash ou de imagens, para garantir uma formatação mais apresentável. Porém, graças ao CSS, é possível construir um menu usando apenas folhas de estilo, formatando-o para que fique com a mesma aparência dos seus antecessores. Para isso, basta criar cada item do menu como se fosse uma lista não ordenada, como o exemplo que segue:

```
<div id="menu_vert">
    <ul>
        <li>Início</li>
        <li>A Empresa</li>
        <li>Portfólio</li>
        <li>Localização</li>
        <li>Fale Conosco</li>
    </ul>
</div> <!--MENU VERTICAL-->
```

O resultado desse código seria o seguinte:

Figura 22 - Resultado da lista não-ordenada



Como observado na figura, trata-se nada mais nada menos do que uma lista não ordenada. Como ele será o principal responsável pela navegação do site, certamente eles devem ser links; então será realizada uma modificação no código:

```
<div id="menu_vert">
    <ul>
        <li><a href="#">Início</a></li>
```

```

        <li><a href="#">A empresa</a></li>
        <li><a href="#">Portfólio</a></li>
        <li><a href="#">Localização</a></li>
        <li><a href="#">Fale Conosco</a></li>
    </ul>
</div> <!--MENU VERTICAL-->

```

Fazendo isso, eles já estão prontos para receberem links. O símbolo “#” será substituído posteriormente pelo endereço da página que a opção de menu receberá.

O próximo passo é a formatação desse menu. No arquivo CSS, realizam-se as seguintes codificações:

```

#menu_vert {
    width: 140px;
    float: left;
}

#menu_vert ul{
    margin: 0;
    padding: 0;
    list-style: none;
    float: left;
}

#menu_vert ul li{
    width: 130px;
}

#menu_vert ul li a{
    text-decoration: none;
    color: #FFF;
    background: #333;
    float: left;
    width: 185px;
}

```

```
padding: 5px;
margin: 1px 0;
font: 14px Tahoma, Arial, Helvetica, sans-serif;
font-weight: bold;
border-left: 6px solid #CC3333;
}
```

```
#menu_vert ul li a:hover{
background: #CC3333;
border-left: 5px solid #333;
```

Analisando determinadas funções do código:

- `#menu_vert` → definido a largura e o float da div mãe. Perceba que o atributo “height” (altura) foi removido;
- `List-style: none` → usado para remover a lista não ordenada das opções;
- `Text-decoration: none` → usado para remover o sublinhado dos links;
- `Font-weight: bold` → usado para colocar o texto em negrito;
- `Margin` → usado para definir o espaço externo do elemento;
- `Padding`: usado para definir o espaço interno do elemento.
- `Hover` → efeito muito semelhante aos menus em flash, que ocorre quando o usuário aponta o mouse para o link e este

Com essas configurações, o menu terá a seguinte aparência:

Figura 23 - Resultado final do menu vertical usando apenas CSS



Pode-se observar na execução da página que ao apontar o mouse para o item “A empresa”, ocorre o efeito hover.

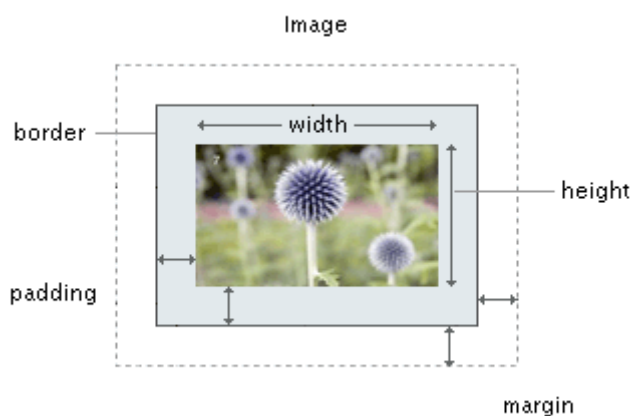
A div do menu vertical será aproveitada para fazer a inserção de uma imagem logo abaixo dele, para preencher o espaço vazio. Para isso, basta inserir o seguinte código logo após o fechamento do último </ul> do menu:

```

```

A figura a seguir retrata bem a diferença entre os principais atributos de formatação:

Figura 24 - Ilustração dos elementos border, padding, height e margin



<http://www.web-toolbox.net/webtoolbox/css/css-bildformatierung.htm> - 13/10/2010 - 19:50

#### 5.5.4.2 MENU HORIZONTAL

Como combinado, será feito também o mesmo menu no sentido horizontal. Como as modificações desse menu para o vertical são poucas, a seguir será descrito apenas o que muda de um para outro.

```
#menu{
    height: 30px;
}
#menu ul{
    width: 100%;
}
#menu ul li{
    display: inline;
}
```

Analisando as alterações do código:

- Height: 30px → diferente do menu vertical, o atributo height deve ser mantido;
- Width: 100% → essa é uma outra forma de definir a largura de um objeto, em porcentagem (ao invés de em pixels)
- Display: inline → opção usada quando o menu é horizontal.

### 5.5.5 CRIANDO AS DIV'S DO CONTEÚDO

No conteúdo principal do nosso site, serão colocados três blocos com três notícias, sendo que cada uma delas irá conter uma imagem, um título e um texto. Como todos os blocos terão a mesma formatação, pode-se criar apenas um código CSS para todos eles. Nesse ponto, temos um problema: dentro de uma página, não pode ocorrer mais de uma div com o mesmo identificador (id); por isso, ela precisará ser substituída por uma classe (class), como mostra o código a seguir (dentro da div conteúdo):

```
<div class="lista">
    <div class="lista_img">
        
    </div><!--LISTA_IMG-->
```

```

<h1>W3C</h1>

<p>O Consórcio World Wide Web (W3C) é um consórcio internacional no
qual organizações filiadas, uma equipe em tempo integral e o público
trabalham juntos para desenvolver padrões para a Web.
(http://www.w3c.br/sobre/)</p>
</div> <!--LISTA-->

```

### Analizando o código:

- Div class="lista" → como comentado anteriormente, serão criadas 3 divs que receberão a mesma formatação, mas não é possível atribuir a mesma identificação (id) para todas elas; por isso é utilizada uma classe (class), fazendo com que possa ser usado o mesmo código CSS, como será visto posteriormente, para as três divs;

- Alt → como comentado no capítulo 3, a W3C recomenda o uso do atributo "alt" nas imagens, para que pessoas com o uso de leitores de tela possam ter acessibilidade total ao site;

Feito isso, basta triplicar o código XHTML, alterando apenas o seu conteúdo para formar as três notícias do conteúdo principal.

