

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

Rodrigo Fagundes

DESENVOLVIMENTO DE UM WEB-BROWSER
COM SISTEMA DE ABAS MULTITHREAD

Florianópolis
2009

Rodrigo Fagundes

**DESENVOLVIMENTO DE UM WEB-BROWSER
COM SISTEMA DE ABAS MULTITHREAD**

Trabalho de Conclusão de Curso submetido à
Universidade Federal de Santa Catarina
como parte dos requisitos para a obtenção do
grau de Bacharel em Ciências da Computação

Professor Dr. José Mazzucco Júnior

**BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CENTRO TECNOLÓGICO
UNIVERSIDADE FEDERAL DE SANTA CATARINA**

Florianópolis
Outubro 2009

Rodrigo Fagundes

DESENVOLVIMENTO DE UM WEB-BROWSER
COM SISTEMA DE ABAS MULTITHREAD

Trabalho de Conclusão de Curso submetido à
Universidade Federal de Santa Catarina como
parte dos requisitos para a obtenção do grau de
Bacharel em Ciências da Computação

Professor Dr. José Mazzucco Júnior
Universidade Federal de Santa Catarina

Professor Dr. Roberto Willrich
Universidade Federal de Santa Catarina

Professor Dr. Vítório Bruno Mazzola
Universidade Federal de Santa Catarina

RESUMO

FAGUNDES, Rodrigo. Desenvolvimento de um web-browser com sistema de abas multithread. Trabalho de conclusão de curso (Graduação em Ciências da Computação) – Departamento de Informática e Estatística da Universidade Federal de Santa Catarina, Florianópolis, 2009.

O presente trabalho trata do desenvolvimento de um web-browser, também chamado de navegador, que utiliza múltiplas threads para o funcionamento de seu sistema de abas. Primeiramente foi focado o desenvolvimento das funcionalidades básicas do navegador, como a troca de mensagens com um servidor pelo protocolo HTTP para ser possível acessar documentos contidos nele, e a interpretação de documentos HTML. Em seguida foi desenvolvido o sistema de abas, permitindo ao usuário abrir várias páginas da Internet sem a necessidade de iniciar uma nova instância do navegador. Por último foi incluído as threads para alcançar as vantagens de um sistema multithreads.

ABSTRACT

FAGUNDES, Rodrigo. Desenvolvimento de um web-browser com sistema de abas multithread. Trabalho de conclusão de curso (Graduação em Ciências da Computação) – Departamento de Informática e Estatística da Universidade Federal de Santa Catarina, Florianópolis, 2009.

The present work deals with the development of a web-browser, also called navegador, that uses multiple threads to the operation of its tab's system. At first was focus the development of the basic features of the browser, such as the exchange messages with a server over the HTTP protocol to be able to access documents contained in it, and the interpretation of HTML documents. Then the tab's system was developed, allowing the users to open multiple Web pages without the need to start a new instance of the browser. Finally it included the threads to achieve the advantages of a multithread system.

LISTA DE ILUSTRAÇÕES

FIGURA 1		
Navegador Nexus		14
FIGURA 2		
Navegador Netscape Navigator		15
FIGURA 3		
Navegador Internet Explorer		17
FIGURA 4		
Navegador Mozilla Firefox		18
FIGURA 5		
Mercado de navegadores		19
FIGURA 6		
Provedor de serviços da Internet		20
FIGURA 7		
Comparação dos modelos OSI e TCP/IP		21
FIGURA 8		
Exemplo de código HTML		23
FIGURA 9		
Exemplo de código CSS		24
FIGURA 10		
Exemplo de código JavaScript		25
FIGURA 11		
Exemplo de processos com threads		26
FIGURA 12		
Estrutura da interface do navegador		27
FIGURA 13		
Diagrama de classes do navegador		28
FIGURA 14		
Tela do navegador		29

LISTA DE ABREVIATURAS E SIGLAS

CSS

Cascade Style Sheet

DARPA

Defense Advanced Research Projects Agency

DNS

Domain Name System

E-MAIL

Electronic-Mail

FTP

File Transfer Protocol

HTML

HyperText Markup Language

HTTP

HyperText Transfer Protocol

IP

Internet Protocol

ISP

Internet Service Provider

ITU

International Telecommunication Union

NCP

Network Control Protocol

NCSA

National Center for Supercomputing Applications

OSI

Open Systems Interconnection

SMTP

Simple Mail Transfer Protocol

SSH

Secure Shell

TCP

Transmission Control Protocol

TELNET

Teletype network

XHTML

Extensible HyperText Markup Language

XML

Extensible Markup Language

W3C

World Wide Web Consortium

WEB

World Wide Web

WHATWG

Web HyperText Application Technology Working Group

SUMÁRIO

RESUMO

ABSTRACT

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

1		
INTRODUÇÃO		11
1.1	Objetivos	12
2		
HISTÓRIA DA INTERNET		13
2.1	Origem da Internet	13
2.2	História da Internet	14
3		
FUNCIONAMENTO DA INTERNET		20
3.1	Estrutura da Internet	20
3.2	Estrutura de um documento da Web	23
4		
THREADS		26
5		
DESENVOLVIMENTO		27

5.1 Testes Comparativos	30
6 CONCLUSÃO	32
6.1 Trabalhos Futuros	32
REFERÊNCIAS	33
APÊNDICES	35
A – Código fonte do navegador	35
B – Documento HTML de exemplo	44
C – Resultados dos testes realizados	45
D - Artigo	47

1 INTRODUÇÃO

Atualmente o uso da Internet é algo corriqueiro para a maior parte das pessoas, e por consequência, o uso de navegadores também é algo comum. Um navegador, ou web-browser, é um aplicativo que permite a visualização de documentos HTML, tais documentos podem estar em servidores na Internet, ou na própria máquina do usuário. Com sua utilização é possível aproveitar ao máximo as possibilidades da Internet, tal qual a troca de informação e a comunicação entre usuários.

Threads dividem as tarefas de um programa, o que pode gerar um aplicativo mais eficiente e que permite a execução de tarefas diferentes simultaneamente, quando há mais de um processador disponível, ou alternadamente, quando há somente um processador disponível. Isso permite que um programa possa executar outras tarefas sem que fique esperando um longo tempo em uma tarefa que demande mais tempo.

Neste trabalho busquei desenvolver um navegador que utilizasse múltiplas threads no seu sistema de abas para tentar obter um aplicativo com desempenho satisfatório, usufruindo das vantagens de um sistema multithread.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Entender e implementar um navegador que utilize múltiplas threads em seu sistema de abas visando obter as vantagens de um sistema multithread e consiga receber e enviar dados através do protocolo HTTP e também interpretar corretamente documentos HTML.

1.1.2 Objetivos Específicos

- Identificar e explorar as partes de um navegador que possam ser executadas com múltiplas threads.
- Conseguir a troca de dados com servidores através do protocolo HTTP.
- Permitir ao navegador a capacidade de interpretar HTML.
- Permitir ao navegador a capacidade de interpretar Folhas de Estilo em Cascata.
- Permitir ao navegador a capacidade de interpretar JavaScript.
- Obter um navegador com desempenho satisfatório.

1.2 ORGANIZAÇÃO DO TEXTO

No capítulo 2 deste trabalho aborda-se resumidamente a história da Internet, incluindo sua origem, evolução e a história dos principais navegadores. O capítulo 3 menciona o modelo TCP/IP e os principais protocolos utilizados pela Internet (TCP, IP e HTTP) e também aborda as principais tecnologias utilizadas por um navegador (HTML, CSS e JavaScript). No capítulo 4 é conceituado um sistema multithread. O capítulo 5 descreve como foi desenvolvido o navegador. Por fim, o capítulo 6 conclui o trabalho e aponta possíveis trabalhos futuros.

2 HISTÓRIA DA INTERNET

2. 1. ORIGEM DA INTERNET

A Internet teve origem a partir de um projeto militar dos Estados Unidos denominado ARPANET. A ARPANET foi desenvolvida em 1969 pela Defense Advanced Research Projects Agency (DARPA), objetivando criar um sistema de comunicação que continuasse funcionando mesmo que fosse parcialmente danificado.

Inicialmente a ARPANET interligava quatro instituições: Universidade da Califórnia de Los Angeles, Universidade da Califórnia de Santa Bárbara, Instituto de Pesquisa de Stanford, e Universidade de Utah. Com isso era possível a troca de informações através do programa SNDMSG, desenvolvido por Ray Tomlinson. Este programa introduziu o caractere @ para separar o nome do usuário do nome da máquina, semelhante ao sistema de correio eletrônico (e-mail) utilizado atualmente. Com o tempo outras instituições tiveram permissão de conectar suas redes internas a ARPANET e ela passa a ser chamada de a rede das redes.

O termo Internet surge em 1974, na primeira especificação completa do Transmission Control Protocol (TCP). Este protocolo substitui o Network Control Protocol (NCP), usado até então pela ARPANET. Durante a década de 80 o número de computadores ligados a ARPANET cresceu rapidamente, incluindo também máquinas de instituições de outros países além dos Estados Unidos. A Internet passa a ser uma rede maior, que engloba a ARPANET.

Na década de 90 a Internet foi liberada para propósitos comerciais. Empresas passam a vender serviços como o e-mail e a transferência de arquivos pelo File Transfer Protocol (FTP). A utilização da Internet aumenta ainda mais nos anos seguintes com a criação do primeiro navegador chamado de WorldWideWeb, renomeado posteriormente para Nexus a fim de evitar confusão com a World Wide Web (Web).

A World Wide Web, que pode ser traduzido para rede de alcance mundial, é um rede de documentos hipertexto interligados que podem ser acessados pela Internet através de um navegador. Ela foi proposta por Tim Bernes-Lee e Robert Cailliau em 1990. Nesta proposta foi definido a arquitetura

da Web, qual seria o papel de um servidor e de um navegador e também metas para o desenvolvimento do projeto (BERNERS-LEE; CAILLIAU, 1990).

Atualmente existem mais de 1,5 bilhões de pessoas em todo mundo que utilizam a Internet, segundo o relatório anual da International Telecommunication Union (2008).

2.2 HISTÓRIA DOS NAVEGADORES

Web-Browsers ou navegadores são ferramentas que permitem a visualização de documentos HTML, que podem estar em servidores na Internet, ou na própria máquina do usuário.

Segundo Berners-Lee (2009), durante o início da década de 1990 o único meio de se acessar documentos na Web era usando o navegador desenvolvido por ele, primeiramente chamado de WorldWideWeb e posteriormente renomeado para Nexus.

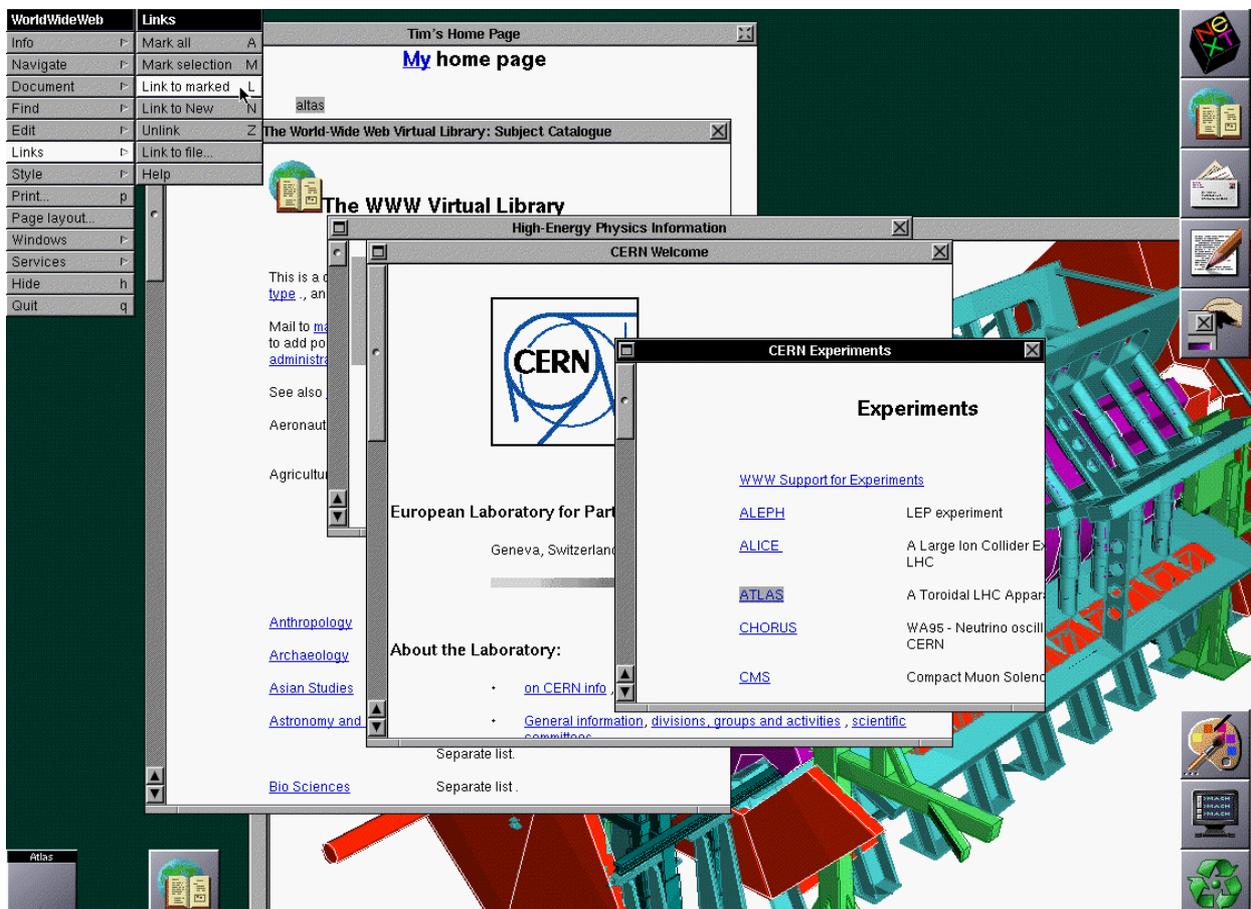


Figura 1: Navegador Nexus (BERNERS-LEE, 2009).

