



UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ

CAMPUS LUIZ MENEGHEL

FABIO MARCEL F. ESPERANDIO

COMPARATIVO PRÁTICO ENTRE HTML E HTML5

Bandeirantes

2013

FABIO MARCEL F. ESPERANDIO

COMPARATIVO PRÁTICO ENTRE HTML E HTML5

Monografia apresentada à Universidade Estadual do Norte do Paraná – *campus* Luiz Meneghel – como requisito parcial para aprovação no Curso de Sistemas de Informação.

Orientador: Prof. Rodrigo Tomaz Pagno

Bandeirantes

2013

FABIO MARCEL F. ESPERANDIO

COMPARATIVO PRÁTICO ENTRE HTML E HTML5

Monografia apresentada à Universidade Estadual do Norte do Paraná – *campus* Luiz Meneghel – como requisito parcial para aprovação no Curso de Sistemas de Informação.

COMISSÃO EXAMINADORA

Prof. Me. Rodrigo Tomaz Pagno
UENP – *Campus* Luiz Meneghel

Prof. Me. Neimar Neitzel
UENP – *Campus* Luiz Meneghel

Prof. Me. Bruno Miguel de Souza
UENP – *Campus* Luiz Meneghel

Bandeirantes, 26 de Novembro de 2013.

AGRADECIMENTOS

Agradeço muita coisa a muitas pessoas.

Agradeço, principalmente, às muitas amizades feitas no decorrer de todos esses anos de estudo. Em especial, aos companheiros Dani Cachorrão, Gustavo Pimpão, Pedro Perereca e Sergio Coyote, por me fazerem compreender o significado da palavra família.

Agradeço também aos parceiros Lucas Rodela, Jean Mofo, Silvana Rainha, Carollyne Princesa, Gustavo Cu, Renato Jahú, Alúcio Careca, Alberto Furquia, Naiara Lontra, Fernando Aedes, Erika Anja, Galeno Rabello Americano do Brasil (não, não é apelido), Neto Cogumelo, Sylvia Docinho, Emiliana, Rodrigo Rogério, Muguilo, Fabio Conchas, Ivan Brabão, Fernandinha Safadinha, Thiago Manfrinhô, Kenia Tantão, Heron BigPutá, Hilário KamenXuxa, por todo o Apoio_Diversão_Aventura_Confusão_Histórias, sem vocês, eu, provavelmente, estaria preso ou morto.

Agradeço, com toda convicção, aos professores Stevan, Della Mura, Christian, Sgarbi, Fernando, Biluka, Bruno, por terem desempenhado bem seus papéis e ensinado coisas realmente úteis. Em especial, à falecida Cristiane Yanase, por ser para sempre minha querida *tia*. Ainda de forma muito especial, ao meu orientador Rodrigo Pagno, por ter surgido como um messias para almas condenadas e pela excelente orientação.

Agradeço ainda a XVII Turma de Sistemas de Informação da Universidade Estadual do Norte do Paraná - Campus Luiz Meneghel – que, de muitas maneiras, foi a melhor turma que eu conheci na faculdade.

Agradeço à cidade de Bandeirantes, que, direta ou indiretamente, ensinou-me muito sobre a vida.

Agradeço a meu pai, Túlio, por ter sido exatamente o que precisava ser; e à minha mãe, Conceição, pela boa vontade.

Agradeço à empresa Google, por tudo que faz para facilitar a vida pelo módico preço da privacidade.

Agradeço aos Johnnies, Cash e Walker, por fazerem do mundo um lugar suportável.

*[...] "So as you read this know my friends
I'd love to stay with you all
Please smile when you think of me
My body's gone, that's all*

*To all the world
To all my friends
I love yo
I have to leav
These are the last words
I'll ever speak
And they'll set me free"[...]
-Megadeth*

RESUMO

Este trabalho refere-se ao estudo das mudanças impostas ao HTML, em sua quinta versão, objetivando analisar as diferenças, quando comparadas à quarta versão. Para tanto, usou-se métodos de pesquisa descritivos, experimentais e exploratórios, a fim de se observar três perspectivas de mudança: *perspectiva de sistema*, através de testes e monitoramento de recursos computacionais; *perspectiva de desenvolvedor*, através de entrevistas, a fim de descobrir quais as opiniões e as expectativas de profissionais da área; *perspectiva de usuário*, através de entrevistas, a fim de extrair dados sobre sua interação com documentos *Web* e suas opiniões sobre o assunto. Embasado na comparação entre os dados obtidos no estudo das duas referidas versões da HTML, espera-se concluir quais são os reais impactos das melhorias na estruturação da linguagem.

PALAVRAS-CHAVE: HTML, Desenvolvimento *Web*, *Plugins*, HTML5, Internet.

ABSTRACT

This work refers to the study of the changes imposed on the HTML in its fifth version, in order to analyze their differences compared to the fourth version. To do so we use descriptive, exploratory and experimental research methods to observe three perspectives on this change, system perspective, consisting of testing and monitoring of computing resources; developer standpoint, through interviews, find out what the opinions and expectations of professional the related field; user perspective through interviews, extract data about your interaction with web documents and their opinions on this. Based upon the comparison between the data obtained for both versions of HTML, is expected to conclude on which the actual impact of improvements in language structure.

KEYWORDS: HTML, *Web Development*, *Plugins*, HTML5, *Internet*.

LISTA DE ABREVIATURAS E SIGLAS

CSS	- <i>Cascading Style Sheet</i>
HTML	- <i>HyperText Markup Language</i>
HTTP	- <i>HyperText Transfer Protocol</i>
URL	- <i>Uniform Resource Locator</i>
W3C	- <i>World Wide Web Consortium</i>
XHTML	- <i>eXtensible HyperText Markup Language</i>
XML	- <i>eXtensible Markup Language</i>
WYSIWYG	- <i>What You See Is What You Get</i>
RAM	- <i>Random Access Memory</i>
HDD	- <i>Hard Disk Drive</i>
ROM	- <i>Read Only Memory</i>

LISTA DE TABELAS

Tabela 1 – Mudanças entre HTML4 e XHTML.....	22
Tabela 2 - Novas <i>Tags</i> de Estrutura HTML5.....	23
Tabela 3 - Novas <i>Tags</i> de Controle de Formulário HTML5.....	25
Tabela 4 - Novos tipos de entrada do usuário HTML5.....	25
Tabela 5 - Média de resultados dos testes.....	56
Tabela 6 - Margens de erro dos testes.....	56
Tabela 7 - Resultado dos testes no Internet Explorer em HTML4.....	57
Tabela 8 - Resultados dos testes no Internet Explorer em HTML5.....	58
Tabela 9 - Resultados dos testes no Firefox em HTML4.....	59
Tabela 10 - Resultados dos testes no Firefox em HTML5.....	60
Tabela 11 - Resultados dos testes no Chrome em HTML4.....	61
Tabela 12 - Resultados dos testes no Chrome em HTML5.....	62

LISTA DE FIGURAS

Figura 1 - Tela de login em HTML4.....	32
Figura 2 - Tela de login em HTML5.....	33
Figura 3 - Tela de texto em HTML4.....	34
Figura 4 - Tela de texto em HTML5.....	34
Figura 5 - Tela de vídeo em HTML4.....	35
Figura 6 - Tela de vídeo em HTML5.....	36
Figura 7 - Início de documento em HTML4	37
Figura 8 - Início de documento em HTML5	37
Figura 9 - Áreas de <i>layout</i> em HTML4	38
Figura 10 - Áreas de <i>layout</i> em HTML5	38
Figura 11 - Exemplo de elemento de áudio em HTML4	39
Figura 12 - Exemplo do elemento de áudio no HTML5	39
Figura 13 - Exemplo do elemento de vídeo em HTML4	40
Figura 14 - Exemplo do elemento de vídeo em HTML5	40
Figura 15 - Gráfico do uso de Threads.....	42
Figura 16 - Gráfico do consumo médio de processamento.....	43
Figura 17 - Gráfico de alocação de memória ROM.....	43
Figura 18 - Gráfico da alocação de memória de trabalho	44

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivo	14
1.2	Objetivo Geral	14
1.2.1	Objetivos Específicos	14
1.3	Justificativa	14
1.4	Organização	15
2	MÉTODO	16
3	REFERENCIAL TEÓRICO	17
3.1	Desenvolvimento de Páginas <i>Web</i>	17
3.2	Navegadores	17
3.2.1	<i>Microsoft Internet Explorer</i>	18
3.2.2	<i>Mozilla Firefox</i>	19
3.2.3	<i>Google Chrome</i>	19
3.3	HTML	20
3.4	HTML4 e XHTML	21
3.5	HTML5	23
3.6	<i>Adobe Flash</i>	26
3.7	<i>JavaScript</i>	27
3.8	<i>Cascading Style Sheet</i>	27
3.9	Editores HTML	28
3.9.1	<i>Notepad++</i>	29
3.9.2	<i>Adobe Dreamweaver</i>	29
4	DESENVOLVIMENTO	31
4.1	Páginas de teste	32
5	AVALIAÇÃO	41
5.1	Perspectiva do Sistema	41
5.2	Perspectiva do Desenvolvedor	45
5.2.1	Desenvolvedor 1	45
5.2.2	Desenvolvedor 2	45

5.2.3	Desenvolvedor 3.....	45
5.2.4	Desenvolvedor 4.....	46
5.2.5	Desenvolvedor 5.....	46
5.2.6	Desenvolvedor 6.....	46
5.2.7	Desenvolvedor 7.....	46
5.3	Perspectiva do Usuário.....	47
5.4	Comparativo das três perspectivas.....	48
6	CONSIDERAÇÕES FINAIS	49
6.1	Dificuldades	49
6.2	Conclusão.....	50
6.3	Trabalhos Futuros.....	51
7	REFERÊNCIAS	52
8	ANEXOS	56

1 INTRODUÇÃO

Em 1989, nascia a linguagem de marcação de hipertexto, o HTML (*HyperText Markup Language*), com o objetivo de facilitar a comunicação e a disseminação de pesquisas e trabalhos científicos. Com o crescimento da *Internet* pública e com as padronizações da linguagem, o HTML tornou-se o principal meio de transmissão e visualização das informações via páginas *Web* no mundo (W3C, 2012).

Como linguagem utilizada na criação de documentos que podem ser interpretados por navegadores, é, portanto, o componente principal da *interface* e de todo conteúdo disponível na *Web*, evoluindo com as necessidades dos usuários até sua versão final mais recente, HTML 4, aprovada em 1999.

A versão 4 não dispõe de meios necessários para a incorporação bem sucedida de recursos mais expressivos e controles de multimídia de forma nativa, sendo preciso que desenvolvedores *Web* recorram a outras tecnologias para fornecer essas funcionalidades, como, por exemplo, o *Microsoft Silverlight* e o *Adobe Flash*.

Batista (2010) ressalta a evolução da *Internet*, salientada pela grande migração de aplicações *desktop* para aplicações *Web*, que não pôde esperar pelos avanços da linguagem HTML. Assim, para agregar novas funcionalidades, foi necessário que se usassem ferramentas externas que, apesar de eficientes, trazem várias dificuldades, como a falta de padronizações e não codificação aberta, uma vez que são ferramentas proprietárias.

Objetivando livrar-se da dependência de outras ferramentas para a reprodução de elementos de mídia, melhorar as condições de navegação para os usuários, simplificar os códigos para os programadores e reduzir os custos do desenvolvimento *Web*, surge a linguagem HTML5.

Segundo Jobs (2010), ferramentas de auxílio multimídia como o *Adobe Flash* tem uma série de falhas ou de incompatibilidades que retardam o progresso *Web*, motivos que levaram a maioria dos produtos da marca *Apple* a não suportarem essas tecnologias. Porém, com a chegada da HTML5, tais problemas talvez pudessem ser eliminados.

Com a introdução de um novo conjunto de elementos nativos, que facilitam a estruturação e o desenvolvimento, a HTML5 pretende simplificar a navegabilidade do usuário, proporcionando páginas mais leves, dinâmicas, seguras e com recursos, sem a necessidade de outras ferramentas (SAMMY, 2011).

Assim sendo, propõe-se a realização de um estudo prático para comparar a atual versão da HTML (versão 4) com a nova versão da linguagem HTML (versão 5), de forma a apontar as principais melhorias e a localizar possíveis pontos fracos para as atuais tendências da *Internet*.

1.1 Objetivo

Neste tópico são descritos as metas e objetivos deste trabalho.

1.2 Objetivo Geral

Este trabalho objetiva realizar uma comparação prática entre a linguagem HTML4 e a versão HTML5, a fim de revelar as diferenças percebidas por usuários, desenvolvedores e sistema, em relação às diferentes versões apresentadas.

1.2.1 Objetivos Específicos

- Estudar as principais linguagens e ferramentas utilizadas para o desenvolvimento e a visualização de documento *Web*;
- Analisar as características dessas linguagens e ferramentas, evidenciando os pontos críticos de interação entre elas;
- Levantar as principais características da HTML, versão 5;
- Desenvolver documentos *Web* semelhantes na versão HTML4 e na versão HTML5, a fim de comparar as diferentes implementações;
- Montar uma estrutura comparativa, a fim de apontar as diferenças encontradas entre as linguagens.