Agora, resta codificar o CSS das duas classes criadas (lista e lista\_img) para aplicar uma formatação. O código CSS dessas classes deve ficar da seguinte forma:

```

.lista{
    padding: 5px;
    background: #F4F4F4;
    margin: 0 0 5px 0;
    border: 1px solid #333;
}

.lista h1{
    font: 18px Tahoma, Arial, Helvetica, sans-serif;
    color: #F60;
    font-weight: bold;
    margin: 0 0 5px 0;
}

```



```
.lista p{
    font: 12px Georgia, Arial, Helvetica, sans-serif;
    color: #333;
    margin: 0 0 40px 0;
}

.lista_img img{
    width: 150px;
    height: 100px;
    padding: 3px;
    border: 1px dashed #999;
    float: left;
    margin-right: 10px;
}
```

Analizando o código:

- `.lista` → como se tratam de classes e não de identificadores (id), devemos atribuir o “.” (ponto) ao invés do sinal de “#”;
- `.lista h1` → declarando dessa forma, está sendo formatado apenas o conteúdo h1 da classe lista. Se fosse colocado apenas h1, sem referenciar a classe, a formatação seria aplicada a todos os elementos h1 da página. O mesmo ocorre com o texto do parágrafo (<p>) e com a imagem (<img>).

### 5.5.6 TOPO DO SITE

Seguindo para a parte final da preparação da página inicial do site, será colocado o logotipo e a frase de apresentação no topo do site.

No código XHTML, modificaremos a div topo para que fique da seguinte forma:

```
<div id="topo">
    <div id="topo_logo">
```

```

        
    </div> <!--LOGO-->

    <div id="topo_texto">
        "Por uma internet melhor!!!"
    </div> <!--TEXTO-->
</div>

```

Vale lembrar que, para o logotipo do site, foi criada uma imagem com as mesmas proporções definidas para a div **topo\_logo** no código CSS (250 X 80 pixels).

O código CSS do topo fica da seguinte forma:

```

#topo{
    height: 110px;
    background: #F3F3F3;
}

#topo_logo{
    width: 250px;
    height: 80px;
    background: #000;
    margin: 10px;
    float: left;
}

#topo_texto{
    width: 250px;
    height: 40px;
    float: right;
    margin: 35px;
}

#topo_texto p{
    font: italic 23px "Monotype Corsiva", Arial, Helvetica, sans-serif;
    color: #25037C;
}

```

```
}
```

### 5.5.7 RODAPÉ

Para terminar a formatação da página inicial, será formatado o rodapé, que conterà a assinatura do site.

O código XHTML para o rodapé é o seguinte:

```
<div id="rodape">
  <p align="center">Monografia - Padrões para Desenvolvimento Web<br />
    FATEC - Americana <br />
    Desenvolvendo um site nos Padrões Web
  </p>
</div>
```

Perceba o uso correto das quebras de linhas nos parágrafos através da marcação “<br />”.

O código CSS para o rodapé é o seguinte:

```
#rodape{
  width: 780px;
  background: #003399;
  height: 80px;
  float: left;
}

#rodape p{
  padding: 15px 0;
  font: 14px Tahoma, Arial, Helvetica, sans-serif;
  color: #FFF;
}
```

Feito isso, a página inicial deve estar com a seguinte aparência:

Figura 25 - Resultado final da página inicial do nosso site

The screenshot shows a website with the following layout:

- Header:** "PADRÕES WEB" in large purple letters, and the slogan "Por uma internet melhor!!!" in a cursive font.
- Navigation Menu:** A blue bar with links: "Início", "A empresa", "Portfólio", "Localização", and "Fale Conosco".
- Left Sidebar:** A vertical stack of five browser icons: Internet Explorer, Firefox, Chrome, Safari, and Opera.
- Main Content Area:** Three boxes with dashed borders:
  - W3C:** Features the W3C logo and text: "O Consórcio World Wide Web (W3C) é um consórcio internacional no qual organizações filiadas, uma equipe em tempo integral e o público trabalham juntos para desenvolver padrões para a Web. (W3C - <http://www.w3c.br/sobre/>)"
  - Governo Eletrônico:** Features the text "Governo Eletrônico" and "e-MAG - Modelo de Acessibilidade de Governo Eletrônico". Text: "O Modelo de Acessibilidade de Governo Eletrônico (e-MAG), consiste em um conjunto de recomendações a ser considerado para que o processo de acessibilidade dos sítios e portais do governo brasileiro seja conduzido de forma padronizada e de fácil implementação. (Governo Eletrônico - <https://www.governoeletronico.gov.br/acoes-e-projetos/e-MAG>)"
  - Web Standards:** Features an image of a mouse cursor clicking on "http://www" and the text "Web Standards". Text: "Web standards são padrões para a construção de sites determinados pelo W3C, órgão dirigido pelo próprio criador da web, Tim Berners Lee, e apoiado pelas maiores empresas de tecnologia da informação e comunicação do mundo, incluindo IBM, Microsoft, Google, HP, America Online, Apple, Adobe-Macromedia, Sun Microsystem, Mozilla Foundation e outras. (FatorW.com - <http://fatorw.com/internet/web-standards/o-que-sao-web-standards/>)"
- Footer:** A blue bar with the text: "Monografia - Padrões para Desenvolvimento Web", "FATEC - Americana", and "Desenvolvendo um site nos Padrões Web".