Em 1993 o código fonte do Nexus é liberado como de domínio público, permitindo assim que qualquer um possa usar o navegador. No mesmo ano, Marc Andreessen e Eric Bina, na época no grupo de software design da National Center for Supercomputing Applications (NCSA), desenvolvem o Mosaic, um navegador para sistemas Unix e posteriormente para Windows e Apple Macintosh, sendo o primeiro navegador para diferentes plataformas. O Mosaic foi o navegador mais popular de sua época.

Posteriormente Marc funda a empresa Mosaic Communications, juntamente com Jim Clark. Mais tarde a empresa é renomeada para Netscape Communications Corporation. No final de 1994 a Netscape lança um navegador com codinome Mozilla, o primeiro navegador a ser comercializado. Posteriormente o nome do navegador muda para Netscape Navigator. Com a inclusão de novas funcionalidades e integração com outros serviços como o e-mail, fez com que o navegador se popularizar rapidamente.

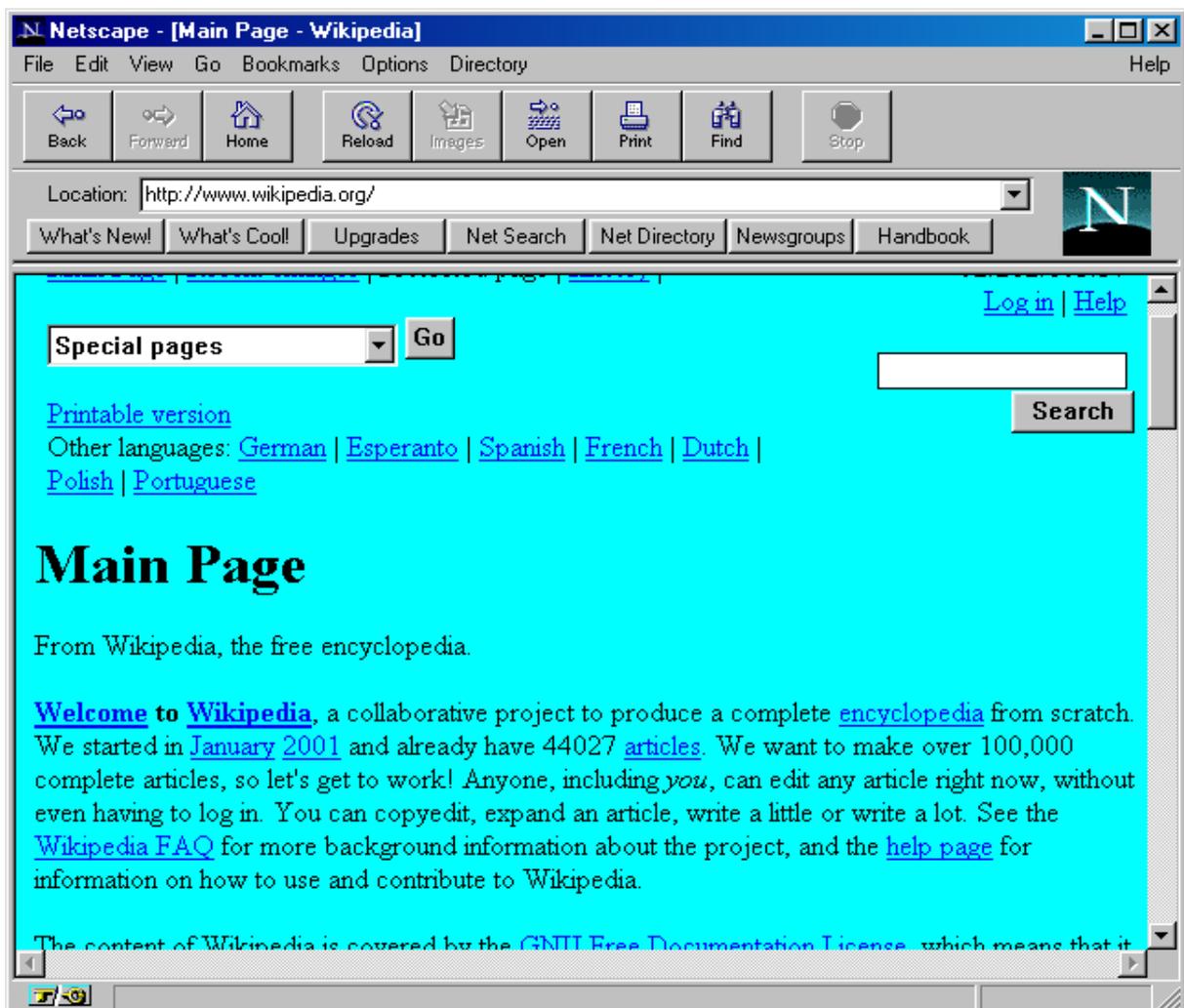


Figura 2: Netscape Navigator (WIKIPEDIA, 2009).

Devido a popularidade do navegador da Netscape e do crescente uso da Web, a Microsoft Corporation resolve lançar seu próprio navegador, comprando da Spyglass Inc., divisão comercial da NCSA, a tecnologia do Mosaic. Tal tecnologia serve de base para o navegador Internet Explorer lançado em Agosto de 1995. Com isso tem início a chamada Guerra dos Browsers, período marcado pela luta pelo domínio do mercado de navegadores entre a Netscape e a Microsoft.

A Guerra dos Browsers ocorre durante o período de 1995 a 2002, com os navegadores Netscape Navigator e Internet Explorer competindo pelo mercado. Inicialmente a Netscape tinha o domínio do mercado, com mais de 90% dos usuários utilizando o Netscape Navigator. No final deste período a Microsoft conseguiu reverter a situação, com 95% dos usuários utilizando o Internet Explorer.

O Internet Explorer teve algumas vantagens em relação ao seu concorrente:

- Era gratuito, enquanto o Netscape cobrava pelo software;
- Teve maior suporte financeiro;
- A partir da versão 4, lançada em 1997, vem integrado ao sistema operacional Microsoft Windows.

Devido a essa disputa, cada empresa começa a criar funcionalidades próprias no interpretador HTML e posteriormente no interpretador Javascript. Isso fez com que os documentos HTML fossem visualizadas de maneira diferente nos navegadores. Em 1998 com o lançamento da quinta versão do Internet Explorer havia cinco interpretadores diferentes de código JavaScript em uso pelas diferentes versões de navegadores disponíveis. Isto significa que fazer uma página que fosse acessível a grande parte dos usuários da Web era impossível.

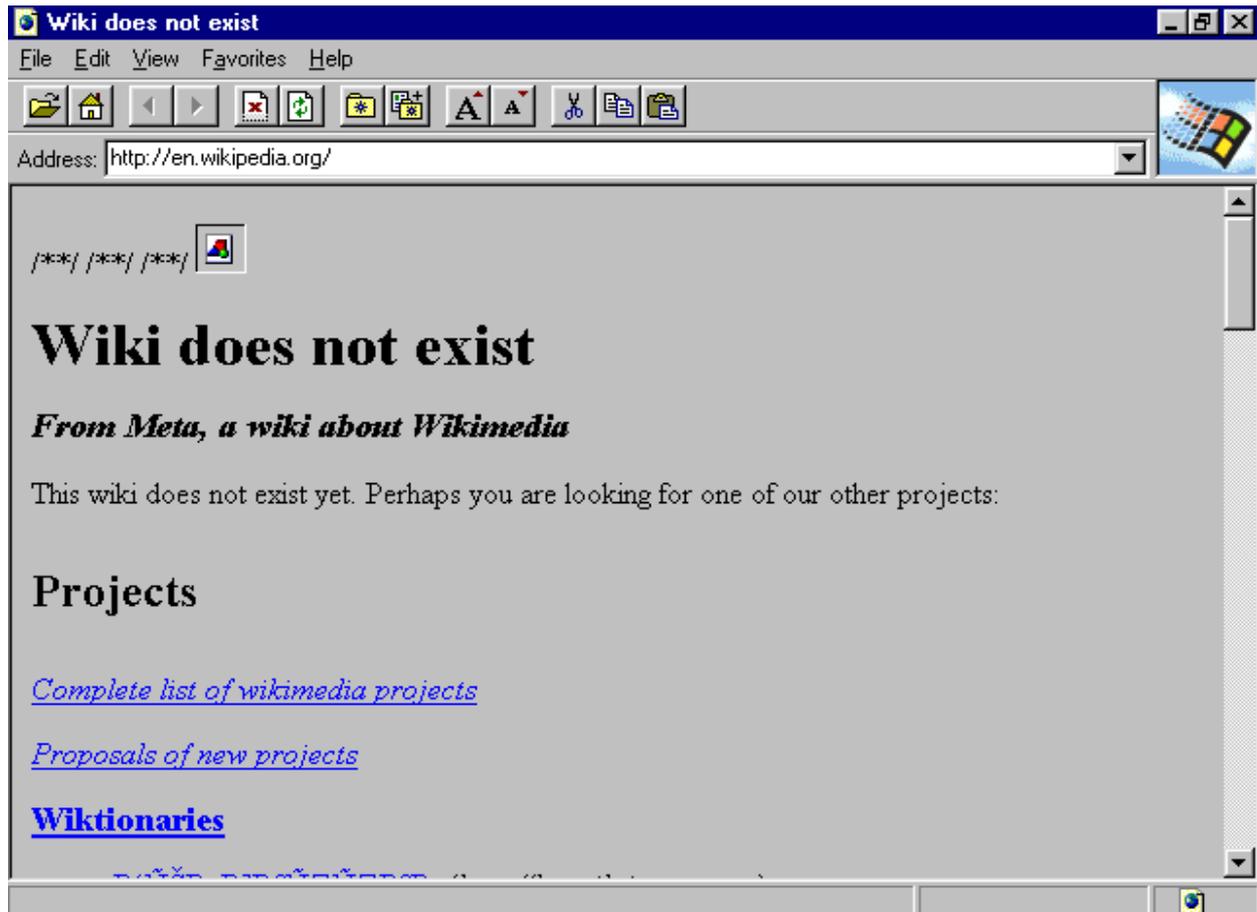


Figura 3: Internet Explorer (WIKIPEDIA, 2009).

De 1998 até 2002 o Internet Explorer foi ganhando mercado rapidamente. Com isso a guerra dos navegadores acaba, e a Microsoft diminui os investimentos no seu navegador. Isso só vai mudar com o início da segunda guerra dos browsers.

Visando organizar a Web, é fundado em 1994 por Berners-Lee o World Wide Web Consortium (W3C). Para acabar com a incompatibilidade de código HTML entre navegadores, é proposto pela W3C diversas recomendações para padronizar o HTML, as folhas de estilo e outras tecnologias usadas na Internet.

Após perder a disputa pelo mercado de navegadores, a Netscape resolve licenciar o código fonte do Netscape Navigator como de código aberto, antes de ser vendida para a empresa America Online. Com base neste código fonte, surge o projeto Mozilla e o navegador de mesmo nome.

Em 2002, um pequeno grupo de desenvolvedores do navegador Mozilla, resolvem criar um navegador menos dependente da Netscape e dão origem ao navegador Mozilla Firefox. Com

diversas inovações este navegador ganha espaço entre os usuários da Internet.