1.3 Justificativa

A linguagem HTML é usada para a criação de documentos *Web*. A necessidade de recursos *Web* tem crescido a cada dia, mas a versão atual, HTML4, não dispõe de

meios necessários para a incorporação bem sucedida de recursos mais expressivos e de controles de multimídia.

Diante das dificuldades encontradas em relação à falta de suporte nativo da linguagem HTML4, como, por exemplo, a necessidade de se trabalhar com os diferentes *plugins*, tanto para o desenvolvedor como para o usuário. Mostra-se útil um comparativo prático entre esses diferentes modos de implementação, entre essas duas versões da HTML, mostrando, de forma prática, as diferentes características de implementação, de navegação e de desempenho.

Dessa forma, mostrar-se-á se houve avanços no sentido de suprir as necessidades de navegação do usuário; se houve melhorias, como velocidade de implementação e facilidade de uso para o desenvolvedor; se houve melhoria de velocidade e melhor prática interativa para o usuário; e, ainda, se houve melhorias no desempenho e menor uso de recursos por parte do sistema.

Ao final deste trabalho, espera-se concluir se as melhorias implementadas no HTML5 vieram a tempo de contribuir com o desenvolvimento *Web* de modo satisfatório ou se chegaram de forma tardia a um cenário onde outras soluções já superaram as dificuldades da linguagem.

1.4 Organização

Este trabalho está organizado da seguinte maneira: a seção 2 apresenta o método de elaboração deste trabalho; a seção 3 apresenta o referencial teórico, com embasamento da pesquisa; a seção 4 contém o desenvolvimento das páginas para teste; a seção 5 exhibe as avaliações sobre os três aspectos de desenvolvimento; a seção 6 mostra as considerações finais e as conclusões acerca dos resultados obtidos.

2 MÉTODO

Este trabalho assume caráter descritivo, visando descrever funcionalidades e ferramentas de auxílio de versões diferentes da linguagem HTML. Assim como caráter exploratório e experimental, procedendo a um estudo comparativo entre páginas desenvolvidas em HTML4 e HTML5, testando-as a nível de sistema e entrevistando usuários e desenvolvedores que interagiram com as páginas de teste, salientando as possíveis vantagens de um método sobre outro, usando funcionalidades semelhantes em quesitos multimídia e estrutural.

De modo claro, este trabalho foi elaborado da seguinte maneira:

- Fundamentação teórica:

- Estudo das versões 4 e 5 da HTML.
- Estudo das ferramentas de auxílio e desenvolvimento em HTML.
- Apresentação das diferenças entre as versões 4 e 5 da HTML.

- Desenvolvimento:

- Levantamento das principais variáveis quanto aos usuários, ao sistema e aos desenvolvedores.
- Desenvolvimento das interfaces nas versões 4 e 5 da HTML.

- Avaliação:

- Estabelecimento e aplicação de métricas de sistema para realização de um comparativo entre as versões.
- Utilização das interfaces nas diferentes versões diante da interação de usuários e aplicação de um questionário.
- Apresentação dos códigos aos desenvolvedores, através de uma entrevista e aplicação de um questionário.
- Análise dos testes, evidenciando as principais variáveis neles aplicadas.

- Considerações finais e conclusões.

3 REFERENCIAL TEÓRICO

Neste trabalho, serão abordados conceitos sobre o desenvolvimento *Web* e a linguagem de marcação de hipertexto para *Web*, assim como navegadores e ferramentas de auxílio ao desenvolvimento *Web*.

3.1 Desenvolvimento de Páginas *Web*

A *Web* pode ser definida como um conjunto de documentos acessíveis por meio da *Internet*. Esses documentos, conhecidos por páginas *Web*, contêm uma tecnologia chamada hipertexto, através da qual é possível percorrer partes do documento e alcançar outros documentos através de ligações entre eles, conhecido como *links* (REIS, 2007).

Segundo Moleiro (2010), a *Web* pode ser descrita como um sistema de informação em hipertexto, gráfico, interativo, distribuído, dinâmico, independente de plataforma e globalizado, utilizado na rede mundial de computadores.

Moleiro (2010), ainda define *páginas Web* como elementos específicos de uma apresentação *Web* dentro de uma estrutura; e *apresentação Web* como um conjunto de páginas *Web*, estruturadas sobre um determinado conteúdo. As apresentações *Web* conjuntas, por sua vez, formam um *site*.

Para a criação de uma página *Web* usam-se textos e comandos especiais que fazem parte da HTML. A finalidade da HTML é formatar o texto exibido para criar ligações entre as páginas *Web*, criando assim documentos com o conceito de hipertexto. Para que o conteúdo HTML possa ser formatado e exibido na *Internet*, existem os navegadores, programas que leem o conteúdo do arquivo, interpretam os comandos e exibem sua página *Web* (REIS, 2007).

3.2 Navegadores

Um navegador, também conhecido por *browser*, é um programa de computador que habilita seus usuários a interagirem com documentos de páginas *Web*, que podem ser escritas em linguagens HTML, XHTML ou HTML5, com ou sem linguagens como o CSS, e que estão hospedadas em um servidor *Web* (BUGS, 2012).

Para serem acessadas, as referidas páginas *Web* armazenadas em servidores, necessitam de um programa que faça o papel de cliente pedindo o serviço de transferência de arquivos. Para que haja uma comunicação eficiente entre o cliente e o servidor, é necessário um protocolo de comunicação, a fim de que ambos consigam entendimento mútuo. Para todo tipo de serviço é utilizado um protocolo próprio. Para a requisição de documentos *Web*, o protocolo mais comum é o HTTP. Assim, utilizando o HTTP, o navegador *Web* consegue, a partir de uma URL (*Uniform Resource Locator*), acessar seu conteúdo no servidor, interpretar e traduzir o código HTML, a fim de se obter um documento contendo a informação (FERREIRA *et al.*, 2007).

Para este trabalho foram escolhidos os três navegadores mais usados no mundo, na atualidade (STATCOUNTER, 2013).

3.2.1 Microsoft Internet Explorer

O *Internet Explorer*, também conhecido como IE, é um navegador de licença proprietária, lançado originalmente pela *Microsoft*, em 23 de agosto de 1995. É atualmente o segundo navegador mais usado. Até 2008, foi o navegador mais utilizado no mundo inteiro, uma vez que é distribuído em cada versão do sistema operacional *Windows*. Porém, desde 2004, vinha perdendo espaço para outros navegadores. Em abril de 2005, a porcentagem de usuários do IE era de 85%; em março de 2013 era o segundo navegador mais usado no mundo, com 29% dos usuários, contra 37% do *Google Chrome* e 21% do *Mozilla Firefox*, segundo dados da *StatCounter* (STATCOUNTER, 2013).

Este navegador é um componente integrado nas versões mais recentes do *Microsoft Windows*. Está disponível como produto grátis e separado para as versões mais antigas do sistema operacional. Acompanha o *Windows* desde a versão 95 OSR2.

Ao longo do tempo, o *Internet Explorer* foi apontado como um *software* com numerosas falhas de segurança. Programas maliciosos ou oportunistas exploravam brechas para roubar informações pessoais. *Softwares* maliciosos, como vírus e *trojans*, exploravam falhas do navegador para controlar e roubar informações, ou direcionar os usuários a determinadas páginas *Web* (CROWLEY, 2010).

Crowley (2010), ainda aponta esses problemas como uma das causas para a perda de mercado, e ressalta que a vulnerabilidade encontrada está relacionada ao grande número de usuários, sendo este o fator predominante para que pessoas mal-intencionadas explorassem erros com fins ilícitos. Um dos motivos do *Internet Explorer* ser alvo de tantas críticas está relacionado ao fato de o mesmo ser de código fechado, dificultando a identificação de problemas antes que algum *software* mal-intencionado seja descoberto.

3.2.2 Mozilla Firefox

O navegador *Mozilla Firefox* é um *software* livre e totalmente compatível com inúmeras plataformas para interpretar e exibir páginas da *Web*, desenvolvido pela *Mozilla Foundation* com ajuda de centenas de colaboradores. Cerca de 40% do código do programa foi totalmente escrito por voluntários. A intenção dessa fundação é desenvolver um navegador leve, seguro, intuitivo e altamente extensível (MOZILLA, 2013).

Segundo o *site* de estatísticas para *Internet StatCounter*, em agosto de 2013, 21,11% de todos os usuários da *Internet* utilizavam o *Mozilla Firefox*, fazendo dele o terceiro navegador mais utilizado no mundo, atrás do *Google Chrome*, que teve 38,12% dos usuários, e do *Microsoft Internet Explorer*, que teve 29,43% dos usuários.

A própria *Mozilla Foundation* diz-se orgulhosa do fato de o *Firefox* apresentar alta compatibilidade com os atuais padrões *Web*, especificados pelo W3C, tais como: HTML, XML, XHTML, CSS, *JavaScript*, DOM, *MathML*, XSL e *XPath*. Há também o suporte à transparência variável em arquivos de imagem PNG. O suporte aos padrões *Web* é constantemente melhorado pelos colaboradores do projeto *Mozilla* (MOZILLA, 2013).

3.2.3 Google Chrome

Em 2008, a gigante de *Mountain View* entrou para o mercado de navegadores com seu *Google Chrome*, um navegador gratuito, de código aberto, com *design* simples e integrado. Até sua vigésima sétima versão, usava o motor gráfico *WebKit*, a partir de sua vigésima oitava versão, continuou com o *WebKit* apenas nas versões para IOS. As demais versões utilizam o motor gráfico *Blink*, desenvolvido pela própria *Google*.

Apesar de ser gratuito e de código aberto, o *Chrome* possui componentes privados, como o visualizador de arquivos PDF e suporte integrado ao *Flash* (GOOGLE, 2011).

Com apenas cinco anos de existência, em 2013, o *Chrome* é, com grande vantagem, o navegador mais usado no mundo. Segundo o *StatCounter* (2013), em outubro de 2013, o *Google Chrome* é usado por 39,14% dos internautas do mundo; em segundo lugar, aparece o *Microsoft Internet Explorer* com 28,83% dos usuários; e em terceiro lugar, aparece o *Mozilla Firefox*, com 20,43% dos usuários.

Vantagem ainda mais nítida é percebida no Brasil. O *StatCounter* (2013) alega que 61,27% dos brasileiros usam o *Google Chrome* como seu navegador padrão, enquanto o *Microsoft Internet Explorer* e o *Mozilla Firefox* apresentam 19,36% e 15,99% respectivamente.

Segundo Malenkovich (2013), o *Google Chrome* conta com excelentes recursos de segurança, como bloqueio de janelas *pop-up*; opções para impedir que *sites* rastreiem sua localização física; possibilidade nativa ao bloqueio do uso de câmera e microfone permanente, ou que requisitem permissão ao usuário; proteção contra *phising* e aviso de página contendo *softwares* mal-intencionados; além da possibilidade do usuário instalar extensões, aplicativos e *plugins* que integrem a segurança do navegador. Apesar dessas funcionalidades, Malenkovich (2013), cita o perigo de qualquer pessoa mal-intencionada poder ter acesso a todas as senhas salvas no navegador de maneira bastante simples, desde que tenha acesso ao navegador *in loquo*.

3.3 HTML

A HTML foi criada pelo físico, Tim Berners-Lee, no início da década de 90, para ser um meio de comunicação e disseminação de trabalho junto com o HTTP (*HyperText Transfer Protocol*). A linguagem permitia que textos e imagens fossem estruturados em um documento de texto através da utilização de *tags* (etiquetas), que são estruturas de instruções breves escritas entre sinais de menor e maior. Por exemplo: " Oi" para que a palavra *Oi* apareça em um navegador em negrito.

Com o passar do tempo, a *Internet* foi expandindo sua função e o HTML não acompanhou essas mudanças. Entre a primeira versão e a versão 3.0, não houve

grandes alterações por causa da grande dificuldade de se alterar tudo o que já havia sido publicado, e, até aquele momento, pouco se via de multimídia na *Internet*.

O W3C (*World Wide Web Consortium*), consórcio formado por instituições comerciais, educacionais, governamentais e empresariais, que objetivam definir padrões para a criação de documentos *Web*, convencionou em 97 a versão 4.0 do HTML, que teve como maior diferença a inserção de objetos dinâmicos, tornando possível a alteração de elementos de páginas HTML sem a necessidade de um novo documento *Web* (W3C, 2013).

Em 2000, o W3C propõe a união do HTML com o XML (*eXtensible Markup Language*), uma recomendação do W3C para padronização de documentos, criando assim o chamado XHTML. Ainda assim, essa padronização não ajudava desenvolvedores em questões de multimídia, surgindo, então, a necessidade de agregar ferramentas para auxílio audiovisual (W3C, 2013).

Essas ferramentas, chamadas *plugins*, aumentam as capacidades ou funções, solucionando a falta nativa de suporte multimídia. Contudo, existiam algumas desvantagens, como, por exemplo, não seguir um padrão único, ser proprietário e não possuir código aberto, limitando o avanço do desenvolvimento *Web* (BATISTA, 2010).