No próximo item, configuraremos as páginas internas do site.

### 5.5.8 CONFIGURANDO PÁGINAS INTERNAS

Criada a página inicial, nos resta a criação das páginas internas do site, que serão baseadas na própria página inicial, pois o conteúdo de todos os links do site aparecerá na div conteúdo. Dessa forma, todo o restante da estrutura será mantido. Como se trata de um site experimental, o texto das páginas internas será retirado do site Lorem Ipsum (<http://www.lipsum.com/>), muito utilizado pelos desenvolvedores para gerar conteúdo de exemplo para colocarem nos sites experimentais. As etapas da criação são:

- Abre-se o arquivo index.html e cria-se um outro arquivo com o nome (empresa.html):
- No arquivo criado, remover todas as div's filhas da div conteúdo;
- Com a div conteúdo vazia, será acoplada a ela uma tag h1 com o texto de "A empresa", que será o título da página;
- Criando algumas div's de parágrafo (<p>), coloca-se o texto de conteúdo da página;
- Ao final, a div conteúdo deve ter a seguinte forma:

```
<div id="conteudo">
    <h1>A empresa</h1>
    <p>TEXTO</p>
    <p>TEXTO</p>
    <p>TEXTO</p>
</div> <!--CONTEÚDO-->
```

Seguindo os passos anteriores para os demais links do site (portolio.html e localização.html), as páginas internas estarão prontas. A página contato.html será criada à parte, no próximo item.

### 5.5.9 CRIANDO A PÁGINA DE FALE CONOSCO

A página Fale Conosco terá uma atenção especial, pois 95% dos sites contêm pelo menos um formulário em sua estrutura, seja ele de contato, de login, de busca ou de qualquer outro propósito que vise dar interatividade ao site. Devido a essa importância, será utilizado o formulário de contato criado no capítulo 4 e aplicaremos formatação CSS sobre ele, tornando-o mais apresentável. Seguem as etapas da criação:

- Abrindo o arquivo index.html e salvando-o com outro nome (contato.html):
- No arquivo criado, remover todas as div's filhas da div conteúdo;
- Com a div conteúdo vazia, adiciona-se a ela uma tag h1 com o texto de "Fale Conosco", que será o título da página;

- Copiar o formulário de contato do capítulo 4, item 4.5.6, dentro da div conteúdo;
- No arquivo estilos.css, será formatado cada elemento desse formulário, ficando da seguinte forma:

```
fieldset {  
    width: 620px;  
    border: none;  
}  
fieldset legend{  
    font: Bold 15px Verdana, Arial, Tahoma;  
}  
label{  
    display: block;  
    font: Bold 12px Verdana, Arial, Tahoma;  
    width: 200px;  
    margin: 2px 0 12px 0;  
}  
input{  
    width: 500px;  
    height: 25px;  
    color: #666;  
    font-weight: bold;  
}
```

A explicação de cada elemento HTML do formulário de contato encontra-se na tabela 3 do item 4.5.6, no capítulo 4.

Ao abrir essa página no navegador, ela deverá ter a seguinte aparência:

Figura 26 - Formulário de contato formatado com CSS

**Fale Conosco**

Dados Pessoais

Nome:

Sobrenome:

Sexo:

Dados de Localização

Estado:

Cidade:

Como se pode notar, os botões “Enviar!” e “Limpar!” e o elemento de texto do campo “Estado” adquiriram a largura estipulada para o elemento “input”, ficando com o mesmo tamanho que os campos de texto do formulário, o que não resultou em uma boa estética, sendo que os botões que se costuma ver nos formulário de contato são menores.

Para resolver esse problema, será novamente utilizado o recurso de criação de classes, pois elas nos permitirão realizar uma formatação em particular para esses dois botões. Para realizar tal efeito, faz-se da seguinte forma:

- No arquivo contato.html, no qual é feita a declaração dos inputs dos botões e do campo de texto Estado, será realizada uma alteração para acrescentar essas classes, ficando da seguinte forma:

```
<label for="Estado">Estado: <input type="text" name="estado"
class="estado" /> </label>
<input type="submit" name="enviar" value="Enviar!" class="envia" />
```

```
<input type="reset" name="limpar" value="Limpar!" class="limpa" />
```

- Feito isso, pode-se voltar ao arquivo estilos.css e realizar a formatação dessas classes<sup>6</sup>, ficando da seguinte forma:

```
.envia, .limpa{  
    width: 60px;  
}  
  
.estado{  
    width: 30px;  
}
```

Como o objetivo era o de definir a mesma largura para os dois botões, eles podem ser feitos na mesma declaração, apenas separados por vírgula, como pode ser observado no código. Feito isso, mesmo esses itens se tratando de inputs, que estão com outra formatação, vão adquirir a forma que foi definida nessas classes, e não no elemento input declarado anteriormente, mostrando a grande importância das classes para solucionarem os problemas com formatação. A nova aparência dessa página será a seguinte:

Figura 27 - Formulário de contato após a aplicação de classes nos botões



**Fale Conosco**

Dados Pessoais

Nome:

Sobrenome:

Sexo:

Dados de Localização

Estado:

Cidade:

<sup>6</sup> O recurso de criação de classes foi visto com mais detalhes no item 5.5.5 desse capítulo.