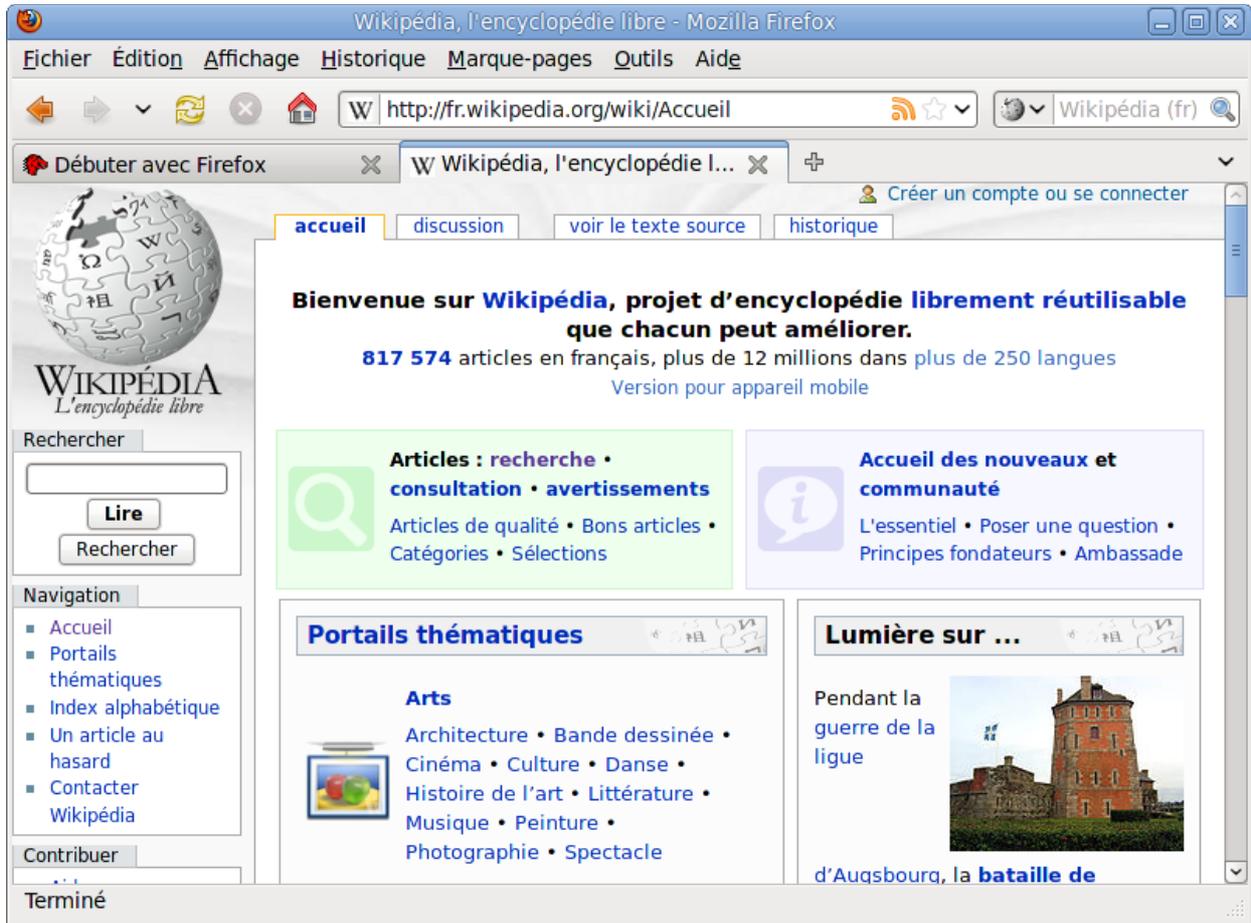


Figura 4: Mozilla Firefox (WIKIPEDIA, 2009).

Com a perda de usuários pela primeira vez desde que dominou o mercado, a Microsoft volta a investir no seu navegador, e depois de cinco anos sem ter uma nova versão, é lançado Internet Explorer 7.

Atualmente a disputa pelo domínio deste mercado continua. Segundo a empresa Net Applications (2009), o Internet Explorer mantém-se dominando o mercado com 65,71% dos usuários, enquanto o Firefox tem 23,75%.

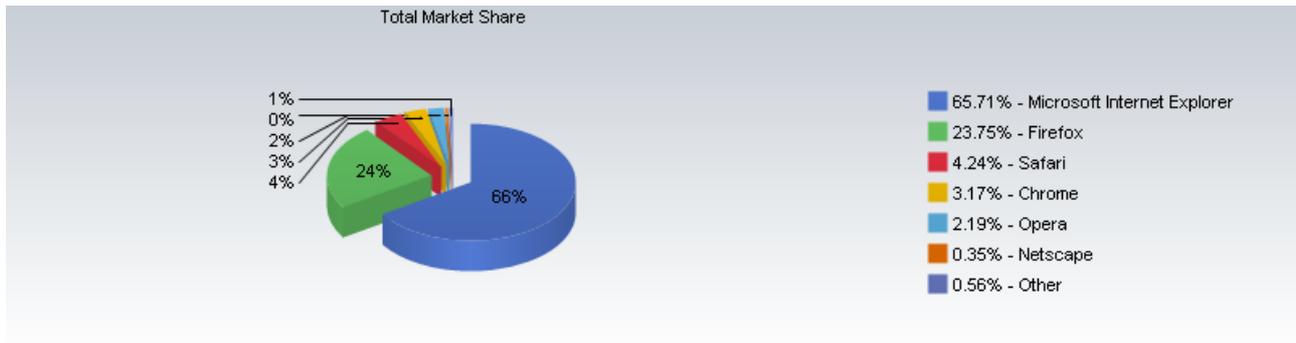


Figura 5: Mercado de navegadores (NET APPLICATION, 2009).

Outros navegadores também tentam ganhar parte do mercado à algum tempo. A empresa Apple investiu na criação de seu próprio navegador, o Safari, lançado em 2003 para o sistema operacional Mac OS X e posteriormente para o Windows.

A empresa norueguesa Opera Software ASA criou o navegador Opera em 1995. Inicialmente apenas para o sistema Windows, em 1998 a empresa começou a explorar novos mercados para seu navegador, como o uso dele em vídeo-games e dispositivos portáteis para acessar a Web.

A última empresa a entrar nesse mercado foi a Google em 2008. Apostando na popularidade dos serviços da empresa, como seu sistema de buscas e o serviço de armazenamento de vídeos, para motivar os usuários a usarem seu navegador, o Google Chrome.

Atualmente a importância dos navegadores tem aumentado, devido a novos serviços disponíveis na Web.

3 FUNCIONAMENTO DA INTERNET

3.1 ESTRUTURA DA INTERNET

Para se conectar a Internet um computador, ou rede local, deve se conectar a um Provedor de Serviços da Internet (ISP). A ISP por sua vez se conecta a ISP maiores que estão conectadas a outras, permitindo assim a navegação na Internet.

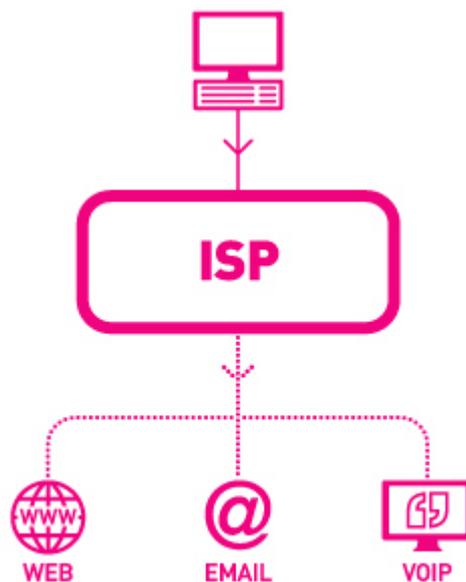


Figura 6: Provedor de serviços da Internet (EXCELL GROUP, 2009).

Para troca de dados e acesso a arquivos e informações contidas na Internet é usado uma série de protocolos que especificam como deve ser feita essa comunicação entre as máquinas. Através da família de protocolos TCP/IP, é feito todo trabalho para o estabelecimento da comunicação entre as máquinas cliente e servidor, e permitindo assim que dados sejam transmitidos.

A Internet usa uma estrutura de rede semelhante ao modelo Open Systems Interconnection (OSI), chamado de modelo TCP/IP. A diferença entre estes modelos é que a camada de Aplicação do modelo da Internet engloba as camadas de Apresentação, Aplicação e Sessão. Conforme Tanenbaum (2003, p.46), o modelo TCP/IP não viu necessidade de incluir estas camadas, pois são

pouco usadas na maioria das aplicações. Esta camada utiliza diversos protocolos. Já a camada de Transportes utiliza os protocolo TCP (Transmission Control Protocol) e UDP (User Datagram Protocol), enquanto a camada de rede utiliza o protocolo IP. As camadas Física e de Enlace do modelo OSI estão englobadas sem distinção na camada de Interface com a Rede do modelo TCP/IP. Uma comparação entre as camadas dos modelos OSI e TCP/IP pode ser visualizada abaixo:

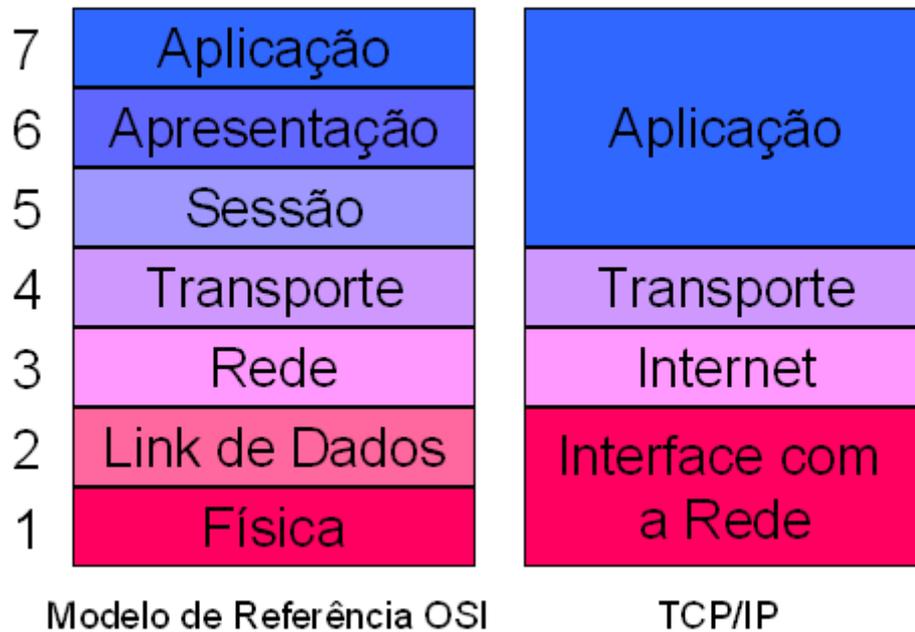


Figura 7: Comparação dos modelos OSI e TCP/IP (TORRES, 2009).

3.1.1 Camada de Interface com a rede

Segundo Tanenbaum (2003, p.47), o modelo TCP/IP não especifica bem esta camada, sendo que o único fato definido é que um “host deve se conectar à rede utilizando algum protocolo”. Também não é claramente definido nenhum protocolo desta camada pelo modelo TCP/IP.

3.1.2 Camada de Internet

Tanenbaum (2003, p.45) diz que, no modelo TCP/IP, “a camada internet define um formato de pacote oficial e um protocolo chamado IP. A tarefa da camada internet é entregar pacotes IP onde eles são necessários”. Segundo DARPA, (1981, tradução minha) o protocolo IP (Internet Protocol) deve “prover as funções necessárias para a entrega de pacotes [...] de uma fonte a um destino através

de um sistema de redes interconectadas.” Também é dito que este protocolo deve servir como interface entre o protocolo de aplicação e o protocolo da rede local.

3.1.3 Camada de Transporte

A camada de transporte do modelo TCP/IP tem como finalidade “permitir que as entidades pares de hosts de origem e de destino mantenham uma conversação, exatamente como acontece na camada de transporte OSI” Tanenbaum (2003, p.46). Esta camada tem dois protocolos, o TCP e o UDP, sendo o primeiro um protocolo orientado a conexões confiável e o segundo não confiável. Isto significa que o TCP é um protocolo que entregará os dados sem erros, enquanto o UDP não garante isso. Na especificação do TCP foi definido que o protocolo "tem como objetivo prover um serviço de comunicação de confiança entre processos finais em um ambiente com redes diversas" (DARPA, 1981, tradução minha). este protocolo então garante a entrega dos pacotes, verifica se eles estão sendo enviados de forma correta, na sequência apropriada e sem erros. Usualmente um navegador utiliza, através do protocolo HTTP, o protocolo TCP.

3.1.4 Camada de Aplicação

A camada de Aplicação do modelo TCP/IP, é a camada utilizada pelos aplicativos para se comunicar pela rede com outros programas. Os protocolos mais usados na Internet são o HTTP e o SMTP, sendo o protocolo SMTP usado para a o envio de e-mails. Já o protocolo HTTP é "um protocolo do nível de aplicação com a leveza e velocidade necessária para ser utilizado com sistemas de informação de hipermídia distribuídos e colaborativos" (BERNERS-LEE; FIELDING; NIELSEN, 1996, tradução minha). Ele é usado para realizar requisições de uma máquina cliente a um servidor e assim o cliente pode acessar arquivos e dados no servidor, como páginas HTML. Outros protocolos desta camada incluem FTP (File Transfer Protocol), SSH (Security Shell), DNS (Domain Name System), TELNET (Teletype network), etc.