Para melhorar a situação, novas empresas uniram-se ao W3C e deu início ao projeto do HTML, versão 5, que tinha como objetivo resolver várias dessas questões, como segurança, recursos multimídia, portabilidade, desempenho com novas *tags* e nova estruturação do documento *Web*, mantendo a retrocompatibilidade de páginas antigas.

3.4 HTML4 e XHTML

Em sua quarta versão, o código HTML acrescenta em suas características nativas o suporte a estruturas de folhas de estilo, *scripts*, objetos integrados, direcionamento de texto, bem como tabelas mais completas e melhorias nos formulários, resultando na possibilidade de criação de documentos com maior acessibilidade (PEDROSO *et al.*, 2007).

Contudo, essa versão ainda não trazia diferencial real para a semântica do código, e também não facilitava a manipulação dos elementos escritos nas linguagens

JavaScript e *CSS*, o que resultava em diferentes problemas por parte da interpretação dos navegadores (W3C, 2013).

Uma alternativa para resolver as dificuldades foi a criação do XHTML, que unia os padrões do XML ao código HTML, facilitando muito para os desenvolvedores, acostumados com as características do HTML, mas apresentando a flexibilidade do padrão XML de interpretação multiplataforma, permitindo aos elementos do XML serem inseridos no conteúdo HTML já existente, além da facilidade de atualização do documento *Web* e melhoria da acessibilidade (MENDES 2009).

Mendes (2009), ainda ressalta a vantagem da portabilidade, já que os elementos contidos no padrão XML são capazes de adaptar o conteúdo exibido dentro de uma codificação que cada plataforma consiga compreender. As principais mudanças entre o HTML4 e o XHTML podem ser vistas no seguinte exemplo:

Tabela 1 – Mudanças entre HTML4 e XHTML

CARACTERÍSTICAS	HTML4	XHTML
ANINHAMENTO	<code><i>Teste</i></code>	<code><i>Teste</i></code>
BEM-FORMADOS	<code><html> teste </html></code>	<code><html> <head><title>ola</title> </head> <body> teste </body> </html></code>
FECHAMENTOS	<code><html><body> <p>oi <p>
 <hr> </body> </html></code>	<code><html><body> <p>oi</p> <p>
 <hr /> </p> </body></html></code>

Os elementos XHTML devem estar sempre aninhados e bem formados, como é mostrado na tabela 1. Os nomes das *tags* devem estar em letras minúsculas e os atributos em letras maiúsculas. Os valores dos atributos devem estar entre aspas, e a minimização de atributos é proibida (W3C, 2013).

3.5 HTML5

O intuito do HTML5 é simplificar a navegabilidade do usuário, ao proporcionar páginas mais leves, dinâmicas e seguras. Seu objetivo, de modo claro é trazer maior compatibilidade e fluidez em tarefas como exibição de vídeos, uso *offline* de aplicações *Web*, elaboração de gráficos e formulários interativos, indexação mais eficiente em mecanismos de busca e facilidade de entendimento e manutenção dos códigos por parte dos desenvolvedores, sem prendê-los a *plugins* que, muitas vezes, trazem maiores dificuldades (HEY, 2010).

As principais mudanças da quinta versão, como cita Hey (2010), estão ligadas diretamente à necessidade de suporte, independente dos novos formatos de conteúdo multimídia, das novas funcionalidades semânticas e da acessibilidade.

Um novo elemento, o *Canvas*, foi incluído para possibilitar o desenho de gráficos em um documento *Web*, função até então permitida apenas com o uso de *plugins*.

Também foram criados elementos para integrar de forma nativa o uso de áudio e vídeo, e dois novos elementos para armazenamento de dados, um para uma sessão ativa e outro sem limite de tempo (W3C, 2013).

Muitos elementos da quarta versão da HTML foram excluídos ou substituídos, por não serem realmente necessários ou por serem usados de forma incorreta, trazendo uma nova gama de elementos para proporcionar uma melhor estruturação e conteúdo multimídia (W3C, 2013).

Estes elementos são mostrados na tabela 2 a seguir.

Tabela 2 - Novas *Tags* de Estrutura HTML5

(Continua)

TAG	DESCRIÇÃO
<article>	Especifica um artigo qualquer.
<aside>	Relaciona um conteúdo ao redor.
<bdi>	Para texto que não esteja diretamente vinculado ao elemento pai.
<command>	Um botão, <i>radioButton</i> ou <i>checkBox</i> .
<summary>	Uma legenda ou resumo dentro do elemento detalhe.

Tabela 2 - Novas Tags de estrutura HTML5

(Conclusão)

TAGS	DESCRIÇÃO
<figure>	Para agrupamento de uma sessão de conteúdo <i>stand-alone</i> .
<figcaption>	Legenda para um elemento figura.
<footer>	Para um rodapé de um documento ou sessão.
<header>	Para uma introdução de um documento ou sessão.
<hgroup>	Para uma sessão de cabeçalhos usando h1 para h6.
<mark>	Para textos que são destaques.
<meter>	Para uma medição. Deve-se conhecer o valor máximo e mínimo.
<nav>	Para uma sessão de navegação.
<progress>	Para mostrar o progresso de uma execução.
<section>	Para uma sessão de um documento como: capítulos, cabeçalhos, rodapés, etc.
<time>	Para definir um tempo e/ou data.
<wbr>	Quebra de linha.

Percebe-se na tabela 2, que as mudanças no código são ligadas à semântica das *tags* e voltadas para melhor interpretação do código, além de uma melhor estruturação.

Houve também a inclusão de novos controles para formulários, algo que a quarta versão da HTML quase não abordava, fazendo com que os desenvolvedores recorressem à *JavaScript*.

Os novos controles são exibidos na tabela 3.

Tabela 3 - Novas *Tags* de Controle de Formulário HTML5

TAG	DESCRIÇÃO
<datalist>	Uma lista de opções de entrada.
<keygen>	Gera chaves para autenticação de usuários.
<output>	Para diferentes tipos de saídas, como as geradas por <i>scripts</i> .

O elemento *input* ganhou novos valores para o atributo *type*, garantindo maior controle sobre a entrada de dados do usuário, além do suporte total ao CSS3 (*Cascading Style Sheets*), o que proporciona maior liberdade criativa aos desenvolvedores, permitindo personalização de estilos, efeitos e recursos de estética da interface, tornando-as muito mais interessantes ao usuário.

Os novos tipos de entrada podem ser vistos na tabela 4.

Tabela 4 - Novos tipos de entrada do usuário HTML5

TYPE	DESCRIÇÃO
Tel	A entrada é um número de telefone.
url	A entrada é uma URL.
Email	A entrada é um ou mais endereço de email.
Datetime	A entrada é uma data e/ou hora.
Date	A entrada é uma data
Month	A entrada é um mês.
Week	A entrada é um dia da semana.
Time	A entrada é uma hora.
datetime-local	A entrada é uma data e hora local.
Number	A entrada é um número.
Range	A entrada é uma faixa de valores.
Color	A entrada é uma cor em hexadecimal como #FF00FF.

3.6 Adobe Flash

O *Flash* foi criado pela empresa *Macromedia*, no começo dos anos 90, inicialmente para gráficos vetoriais. Seu desenvolvimento tornou-o uma ferramenta interativa de criações multimídia, agregou ao longo do tempo características de programação, como estruturas repetidoras e condicionais, fez do *Adobe Flash* uma das principais ferramentas de controle e criação de recursos audiovisuais na *Internet* (KARLINS, 2010).

A ferramenta *Flash* evoluiu bastante. De uma ferramenta simples de desenho e animações para a *Internet* a um ambiente completo de desenvolvimento multimídia com mais de quinhentos mil produtores de conteúdos e mais de 325 milhões de utilizadores do *Adobe Flash Player*. É até possível que, hoje em dia, o *Adobe Flash Player* seja o *software* mais distribuído através da *Internet* (GAY, 2013).

Muller (2011), dá crédito ao poder de processamento multimídia e à importante vantagem de gerar normalmente arquivos de tamanho pequeno, os principais motivos para que o *Flash* tenha sido tão difundido e utilizado por desenvolvedores para criação de jogos, *banners* e animações que podem interagir ou não com o usuário, entre tantas outras possibilidades multimídia. Muller (2011), resalta ainda que a ferramenta é baseada em imagens vetoriais, permitindo que todos os recursos multimídia sejam apresentados nos mais diferentes navegadores de forma bastante similar. Como as imagens são geradas através de fórmulas matemáticas, e não por combinações de pontos, elas garantem a vantagem que as imagens não sofrerão deformações caso sejam redimensionadas.

Porém, essa ferramenta também pode resultar em algumas complicações, além das gerais de um *framework* privado, como cita Jobs (2010). A maioria dos vídeos em *flash* ainda usam decodificadores antigos, resultando em maior consumo de recursos do sistema. O *Flash* também já obteve os piores resultados de segurança, causando travamentos e outros problemas ao sistema. Outro problema apontado é a tecnologia *Flash* ter sido criada na era dos computadores e muitos recursos utilizam-se do *mouse*, acessório que não está presente na crescente tendência de aparelhos com suporte a toque, embasando assim o fato de nenhum dispositivo móvel da *Apple* ter suporte ao *Flash*.

3.7 JavaScript

A *JavaScript* foi criada inicialmente com o nome de *LiveScript*, nome oficial da linguagem em seu lançamento na versão beta do navegador *Netscape 2.0*, em setembro de 1995. Porém, teve seu nome mudado em um anúncio conjunto com a *Sun Microsystems*, em dezembro de 1995, quando foi implementado no navegador *Netscape* versão 2.0B3 (FLANAGAN, 2011).

A mudança de nome de *LiveScript* para *JavaScript* deu-se na mesma época em que a *Netscape* adicionou suporte à tecnologia *Java* em seu navegador, resultando na impressão equivocada de que a linguagem foi baseada em *Java*. Mas, na realidade, foi uma estratégia de *marketing* da *Netscape* para aproveitar a popularidade da recém-lançada linguagem *Java* (FLANAGAN, 2011).

Seu objetivo original era ser implementado como parte dos navegadores *Web* para que *scripts* pudessem ser executados do lado do cliente e interagissem com o usuário, sem a necessidade de este script passar pelo servidor, controlando o navegador, realizando comunicação assíncrona e alterando o conteúdo do documento exibido.

Concebida para ser uma linguagem *script* com orientação a objetos baseada em protótipos, tipagem fraca e dinâmica, possui suporte à programação funcional e apresenta recursos como fechamentos e funções de alta ordem comumente indisponíveis em linguagens populares como C++ (ECMA, 2011).

A linguagem *JavaScript* tem como características o suporte a elementos de sintaxe de programação estruturada e a diferenciação entre expressões e comandos, tipagem dinâmica das variáveis, tendo por base objetos que podem ter suas propriedades e valores adicionados, editados ou apagados em tempo de execução.

Os objetos que possuem propriedades e métodos podem ser passados como argumentos, serem atribuídos a variáveis ou retornados como qualquer outro objeto. Conta também com funções aninhadas para fechamentos (JAVASCRIPT, 2012).

3.8 Cascading Style Sheet

Em 1996, o W3C lançava oficialmente a recomendação de separação de folha de estilo e formatação do código HTML padrão, o CSS, no intuito de separar a

formatação da informação do resto do código HTML, evitando problemas de interpretação dos navegadores e facilitando a compreensão do código por parte dos desenvolvedores (BARROS e SANTOS, 2008).

O CSS formata a informação entregue pelo HTML, não importando o formato de mídia que ela se encontre, seja em imagem, texto, vídeo, áudio ou qualquer outro elemento criado. Essa formatação, na maioria das vezes, é visual, mas não necessariamente, e prepara essa informação para que ela seja consumida da melhor maneira possível (W3C, 2013).

Em sua terceira versão, está presente uma nova característica, que se baseia na modularização da linguagem, na qual cada módulo tem atualização independente dos outros, tornando a melhora do código contínua e mais rápida. Outra vantagem da modularização é a capacidade de dispositivos específicos terem a possibilidade de escolher quais módulos suportam, assim focando recursos em funções mais importantes ao sistema (EIS e FERREIRA, 2012).

3.9 Editores HTML

Um editor HTML é um *software* para a criação de páginas. Mesmo que a edição em HTML de uma página *Web* possa ser feita com qualquer editor de texto, editores HTML específicos oferecem vários recursos próprios para auxiliar na criação de páginas *Web*, além de outras funcionalidades. Editores são ótimas ferramentas para profissionais de desenvolvimento para a *Internet*, visto que lhes poupam trabalhos pequenos, permitindo assim manter o foco nos detalhes mais importantes.

Existem, além dos editores de texto comuns, que também podem ser utilizados, como, por exemplo, o *Microsoft Notepad*, editores próprios para páginas *Web* chamados *WYSIWYG* (O que você vê é o que você tem, em livre tradução), que permitem a visualização em tempo real do que se está editando ou desenvolvendo (GOOGLE, 2013).

As vantagens de se usar um *WYSIWYG* são a melhora no tempo de implementação e a comodidade de não ter que conhecer o código a fundo, ideal para desenvolvedores ainda sem muita experiência.