Agora sim, nosso formulário está com uma aparência muito mais agradável. Vale lembrar que, mesmo para os designers, que têm seu foco principal na formatação, é muito importante ter um conhecimento ainda que básico em alguma linguagem de programação de servidor, como o PHP, ASP, JSP, etc. para trabalhar juntamente com o XHTML e o CSS, pois a manutenção do código seria mais fácil. Como o foco dessa monografia é o uso de padrões, essas linguagens não foram abordadas. Ainda assim podemos citar o fato de que, por exemplo, a utilização do PHP facilitaria muito na criação do menu, pois da maneira como foi apresentada, se fosse necessário inserir um item novo neste, seria necessário realizar a alteração em todas as páginas do site. Com a utilização do PHP isso não seria preciso, pois o menu seria colocado em um arquivo à parte e as páginas fariam apenas referência a esse menu, sendo que, se fosse preciso incluir um item, isso seria feito no arquivo e automaticamente todas as páginas seriam atualizadas com o novo menu. Para um maior aprofundamento na linguagem, o site oficial do PHP (<http://www.php.net/>) possui a documentação completa da linguagem.

Como comprovado, este site foi realizado seguindo os padrões exigidos pela W3C para a sua melhor visualização em todos os tipos de dispositivos, e analisando-o cuidadosamente, pode ser notada a grande melhoria frente ao antigo método de desenvolvimento, no que diz respeito aos seguintes aspectos:

- Topo sem animações: apenas com uma imagem e um texto, o topo foi bem definido e fugiu um pouco dos que anteriormente eram feitos com animações em flash, deixando o carregamento mais lento e chamando a atenção mais ao topo do que ao conteúdo;
- Menus com lista numerada: um menu criado inteiramente com HTML e CSS, sem necessidade de utilização de Flash ou de JavaScript;
- Estruturação sem tabelas: a divisão dos blocos do site, que antes só era conseguida com a utilização de tabelas, foi feita inteiramente com CSS através de divs, gerando um significativo aumento no carregamento da página;
- Separação de estrutura e formatação: para uma facilidade maior de manutenção, o código da estrutura é completamente separado da sua formatação, fazendo com que em poucos minutos uma manutenção possa ser feita.

A criação do nosso site de demonstração está completa. No capítulo seguinte, serão feitos testes nos principais navegadores do mercado para comprovar se a sua visualização ocorrerá de forma correta e em seguida o XHTML e o CSS do site serão validados por mecanismos do W3C para comprovar se os padrões foram aceitos.

## 6 TESTES E VALIDAÇÕES

Após a criação do site, inicia-se uma etapa muito importante do desenvolvimento: os testes. Como descrito no capítulo 2, devido à grande quantidade de navegadores existente no mercado e ao fato de cada um empregar o seu próprio método de interpretação de um site, é necessário fazer os testes de validação para ser comprovado que nosso site será visualizado corretamente, independentemente do navegador ou dispositivo utilizado. Para os testes, serão utilizados os seguintes navegadores:

- Internet Explorer (versões 6, 7 e 8), presente no Sistema Operacional Windows da Microsoft. Desde a versão do Windows Vista, a Microsoft retirou a versão 6 do seu sistema, mas como ainda existem muitos usuários da versão XP (versão em que o Internet Explorer 6 vem instalado por padrão) e não se preocupam em fazer a atualização para uma versão mais recente, ainda se deve ter a preocupação com os testes nesta versão. Para baixar a versão mais recente, basta acessar o endereço <http://www.microsoft.com/brasil/windows/internet-explorer>. Para os testes em outros navegadores, pode-se utilizar o programa IETester (<http://www.my-debugbar.com/wiki/IETester/HomePage>), que faz a instalação do Internet Explorer 6 e 7.
- Firefox, da empresa Mozilla, navegador que apareceu como um dos principais concorrentes do Internet Explorer. Para os testes, foram utilizadas as versões 2.0 até a 3.6.11. Para baixar o Firefox, basta acessar o endereço <http://br.mozdev.org/download/>.
- Chrome, da empresa Google, browser que vem ganhando cada vez mais usuários no mercado, testado nas versões 6 e 7. Para baixar o Chrome, basta acessar o endereço <http://www.google.com/chrome>.
- Safari, da empresa Apple, não pode ficar fora dos testes por ser o principal browser dos usuários Mac, Sistema Operacional da Apple, mas também disponível para o Sistema Operacional Windows, no qual foi utilizado para testes nas versões 4 e 5. Para baixar o Safari, basta acessar o endereço <http://www.apple.com/br/safari>.

- Opera, da empresa Opera Software, navegador muito utilizado e tido por muitos como o primeiro navegador a ser 100% padronizado de acordo com os padrões da W3C e o mais seguro segundo a empresa de segurança Symantec<sup>7</sup>. Para baixar o Opera, basta acessar o endereço <http://www.opera.com>.
- Opera Mobile, browser da Opera Software usado na maior parte dos dispositivos móveis. Para utilizar o Opera Mobile, basta acessar o endereço <http://www.opera.com/mobile/demo>.

Felizmente, a maioria desses navegadores estão de acordo com as normas propostas pela W3C e geralmente não são comuns problemas com eles. A maior preocupação dos desenvolvedores são as versões 6 e 7 do internet Explorer, pois como são versões que não seguem corretamente os padrões e muitos ainda as utilizam, para elas devem-se fazer testes mais específicos.

## 6.1 FIREFOX, CHROME, OPERA E SAFARI

Para acessar o site criado, basta dar um clique-duplo sobre o arquivo `index.html` no diretório utilizado para armazenar os arquivos do site e automaticamente o seu navegador padrão será aberto e exibirá a tela inicial. Feito isso, para testar em outros navegadores, basta copiar o caminho do site na barra de endereços e colar na barra do outro navegador a ser testado, seguido da tecla Enter.