3.2 ESTRUTURA DE UM DOCUMENTO DA WEB

Um documento, ou página, da Web é um documento disponível na Internet, normalmente estruturado em HTML. Um ou mais destes documentos formam um web site, ou simplesmente site. Web site pode ser traduzido para o português como local na teia, com teia significando rede.

3.2.1 HTML

O HyperText Markup Language (HTML), que pode ser traduzido para Linguagem de Marcação de Hipertexto, é “a linguagem para descrever a estrutura dos documentos Web” (WORLD WIDE WEB CONSORTIUM, 2009a). O código HTML é interpretado pelo navegador que monta página para ser visualizada pelo usuário. Criada por Tim Berners-Lee para facilitar a comunicação entre ele e seus colegas, atualmente a sua especificação é mantida pela W3C. Antes da criação das folhas de estilo em cascata, o HTML também era responsável pela formatação da página.

Exemplo de documento HTML:

```
01 <!DOCTYPE html>
02 <html>
03 <head>
04 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
05 <title>Título da página</title>
06 </head>
07 <body>
08 <!-- Comentário... -->
09 Conteúdo da página.
10 </body>
11 </html>
```

Figura 8: Exemplo de código HTML.

Atualmente o HTML está na versão 4. Este deveria ser substituído pelo HTML estendido (Extensible HyperText Markup Language – XHTML). Pela definição da W3C (WORLD WIDE WEB CONSORTIUM, 2009a)

“XHTML é um variante do HTML que usa a sintaxe do XML, a Linguagem de Marcação Estendida. XHTML tem os mesmos elementos (para parágrafos, etc.)

que o HTML, mas a sintaxe é levemente diferente. Devido a XHTML ser uma aplicação XML, pode-se usar outras ferramentas XML com ela [...]”.

Entretanto, em 2004, Apple, Mozilla e Opera criam o Web Hypertext Application Technology Working Group (WHATWG), pois estavam descontentes com algumas propostas do W3C, especialmente a falta de preocupação com as necessidades dos desenvolvedores Web. A WHATWG começou a trabalhar no HTML 5, com a intenção de adequar o HTML as necessidades reais dos desenvolvedores. Eventualmente a W3C aceitou discutir o HTML 5 e postergou o desenvolvimento da segunda versão do XHTML. Em Janeiro de 2009 a W3C publicou a primeira especificação da quinta versão do HTML, ainda em versão de testes. Segundo a especificação (WORLD WIDE WEB CONSORTIUM, 2009b) o HTML 5 substituirá tanto o HTML 4 quanto o XHTML 1.

3.2.2 Folhas de Estilo

As folhas de estilo em cascata (Cascade Style Sheet - CSS), servem para “descrever a apresentação dos documentos Web, incluindo cores, leiaute, e informações da fonte, assim como alterar a apresentação em diferentes dispositivos como aqueles com grandes telas, pequenas telas ou impressoras” (WORLD WIDE WEB CONSORTIUM, 2009a). Uma folha de estilo pode ser escritas dentro do documento HTML ou em um documento externo referenciado pelo documento HTML. A grande inovação das folhas de estilo é que permitem definir a aparência de diversas páginas em um só documento.

Exemplo de documento de folha de estilo:

```
01 @CHARSET "UTF-8";
02
03 body {
04  /* Comentário... */
05  background-color: black;
06  color: white;
07  font-size: 20px;
08  text-align: center;
09 }
```

Figura 9: Exemplo de código CSS.

Silva (2008, p. 418) apresenta algumas vantagens obtidas com o uso de folhas de estilo como o controle total sobre a formatação de um site em um único arquivo central, a agilização da manutenção e reestruturação de um site, e a redução do tempo de carga dos documentos Web.

Para demonstrar a versatilidade das folhas de estilo, Dave Shea criou em 2003 o ZenGarden. Este site é composto de apenas um documento HTML com diversas folhas de estilo que modificam totalmente a aparência da página. Esta página pode ser conferida em (SHEA, 2009).

3.2.3 JavaScript

Para possibilitar maiores recursos para os documentos HTML, foi criada a linguagem de script de programação JavaScript. Assim como as folhas de estilo, o JavaScript pode ser escrito dentro do documento HTML ou em um documento externo referenciado pelo documento HTML.

Exemplo de documento de JavaScript:

```
01 var x = 7;
02 var y = x * x;
03
04 function uma_funcao(z) {
05     // Comentário...
06     variavel = (x + y) / z;
07     if (variavel > 1) {
08         alert("exemplo...");
09     }
10     document.write("Variavel = " + variavel);
11 }
```

Figura 10: Exemplo de código JavaScript.

O JavaScript foi criado numa parceria entre a Netscape e a Sun Microsystems em 1995, tendo sintaxe semelhante a linguagem de programação Java. Em conjunto com as folhas de estilo permitem criar páginas HTML dinâmicas, podendo, através de uma ação do usuário, alterar a apresentação da página, validar dados, executar operações matemáticas, ou mesmo carregar toda uma nova página.

4 THREADS

Oliveira, Carissimi e Toscani (2004, p. 65) definem thread como "um fluxo de execução" de um processo. Já o processo é definido por Tanenbaum e Steen (2007, p.42) como "um programa em execução".

Um sistema Multithreading "[...] separa um processo em várias threads. Cada uma dessas threads é executada independentemente." (SUN MICROSYSTEMS, 2008, p. 17, tradução minha). Isso acarreta em vantagens e desvantagens em comparação a um sistema monothread.

Tanenbaum e Steen (2007, p. 43) citam como principal vantagem o "[...] fato de que, em um processo monothread, sempre que for executada uma chamada bloqueadora de sistema, o processo é bloqueado como um todo", enquanto que um processo multithread teria apenas uma de suas threads bloqueadas.

Por exemplo, em um aplicativo de planilhas, se este aplicativo for monothread quando o usuário entra com alguma alteração num campo que acarretará em cálculos em outros campos o usuário terá que esperar pela conclusão dos cálculos para poder realizar outra atividade. Já se o aplicativo for multithread enquanto uma thread se encarrega de realizar os cálculos, outra poderia permitir que o usuário realizasse uma outra. atividade sem a necessidade de esperar pela conclusão da primeira thread.

A figura abaixo, mostra como é representado um processo monothread, e um processo multithread.



Figura 11: Exemplo de processo monothread e multithread.

5 DESENVOLVIMENTO

O navegador foi desenvolvido utilizando-se a linguagem de programação Java e sua biblioteca de classes para implementar algumas das funcionalidades do navegador.

O ambiente montado para o desenvolvimento deste software consiste em:

- Sistema Operacional Ubuntu Hardy Heron (8.04)
- Ambiente de Desenvolvimento Integrado Eclipse Europa (3.3)
- Java Development Kit 6

Primeiramente foi construído um protótipo da interface do programa, utilizando-se dos componentes da biblioteca SWING. Na figura abaixo está a estrutura usada para criar o protótipo.

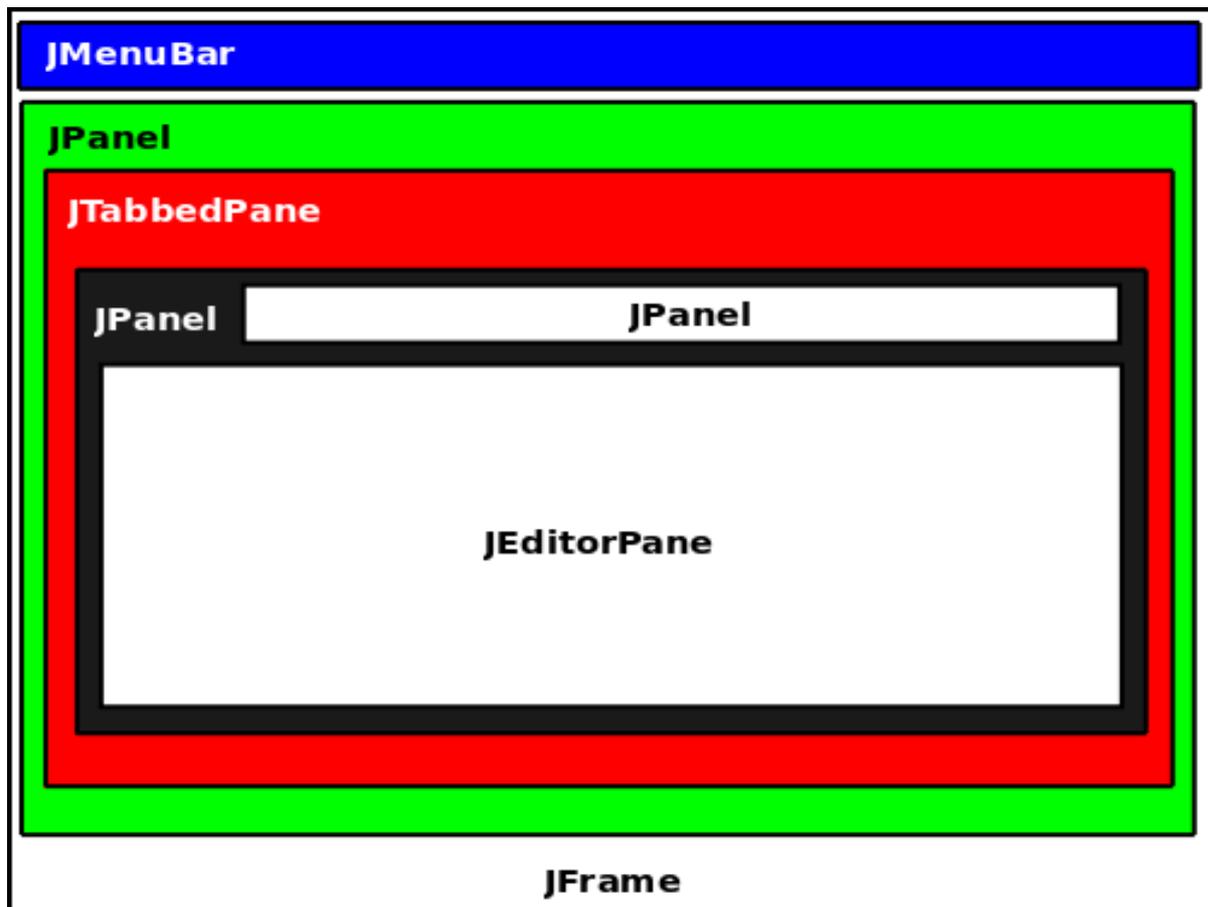


Figura 12: Estrutura da interface do navegador.

Com base nessa estrutura foram criadas as classes que compõe o navegador. O diagrama de classes abaixo demonstra as classes implementadas.

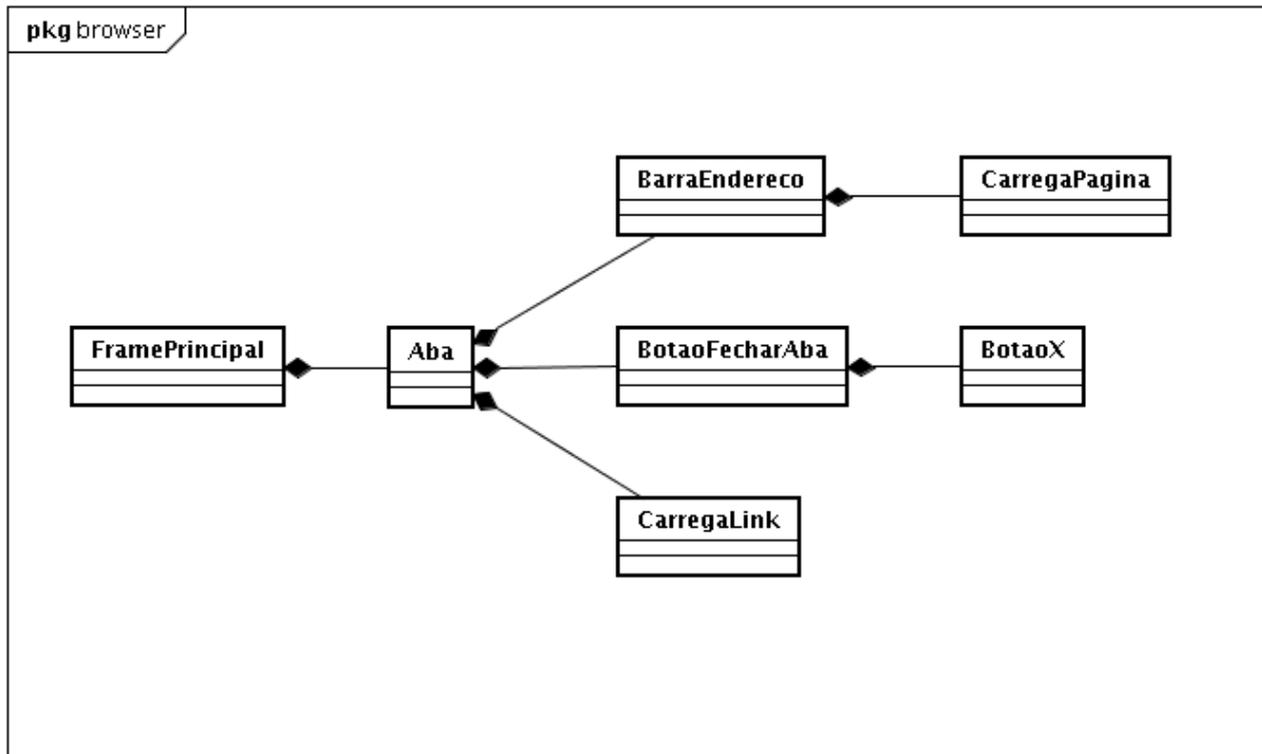


Figura 13: Diagrama de classes simplificado.