No entanto, existe a desvantagem de poder gerar *lixo de código*, trechos de código inseridos desnecessariamente no documento, que não são visíveis no momento da criação do documento *Web*, o que pode dificultar a compreensão do código por parte dos desenvolvedores e consumir maiores recursos do sistema desnecessariamente (TECMUNDO, 2013).

3.9.1 **Notepad++**

O *Notepad++* é um *software* livre para edição de texto e de código fonte, que suporta várias linguagens de programação como *C*, *C++*, *C#*, *Java*, *XML*, *HTML*, *PHP*, *JavaScript*, *ASCII art*, *Doxygen*, *ASP*, *VB/VBScript*, *Unix Shell Script*, *BAT*, *SQL*, *CSS*, *Pascal*, *Perl*, *Python*, *Assembly*, *Ruby*, *Lisp*, *Scheme*, *PostScript*, *VHDL*, entre outras.

Este editor é baseado no componente *Scintilla*, uma biblioteca livre com ênfase em recursos avançados na edição de códigos; é escrito em *C++*, utilizando a *API Win32* e usa a *STL*, que asseguram a mais alta velocidade de execução e o menor tamanho para o *software*. É seu objetivo oferecer uma ferramenta leve e eficiente com uma interface gráfica totalmente modificável.

Além disso, usuários podem definir suas próprias linguagens, usando um *sistema de definição de linguagem* integrado, que faz do *Notepad++* extensível, para ter realce de sintaxe e compactação de trechos de código.

O *Notepad++* possui suporte à função autocompletar, busca e substituição de expressões regulares; divisão de tela; *zoom*; suporte para macros e *plugins*, entre outros.

Em consequência da interface simplificada e dos diversos recursos de manipulação de arquivos que o programa possui, serve tanto para usuários leigos, como para usuários avançados (*NOTEPAD++ 2013*).

3.9.2 **Adobe Dreamweaver**

O *Adobe Dreamweaver* é um *software* fechado de licença proprietária, de desenvolvimento voltado para a *Web*. Em suas versões iniciais, servia apenas como um simples editor *HTML* com característica *WYSIWYG*. Mas, em suas versões posteriores, incorporou um notável suporte para várias tecnologias *Web*, tais

como: XHTML, CSS, JavaScript, Ajax, PHP, ASP, ASP, Dot.NET, JSP, ColdFusion e outras linguagens para o servidor (FURLAN, 2010).

Enquanto o desenvolvedor cria uma página, é possível visualizar somente o código, com a facilidade das *tags* poderem ser adicionadas através de menus. Também é possível uma visualização por *design*, que torna a ferramenta WYSIWYG, e mostra na tela diretamente como está a sua página em tempo real. Ainda é possível uma visualização em tela dividida, que apresenta tanto o código quanto o *design*, para que o desenvolvedor construa algo no *design* e veja como está sendo criado o código, e vice-versa, este recurso de visualização é bastante usado para se evitar um dos problemas mais citados com o *Dreamweaver*, os chamados *códigos sujos*, trechos de código desnecessários que podem tornar a aplicação mais dispendiosa ao sistema e atrapalhar a leitura por parte do desenvolvedor e, por vezes, do navegador (ADOBE, 2013).

Outra funcionalidade do *Dreamweaver* é possibilitar uma previsão da visualização da saída HTML da página diretamente no navegador de destino selecionado. O *software* possui também algumas ferramentas de gerenciamento e transferência de projetos, como a capacidade de encontrar e substituir, no projeto inteiro, linhas de texto ou código através de parâmetros especificados. O painel de comportamentos também permite a criação de JavaScript básico, sem qualquer conhecimento de codificação (ADOBE, 2013).

4 DESENVOLVIMENTO

Tendo em vista que, para o desenvolvimento deste trabalho, foram estudadas as melhores condições para a realização do comparativo prático e analisados a fundo os resultados obtidos no referido estudo, chegou-se à conclusão de que um modo de se realizar tais testes seria criar duas páginas *Web* similares, uma usando a linguagem HTML4 e ferramentas de auxílio como o *Adobe Flash* e *JavaScript*, e outra desenvolvida em HTML5, tentando substituir as ferramentas de auxílio da quarta versão do HTML pelas novas implementações da quinta versão.

A ferramenta *Adobe Flash* foi escolhida por ser a ferramenta proprietária que mais relevância apresenta com as atualizações do HTML. Muitos desenvolvedores e estudiosos do desenvolvimento *Web* indicam que as mudanças de suporte nativo do HTML5 resultariam no abandono da ferramenta *Adobe Flash*. Por outro lado, a própria empresa *Adobe* e outros profissionais relacionados alegam que nem o HTML5 pode substituir todas as funcionalidades do *Adobe Flash* nem essa ferramenta deixará de ser usada por desenvolvedores já habituados a ela.

Independente a estas análises, é indiscutível a força que a ferramenta *Adobe Flash* tem no cenário de desenvolvimento *Web* há anos. Segundo a própria *Adobe*, mais de um bilhão de computadores usam o *Adobe Flash* em seus navegadores e sistemas operacionais todos os dias; mais de quinhentos milhões de dispositivos móveis têm suporte à ferramenta; mais de vinte mil aplicativos em mercados móveis são feitos em *Flash*; vinte e quatro dos vinte e cinco jogos com mais usuários no *Facebook* são criados em *Flash*; e mais de três milhões de desenvolvedores trabalham com o *Adobe Flash* para produzir conteúdo *Web*.

Esses números refletem uma ferramenta que, no ano de 2011, estava presente em 99% dos computadores conectados a *Internet*. Devido a sua grande importância e participação na *Internet*, muitos especialistas acreditam que, se a ferramenta fosse de código aberto, haveria melhorias e correções mais rapidamente, pois contaria com o apoio de milhares de desenvolvedores ao redor do mundo.

Para tentar suprir tal necessidade, surge o HTML5, que atribui mudanças em sua estrutura e funcionalidade de bibliotecas *JavaScript*, de forma nativa e embutida,

embasando assim a escolha de se usar essa linguagem de auxílio nos testes realizados na tentativa de igualar os códigos das páginas *Web* sob tais perspectivas.

Para o desenvolvimento dos referidos documentos *Web*, foi utilizado o editor de HTML *Adobe Dreamweaver*, pelo potencial que apresenta em melhorar o desempenho de desenvolvedores com pouca experiência e ser comumente usado tanto pelos que estão começando, como pelos mais experientes, que sabem lidar com as possíveis desvantagens dessa ferramenta.

4.1 Páginas de teste

Na perspectiva visual, poucas são as diferenças entre as implementações, como pode ser notado nas imagens a seguir.

Figura 1 - Tela de login em HTML4

Apresentando
TCC
Site Teste

Site em HTML 4

LOGIN

Login

Senha

Ok

Av. xxx yyy zzz, No 666
Bandeirantes - PR - CEP: 86360-000
Email: fabioTcc@uenp.com.br

Fonte: Autoria própria.

Figura 2 - Tela de login em HTML5

Apresentando
TCC
Site Teste

Site em HTML 5

LOGIN

Login

Senha

Ok

Av. xxx yyy zzz, No 666
Bandeirantes - PR - CEP: 86360-000
Email: fabioTcc@uenp.com.br

Fonte: Autoria própria.

A tela de *login*, como pode ser vista nas figuras 1 e 2, apresentam um logo da página *Web* no canto superior esquerdo. O *layout* consiste em cabeçalho e rodapé, além de três colunas, modelo bastante comum em páginas *Web*, com uma coluna abrigando o *menu* do *site*; outra, a informação; e uma terceira que, geralmente, mostra os *sites* parceiros, os campos de busca, entre outras funções.

A validação dos campos *login* e senha é feita através de um trecho em *JavaScript*, onde ocorre a confirmação de que os campos estão devidamente preenchidos com os valores *admin* e *admin*. Em resposta negativa a essa condição, é exibido uma janela *pop-up* informando que algum dos campos foi preenchido incorretamente. Ao dar entrada corretamente, o usuário é direcionado à página de texto e áudio, como é apresentado a seguir.

Figura 3 - Tela de texto em HTML4

Apresentando
TCC
Site Teste

Site em HTML 4

[Principal](#)

[Video](#)

[Produtos](#)

[Serviços](#)

[Contato](#)

[A Tout le Monde](#)

Por que a TI é a profissão do futuro

Nesse momento de pós-industrialização, a informação vale mais do que dinheiro. Por exemplo, se tivermos a informação que no próximo ano haverá uma estiagem na Califórnia, poderemos plantar mais soja e milho no Brasil, pois os preços irão subir. Invisto em ações de empresas que produzem grãos e ganharei muito dinheiro.

A informação vale hoje muito mais do que ouro e petróleo, pois com ela consigo achar o ouro e o petróleo. E o profissional de tecnologia da informação é a pessoa que sabe transformar dados aparentemente sem consistência em informação útil e precisa.

**Henrique Gerstner é Diretor de Educação do Grupo QI*

Por que a TI é a profissão do futuro

[Leia o artigo completo](#)

Conheça as 5 profissões do futuro

[Olhar Digital](#)

Av. xxx yyy zzz, No 666
Bandeirantes - PR - CEP: 86360-000
Email: fabioTcc@uenp.com.br

Fonte: Autoria própria.

Figura 4 - Tela de texto em HTML5

Apresentando
TCC
Site Teste

Site em HTML 5

[Principal](#)

[Video](#)

[Produtos](#)

[Serviços](#)

[Contato](#)

0:11

Por que a TI é a profissão do futuro

Nesse momento de pós-industrialização, a informação vale mais do que dinheiro. Por exemplo, se tivermos a informação que no próximo ano haverá uma estiagem na Califórnia, poderemos plantar mais soja e milho no Brasil, pois os preços irão subir. Invisto em ações de empresas que produzem grãos e ganharei muito dinheiro.

A informação vale hoje muito mais do que ouro e petróleo, pois com ela consigo achar o ouro e o petróleo. E o profissional de tecnologia da informação é a pessoa que sabe transformar dados aparentemente sem consistência em informação útil e precisa.

**Henrique Gerstner é Diretor de Educação do Grupo QI*

Por que a TI é a profissão do futuro

[Leia o artigo completo](#)

Conheça as 5 profissões do futuro

[Olhar Digital](#)

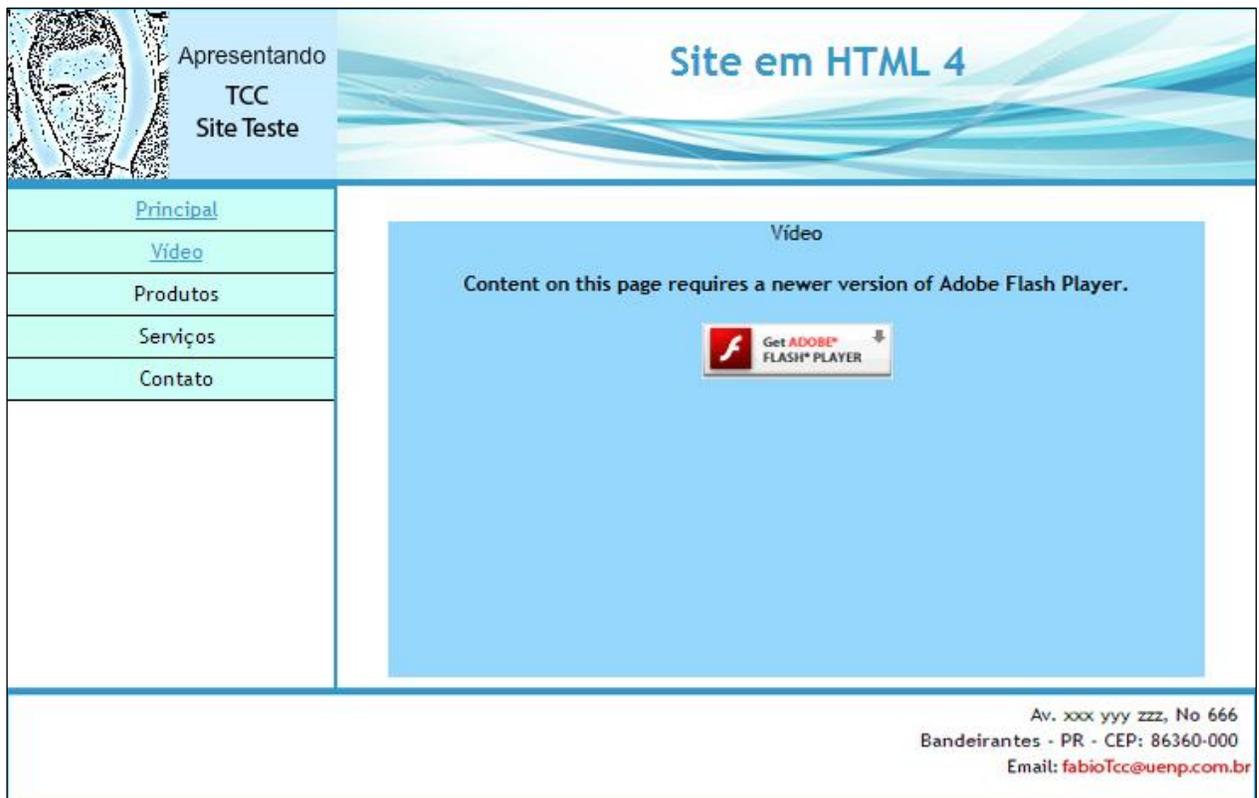
Av. xxx yyy zzz, No 666
Bandeirantes - PR - CEP: 86360-000
Email: fabioTcc@uenp.com.br

Fonte: Autoria própria.