Ao acessar o site nesses 4 navegadores, nenhuma propriedade definida no código CSS foi perdida, e o site apresentou o layout definido inicialmente, ficando com o seguinte aspecto:

---

<sup>7</sup> Disponível em <http://my.opera.com/operabrasil/blog/2010/04/20/opera-o-browser-mais-seguro-segundo-a-symantec>

Figura 28 - Site nos navegadores Firefox, Chrome, Opera e Safari



## 6.2 INTERNET EXPLORER

Como apresentado no início desse capítulo, as versões 6, 7 e 8 do Internet Explorer participaram do teste, sendo que na versão 8, nenhum erro foi encontrado. Já nas versões 6 e 7, um pequeno desvio foi ocorrido no menu vertical, no qual o espaçamento ficou maior entre os itens que fazem parte deste menu, como comprova a figura a seguir:

Figura 29 - Problema com a margem no Internet Explorer 6 e 7



Isso ocorreu porque o Internet Explorer não trabalha bem com a propriedade `margin`. Para corrigir esse problema, deve-se criar uma folha de estilo paralela, somente para o caso de o navegador ser o Internet Explorer 6, e tentar corrigir o problema. Este caso pode ser solucionado da seguinte forma:

- Cria-se uma cópia do arquivo `estilos.css` e se aplica a ele o nome de `estilosie6.css`;
- Na chamada para esse arquivo nas páginas do site, coloca-se um comentário condicional, logo abaixo da chamada principal do CSS, indicando que se o navegador for o Internet Explorer 6, ele deve executar o arquivo `estilosie6.css`;

```
<!--[if IE 6]>
<link rel="stylesheet" href="estilosie6.css"
type="text/css" />
<![endif]-->
```

- No arquivo `estilosie6.css`, localizar a formatação CSS `#menu_vert ul li a`, e alterar a sua propriedade `margin` para "0";
- Ao visualizar o arquivo, o tamanho da margem terá diminuído;
- Fazer os mesmos passos para realizar a validação caso o browser seja o Internet Explorer 7.

Nesse caso, o erro foi solucionado apenas alterando o valor da propriedade `margin`, mas os erros poderiam ser maiores na medida em que o site aumenta, e deveria ser seguida a mesma ideia para solucionar os outros problemas.

Figura 30 - Visualização do menu corrigido no Internet Explorer 6 e 7



### 6.3 OPERA MOBILE

Para realizar os testes para dispositivos Mobile, será utilizado um aplicativo que faz a simulação do browser de um dispositivo móvel, o Opera Mobile<sup>8</sup>. Ele é muito útil para que os desenvolvedores façam os testes de suas páginas no próprio PC, já sabendo como ele ficará quando executado em um dispositivo *Mobile*. A escolha desse browser ocorreu devido ao grande número de celulares com o Opera e pela eficiência com que são tratadas as páginas, reconhecendo totalmente as folhas de estilo CSS.

Como o teste será realizado diretamente do site do Opera, o site de exemplo deve estar hospedado em algum provedor para poder ser visualizado. Com uma rápida busca no Google, encontram-se centenas de provedores gratuitos, e como não foi utilizada nenhuma tecnologia de servidor (como o PHP, por exemplo), esse número é ainda maior. Após fazer a hospedagem, basta digitar o caminho do site na barra de endereços do Opera Mobile e teclar “Enter” para que o site seja carregado na tela do mini navegador.

---

<sup>8</sup> O Opera Mobile pode ser encontrado em <http://www.opera.com/mobile/demo/>

### 6.3.1 OPÇÕES DE VISUALIZAÇÃO

Com o Opera mobile, pode-se visualizar nosso site de maneiras diferentes. Ao entrar no endereço de nosso site, ele aparecerá com a seguinte visualização:

Figura 31 - Nosso site sendo visualizado no Opera Mobile



Fonte: <http://www.opera.com/mobile/demo/> (25/10/2010)

Como se pode observar, o site é carregado na sua totalidade, mas ao clicar em algum lugar da tela, automaticamente é feita a ampliação para que possa ser visualizado seu conteúdo, bastando um clique-arraste com o mouse para mover a página e clicar sobre um link para ver o seu conteúdo. Uma outra maneira de movimentar o site é com as setas de movimentação do teclado e teclando Enter para acessar um link.



Figura 32 - Ampliação de tela no Opera Mobile



Fonte: <http://www.opera.com/mobile/demo/> (25/10/2010)

O Opera Mobile também apresenta uma outra forma de visualização que traz as informações do site de uma maneira diferente da que definimos, mas em que é completamente possível visualizar o seu conteúdo e é também a maneira que alguns celulares oferecem para visualização. Para conhecermos esse novo método, basta clicarmos no último botão da barra de ferramentas inferior do navegador Mobile e em seguida clicarmos em “Settings”. Das opções que surgem, ativamos o item “Mobile View” e clicamos no botão “Back”. Aparecerá uma mensagem perguntando se queremos recarregar a página e, clicando em “Yes”, aparecerá o novo modo de visualização.

Figura 33 - Visualização no modo "Mobile View"



Fonte: <http://www.opera.com/mobile/demo/> (25/10/2010)

Nesse novo método, a página é toda disposta apenas na vertical, e podemos movê-la com um clique-arraste no mouse ou com as setas de movimentação. Podemos perceber que, apesar dessa nova disposição, a página é 100% navegável e podemos visualizar corretamente o seu conteúdo, tudo isso graças ao uso dos padrões utilizados para o desenvolvimento.

#### 6.4 VALIDANDO O SITE

Após realizarmos os testes nos principais navegadores disponíveis, chegou o momento de descobrirmos se nosso site realmente é 100% validado de acordo com as normas da W3C, e para isso, utilizaremos as ferramentas disponíveis pelo próprio consórcio para validarmos tanto o XHTML quanto o CSS.

### 6.4.1 VALIDANDO O XHTML

Para validarmos o XHTML, usaremos uma ferramenta oferecida pela W3C, o Markup Validation Service, que pode ser acessado pelo endereço <http://validator.w3.org/>. Veremos que a ferramenta nos oferece três modos de validação: Validate by URI (que nos permite validar simplesmente colocando o endereço do site<sup>9</sup>), Validate by File Upload (no qual podemos selecionar o arquivo que queremos validar) e finalmente Validate by direct input (no qual colamos todo o código da página a ser validada e é a opção que vamos utilizar). Após inserir o código no campo e clicar em “Check”, será iniciada a validação e apresentado o resultado final. Se nosso site foi corretamente validado, aparecerá a seguinte mensagem:

Figura 34 – Validação com sucesso

A green rectangular banner with white text that reads: "This document was successfully checked as XHTML 1.0 Transitional!"