A classe *FramePrincipal* quando instanciada cria o menu, o painel que conterá as abas e uma aba inicial. Através do método *novaAba*, esta classe gera uma nova aba e a adiciona ao painel sempre que solicitado. Uma aba é representada pela classe *Aba* que contém a estrutura mínima para ser possível a abertura de uma página da Internet.

Todo o trabalho de comunicação entre o navegador e os servidores da Internet é feita pela classe *JEditorPane*, pertencente a biblioteca SWING. Essa mesma classe se encarrega de interpretação das páginas HTML. Foi criada duas classes que utilizam os métodos da classe *JEditorPane*, *CarregaPagina* e *CarregaLink*. Tais classes utilizam um controle de eventos para saber quando é feito a solicitação de uma página por parte do usuário.

A classe *Aba* é filha da classe *Thread*, sendo assim, cada aba criada representa uma thread do navegador. Isso torna cada aba passível do chaveamento de threads dentro do processo do navegador. Quando a aba é fechada a thread é destruída. Cada aba tem sua barra de endereços para

receber uma url de destino, e uma tela para mostrar a página HTML, esta tela sendo um *JEditorPane*.

Abaixo, a figura 14 mostra o navegador funcionando, com duas abas abertas, sendo que a aba selecionada mostra uma página de exemplo criada para executar testes com o navegador. Esta página HTML de exemplo pode ser consultada no apêndice B.

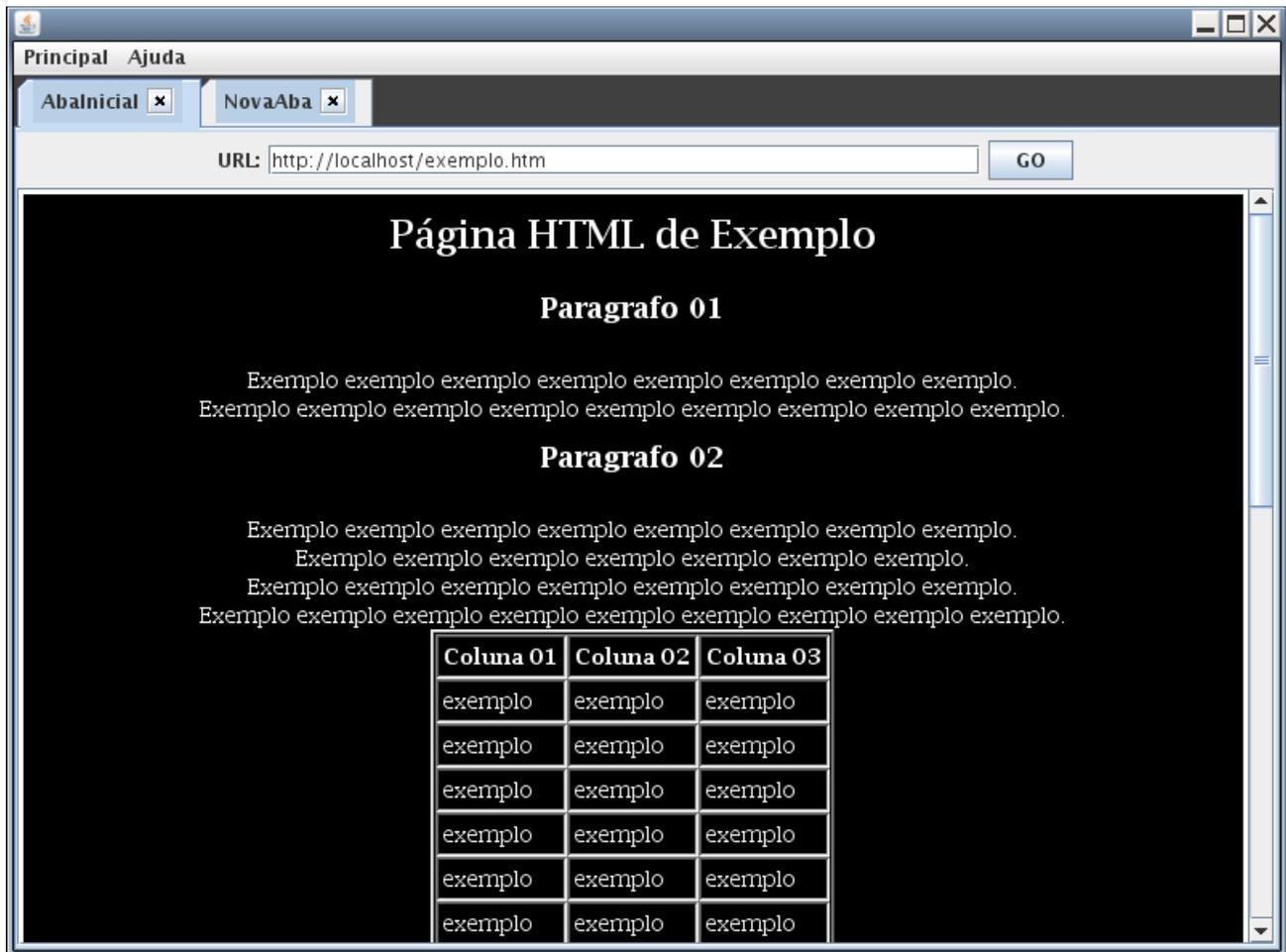


Figura 14: Tela do Navegador.

5.1 TESTES COMPARATIVOS

5.1.1 Comparação entre navegador usando threads e sem usa-las

Foram realizados testes em dois computadores, um com processador de dois núcleos e outro com processador de um núcleo. Foi comparado o tempo de criação de uma aba entre o navegador implementado e uma adaptação do mesmo que não utiliza threads. Sabe-se que processadores com mais de um núcleo tendem a aproveitar melhor um aplicativo multithread, então é esperado que o navegador com threads leve vantagem no processador de dois núcleos. Os testes realizados comprovaram isso, mostrando que na média a criação e execução de uma nova aba tende a ser mais rápida no navegador multithread quando está trabalhando com o processador de dois núcleos. Já no processador de um núcleo, o desempenho foi semelhante ao navegador sem múltiplas threads.

5.1.2 Comparação entre o navegador e um navegador monothread comercial

Foram realizados testes para comparar o desempenho do navegador com um navegador monothread comercial. Infelizmente o único navegador asseguradamente monothread encontrado foi o Internet Explorer 6. Infelizmente porque este navegador, sendo nativo do sistema operacional Windows, utiliza artefatos do próprio sistema para ter uma execução otimizada, sendo assim não é um navegador adequado para comparar com o navegador criado, já que este não pode utilizar os mesmos artefatos.

Nestes testes notou-se que quando abria-se uma nova janela do navegador monothread, a memória consumida aumenta em 1000k na média, já o navegador multithread consumiu 5000k cada vez que uma nova aba foi utilizada com a mesma url. É possível que isto se deva ao fato do navegador Internet Explorer guardar em cache os dados da página acessada, enquanto que o navegador multithread não implementa esta funcionalidade e necessita buscar do servidor os dados a cada solicitação. Apesar do consumo de memória ser maior em comparação com o navegador monothread, o uso da Web foi melhor no navegador multithread, sendo mais rápido a abertura de uma nova janela.

5.1.2 Comparação de uso com outros navegadores

Comparando a utilização do navegador desenvolvido com outros navegadores comerciais, notou-se um desempenho semelhante. As principais diferenças notadas são devidas a outros recursos incluídos nos navegadores comerciais. Por ser escrito em java, o navegador desenvolvido tem a vantagem de funcionar em qualquer computador com uma máquina virtual java.

Os resultados de todos os testes podem ser conferidos no apêndice C.

6 CONCLUSÃO

Com o presente trabalho desenvolveu-se um navegador que consegue solicitar documentos da Web de servidores da Internet e que consegue interpretar satisfatoriamente o seu HTML, mostrando a página aos usuário. Também se introduziu um sistema de abas no navegador e Threads neste sistema. Com isso conseguiu-se obter as vantagens que um sistema de múltiplas Threads trás.

A maior vantagem obtida com o uso de multithreads no sistema de abas é que não é necessário esperar a conclusão de uma solicitação de uma página HTML por parte do navegador para fazer outra solicitação, ou visualizar o resultado da interpretação do código de outra aba. Com o uso de múltiplas threads é possível que diversas abas solicitem páginas da Web sem bloquear o uso do navegador.

Entretanto, poderia ser ainda melhor o aproveitamento das threads pelo navegador, explorando ainda mais suas vantagens, conforme visto a seguir.

6.1 Trabalhos futuros

- Implementação de um interpretador de HTML e CSS multithreads;
Assim o navegador poderia dar uma resposta mais rápida ao usuário, pois poderia processar o HTML enquanto outras threads se preocupariam com a solicitação de imagens, scripts, etc.
- Implementação de um interpretador JavaScripts;
Para que funcionasse em conjunto com o interpretador HTML e CSS, permitindo ao navegador exibir corretamente as páginas HTML e dar uma melhor resposta ao usuário.
- Inclusão de novas funcionalidades ao navegador que tirem proveito das múltiplas threads.
Tal qual outros navegadores, poderia ser criado funcionalidades adicionais que rodariam em threads, aumentando a utilidade do navegador.

REFERÊNCIAS

BERNERS-LEE, Tim; CAILLIAU, Robert. **WorldWideWeb**: Proposal for a HyperText Project. Disponível em: <<http://www.w3.org/Proposal.html>>. Acesso em: 02 out. 2009.

INTERNATIONAL TELECOMMUNICATION UNION. **Trends in Telecommunication Reform 2008**: Six Degrees of Sharing. Disponível em: <http://www.itu.int/newsroom/press_releases/2008/35.html>. Acesso em: 02 out. 2009.

BERNERS-LEE, Tim. **The WorldWideWeb browser**. Disponível em: <<http://www.w3.org/People/Berners-Lee/WorldWideWeb.html>>. Acesso em: 02 out. 2009.

WIKIPEDIA. **Wikipedia**. Fonte das imagens dos navegadores Netscape, Internet Explorer e Firefox. Disponível em: <<http://www.wikipedia.org/>>. Acesso em: 13 out. 2009.

NET APPLICATIONS. **Browser Market Share**. Disponível em: <<http://marketshare.hitslink.com/report.aspx?qprid=0>>. Acesso em: 13 out. 2009.

EXCELL GROUP. **Internet Service Provider**. Fonte da imagem ISP. Disponível em: <<http://www.excellgroup.com/generated/0001/000151/v003/transform10/cache186/ISP.jpg>>. Acesso em: 13 out. 2009.

TORRES, Gabriel; LIMA, Cássio. **Comparação entre TCP/IP e OSI**. Fonte da imagem de comparação entre as camadas do modelo TCP/IP e do modelo OSI. Disponível em: <<http://www.clubedohardware.com.br/imageview.php?image=17520>>. Acesso em: 13 out. 2009.

TANENBAUM, Andrew Stuart. **Redes de Computadores**. 4. ed. Rio de Janeiro: Elsevier, 2003. 945 p.

DARPA (Virginia, Estados Unidos) (Org.). **RFC 791**: Internet Protocol, 1981. Disponível em: <<http://tools.ietf.org/html/rfc791>>. Acesso em: 07 out. 2009.

DARPA (Virginia, Estados Unidos) (Org.). **RFC 793**: Transmission Control Protocol, 1981. Disponível em: <<http://www.ietf.org/rfc/rfc793.txt>>. Acesso em: 07 out. 2009.

BERNERS-LEE, Tim; FIELDING, Roy T.; NIELSEN, Henrik Frystyk. **RFC 1945**: Hypertext Transfer Protocol, 1996. Disponível em: <<ftp://ftp.rfc-editor.org/in-notes/rfc1945.txt>>. Acesso em: 07 out. 2009.

WORLD WIDE WEB CONSORTIUM (Org.). **HTML & CSS**. Disponível em: <<http://www.w3.org/standards/webdesign/htmlcss>>. Acesso em: 17 out. 2009.

WORLD WIDE WEB CONSORTIUM (Org.). **HTML 5 Draft**. Disponível em: <<http://www.w3.org/TR/2008/WD-html5-20080122/>>. Acesso em: 20 out. 2009.

SILVA, Maurício Samy. **Construindo Sites com CSS e (X)HTML**: Sites Controlados por Folhas de Estilo em Cascata. São Paulo: Novatec, 2008. 446 p.