Nas figuras 3 e 4, são exibidas as telas onde se encontram um texto na coluna central e *links* nas colunas laterais. Como citado anteriormente, na coluna à esquerda localiza-se o *menu* do *site*, com links para outras páginas, também em *localhost*. Importante perceber que na figura 3, o último *link* do *menu* é, na verdade, um *link* para um arquivo de áudio que, no HTML4, abre outra janela, exibindo apenas o controle de mídia. Em contra partida na figura 4, na HTML5, os controles são exibidos na mesma janela, tornando melhor a experiência do usuário ao acessar a página *Web*. Na coluna à direita, temos resumos de notícias e *links* para *sites* externos.

A segunda opção no menu à esquerda leva a próxima página onde é exibido um vídeo, como é visto a diante.

Figura 5 - Tela de vídeo em HTML4



Fonte: Autoria própria.

Figura 6 - Tela de vídeo em HTML5



Fonte: Autoria própria.

É interessante notar na figura 5 que, em HTML4, para se utilizar o recurso de vídeo nos navegadores *Microsoft Internet Explorer* e *Mozilla Firefox*, o usuário deve obter uma versão do *plugin Adobe Flash*, suporte que poucos navegadores oferecem de modo nativo, como é o caso do navegador *Google Chrome*.

Por outro lado na figura 6, a versão desenvolvida em HTML5, os navegadores *Microsoft Internet Explorer*, apenas emite uma notificação requisitando ao usuário permissão para exibir conteúdo multimídia, mesmo que o vídeo esteja em *localhost*.

Já os navegadores *Mozilla Firefox* e *Google Chrome* exibem o vídeo e seus controles perfeitamente sem necessidade de notificações ou instalações.

Agora quanto aos códigos das páginas HTML, têm-se muitas partes iguais como questões de *design* e *layout*, porém podem-se observar maiores diferenças entre as duas versões principalmente na tela de vídeo, como é mostrado em seguida.

Figura 7 - Início de documento em HTML4

```

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=ISO-8859-1" />
5 <title>Site para apresentação TCC Fabio Marcel Esperandio</title>

```

Fonte: Autoria própria

Figura 8 - Início de documento em HTML5

```

1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <meta charset="utf-8"/>
5 <title>Site para apresentação TCC Fabio Marcel Esperandio</title>

```

Fonte: Autoria própria.

Logo de início, já é possível observar, nas figuras 7 e 8, a diferença na declaração do documento HTML. Na figura 8, a versão cinco da HTML, é muito mais fácil e prático declarar o documento, pois não há necessidade de se decorar as URLs do W3C, fato que, além de ineficiente, ainda poderia causar erros ao documento. O *doctype* indica para o navegador qual especificação de código deve ser utilizada. Nas versões anteriores à quinta, era necessário declarar a definição de tipo de documento diretamente no código do *doctype*, como visto na figura 7. Com a HTML5, a declaração de definição de tipo de documento a ser utilizado é responsabilidade do navegador.

Outra simplificação foi na declaração do conjunto de caracteres. Na nova versão, basta declarar qual codificação será utilizada.

Figura 9 - Áreas de *layout* em HTML4

```

<div id="direita">
  <div class="secao_direita">
  </div>

</div>
<div style="clear: both;"></div>
</div>

```

Fonte: Autoria própria.

Figura 10 - Áreas de *layout* em HTML5

```

<aside id="direita">

</aside>
<div style="clear: both;"></div>
</article>
<footer id="rodape">

```

Fonte: Autoria própria.

As figuras 9 e 10 mostram como a quinta versão do HTML deixa o código muito mais organizado. Com *tags* como *footer*, *article* e *aside* torna-se muito mais fácil a leitura do código por outros desenvolvedores, faz com que a estruturação de uma página Web seja quase autoexplicativa, diminuindo muito o uso da tag *div*.

Na implementação em HTML4, como é mostrado na figura 9, o uso da *tag div* é extremamente necessário para segmentar o *layout* do documento Web, não havendo outra opção nativa para a estruturação e organização da informação exibida.

Com isso mostra-se a relevância da nova organização como exemplificado na figura 10, as *tags* mostradas e as outras adicionadas ao código HTML5 fazem do desenvolvimento algo até intuitivo para alguns desenvolvedores, pelo nome das novas *tags*.

Figura 11 - Exemplo de elemento de áudio em HTML4

```

153 <div class="itemMenu"><a href="html 4.html"> Principal</a></div>
154 <div class="itemMenu"><a href="video.html">Video</a></div>
155 <div class="itemMenu">Produtos</div>
156 <div class="itemMenu">Serviços</div>
157 <div class="itemMenu">Contato</div>
158 <div class="itemMenu">
159 <a href="D:\Fm\Music\Megadeth\A Tout le Monde.mp3">A Tout le Monde </a>
160 </div>

```

Fonte: Autoria própria.

Figura 12 - Exemplo do elemento de áudio no HTML5

```

<audio autoplay="true" controls src="D:\Fm\Music\Megadeth\A Tout le Monde.mp3">
  <source src="D:\Fm\Music\Megadeth\A Tout le Monde.mp3" type="audio/mp3"/>
</center>
</audio>

</div>
</div>
<div id="miolo">

```

Fonte: Autoria própria.

Nas figuras 11 e 12, é mostrado o trecho em que é inserida uma referência a um arquivo de áudio. Além da mudança no código, também foi alterada a experiência do usuário de maneira a ser mais agradável.

Observa-se na figura 11, a HTML4 usa a *tag* de *hiperlink* de referência, causando assim a abertura de uma nova janela, apenas com o arquivo de áudio e seus controles.

Já na figura 12, nota-se na HTML5 uma *tag* específica para isso, resultando na exibição dos controles na mesma janela, além de alguns recursos como reprodução automática e modo de repetição de mídia, evidenciando o maior controle do desenvolvedor sobre seus recursos multimídia na quinta versão da HTML.

Na implementação em HTML4 não existe modo de controle de mídia nativo pré-definido, o desenvolvedor deve optar por usar uma ferramenta auxiliar que ajude na criação destes controles, ou implementá-los no modo de funções em linhas de código.

Figura 13 - Exemplo do elemento de vídeo em HTML4

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" width="480" height="310" id="FlashID2" title="video">
  <param name="movie" value="flash/Untitled-2.swf" />
  <param name="quality" value="high" />
  <param name="wmode" value="opaque" />
  <param name="swfversion" value="6.0.65.0" />
  <param name="expressinstall" value="Scripts/expressInstall.swf" />
  <object type="application/x-shockwave-flash" data="flash/Untitled-2.swf"
    width="480" height="310">
```

Fonte: Autoria própria.

Figura 14 - Exemplo do elemento de vídeo em HTML5