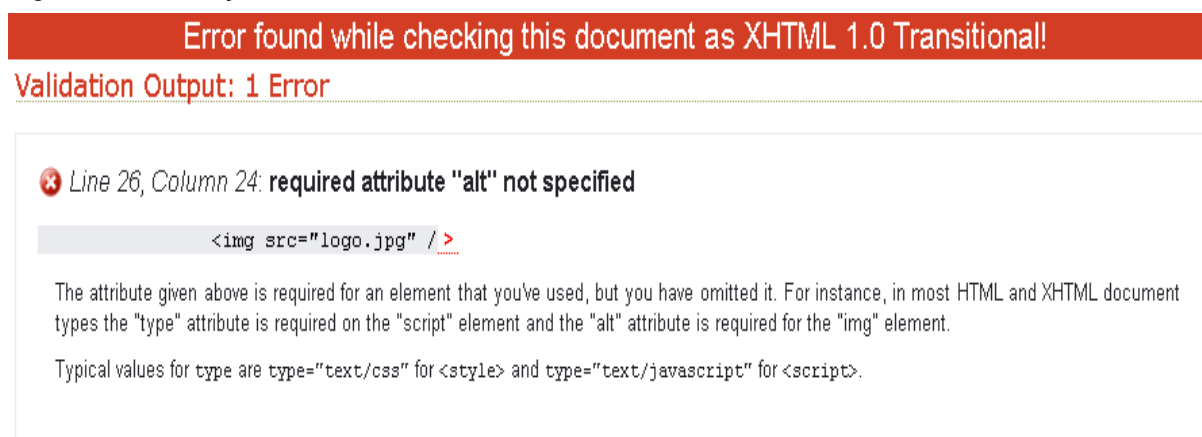
Fonte: <http://validator.w3.org/> - 27/10/2010)

Essa mensagem indica que nosso documento passou com sucesso na validação. Para testarmos se o que foi passado na teoria é realmente correto, vamos propositalmente eliminar o atributo “alt” da tag de inserção de imagem do logo e validaremos novamente o site. Podemos perceber que a mensagem de resultado indicará que foi encontrado um erro na validação, e ao rolarmos a página, ele demonstra qual foi o erro encontrado.

---

<sup>9</sup> Não indicado para site hospedados em servidores gratuitos, pois alguns desses servidores inserem linhas de código no site, atrapalhando a validação.

Figura 35 – Validação com erro



Fonte: <http://validator.w3.org/> - 27/10/2010)

Como comprovado, o erro acusado pela página é exatamente a não especificação do atributo “alt” na tag de inserção da imagem.

#### 6.4.2 VALIDANDO O CSS

Após validarmos corretamente o HTML e o XHTML da página, nos resta validarmos a folha de estilo. Também utilizaremos uma ferramenta oferecida pela própria W3C, o CSS Validation Service, que pode ser acessado pelo site <http://jigsaw.w3.org/css-validator/>. Esta ferramenta dispõe das mesmas opções do validador de XHTML, então também faremos a opção de “Validar por entrada direta”, como é apresentado na página.

Após acessarmos o endereço, copiamos todo o código do arquivo estilos.css, que contém nossa folha de estilos, colamos no campo de entrada e clicamos em “verificar”. Em seguida, é exibida a página de resultados da validação, na qual podemos descobrir se foram encontrados erros na nossa folha de estilos ou não. Como nosso site não apresentou nenhum problema, podemos ver a seguinte mensagem:

Figura 36 - Correta validação da Folha de Estilo



Fonte: <http://validator.w3.org/> - 27/10/2010)

Podemos realizar um teste parecido com o do item anterior, modificando propositalmente a propriedade do elemento “float” para “center”, e como float nunca pode ser center, veremos que o resultado da validação será o seguinte:

Figura 37 - Validação incorreta da Folha de Estilo

Lamento! Encontramos os seguintes erros (2)		
URI : TextArea		
34	#topo_logo	Erro de valor : <a href="#">float</a> center não é um valor float : center center
137	#conteudo	Erro de valor : <a href="#">float</a> center não é um valor float : center center

Fonte: <http://validator.w3.org/> - 27/10/2010)

Como comprovado, o erro retornado pela página é exatamente o valor center definido para o float.

Com isso, temos nosso site 100% validado de acordo com as normas da W3C. Isso nos mostra que sempre devemos usar as boas técnicas de desenvolvimento utilizando os padrões, para que sempre possamos validar nossos sites e fazer com que eles sejam visualizados em todos os browsers e dispositivos, com uma significativa rapidez no carregamento das páginas.

## 7 CONSIDERAÇÕES FINAIS

Estabelecer padrões Web para produzir sites acessíveis nos dias de hoje tornou-se imprescindível para controlar toda a evolução do uso da Internet.

Permitir que uma maior quantidade de pessoas tenham acesso a todo e qualquer tipo de informação, independentemente do tipo de tecnologias, sistemas operacionais, navegadores e condições pessoais, possibilita uma experiência enriquecedora tanto para usuários como para programadores.

Como foi demonstrado no estudo de caso para a criação de um site seguindo os padrões, cada elemento deve ser corretamente estruturado para que os padrões sejam mantidos e que o site seja corretamente visualizado em todos os tipos de navegadores e dispositivos, causando um significativo aumento no tempo de carregamento da página, devido à retirada de recursos como animações e efeitos que antes só eram possíveis com a utilização de uma grande quantidade de scripts, porém que hoje são possíveis apenas com um XHTML bem estruturado e formatação com folhas de estilo CSS.