SHEA, Dave. **Zen Garden**: The Beauty of CSS Design. Disponível em: <<http://www.csszengarden.com/>>. Acesso em: 16 out. 2009.

OLIVEIRA, Rômulo Silva de; CARISSIMI, Alexandre da Silva; TOSCANI, Simão Sirineo. **Sistemas Operacionais**. 3. ed. Porto Alegre: Sagra Luzzatto, 2004. 274 p.

TANENBAUM, Andrew Stuart e STEEN, Maarten Van. **Sistemas Distribuídos**: Princípios e paradigmas. 2. ed. São Paulo: Pearson Prentice Hall, 2007. 402p.

SUN MICROSYSTEMS (California. Estados Unidos) (Org.). **Multithreaded Programming Guide**. Santa Clara: Sun, 2008. 270 p. Disponível em: <<http://dlc.sun.com/pdf/816-5137/816-5137.pdf>>. Acesso em: 08 out. 2009.

APÊNDICE A – Código fonte do navegador

```
===== PRINCIPAL =====  
  
package browser;  
  
public class Principal {  
  
    public static void main(String args[]) {  
        System.out.println("Programa iniciado.");  
        new FramePrincipal();  
    }  
  
}
```

```
=====
```

```

===== FRAMEPRINCIPAL =====
/*****
* Classe responsável pelo Frame Principal da interface do navegador. *
*****/

package browser;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Container;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTabbedPane;

public class FramePrincipal extends JFrame {

    private static final long serialVersionUID = 1L;
    private JMenuBar barraMenu;
    private JMenu menuPrincipal, menuAjuda;
    private JMenuItem abrirAba, sair, sobre;
    private JPanel painelPrincipal;
    private JTabbedPane painelAbas;

    /** Cria a interface do navegador e seus componentes. */
    public FramePrincipal() {
        montaMenu();
        montaTela();
        mostraBrowser();
    }

    /** Cria a estrutura de menus do navegador. */
    private void montaMenu() {
        barraMenu = new JMenuBar();
        menuPrincipal = new JMenu();
        menuAjuda = new JMenu();
        abrirAba = new JMenuItem();
        sair = new JMenuItem();
        sobre = new JMenuItem();

        menuPrincipal.setText("Principal");
        abrirAba.setText("Nova Aba");
        sair.setText("Sair");
        menuPrincipal.add(abrirAba);
        menuPrincipal.add(sair);

        menuAjuda.setText("Ajuda");
        sobre.setText("Sobre o Browser");
        menuAjuda.add(sobre);

        barraMenu.add(menuPrincipal);
        barraMenu.add(menuAjuda);

        abrirAba.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                novaAba("NovaAba");
            }
        });
    }
}

```

```

    }
});

sair.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
});

sobre.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Versão 0.9", "Sobre:",
JOptionPane.INFORMATION_MESSAGE);
    }
});

this.setJMenuBar(barraMenu);
}

/** Cria os componentes que formam a tela do navegador. */
private void montaTela() {
    Container corpoFrame = getContentPane();
    painelPrincipal = new JPanel();
    painelAbas = new JTabbedPane();

    novaAba("AbaInicial");
    painelPrincipal.setLayout(new BorderLayout());
    painelPrincipal.add(painelAbas);

    corpoFrame.add(painelPrincipal);
}

/** Cria uma nova aba no browser. */
private void novaAba(String titulo) {
    Color cor = painelAbas.getBackground();
    Aba aba = new Aba();
    JPanel painelAba;
    BotaoFecharAba botaoAba = new BotaoFecharAba(painelAbas, titulo, cor);
    int indiceAba;

    aba.start();
    painelAba = aba.getPainel();
    painelAbas.addTab(null, painelAba);
    indiceAba = painelAbas.indexOfComponent(painelAba);
    painelAbas.setTabComponentAt(indiceAba, botaoAba);
}

/** Seta a aparência do browser e torna-o visível. */
private void mostraBrowser() {
    painelPrincipal.setBackground(Color.DARK_GRAY);
    this.setSize(800, 600); //TODO Verificar se tem como setar MAX.
    this.setVisible(true);
}
}
}
=====

```

```

===== ABA =====
/*****
* Classe responsável pela estrutura das abas.
*****/

package browser;

import java.awt.BorderLayout;
import java.awt.Color;

import javax.swing.JEditorPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextField;

public class ABA extends Thread {

    private static final long serialVersionUID = -2848601238742726009L;

    private JScrollPane telaHTMLScroll;
    private JEditorPane telaHTML;
    private JTextField campoURL;
    private CarregaLink carregador;
    private BarraEndereco barraEndereco;
    private JPanel painel;

    public ABA() {
        painel = new JPanel(new BorderLayout());
        painel.setBackground(Color.CYAN);
    }

    public JPanel getPainel() {
        return painel;
    }

    public void run() {
        try {
            System.out.println("ABA iniciada");
            // Campo para entrada da URL.
            campoURL = new JTextField(40);

            // Monta tela de visualização da página HTML.
            telaHTML = new JEditorPane();
            telaHTML.setEditable(false);

            carregador = new CarregaLink(telaHTML, campoURL);
            telaHTML.addHyperlinkListener(carregador);
            telaHTMLScroll = new JScrollPane(telaHTML);

            // Campo e tela precisam ser inicializados antes de criar a classe na
            função.
            barraEndereco = new BarraEndereco(telaHTML, campoURL);

            painel.add(barraEndereco, "North");
            painel.add(telaHTMLScroll, "Center");
        } catch (Exception erro) {
            System.out.println(erro);
        }
    }
}
=====

```

```

===== BARRAENDEREÇO =====
/*****
* Classe responsável pela criação dos componentes que formam a barra de
* endereço da tela do navegador.
*****/

package browser;

import java.awt.FlowLayout;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JEditorPane;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class BarraEndereco extends JPanel {

    private static final long serialVersionUID = -6324350801813261701L;
    private JLabel label;
    private JButton botao;

    ActionListener carregaPaginas;

    /** As variáveis tela e campo são necessárias para a conf. das ações entre
     * elas e a barra de endereços. */
    public BarraEndereco(JEditorPane tela, JTextField campo) {
        super(new FlowLayout());
        label = new JLabel("URL:");
        //campo = new JTextField(40);
        botao = new JButton("GO");

        CarregaPagina loadPage = new CarregaPagina(tela, campo);
        botao.addActionListener(loadPage);

        add(label);
        add(campo);
        add(botao);
    }
}
=====

```

```

===== BOTAOFECHARABA =====
/*****
* Classe que representa um painel com o título e botão p/ fechar de uma aba.  *
*****/

package browser;

import java.awt.Color;
import java.awt.FlowLayout;

import javax.swing.JPanel;
import javax.swing.JTabbedPane;
import javax.swing.JLabel;

public class BotaoFecharAba extends JPanel {

    private static final long serialVersionUID = 946979934315545871L;

    private JTabbedPane painelAbas;
    private JLabel tituloAba;
    private BotaoX botaoX;

    public BotaoFecharAba(JTabbedPane painel, String titulo, Color corDeFundo) {
        super(new FlowLayout());
        painelAbas = painel;
        tituloAba = new JLabel(titulo);
        botaoX = new BotaoX(painelAbas, this);

        setBackground(corDeFundo);
        add(tituloAba);
        add(botaoX);
    }
}
=====

```

```

===== BOTAOX =====
/*****
 * Classe que cria um botão com forma de X, com a função de fechar uma Aba. *
 *****/

package browser;

import javax.swing.BorderFactory;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.Dimension;
import javax.swing.JTabbedPane;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.BasicStroke;
import java.awt.Color;

public class BotaoX extends JButton implements ActionListener {

    private static final long serialVersionUID = 177L;
    private JTabbedPane painelAba;
    private BotaoFecharAba painelBotao;

    /** Necessita do painel de Abas e do painel do botão para poder fechar
     * a aba que representa. */
    public BotaoX(JTabbedPane painel_aba, BotaoFecharAba painel_botao) {
        painelAba = painel_aba;
        painelBotao = painel_botao;
        setPreferredSize(new Dimension(17, 17));
        setToolTipText("Fecha esta aba.");
        setFocusable(false);
        setBorder(BorderFactory.createEtchedBorder());
        addActionListener(this);
    }

    /** Cria o desenho (X) do botão. */
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2 = (Graphics2D) g.create();

        // Muda a imagem quando o botão é pressionado.
        if (getModel().isPressed()) { g2.translate(1, 1); }
        g2.setStroke(new BasicStroke(2));
        g2.setColor(Color.BLACK);
        if (getModel().isRollover()) { g2.setColor(Color.MAGENTA); }
        int delta = 6;
        g2.drawLine(delta, delta, getWidth() - delta-1, getHeight() - delta-1);
        g2.drawLine(getWidth() - delta-1, delta, delta, getHeight() - delta-1);
        g2.dispose();
    }

    /** Adiciona a ação do botão (fechar a aba) quando ocorre um evento. */
    public void actionPerformed(ActionEvent e) {
        int index = painelAba.indexOfTabComponent(painelBotao);
        if (index != -1) { painelAba.remove(index); }
    }
}
=====

```

```

===== CARREGALINK =====
/*****
* Classe responsável pela implementação do evento de carregar uma página HTML *
* quando solicitado por outra página HTML. *
*****/

package browser;

import java.io.IOException;

import javax.swing.JEditorPane;
import javax.swing.JTextField;
import javax.swing.event.HyperlinkEvent;
import javax.swing.event.HyperlinkListener;

public class CarregaLink implements HyperlinkListener {

    private JEditorPane telaHTML;
    private JTextField campoURL;

    public CarregaLink(JEditorPane tela, JTextField campo) {
        telaHTML = tela;
        campoURL = campo;
    }

    public void hyperlinkUpdate(HyperlinkEvent event) {
        if(event.getEventType() == HyperlinkEvent.EventType.ACTIVATED) {
            try {
                //urlStack.push(event.getURL().toString());
                campoURL.setText(event.getURL().toString());
                telaHTML.setPage(event.getURL());
            } catch(IOException e) {
                telaHTML.setText("Error: " + e);
            }
        }
    }
}
=====

```

```

===== CARREGAPAGINA =====
/*****
* Classe responsável pela implementação do evento de carregar uma página HTML *
* quando solicitado pela entrada de uma URL no navegador. *
*****/

package browser;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JEditorPane;
import javax.swing.JTextField;

public class CarregaPagina implements ActionListener {

    private JEditorPane telaHTML;
    private JTextField campoURL;

    public CarregaPagina(JEditorPane tela, JTextField campo) {
        telaHTML = tela;
        campoURL = campo;
    }

    public void actionPerformed(ActionEvent event) {
        try {
            //urlStack.push(url.getText()); // remember URL for back button
            telaHTML.setPage(campoURL.getText());
        } catch (Exception e) {
            telaHTML.setText("Error: " + e);
        }
        System.out.println("Pedido de loadPage");
    }
}
=====

```

APÊNDICE B – Documento HTML de exemplo

```

<html>
<head>
  <title>Exemplo de página HTML</title>
  <style>body { text-align: center; }</style>
</head>
<body bgcolor="#000000" color="#FFFFFF">
  <h1>Página HTML de Exemplo</h1>
  <h2>Paragrafo 01</h2>
  <p>
    Exemplo exemplo exemplo exemplo exemplo exemplo exemplo exemplo.<br>
    Exemplo exemplo exemplo exemplo exemplo exemplo exemplo exemplo.
  </p>
  <h2>Paragrafo 02</h2>
  <p>
    Exemplo exemplo exemplo exemplo exemplo exemplo exemplo exemplo.<br>
    Exemplo exemplo exemplo exemplo exemplo exemplo exemplo.<br>
    Exemplo exemplo exemplo exemplo exemplo exemplo exemplo exemplo.<br>
    Exemplo exemplo exemplo exemplo exemplo exemplo exemplo exemplo.
  </p>
  <table border="1">
    <tr><th>Coluna 01</th><th>Coluna 02</th><th>Coluna 03</th></tr>
    <tr><td>exemplo</td><td>exemplo</td><td>exemplo</td></tr>
    <tr><td>exemplo</td><td>exemplo</td><td>exemplo</td></tr>
    <tr><td>exemplo</td><td>exemplo</td><td>exemplo</td></tr>
    <tr><td>exemplo</td><td>exemplo</td><td>exemplo</td></tr>
    <tr><td>exemplo</td><td>exemplo</td><td>exemplo</td></tr>
  </table>
  <h2>Paragrafo 03</h2>
  <p>
    Exemplo exemplo exemplo exemplo exemplo exemplo exemplo exemplo.<br>
    Exemplo exemplo exemplo exemplo exemplo exemplo exemplo exemplo.<br>
    Exemplo exemplo exemplo exemplo exemplo exemplo exemplo exemplo.
  </p>
  <table border="1">
    <tr><th>Coluna 01</th><th>Coluna 02</th></tr>
    <tr><td>exemplo</td><td>exemplo</td></tr>
    <tr><td>exemplo</td><td>exemplo</td></tr>
    <tr><td>exemplo</td><td>exemplo</td></tr>
    <tr><td>exemplo</td><td>exemplo</td></tr>
    <tr><td>exemplo</td><td>exemplo</td></tr>
  </table>
</body>
</html>

```

APÊNDICE C – Resultados dos testes realizados

Processador com dois núcleos:

Sem utilizar Threads:

Programa iniciado.