```
192 <video controls>
193   <source src="RedeTV News Profissoes saiba mais sobre tecnologia da
194     informacao.mp4" type="video/mp4">
195   Seu navegador não suporta o elemento <code>video</code>.
196 </video>
```

Fonte: Autoria própria.

Nos trechos de código mostrados nas figuras 13 e 14, a incorporação do vídeo, torna-se gritante a diferença entre as implementações em HTML4 e HTML5.

A figura 13 mostra 9 linhas de código, necessárias em HTML4, para uma incorporação simples em *Flash* de um arquivo de vídeo, definindo características do reprodutor de vídeo e referenciando o arquivo.

Por outro lado, a figura 14, mostra 4 linhas de código, necessárias em HTML5, para se executar a mesma funcionalidade, além de incluir um aviso caso o navegador não tenha suporte ao determinado tipo de arquivo de vídeo.

5 AVALIAÇÃO

A avaliação de linguagens e modos de implementação foi realizada de modo a cobrir três diferentes perspectivas.

Perspectiva de Sistema, para avaliar se, ao sistema, é mais dispendioso alguma das versões testadas, em relação a recursos como uso de processador, de memória RAM e HDD.

Perspectiva do Desenvolvedor, para conhecer as opiniões e expectativas dos desenvolvedores em torno das mudanças trazidas pela HTML5.

Perspectiva do Usuário Final, para descobrir quais as diferenças percebidas pelas pessoas comuns, em geral o público consumidor.

5.1 Perspectiva do Sistema

Para a análise do consumo de recursos de sistema, foram executadas trinta e cinco execuções de rotinas das páginas *Web* criadas para teste, em *localhost*, como é consenso em testes deste tipo entre os engenheiros e desenvolvedores.

Foram usados os navegadores *Microsoft Internet Explorer* em sua décima primeira versão; *Mozilla Firefox* em sua vigésima quinta versão; e *Google Chrome* em sua trigésima primeira versão.

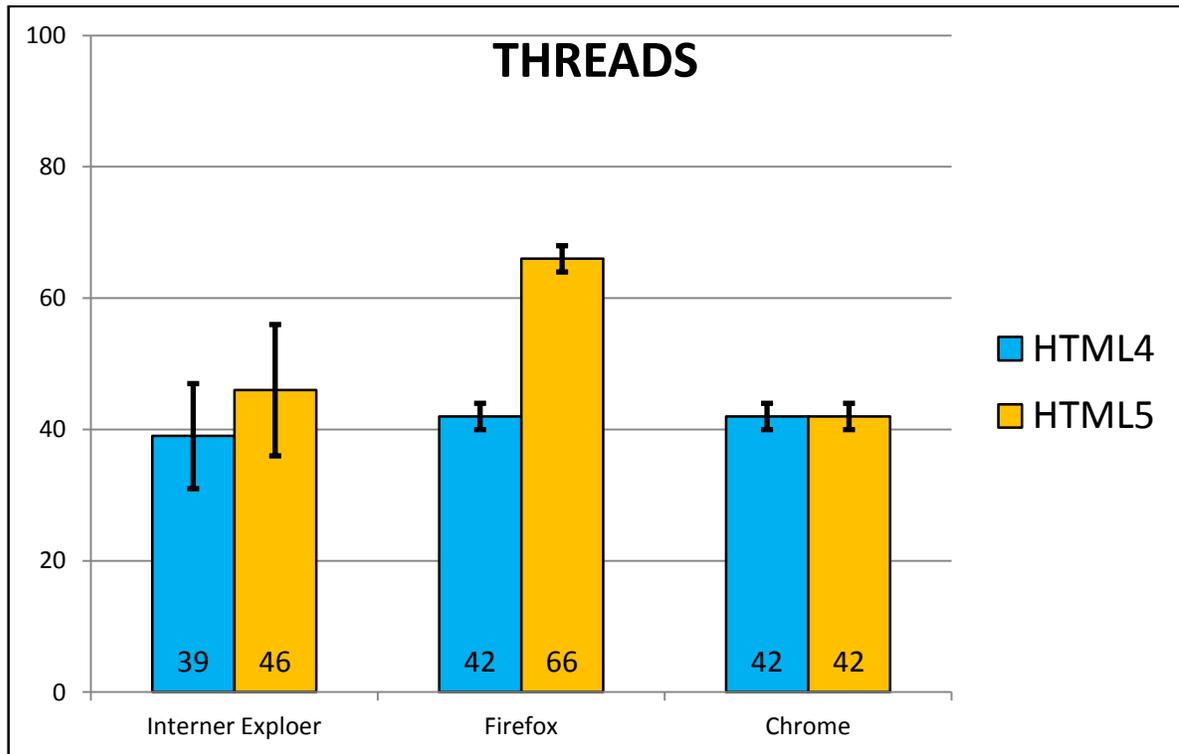
Foi usada a configuração de processador *Core i7-2670QM*, 8 Gb de memória RAM e HDD de 750 Gb.

O monitoramento de recursos foi registrado pelo monitor de recursos nativo do sistema operacional *Microsoft Windows 7*, registrando os resultados obtidos a cada interação e compilando os dados na forma de gráficos, mostrados a seguir.

O uso dos recursos de sistema foi analisado da seguinte forma: foi criada uma rotina que consistia em abrir a página *Web* de *login*, entrar com os valores corretos, passar para página de texto, em seguida para página de vídeo e executa-lo, e, finalmente, retornar à página de texto para executar o arquivo de áudio.

Os resultados podem ser observados nos gráficos seguintes e seus números absolutos estão em anexo ao final deste trabalho.

Figura 15 - Gráfico do uso de Threads



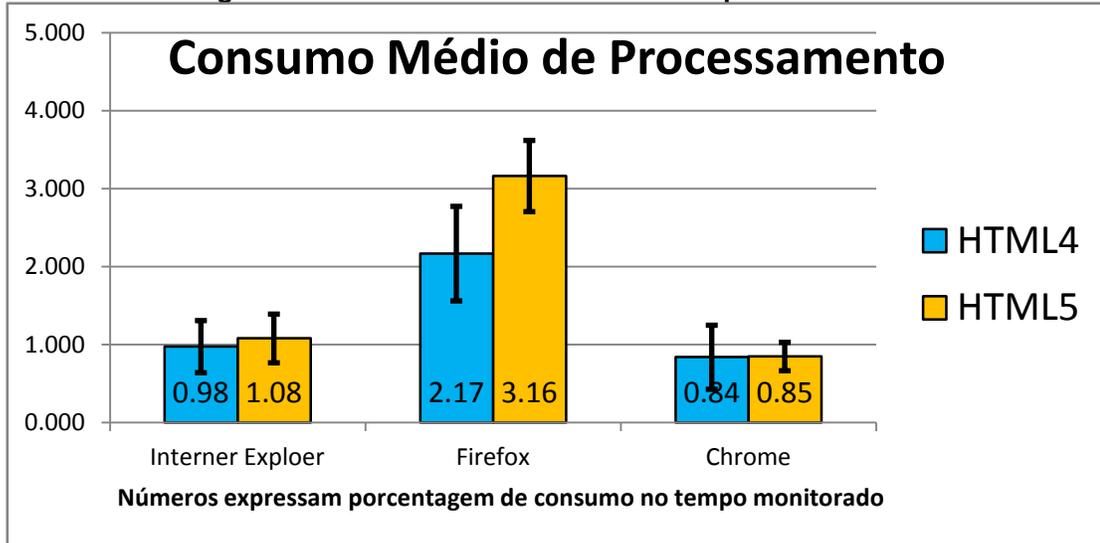
Fonte: Autoria própria.

Na figura 15 é mostrado, em forma de gráfico, a média do uso de *threads* do processador e os números de desvio padrão estão em anexo ao final do trabalho.

Threads são as divisões que um processo de *software* cria para um melhor aproveitamento do potencial do processador.

Exceto pelo *Google Chrome*, os navegadores apresentaram uma maior geração de *threads* na versão HTML5.

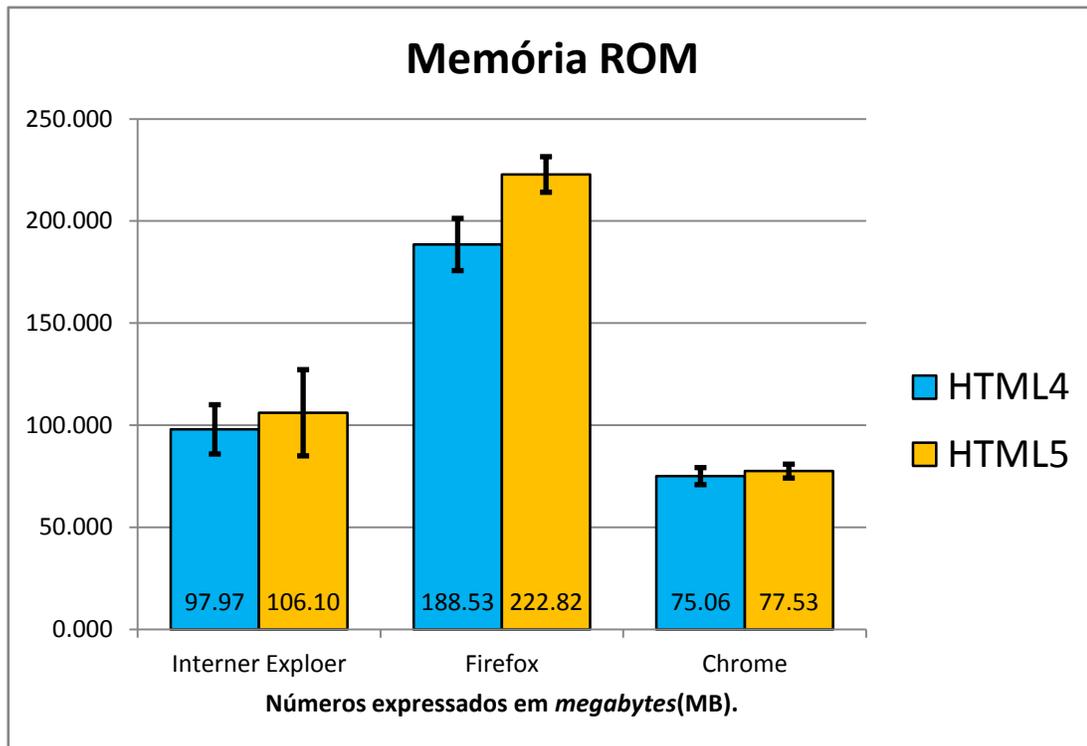
Figura 16 - Gráfico do consumo médio de processamento



Fonte: Autoria própria.

Já na figura 16, pode-se observar o consumo médio de processamento que, mesmo o desvio padrão dos resultados sendo menor na versão HTML5, esta tem um consumo maior nos três navegadores testados.

Figura 17 - Gráfico de alocação de memória ROM

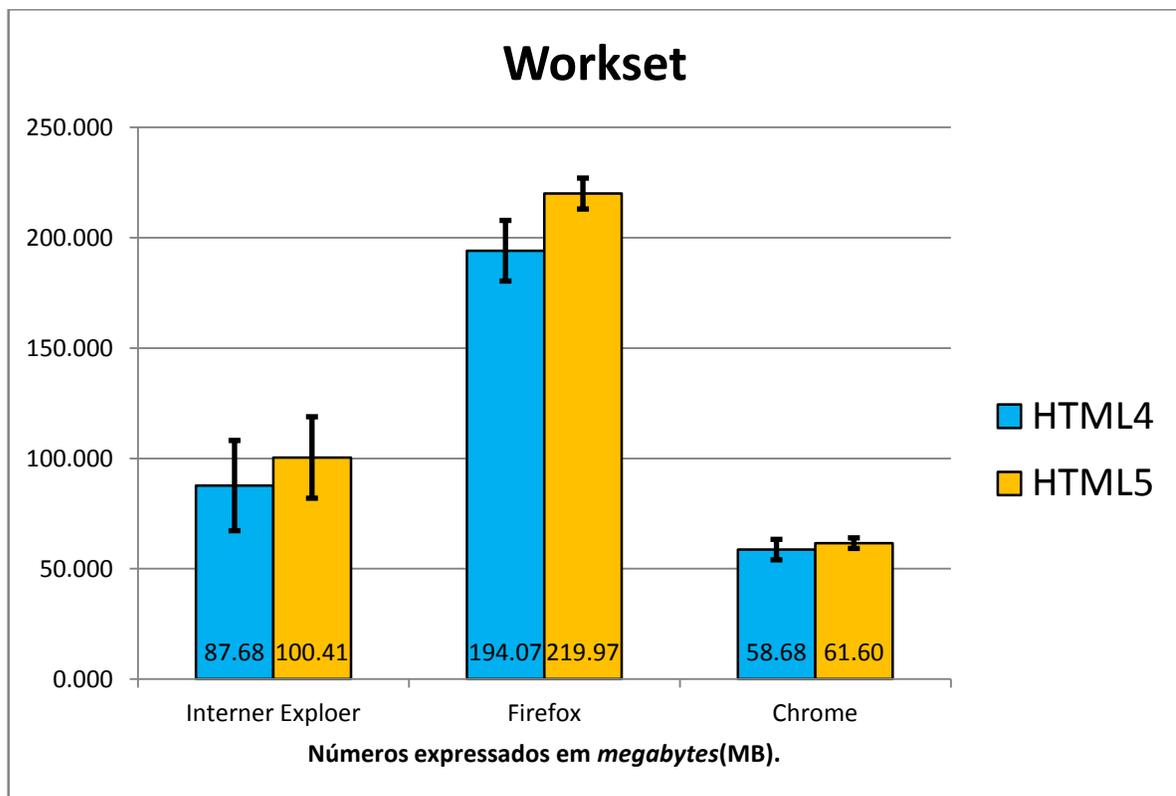


Fonte: Autoria própria.

No gráfico mostrado na figura 17 é demonstrado o aumento médio na alocação de espaço em disco, para a execução do navegador interpretando o código em HTML5. A alocação de memória é o processo onde um *software* solicita e utiliza memória para sua execução.

Mesmo que os três navegadores usem alocação dinâmica para sua execução, a média revelada nos testes ainda demonstra um aumento no consumo por parte da interpretação do código HTML5.

Figura 18 - Gráfico da alocação de memória de trabalho



Fonte: Autoria própria.

Na figura 18, observa-se o consumo médio da alocação de memória do conjunto de trabalho resultantes dos testes.

O conjunto de memória de trabalho é a somatória da memória privada, quantidade indispensável à execução do *software*, e memória compartilhável, quantidade alocada como reserva, que caso não esteja em uso pode ser dividida com outros processos com maior prioridade.

5.2 Perspectiva do Desenvolvedor

Para conhecer as opiniões e expectativas dos desenvolvedores, foram realizadas entrevistas com um grupo de sete desenvolvedores de conteúdo *Web*, profissionais que atuam na área há pelo menos dois anos, todos com perfis bem variados, garantindo a diversidade de opiniões e evitando a realização de uma pesquisa tendenciosa.

Os resultados obtidos nas entrevistas serão transcritos aqui, na forma de citação indireta. E os desenvolvedores serão numerados, de modo a proteger suas identidades.

5.2.1 Desenvolvedor 1

O *desenvolvedor 1* é uma mulher que trabalha há cinco anos como *Web designer*. Ela já trabalhou para a empresa de jogos *LevelUp* e, atualmente, trabalha para a empresa *Decolar.com*. Em sua opinião, o HTML5 pode ser mesmo uma melhoria. Mas acrescenta que ainda é muito cedo para abandonar tecnologias que já eram bem dominadas. Ela acredita que as grandes ferramentas de auxílio à linguagem HTML4 não cairão em desuso, mas se adaptarão à nova realidade do desenvolvimento *Web*, e continuarão a facilitar a vida dos desenvolvedores.

5.2.2 Desenvolvedor 2

Com experiência de seis anos na área de desenvolvimento *Web*, o *desenvolvedor 2* tem uma empresa de criação de aplicações *Web* e presta serviço terceirizado para mais de vinte empresas no interior do estado de São Paulo. Ele diz que a atualização na linguagem HTML vem de forma tardia, acrescentando que já seria necessário o planejamento de atualização para o HTML5. Argumenta que as ferramentas de auxílio são bastante completas e que, na maioria dos casos, tem sido mais vantajoso trabalhar sem o uso do HTML5. Contudo, afirma que a nova versão é muito mais organizada e didática, o que pode vir a aumentar o número de desenvolvedores e, por consequência, melhorar a *Web* no geral.

5.2.3 Desenvolvedor 3

Há três anos trabalhando para a empresa *HP* e chefiando um grupo de outros quatro desenvolvedores, o *desenvolvedor 3*, conta que já utiliza e tem estudado bastante a linguagem HTML5. Acrescenta que também tem conseguido convencer seu

grupo a fazer o mesmo; pois, para ele, é preciso alcançar todos os clientes e seus dispositivos cada vez mais variados. Resalta que o HTML5 ajudaria nesse intento.

5.2.4 Desenvolvedor 4

O *desenvolvedor 4* tem pouco mais de 2 anos de experiência e leciona HTML em uma franquia de escolas de Informática. Segundo ele, o fato da quinta versão trazer suporte nativo à multimídia e maior organização à estrutura da linguagem, melhorou bastante sua didática e despertou maior interesse dos alunos no aprendizado, pois os mesmos percebem os resultados mais facilmente.

5.2.5 Desenvolvedor 5

Iniciando sua empresa de consultoria *Web*, o *desenvolvedor 5* tem seis anos de experiência na área. Ele defende energicamente o uso de ferramentas de auxílio a desenvolvimento. Também resalta que o HTML, mesmo em sua quinta versão, sem o uso de completos como *CSS* e *JavaScript*, fazia parte da *Web 1.0*. Acrescenta que o atual cenário de desenvolvimento *Web*, de modo voraz, selecionando apenas os conteúdos mais interessantes, consegue ter um melhor visual, de maneira mais eficaz, de um modo que interesse ao usuário final.

5.2.6 Desenvolvedor 6

O *desenvolvedor 6*, com cinco anos e meio de experiência, abandonou seu Mestrado na Universidade de São Paulo para dedicar-se exclusivamente ao seu sistema de comércio eletrônico. Segundo ele, o HTML5 é uma ótima e bem-vinda melhoria, que realmente cumpre seu papel: ser base de uma página *Web* e prover recursos para outras ferramentas. Ele também conta que atualizou seus documentos *Web* há alguns meses e os resultados foram melhores do que o esperado. Disse que as novas *tags*, como a geolocalização, são perfeitas para quem pretende alcançar clientes que passam a maior parte do tempo conectado por dispositivos móveis.

5.2.7 Desenvolvedor 7

A *desenvolvedora 7* trabalha há três anos para uma empresa especializada na criação de conteúdo interativo para a *Web*. Ela conta que ainda não se aprofundou

muito nas mudanças no HTML, mas dúvida que, mesmo nos melhores casos, o suporte nativo supere ou, pelo menos, iguale-se ao poder de ferramentas externas; mas reconhece que ferramentas que acompanham essa atualização tendem a se tornarem mais ágeis e melhores.

5.3 Perspectiva do Usuário

Para colher as informações sobre a percepção do usuário final, foi pedido a um grupo de onze pessoas que usassem as duas versões de teste de páginas *Web* nos três navegadores. Logo após o uso, foi realizada uma entrevista para se conhecer as percepções do usuário. Foram abordadas onze pessoas também de perfis variados, de modo a não comprometer os resultados obtidos.

Foram requisitadas estas pessoas para que usassem as páginas *Web* implementadas para os testes, nos três navegadores. As mesmas deveriam tentar o *login* errado e, em seguida, *logarem* corretamente. Ao entrarem na tela de texto, deveriam reproduzir o arquivo de áudio e, posteriormente, irem para a tela de vídeo e reproduzirem o mesmo, na necessidade do *plugin* Flash, foi pedido que cada usuário o instalasse seguindo as instruções dadas pelo navegador e, finalmente voltassem à tela de texto.

De modo unânime, os usuários alegaram preferir a implementada em HTML5, pois, de modo geral, o *logo* animado e o vídeo não sofreram qualquer tipo de engasto ou *lag*, e o reprodutor de multimídia, além de uma aparência amigável e resposta mais rápida, em especial o reprodutor de áudio que não abriu o arquivo em uma nova janela ou aplicação externa. Ainda foi ressaltada, por todos os usuários, a praticidade de reproduzir o conteúdo multimídia sem ter que acessar outra página e baixar um *plugin*. Isso foi lembrado, principalmente, pelos usuários com perfil mais leigo, pois muitos pensaram estar correndo riscos de serem vítimas de *softwares* mal-intencionados.