Nos testes foram demonstradas as diferenças que podem ocorrer entre um navegador e outro e as formas que podem ser lançadas para corrigi-las, fazendo com que todos os usuários tenham acesso não apenas às mesmas informações, mas também ao mesmo layout do site.

Mesmo com a grande relutância de alguns desenvolvedores em deixarem o método tradicional de desenvolvimento de lado e aderirem aos novos métodos, as vantagens dos padrões Web são muitas e tendem a aumentar ainda mais com os lançamentos da versão 5 do HTML e da versão 3 do CSS, que ainda estão no início e, portanto, poucos navegadores oferecem atualmente suporte à essas tecnologias, mas que certamente são muito promissoras por garantirem ainda uma maior facilidade de utilização dos elementos, aumentando ainda mais a interatividade nos websites.

Sendo assim, deixo como sugestão para futuros estudos as novidades da versão 5 do HTML em conjunto com a versão 3 do CSS, pois são tecnologias que,

juntamente com os conceitos de acessibilidade e usabilidade, formarão o futuro da Web.

## 8 REFERÊNCIAS BIBLIOGRÁFICAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. Citação: NBR-10520/ago - 2002. Rio de Janeiro: ABNT, 2002.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. Referências: NBR-6023/ago. 2002. Rio de Janeiro: ABNT, 2002.

BRASIL. IBGE Censo 2000. **IBGE e CORDE abrem encontro internacional de estatísticas sobre pessoas com deficiência.** Disponível em <[http://www.ibge.gov.br/home/presidencia/noticias/noticia\\_visualiza.php?id\\_noticia=438&id\\_pagina=1](http://www.ibge.gov.br/home/presidencia/noticias/noticia_visualiza.php?id_noticia=438&id_pagina=1)> . Acesso em 20/10/2010. 20h15.

Discovery Brasil. **A Internet, seis décadas.** 2009a. Disponível em <<http://www.discoverybrasil.com/internet/interactivo.shtml>>. Acesso em 15/09/2010. 19h50.

FERREIRA, E.L.; EIS, D. A. **Palestra Padrões Web Tableless.** Disponível em <<http://www.tableless.com.br/video-palestra-de-tableless>>. Acesso em: 31/08/2010. 21h05.

FREEMAN, E; FREEMAN, Eric. **Use a Cabeça – HTML com CSS e XHTML.** 2ª edição. Tradução Betina Macêdo. Rio de Janeiro, 2008

Gameleira, F. **Cartilha de Acessibilidade.** 2002a. Disponível em <<http://www.lupadigital.info/>>. Acesso em 21/09/2010. 22h15.

GODINHO, F. (Coord.). **Internet para Necessidades Especiais.** Edição UTAD/GUIA, 1999. Disponível em: <http://www.acessibilidade.net/web/ine/livro.html>. Acesso em: 27/09/2010. 19h40.

LEINER, B M.; CERT, V. G.; CLARK, D. D.; KAHN, R. E.; KLEINROCK, L.; LYNCH, D. C.; POSTEL, J.; ROBERTS, L. G.; WOLFF, S. **A Brief History of the Internet.** Disponível em <<http://www.isoc.org/internet/history/brief.shtml>>. Tradução Aísa Pereira. Disponível em <<http://www.aisa.com.br/index1.html>>. Acesso em 03/09/2010. 12h35.

LEITE. R. V; Upinside Tecnologia. Disponível em <http://www.upinside.com.br>. Acesso em 01/09/2010. 20h00.

W3C – World Wide Web Consortium, Escritório Brasil. **Guia de Referência XHTML.** 2009/15p.



W3C – World Wide Web Consortium, Escritório Brasil. **Guia de Referência CSS**. 2009/22p.

W3C – World Wide Web Consortium. **Sobre o Consórcio World Wide Web**. Disponível em <<http://www.w3c.br/sobre>>. Acesso em 05/09/2010. 20h20.

## ANEXO

### Código fonte completo do site de exemplo criado no capítulo 5

index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
  <meta name="keywords" content="web, padres, W3C" />
  <meta name="description" content="Criando um site nos padres Web" />
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <link rel="stylesheet" href="estilos.css" type="text/css" />
  <!--[if IE 6]>
  <link rel="stylesheet" href="estilosie6.css" type="text/css" />
  <![endif]-->

  <!--[if IE 7]>
  <link rel="stylesheet" href="estilosie7.css" type="text/css" />
  <![endif]-->
  <title>Padrões para Desenvolvimento Web</title>
</head>
<body>
<div id="box">

  <div id="topo">
    <div id="topo_logo">
      <a href="index.html"></a>
    </div> <!--LOGO-->
    <div id="topo_texto">
      <p>"Por uma internet melhor!!!"</p>
    </div> <!--TEXTO-->
  </div>
  <div id="menu">
    <ul>
      <li><a href="index.html">Início</a></li>
      <li><a href="empresa.html">A empresa</a></li>
      <li><a href="portfolio.html">Portfólio</a></li>
```

```

        <li><a href="localizacao.html">Localização</a></li>
        <li><a href="contato.html">Fale Conosco</a></li>
    </ul>
</div> <!--MENU HORIZONTAL-->
<div id="principal">
    <div id="menu_vert">
        <ul>
            <li><a href="index.html">Início</a></li>
            <li><a href="empresa.html">A empresa</a></li>
            <li><a href="portfolio.html">Portfólio</a></li>
            <li><a href="localizacao.html">Localização</a></li>
            <li><a href="contato.html">Fale Conosco</a></li>
        </ul>
        
    </div> <!--MENU VERTICAL-->
    <div id="conteudo">
        <div class="lista">
            <div class="lista_img">
                
            </div><!--LISTA_IMG-->
            <h1>W3C </h1>
            <p>O Consórcio World Wide Web (W3C) é um consórcio
internacional no qual organizações filiadas, uma equipe em tempo integral e
o público trabalham juntos para desenvolver padrões para a Web. (W3C -
http://www.w3c.br/sobre/)</p>
        </div> <!--LISTA-->
        <div class="lista">
            <div class="lista_img">
                