Aba iniciada 13ms

Aba iniciada 15ms

Aba iniciada 16ms

Aba iniciada 20ms

Aba iniciada 15ms

Aba iniciada 11ms

Aba iniciada 19ms

Aba iniciada 11ms

Aba iniciada 10ms

Aba iniciada 11ms

Utilizando Threads:

Programa iniciado.

Aba iniciada 13ms

Aba iniciada 9ms

Aba iniciada 7ms

Aba iniciada 6ms

Aba iniciada 4ms

Aba iniciada 24ms

Aba iniciada 12ms

Aba iniciada 5ms

Aba iniciada 12ms

Processador com um núcleo:

Sem utilizar Threads:

Programa iniciado.

Aba iniciada 140ms

Aba iniciada 16ms

Aba iniciada 16ms

Aba iniciada 15ms

Aba iniciada 16ms

Aba iniciada 16ms

Aba iniciada 15ms

Aba iniciada 16ms

Aba iniciada 16ms

Aba iniciada 16ms

Aba iniciada 15ms

Utilizando Threads:

Programa iniciado.

Aba iniciada 31ms

Aba iniciada 16ms

Aba iniciada 15ms

Aba iniciada 17ms

Aba iniciada 16ms

Aba iniciada 15ms

Aba iniciada 16ms

Aba iniciada 17ms

Aba iniciada 16ms

Comparação com o Internet Explorer 6:

Internet Explorer:

1 janela 26600k

2 janelas 27600k

3 janelas 28000k

4 janelas 29000k

Navegador multithread:

1 janela 33000k

2 janelas 38000k

3 janelas 43500k

4 janelas 48300k

APÊNDICE D – Artigo

DESENVOLVIMENTO DE UM WEB-BROWSER COM SISTEMA DE ABAS MULTITHREAD

Rodrigo Fagundes

Departamento de Informática e Estatística – Universidade Federal de Santa Catarina
Caixa Postal 476 – CEP 88040-900 – Florianópolis, SC – Brasil

rfag@inf.ufsc.br

Abstract. *The present work deals with the development of a web-browser, also called navegador, that uses multiple threads to the operation of its tab's system. At first was focus the development of the basic features of the browser, such as the exchange messages with a server over the HTTP protocol to be able to access documents contained in it, and the interpretation of HTML documents. Then the tab's system was developed, allowing the users to open multiple Web pages without the need to start a new instance of the browser. Finally it included the threads to achieve the advantages of a multithread system.*

Resumo. *O presente trabalho trata do desenvolvimento de um web-browser, também chamado de navegador, que utiliza múltiplas threads para o funcionamento de seu sistema de abas. Primeiramente foi focado o desenvolvimento das funcionalidades básicas do navegador, como a troca de mensagens com um servidor pelo protocolo HTTP para ser possível acessar documentos contidos nele, e a interpretação de documentos HTML. Em seguida foi desenvolvido o sistema de abas, permitindo ao usuário abrir várias páginas da Internet sem a necessidade de iniciar uma nova instância do navegador. Por último foi incluído as threads para alcançar as vantagens de um sistema multithreads.*

1 INTRODUÇÃO

Atualmente o uso da Internet é algo corriqueiro para a maior parte das pessoas, e por consequência, o uso de navegadores também é algo comum. Um navegador, ou web-browser, é um aplicativo que permite a visualização de documentos HTML, tais documentos podem estar em servidores na Internet, ou na própria máquina do usuário. Com sua utilização é possível aproveitar ao máximo as possibilidades da Internet, tal qual a troca de informação e a comunicação entre usuários.

Threads dividem as tarefas de um programa, o que pode gerar um aplicativo mais eficiente e que permite a execução de tarefas diferentes simultaneamente, quando há mais de um processador disponível, ou alternadamente, quando há somente um processador disponível. Isso permite que um programa possa executar outras tarefas sem que fique esperando um longo tempo em uma tarefa que demande mais tempo.

Neste trabalho busquei desenvolver um navegador que utilizasse múltiplas threads no seu sistema de abas para tentar obter um aplicativo com desempenho satisfatório, usufruindo das

vantagens de um sistema multithread.

2 HISTÓRIA DA INTERNET

2.1. ORIGEM DA INTERNET

A Internet teve origem a partir de um projeto militar dos Estados Unidos denominado ARPANET. A ARPANET foi desenvolvida em 1969 pela Defense Advanced Research Projects Agency (DARPA), objetivando criar um sistema de comunicação que continuasse funcionando mesmo que fosse parcialmente danificado.

Inicialmente a ARPANET interligava quatro instituições: Universidade da Califórnia de Los Angeles, Universidade da Califórnia de Santa Bárbara, Instituto de Pesquisa de Stanford, e Universidade de Utah. Com isso era possível a troca de informações através do programa SNDMSG, desenvolvido por Ray Tomlinson. Este programa introduziu o caractere @ para separar o nome do usuário do nome da máquina, semelhante ao sistema de correio eletrônico (e-mail) utilizado atualmente. Com o tempo outras instituições tiveram permissão de conectar suas redes internas a ARPANET e ela passa a ser chamada de a rede das redes.

O termo Internet surge em 1974, na primeira especificação completa do Transmission Control Protocol (TCP). Este protocolo substitui o Network Control Protocol (NCP), usado até então pela ARPANET. Durante a década de 80 o número de computadores ligados a ARPANET cresceu rapidamente, incluindo também máquinas de instituições de outros países além dos Estados Unidos. A Internet passa a ser uma rede maior, que engloba a ARPANET.

Na década de 90 a Internet foi liberada para propósitos comerciais. Empresas passam a vender serviços como o e-mail e a transferência de arquivos pelo File Transfer Protocol (FTP). A utilização da Internet aumenta ainda mais nos anos seguintes com a criação do primeiro navegador chamado de WorldWideWeb, renomeado posteriormente para Nexus a fim de evitar confusão com a World Wide Web (Web).

A World Wide Web, que pode ser traduzido para rede de alcance mundial, é um rede de documentos hipertexto interligados que podem ser acessados pela Internet através de um navegador. Ela foi proposta por Tim Bernes-Lee e Robert Cailliau em 1990. Nesta proposta foi definido a arquitetura da Web, qual seria o papel de um servidor e de um navegador e também metas para o desenvolvimento do projeto (BERNERS-LEE; CAILLIAU, 1990).

Atualmente existem mais de 1,5 bilhões de pessoas em todo mundo que utilizam a Internet, segundo o relatório anual da International Telecommunication Union (2008).

2.2 HISTÓRIA DOS NAVEGADORES

Web-Browsers ou navegadores são ferramentas que permitem a visualização de documentos HTML, que podem estar em servidores na Internet, ou na própria máquina do usuário.

Segundo Berners-Lee (2009), durante o início da década de 1990 o único meio de se acessar documentos na Web era usando o navegador desenvolvido por ele, primeiramente chamado de

WorldWideWeb e posteriormente renomeado para Nexus.

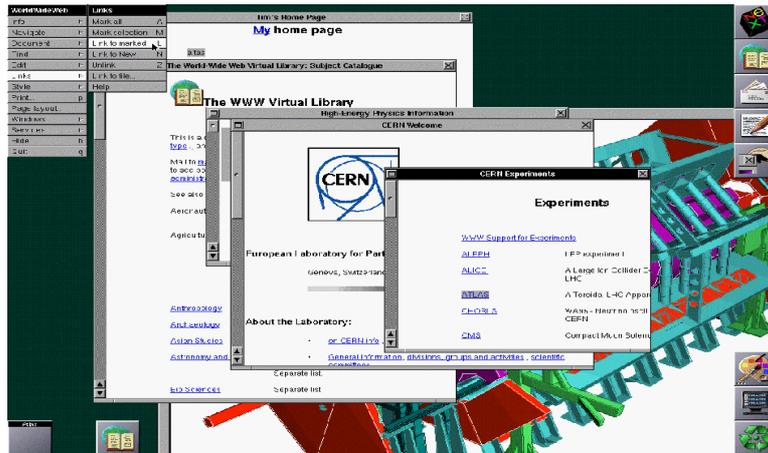


Figura 1: Navegador Nexus (BERNERS-LEE, 2009).

Em 1993 o código fonte do Nexus é liberado como de domínio público, permitindo assim que qualquer um possa usar o navegador. No mesmo ano, Marc Andreessen e Eric Bina, na época no grupo de software design da National Center for Supercomputing Applications (NCSA), desenvolvem o Mosaic, um navegador para sistemas Unix e posteriormente para Windows e Apple Macintosh, sendo o primeiro navegador para diferentes plataformas. O Mosaic foi o navegador mais popular de sua época.

Posteriormente Marc funda a empresa Mosaic Communications, juntamente com Jim Clark. Mais tarde a empresa é renomeada para Netscape Communications Corporation. No final de 1994 a Netscape lança um navegador com codinome Mozilla, o primeiro navegador a ser comercializado. Posteriormente o nome do navegador muda para Netscape Navigator. Com a inclusão de novas funcionalidades e integração com outros serviços como o e-mail, fez com que o navegador se popularizar rapidamente.

Devido a popularidade do navegador da Netscape e do crescente uso da Web, a Microsoft Corporation resolve lançar seu próprio navegador, comprando da Spyglass Inc., divisão comercial da NCSA, a tecnologia do Mosaic. Tal tecnologia serve de base para o navegador Internet Explorer lançado em Agosto de 1995. Com isso tem início a chamada Guerra dos Browsers, período marcado pela luta pelo domínio do mercado de navegadores entre a Netscape e a Microsoft.

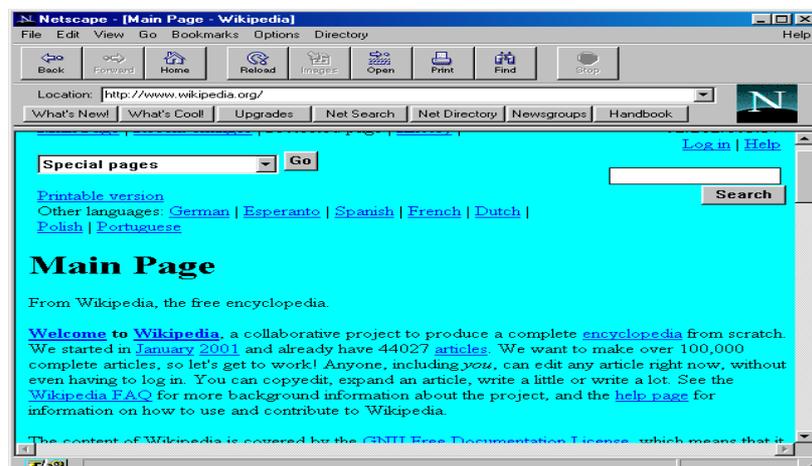


Figura 2: Netscape Navigator (WIKIPEDIA, 2009).

A Guerra dos Browsers ocorre durante o período de 1995 a 2002, com os navegadores Netscape Navigator e Internet Explorer competindo pelo mercado. Inicialmente a Netscape tinha o domínio do mercado, com mais de 90% dos usuários utilizando o Netscape Navigator. No final deste período a Microsoft conseguiu reverter a situação, com 95% dos usuários utilizando o Internet Explorer.

O Internet Explorer teve algumas vantagens em relação ao seu concorrente:

- Era gratuito, enquanto o Netscape cobrava pelo software;
- Teve maior suporte financeiro;
- A partir da versão 4, lançada em 1997, vem integrado ao sistema operacional Microsoft Windows.

Devido a essa disputa, cada empresa começa a criar funcionalidades próprias no interpretador HTML e posteriormente no interpretador Javascript. Isso fez com que os documentos HTML fossem visualizadas de maneira diferente nos navegadores. Em 1998 com o lançamento da quinta versão do Internet Explorer havia cinco interpretadores diferentes de código JavaScript em uso pelas diferentes versões de navegadores disponíveis. Isto significa que fazer uma página que fosse acessível a grande parte dos usuários da Web era impossível.