Ainda foi constatada a preferência de nove dentre os onze usuários ao *Google Chrome*, mesmo alguns que não sabiam por quais razões, ainda o julgavam o melhor dentre os três navegadores utilizados nos testes. Dois dos usuários preferiram a versão no *Microsoft Internet Explorer*, alegando serem acostumados ao uso deste navegador, demonstrando até certo receio de interagirem com outros navegadores.

5.4 Comparativo das três perspectivas

Nos testes realizados, foi possível analisar dados sobre o impacto da atualização da linguagem HTML, quando se constatou que a versão HTML4 consome menos recursos do sistema de modo geral, o que, deveria fazer com que tivesse melhor desempenho. Mesmo assim, a preferência dos usuários recaiu sobre a implementação da versão HTML5, que rendeu uma melhor experiência de navegação. Tal constatação serviu para confirmar o que foi dito pelos desenvolvedores de modo geral.

Foi observado que os desenvolvedores com maior experiência não compartilharam da mesma empolgação dos menos experientes. No entanto, em sua maioria, os desenvolvedores concordam que existe uma real necessidade de incrementar o código, de modo a aperfeiçoar a experiência do usuário ao navegar por páginas *Web* mais interessantes e atrativas.

Os usuários leigos mal notaram as diferenças nas implementações; enquanto os outros, com mais conhecimento, puderam elencar com nitidez suas opiniões a respeito das mudanças da linguagem e diferenças entre navegadores.

Quanto ao sistema pode-se analisar que o navegador é decisivo no quesito de desempenho, como o navegador com maior variância nos resultados, o *Microsoft Internet Explorer*, pode causar eventuais *lags* ou travamentos em sistemas com configurações mais modestas.

O navegador *Mozilla Firefox*, intermedeia os testes com grande consumo de recursos e uma boa navegabilidade, mesmo ainda requerendo a instalação de *plugins* e exigindo uma configuração de sistema mais robusta para sua execução ideal.

Por outro lado, o navegador *Google Chrome*, se mostrou o menor consumidor de recursos dos testes, e obteve a menor variância dos resultados, onde se pode notar que mesmo em alguns casos onde a implementação em HTML5 consumiu mais recursos, ainda sim as diferenças foram mínimas. Assim de modo geral pode ser dito que o navegador *Google Chrome*, é o mais eficiente por não resultar em experiências diferentes ao usuário dependendo da linguagem utilizada, ainda se mostrou a melhor escolha para todos os tipos de configurações de sistema.

6 CONSIDERAÇÕES FINAIS

Este capítulo final trata das principais dificuldades na realização deste trabalho, do que é possível concluir do trabalho realizado, do que foi abordado na realização do mesmo e das propostas para trabalhos futuros.

6.1 Dificuldades

Inúmeras dificuldades foram encontradas no decorrer deste trabalho, principalmente com relação às escolhas sobre o que testar e de como testar.

O que testar, por existir um grande leque de opções disponíveis para testar as diferenças entre as versões da linguagem HTML. Após muita reflexão, chegou-se à conclusão de que um modo adequado e satisfatório seria confrontar a quarta versão e suas necessárias ferramentas com a quinta versão e suas novas opções, substituindo as ferramentas externas.

Nesse ponto, foi necessário escolher quais ferramentas deveriam ser usadas nos testes. Decidiu-se pelo uso do *Adobe Flash*, por ser a ferramenta mais relevante nessa questão, embora existissem muitas outras. Os desenvolvedores usam uma variedade incrível de ferramentas de auxílio, o que acabou sendo levado em consideração no momento dessa decisão.

Foi preciso ainda fazer um estudo sobre os navegadores, levando em conta muitas variáveis que, em alguns, poderiam ser analisadas e, em outros, não, por serem navegadores proprietários de código fechado. Foi preciso ainda estudar o funcionamento dos navegadores, a fim de se escolher as métricas para se determinar se houvessem melhorias ou não.

Como testar foi outra escolha que precisou de muita reflexão e análise. Primeiramente, foram escolhidos quais recursos do sistema que deveriam ser testados, optando-se por não testar o processamento gráfico, pois ele dependia exclusivamente de cada navegador. Depois, foi preciso encontrar uma ferramenta que atendesse à necessidade do teste de registrar o consumo em processador e memórias *ROM* e *RAM*.

Por fim, decidiu-se que aferir os recursos de sistema não bastaria para chegar a uma conclusão convincente. Para isso, seria necessário conhecer a perspectiva dos

desenvolvedores e usuários e, ainda, escolher perfis variados, no intuito de realizar um estudo imparcial sobre o conceito de quem conhece a linguagem e de quem a usa.

Em seguida, criou-se um roteiro para ser utilizado nas entrevistas, de modo a coletar as informações pertinentes aos testes da melhor maneira possível, para depois transcrevê-las de maneira resumida e relevante.

Após a realização de todos os testes e entrevistas, ainda foi preciso um estudo mais aprofundado sobre a interação dos navegadores com as versões da linguagem e com o sistema em âmbito geral.

Com os conhecimentos obtidos através desses estudos, foi preciso realizar uma análise e reflexão minuciosas sobre os resultados, a fim de se chegar a uma conclusão relevante e responsável.

6.2 Conclusão

Fundamentado nos dados obtidos neste trabalho, é possível chegar a uma série de conclusões.

Em primeiro lugar, os resultados dos testes provaram que o HTML5 consome mais processamento. Contudo, deve-se levar em consideração qual o navegador utilizado, pois como foi o caso do navegador *Google Chrome*, o aumento de consumo é ínfimo, tornando imperceptível a diferenciação de linguagens no quesito navegabilidade.

Outra constatação foi verificar que os desenvolvedores, mesmo os menos empolgados com as mudanças na linguagem, percebem as vantagens que elas trazem, embora estejam convictos de que o desenvolvimento em puro HTML5 não pode atender ao mercado que apresenta uma demanda cada vez mais exigente.

Além disso, fazem prognósticos otimistas sobre o futuro da *Internet*. Com o código mais simples, organizado e didático, o número de desenvolvedores tende a crescer, conseqüentemente, aumentará a competição pelo número de usuários e a qualidade dos conteúdos *Web*.

Notou-se ainda que, no caso do usuário final, aqueles que possuem conhecimento na área, percebem facilmente as melhorias, enquanto os leigos não percebem tão facilmente as mudanças. Ainda assim, tem menos preocupações, pois

evita redirecionamentos e notificações de falhas no conteúdo, situações que usuários mais leigos tendem a relacionar com *softwares* maliciosos. Ainda podendo sentir as vantagens quanto à usabilidade do conteúdo *Web*, acessado em seus dispositivos móveis e interfaces de interação *touchscreen*.

Assim sendo, torna-se evidente que as mudanças quanto à linguagem HTML chegam a tempo de alavancar a evolução da *Internet*, mas é preciso que tal melhoria não seja apenas da HTML, mas de todas as ferramentas que melhoram o desenvolvimento e o acesso à informação. Tais mudanças na linguagem não vieram para *matar* nenhuma tecnologia, muito menos para tomar o lugar de alguma ferramenta. Tais mudanças vieram sim para forçar uma melhoria em todo o cenário *Web*, desde a base, no desenvolvimento, até o final na experiência do usuário.

6.3 Trabalhos Futuros

Para melhorar os resultados obtidos e integralizar o conhecimento seria interessante que se fosse dada continuidade a este estudo, considerando *o que e quanto* o uso de diferentes navegadores pode alterar os resultados obtidos.

Seria interessante também uma análise dos motores gráficos dos navegadores e como eles interferem no desempenho do sistema e na experiência do usuário, analisando os recursos necessários tanto em HTML4 quanto em HTML5..

7 REFERÊNCIAS

ADOBE. Dreamweaver Help. Disponível em:

<<http://helpx.adobe.com/en/dreamweaver.html>>. Acessado em 06 set. 2013.

ADOBE. Dreamweaver Reviews. Disponível em:

<<http://www.adobe.com/en/products/dreamweaver/reviews.html>>. Acessado em 06 set. 2013.

BARROS, I. G. M. O; SANTOS, C. F. A. PETele – Apostila de Introdução ao CSS. UFF, Niterói – RJ, 2008.

BATISTA, D. C. T. O Impacto do HTML5 no Desenvolvimento para a *Internet*. Cornélio Procópio, Paraná, 2010.

BUGS, W. *Internet e Navegadores | Browsers*. Disponível em

<<http://www.wagnerbugs.com.br/arquivos/material/browser.pdf>>. Acessado em 04 set. 2013.

CROWLEY, M. D. Professional *Internet Explorer* Development. Editora Apress. Ohio, 2010.

ECMAScript. ECMAScript Language Specification. Disponível em

<<http://www.ecmascript.org/docs.php>>. Acessado em 30 ago. 2013.

EIS, D; FERREIRA, E. HTML5 e CSS3 com farinha e pimenta. Editora Tableless. São Paulo, 2012.

FERREIRA, Y. M. P; MAGALHÃES, P. R; CÉSAR, I. Navegadores Web. UFF, Niterói - RJ, 2007.

FLANAGAN, D. JavaScript: The Definitive Guide, 6th Ed. 2011.

FURLAN, M. P. Dreamweaver CS5. Disponível em <salernopublicidade.com.br/arquivos/manuais/ApostilaDreamweaverCS5.pdf>. Acessado em 06 set. 2013.

GAY, J. The History of Flash. Disponível em <http://www.adobe.com/macromedia/events/john_gay/index.html>. Acessado em 23 ago. 2013.

GOOGLE. Google Releases. Disponível em <<http://googlechromereleases.blogspot.com/>>. Acessado em 06 set. 2013.

GOOGLE. The Chromium Projects. Disponível em <<http://dev.chromium.org/developers/how-tos/get-the-code>>. . Acessado em 06 set. 2013.

GOOGLE. Dúvidas sobre o AdSense. Disponível em <<https://support.google.com/adsense/answer/181949?hl=pt-BR>>. Acessado 06 set. 2013.

HEY, D. F. Estudo de Viabilidade do HTML5 para Desenvolvimento Web. Maringá, Paraná, 2010.

JAVASCRIPT SOURCE. Core JavaScript Reference. Disponível em <http://www.javascriptsource.com/reference/core_ref/contents.html>. Acessado em 30 ago. 2013.

JOBS, S. Thoughts on Flash. Disponível em <<http://www.apple.com/hotnews/thoughts-on-flash>>. Acessado em 23 ago. 2013.

KARLINS, D. Adobe Creative Suite 5 Web Premium How-Tos:100 Essential. San Jose, Califórnia, 2010.

MALENKOVICH, S. Segurança do navegador: Google Chrome. Disponível em <<http://blog.kaspersky.com.br/seguranca-do-navegador-google-chrome/>>. Acessado em 06 set. 2013.

MOZILLA FOUNDATION. Get to know Mozilla. Disponível em <<http://www.mozilla.org/en-US/about/>>. Acessado em 04 set. 2013.

MOZILLA FOUNDATION. The Mozilla Manifesto. Disponível em <<https://www.mozilla.org/en-US/about/manifesto/>>. Acessado em 04 set. 2013.