            </div><!--LISTA_IMG-->
            <h1>e-MAG - Modelo de Acessibilidade de Governo
Eletrônico</h1>
            <p>O Modelo de Acessibilidade de Governo Eletrônico (e-
MAG), consiste em um conjunto de recomendações a ser considerado para que o
processo de acessibilidade dos sítios e portais do governo brasileiro seja
conduzido de forma padronizada e de fácil implementação. (Governo
Eletrônico - https://www.governoeletronico.gov.br/acoes-e-projetos/e-
MAG)</p>
        </div> <!--LISTA-->
        <div class="lista">
            <div class="lista_img">

```

```

        
    </div><!--LISTA_IMG-->
    <h1>Web Standards</h1>
    <p>Web standards são padrões para a construção de sites
determinados pelo W3C, órgão dirigido pelo próprio criador da web, Tim
Berners Lee, e apoiado pelas maiores empresas de tecnologia da informação e
comunicação do mundo, incluindo IBM, Microsoft, Google, HP, America Online,
Apple, Adobe-Macromedia, Sun Microsystem, Mozilla Foundation e outras.
(FatorW.com - http://fatorw.com/internet/web-standards/o-que-sao-web-
standards/)</p>
    </div> <!--LISTA-->
    </div> <!--CONTEDO PRINCIPAL-->
</div> <!--PRINCIPAL-->
<div id="rodape">
    <p align="center">Monografia - Padrões para Desenvolvimento
Web<br />
    FATEC - Americana <br />
    Desenvolvendo um site nos Padrões Web</p>
</div>
</div> <!--BOX-->
</body>
</html>

```

## Estilos.css

```
/*Tirando a margin e o padding da página toda*/
* {
    margin: 0;
    padding: 0;
}

img{
    border: 0;
}

body{
    background: #F3F3F3;
    color: #000;
    font: 13px Verdana, Arial;
}

#box{
    width: 780px;
    /*height: 500px;*/
    margin: 0 auto;
    /*border: 1px solid #FFFFFF;*/
    background: #FFF; /*url(fundo_bg.jpg) repeat-y;*/
}

/*CABEÇALHO*/
#topo{
    height: 110px;
    background: #F3F3F3;
}

#topo_logo{
    width: 250px;
    height: 90px;
    margin: 10px;
    float: left;
}
}
```

```
#topo_logo img{
    border: 0px;
    background: #FFF;
}

#topo_texto{
    width: 250px;
    height: 40px;
    float: right;
    margin: 35px;
}

#topo_texto p{
    font: italic 23px "Monotype Corsiva", Arial, Helvetica, sans-serif;
    color: #25037C;
}

/*MENU HORIZONTAL*/
#menu{
    height: 30px;
    background: #003399;
}

#menu ul{
    margin: 0;
    padding: 0;
    list-style: none;
    width: 100%;
    float: left;
}

#menu ul li{
    display: inline;
    float: left;
}

#menu ul li a{
    text-decoration: none;
    padding: 6px 10px;
    font: 14px Tahoma, Arial, Helvetica, sans-serif;
    color: #FFF;
}
```

```
        font-weight: bold;
        float: left;
        margin: 0 1px;
    }

#menu ul li a:hover{
    color: #09F;
    background: #FFF;
}

#principal{
    width: 780px;
    background: #F3F3F3;
}

/*MENU VERTICAL*/
#menu_vert {
    width: 140px;
    /*background: #006644;*/
    /*height: 280px;*/
    float: left;
}

#menu_vert ul{
    margin: 0;
    padding: 0;
    list-style: none;
    float: left;
}

#menu_vert ul li{
    width: 130px;
}

#menu_vert ul li a{
    text-decoration: none;
    color: #FFF;
    background: #333;
    float: left;
    width: 130px;
    padding: 5px;
}
```

```
margin: 1px 0 1px 0;
/*_margin: 0;
*margin: 0; apenas para IE6 e 7*/
font: 14px Tahoma, Arial, Helvetica, sans-serif;
font-weight: bold;
border-left: 6px solid #CC3333;
}

#menu_vert ul li a:hover{
background: #CC3333;
border-left: 5px solid #333;
}

#conteudo {
width: 620px;
background: #F3F3F3;
/*height: 280px;*/
float: right;
padding: 5px 0;
}

.lista{
padding: 5px;
background: #F4F4F4;
margin: 0 0 10px 0;
border: 1px solid #333;
}

.lista h1{
font: 18px Tahoma, Arial, Helvetica, sans-serif;
color: #F60;
font-weight: bold;
margin: 0 0 5px 0;
}

.lista p{
font: 12px Arial, Helvetica, sans-serif;
color: #333;
margin: 0 0 20px 0;
}
```



```
.lista_img img{
    width: 150px;
    height: 80px;
    padding: 3px;
    border: 1px dashed #999;
    float: left;
    margin-right: 10px;
}

#rodape{
    width: 780px;
    background: #003399;
    height: 80px;
    float: left;
}

#rodape p{
    padding: 15px 0;
    font: 14px Tahoma, Arial, Helvetica, sans-serif;
    color: #FFF;
}

/*FORMULÁRIO*/
fieldset {
    width: 620px;
    border: none;
}

fieldset legend{
    font: Bold 15px Verdana, Arial, Tahoma;
}

label{
    display: block;
    font: Bold 12px Verdana, Arial, Tahoma;
    width: 200px;
    margin: 2px 0 12px 0;
}

input{
```

```
        width: 500px;
        height: 25px;
        color: #666;
        font-weight: bold;
    }

    .envia, .limpa{
        width: 60px;
    }

    .estado{
        width: 30px;
    }

    input:focus, select:focus{
        border: 2px solid #0066FF;
        background: #00CC99;
    }
```