MÜLLER, E. Apresentando as Características da Tecnologia Flash. UDESC, Joinville – SC, 2011.

NOTEPAD++. About and Features. Disponível em <<http://notepad-plus-plus.org/features/>>. Acessado em 07 set. 2013.

PEDROSO, R. P; GOMES, P. A; RIBEIRO, R. O. PETele – Apostila de HTML. UFF, Niterói – RJ, 2007.

REIS, R. T. Desenvolvimento Web com o Uso de Padrões: Tecnologias e Tendências. UFJF, Juiz De Fora – MG. 2007.

SAMY, M. S. Html5 - a Linguagem da Marcação Que Revolucionou a Web. São Paulo, 2011.

STATCOUNTER. StatCounter Global Stats. Disponível em <<http://gs.statcounter.com/>>. Acessado 06 set. 2013.

TECMUNDO. Vantagens De Um Editor WYSIWYG. Disponível em <<http://www.tecmundo.com.br/institucional/2057-vantagens-do-wysiwyg-.htm> >.

Acessado 06 set. 2013

W3C. HTML5 differences from HTML4. Disponível em <<http://www.w3.org/TR/html5-diff/>>. Acessado 23 ago. 2013.

W3C. Princípios HTML5. Disponível em <<http://www.w3.org/TR/html-desing-principles>>.
Acessado em 28 ago. 2013.

W3Schools. Tutorial de HTML5. Disponível em <<http://www.w3schools.com/html5>>.
Acessado 28 ago. 2013.

8 ANEXOS

Em anexo, seguem-se as tabelas com os resultados de todas as interações realizadas nos testes.

Tabela 5 - Média de resultados dos testes

Média (Resultados de memória em Mb)						
Internet Explorer	Threads	CPU	ROM	workset	Share	PVT
HTML4	39	0.976	97.968	87.678	34.291	53.387
HTML5	46	1.081	106.098	100.413	36.331	64.081
Firefox	Threads	CPU	ROM	workset	Share	PVT
HTML4	42	2.169	188.529	194.074	34.070	161.484
HTML5	66	3.163	222.824	219.971	41.086	179.227
Chrome	Threads	CPU	ROM	workset	Share	PVT
HTML4	42	0.840	75.060	58.683	22.191	36.492
HTML5	42	0.849	77.528	61.598	22.850	38.748

Tabela 6 - Margens de erro dos testes

Desvio Padrão (Resultados de memória em Mb)						
Internet Explorer	Threads	CPU	ROM	workset	Share	PVT
HTML4	8	0.334	12.070	20.439	6.192	16.891
HTML5	10	0.312	21.114	18.450	2.023	17.693
Firefox	Threads	CPU	ROM	workset	Share	PVT
HTML4	2	0.606	12.791	13.737	4.338	13.340
HTML5	2	0.457	8.714	7.008	2.452	8.548
Chrome	Threads	CPU	ROM	workset	Share	PVT
HTML4	2	0.411	4.178	4.636	1.598	3.149
HTML5	2	0.182	3.402	2.391	0.818	1.787

Tabela 7 - Resultado dos testes no Internet Explorer em HTML4

Processamento HTML4		Memorias HTML 4 (Resultados exibidos em Kb)			
Threads	Com. Médi. CPU	reserv	workset	compar.	pvt
31	0.57	82752	84600	39928	44672
33	0.63	98320	96288	34684	61604
32	0.88	81888	80604	37904	42700
33	0.45	97144	95144	34852	60292
55	1.04	111952	105088	33768	71320
45	0.69	85440	80652	37896	42756
25	0.77	93344	92812	34668	58144
52	0.89	118896	108312	35036	73276
51	1.32	114768	106244	36056	70188
36	0.90	99480	97924	34748	63176
54	0.69	116212	105496	33832	71664
42	0.99	116140	105772	33940	71832
36	0.69	87008	80208	37608	42600
32	1.02	83392	80436	37736	42700
31	0.81	93772	93748	34856	58892
42	1.35	113048	101776	32668	69108
44	1.50	111448	103304	36068	67236
41	0.69	87168	81804	37808	43996
31	0.60	92852	93736	37736	56000
53	1.34	115928	105956	34472	71484
39	1.35	105996	83636	27052	56584
42	1.17	87104	76832	32972	43860
40	1.06	83232	81048	37856	43192
37	0.90	87996	41924	25472	16452
43	0.72	88944	96396	36968	59428
31	0.70	97376	95884	35172	60712
39	0.66	98852	31560	16680	14880
34	1.59	99584	26756	18972	7784
50	1.11	112220	104744	29024	75720
45	1.20	116356	106568	34828	71740
33	1.86	108940	125688	50944	74744
37	0.65	108248	85812	38496	47316
34	0.85	114980	102804	47928	54876
38	1.25	111096	113248	40252	72996
37	1.45	105708	83332	42680	40652
36	0.80	83904	76032	32540	43492

Tabela 8 - Resultados dos testes no Internet Explorer em HTML5

Processamento HTML5		Memorias HTML 5 (Resultados exibidos em Kb)			
Threads	Com. Médi. CPU	reserv.	workset	compar.	pvt
37	0.99	88012	89816	37088	52728
52	1.27	117564	112656	35832	76824
53	0.69	119312	117620	37036	80584
37	0.84	83104	80376	33952	46424
52	1.86	130128	117012	37620	79392
53	1.47	131412	118236	37220	81016
49	1.35	91212	92044	37636	54408
65	1.68	157820	138536	41256	97280
53	1.10	128560	126108	39496	86612
35	0.87	98784	98352	42296	56056
54	1.66	126676	115596	39888	75708
53	1.14	90852	91744	37500	54244
38	0.72	83168	80952	38856	42096
58	1.31	145768	127148	40904	86244
41	0.87	92920	91512	33620	57892
33	1.02	88384	82860	36456	46404
36	1.14	89248	82380	35424	46956
56	0.87	122484	121368	36566	84802
44	0.64	94104	93792	34724	59068
28	0.76	82912	85956	38680	47276
55	1.24	118254	110348	39452	70896
43	0.94	89678	87516	36200	51316
31	0.80	85824	81072	33232	47840
55	1.18	122536	120876	35924	84952
53	1.12	134460	128848	37376	91472
47	1.07	125996	114108	37652	76456
30	0.86	102144	81564	34256	47308
48	0.67	90128	91172	37428	53744
51	0.98	108044	106200	36228	69972
36	1.04	86400	74856	36676	38180
56	1.17	131560	128884	37108	91776
62	1.76	141572	134280	38080	96200
38	0.84	81472	80640	37980	42660
32	0.76	88992	80076	37536	42540
46	1.04	127164	114940	37276	77664
54	1.20	114540	102172	34864	67308

Tabela 9 - Resultados dos testes no Firefox em HTML4

Processamento HTML4		Memórias HTML 4 (Resultados exibidos em Kb)			
Threads	Com. Médi. CPU	reserv	workset	compar.	pvt
41	1.58	198416	203540	35412	168128
43	2.63	192984	206808	42804	164004
43	1.61	180884	197032	44416	152616
45	1.67	196944	215912	44176	171736
44	2.63	204332	214064	34276	179788
43	2.75	191464	202196	36464	165732
44	1.94	197304	206096	36172	169924
41	1.88	159036	172404	38912	133492
44	2.21	173060	190748	45120	145628
44	1.96	189376	198812	35556	163256
42	1.21	199452	211152	36216	174936
43	1.17	203760	217568	38284	179284
41	0.80	207296	221000	38308	182692
42	2.06	172212	189852	45280	144572
44	1.70	180720	196716	42500	154216
41	1.40	162920	157468	29744	127724
41	1.13	189996	200248	35256	164992
41	2.52	193124	194052	30800	163252
43	3.01	199212	204968	33176	171792
42	2.46	195368	197956	32228	165728
40	2.07	183624	196176	42784	153392
43	1.88	179700	184152	34172	149980
43	2.20	180924	181480	32512	148968
37	3.35	152492	157456	30864	126592
44	2.03	192488	195388	32660	162728
43	2.59	187740	192764	35052	157712
42	2.63	196736	198352	31580	166772
44	2.60	195832	198652	31724	166928
40	3.03	202032	204856	32760	172096
41	2.52	176912	181332	34504	146828
41	2.99	196088	201252	35036	166216
41	2.28	196916	201068	34740	166328
40	1.98	196068	201468	34728	166740
40	2.47	203812	205860	32192	173668
41	2.96	200244	202488	32544	169944
44	2.17	188516	192740	34504	158236

Tabela 10 - Resultados dos testes no Firefox em HTML5

Processamento HTML5		Memorias HTML 5 (Resultados exibidos em Kb)			
Threads	Com. Médi. CPU	reserv.	workset	compar.	pvt
63	2.72	246900	245080	40868	204212
65	2.68	226568	224412	42256	182156
67	2.73	214920	220196	43760	176436
67	3.30	228048	224036	41816	182220
66	3.64	220592	224480	42572	181908
69	3.27	231544	228040	41420	186620
65	2.73	232080	227648	41316	186332
65	2.33	223308	221924	44076	177848
67	3.37	231504	228048	40056	187992
68	3.20	227580	226444	42972	183472
65	2.59	232376	229176	41184	187992
66	3.47	229008	225636	42052	183584
67	2.75	222124	221312	43912	177400
67	2.34	244056	239996	40496	199500
65	3.42	229416	224884	41492	183392
65	3.51	227932	225616	41968	183648
68	3.80	231096	226568	41892	184676
66	2.69	242080	240176	42092	198084
63	3.45	229044	224784	41164	183620
65	3.66	223416	222876	44072	178804
66	3.16	229720	226320	41564	184756
66	3.07	244300	241884	42160	199724
62	3.12	228472	225648	41760	183888
65	2.86	229512	227204	42316	184888
66	2.59	224352	223372	43656	179716
65	3.94	223840	222360	43968	178392
66	3.34	217832	222208	49132	173076
63	2.92	247808	244084	41284	202800
66	3.87	228024	223696	41468	182228
66	3.23	224084	222784	43868	178916
65	3.20	228296	225712	41452	184260
66	2.95	212860	218568	51604	166964
63	3.66	248292	241572	39716	201856
66	4.20	223516	221976	43888	178088
63	2.88	220996	221188	44108	177080
67	3.24	217760	221992	49228	172764

Tabela 11 - Resultados dos testes no Chrome em HTML4

Processamento HTML4		Memorias HTML 4 (Resultados exibidos em Kb)			
Threads	Com. Médi. CPU	reserv	workset	compar.	pvt
47	2.36	86332	79684	30656	49028
40	0.87	79176	60392	21728	38664
41	0.61	77972	60600	22524	38076
43	0.73	79972	63284	23176	40108
41	0.70	77200	59768	22524	37244
37	1.80	63780	48272	19648	28624
42	0.73	77564	61284	23696	37588
42	0.70	77812	60740	22548	38192
41	0.62	76396	58744	22488	36256
41	0.54	76960	61276	23936	37340
42	0.66	76760	59880	22708	37172
40	0.63	76428	59156	22672	36484
44	0.77	74932	57852	22336	35516
44	0.63	78572	60920	22736	38184
44	0.74	78404	61416	22740	38676
42	0.70	78112	60964	22652	38312
42	1.12	78776	61128	22704	38424
43	1.39	77396	59880	22624	37256
42	0.75	77568	60060	22660	37400
41	0.74	77424	59976	22712	37264
39	0.81	63480	49168	20000	29168
40	0.84	77424	59892	22668	37224
41	0.65	77116	59700	22572	37128
42	0.84	78132	60516	22680	37836
40	0.65	78196	60600	22636	37964
42	0.69	77100	59616	22612	37004
42	0.61	78028	61088	22772	38316
41	2.04	64708	49728	19896	29832
39	0.73	78660	61132	22696	38436
44	0.56	77712	60772	22676	38096
41	0.61	77956	60448	22688	37760
41	0.63	78248	60760	22596	38164
41	0.74	78056	60544	22696	37848
43	0.65	78448	62868	24108	38760
41	0.67	78264	60772	22588	38184
41	0.73	77960	60404	22680	37724

Tabela 12 - Resultados dos testes no Chrome em HTML5

Processamento HTML5		Memorias HTML 5 (Resultados exibidos em Kb)			
Threads	Com. Médi. CPU	reserv.	workset	compar.	pvt
45	0.67	79260	64408	24620	39788
44	0.78	80496	63388	22616	40772
39	1.69	64484	49716	19904	29812
39	0.56	80588	65064	23756	41308
40	0.72	79332	64096	24552	39544
42	0.73	79720	63072	23388	39684
43	0.79	79392	62676	23472	39204
43	0.96	79728	62488	23388	39100
41	0.82	79964	63064	23392	39672
44	0.70	66492	64112	24332	39780
41	0.85	80584	63104	23320	39784
40	0.80	79828	62932	23372	39560
45	0.69	79452	62640	23344	39296
41	0.91	80976	65536	24600	40936
44	0.77	81500	62716	23588	39128
43	0.85	80156	63436	23400	40036
42	0.92	78724	63484	24752	38732
41	1.13	80892	63548	23432	40116
44	0.88	79776	62392	23420	38972
42	0.81	80708	63732	23396	40336
39	0.92	80388	63524	23392	40132
40	0.80	79824	63012	23416	39596
41	0.94	80744	63336	23400	39936
44	1.01	80844	63396	23452	39944
42	0.75	79964	62112	22652	39460
43	0.82	80564	66284	24516	41768
43	0.80	81364	63864	23392	40472
41	0.72	79908	62676	23408	39268
42	0.70	80332	63684	23476	40208
42	1.04	80544	63032	22912	40120
42	0.82	80864	63988	23412	40576
41	0.88	80256	62916	23488	39428
43	0.86	80472	63560	23412	40148
41	0.79	79392	62404	22332	40072
43	0.81	79832	63848	22224	41624
44	0.88	80656	63496	23396	40100