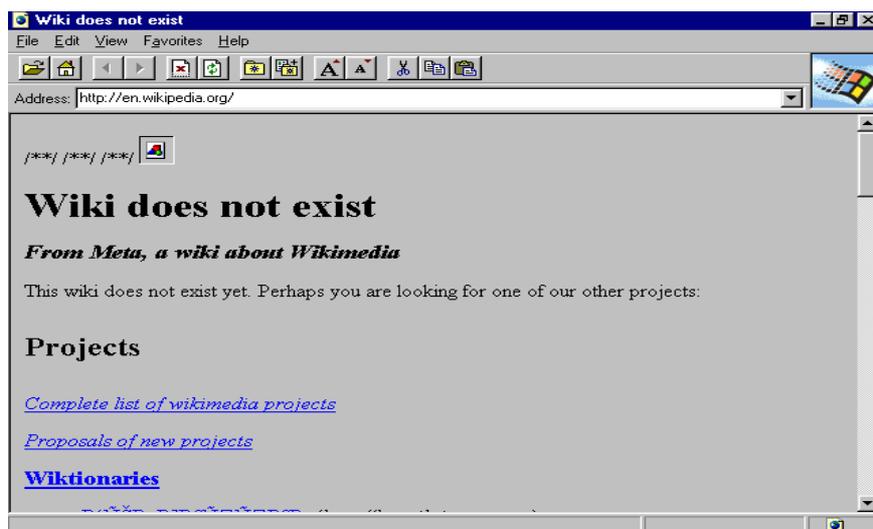


Figura 3: Internet Explorer (WIKIPEDIA, 2009).

De 1998 até 2002 o Internet Explorer foi ganhando mercado rapidamente. Com isso a guerra dos navegadores acaba, e a Microsoft diminui os investimentos no seu navegador. Isso só vai mudar com o início da segunda guerra dos browsers.

Visando organizar a Web, é fundado em 1994 por Berners-Lee o World Wide Web Consortium (W3C). Para acabar com a incompatibilidade de código HTML entre navegadores, é proposto pela W3C diversas recomendações para padronizar o HTML, as folhas de estilo e outras tecnologias usadas na Internet.

Após perder a disputa pelo mercado de navegadores, a Netscape resolve licenciar o código fonte do Netscape Navigator como de código aberto, antes de ser vendida para a empresa America Online. Com base neste código fonte, surge o projeto Mozilla e o navegador de mesmo nome.

Em 2002, um pequeno grupo de desenvolvedores do navegador Mozilla, resolvem criar um navegador menos dependente da Netscape e dão origem ao navegador Mozilla Firefox. Com diversas inovações este navegador ganha espaço entre os usuários da Internet.

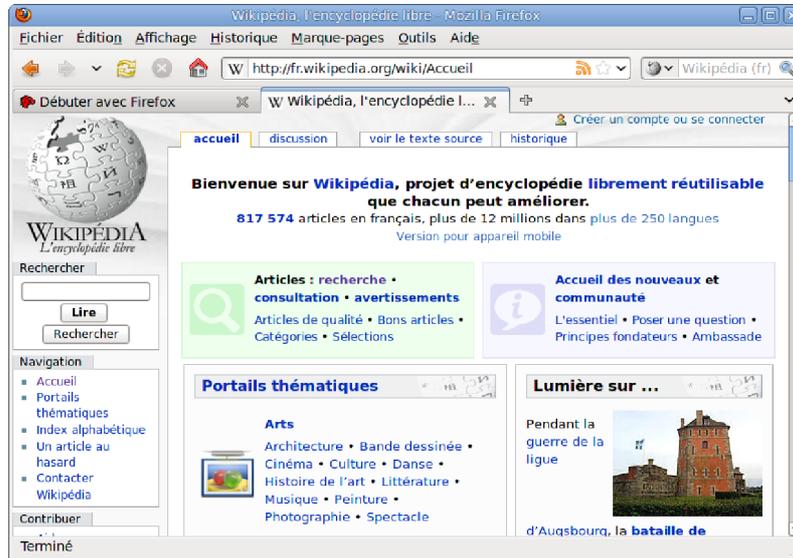


Figura 4: Mozilla Firefox (WIKIPEDIA, 2009).

Com a perda de usuários pela primeira vez desde que dominou o mercado, a Microsoft volta a investir no seu navegador, e depois de cinco anos sem ter uma nova versão, é lançado Internet Explorer 7.

Atualmente a disputa pelo domínio deste mercado continua. Segundo a empresa Net Applications (2009), o Internet Explorer mantém-se dominando o mercado com 65,71% dos usuários, enquanto o Firefox tem 23,75%.

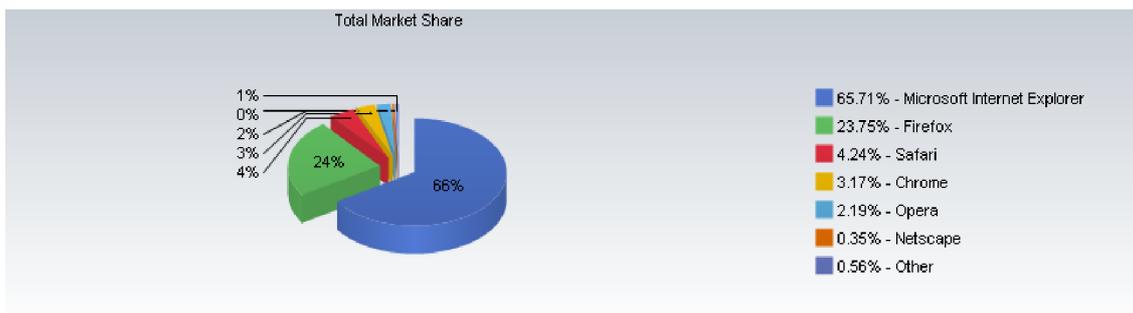


Figura 5: Mercado de navegadores (NET APPLICATION, 2009).

Outros navegadores também tentam ganhar parte do mercado à algum tempo. A empresa Apple investiu na criação de seu próprio navegador, o Safari, lançado em 2003 para o sistema operacional Mac OS X e posteriormente para o Windows.

A empresa norueguesa Opera Software ASA criou o navegador Opera em 1995. Inicialmente apenas para o sistema Windows, em 1998 a empresa começou a explorar novos mercados para seu navegador, como o uso dele em vídeo-games e dispositivos portáteis para acessar a Web.

A última empresa a entrar nesse mercado foi a Google em 2008. Apostando na popularidade dos serviços da empresa, como seu sistema de buscas e o serviço de armazenamento de vídeos, para motivar os usuários a usarem seu navegador, o Google Chrome.

Atualmente a importância dos navegadores tem aumentado, devido a novos serviços disponíveis na Web.

3 FUNCIONAMENTO DA INTERNET

3.1 ESTRUTURA DA INTERNET

Para se conectar a Internet um computador, ou rede local, deve se conectar a um Provedor de Serviços da Internet (ISP). A ISP por sua vez se conecta a ISP maiores que estão conectadas a outras, permitindo assim a navegação na Internet.

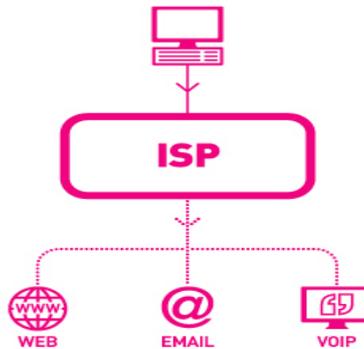


Figura 6: Provedor de serviços da Internet (EXCELL GROUP, 2009).

Para troca de dados e acesso a arquivos e informações contidas na Internet é usado uma série de protocolos que especificam como deve ser feita essa comunicação entre as máquinas. Através da família de protocolos TCP/IP, é feito todo trabalho para o estabelecimento da comunicação entre as máquinas cliente e servidor, e permitindo assim que dados sejam transmitidos.

A Internet usa uma estrutura de rede semelhante ao modelo Open Systems Interconnection (OSI), chamado de modelo TCP/IP. A diferença entre estes modelos é que a camada de Aplicação do modelo da Internet engloba as camadas de Apresentação, Aplicação e Sessão. Conforme Tanenbaum (2003, p.46), o modelo TCP/IP não viu necessidade de incluir estas camadas, pois são pouco usadas na maioria das aplicações. Esta camada utiliza diversos protocolos. Já a camada de Transportes utiliza os protocolos TCP (Transmission Control Protocol) e UDP (User Datagram Protocol), enquanto a camada de rede utiliza o protocolo IP. As camadas Física e de Enlace do modelo OSI estão englobadas sem distinção na camada de Interface com a Rede do modelo TCP/IP. Uma comparação entre as camadas dos modelos OSI e TCP/IP pode ser visualizada abaixo:

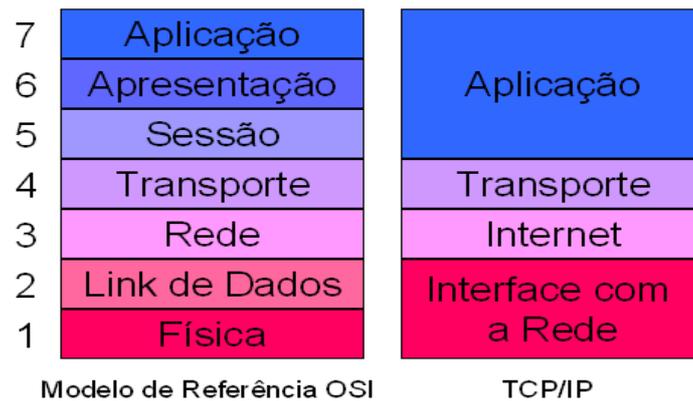


Figura 7: Comparação dos modelos OSI e TCP/IP (TORRES, 2009).

3.1.1 Camada de Interface com a rede

Segundo Tanenbaum (2003, p.47), o modelo TCP/IP não especifica bem esta camada, sendo que o único fato definido é que um “host deve se conectar à rede utilizando algum protocolo”. Também não é claramente definido nenhum protocolo desta camada pelo modelo TCP/IP.

3.1.2 Camada de Internet

Tanenbaum (2003, p.45) diz que, no modelo TCP/IP, “a camada internet define um formato de pacote oficial e um protocolo chamado IP. A tarefa da camada internet é entregar pacotes IP onde eles são necessários”. Segundo DARPA, (1981, tradução minha) o protocolo IP (Internet Protocol) deve “prover as funções necessárias para a entrega de pacotes [...] de uma fonte a um destino através de um sistema de redes interconectadas.” Também é dito que este protocolo deve servir como interface entre o protocolo de aplicação e o protocolo da rede local.

3.1.3 Camada de Transporte

A camada de transporte do modelo TCP/IP tem como finalidade “permitir que as entidades pares de hosts de origem e de destino mantenham uma conversação, exatamente como acontece na camada de transporte OSI” Tanenbaum (2003, p.46). Esta camada tem dois protocolos, o TCP e o UDP, sendo o primeiro um protocolo orientado a conexões confiável e o segundo não confiável. Isto significa que o TCP é um protocolo que entregará os dados sem erros, enquanto o UDP não garante isso. Na especificação do TCP foi definido que o protocolo “tem como objetivo prover um serviço de comunicação de confiança entre processos finais em um ambiente com redes diversas” (DARPA, 1981, tradução minha). este protocolo então garante a entrega dos pacotes, verifica se eles estão sendo enviados de forma correta, na sequência apropriada e sem erros. Usualmente um navegador utiliza, através do protocolo HTTP, o protocolo TCP.

3.1.4 Camada de Aplicação

A camada de Aplicação do modelo TCP/IP, é a camada utilizada pelos aplicativos para se comunicar pela rede com outros programas. Os protocolos mais usados na Internet são o HTTP e o SMTP, sendo o protocolo SMTP usado para a o envio de e-mails. Já o protocolo HTTP é “um protocolo do nível de aplicação com a leveza e velocidade necessária para ser utilizado com

sistemas de informação de hipermídia distribuídos e colaborativos" (BERNERS-LEE; FIELDING; NIELSEN, 1996, tradução minha). Ele é usado para realizar requisições de uma máquina cliente a um servidor e assim o cliente pode acessar arquivos e dados no servidor, como páginas HTML. Outros protocolos desta camada incluem FTP (File Transfer Protocol), SSH (Security Shell), DNS (Domain Name System), TELNET (Teletype network), etc.

3.2 ESTRUTURA DE UM DOCUMENTO DA WEB

Um documento, ou página, da Web é um documento disponível na Internet, normalmente estruturado em HTML. Um ou mais destes documentos formam um web site, ou simplesmente site. Web site pode ser traduzido para o português como local na teia, com teia significando rede.

3.2.1 HTML

O HyperText Markup Language (HTML), que pode ser traduzido para Linguagem de Marcação de Hipertexto, é “a linguagem para descrever a estrutura dos documentos Web” (WORLD WIDE WEB CONSORTIUM, 2009a). O código HTML é interpretado pelo navegador que monta página para ser visualizada pelo usuário. Criada por Tim Berners-Lee para facilitar a comunicação entre ele e seus colegas, atualmente a sua especificação é mantida pela W3C. Antes da criação das folhas de estilo em cascata, o HTML também era responsável pela formatação da página.

Exemplo de documento HTML:

```

01 <!DOCTYPE html>
02 <html>
03 <head>
04 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
05 <title>Título da página</title>
06 </head>
07 <body>
08 <!-- Comentário... -->
09 Conteúdo da página.
10 </body>
11 </html>

```

Figura 8: Exemplo de código HTML.

Atualmente o HTML está na versão 4. Este deveria ser substituído pelo HTML estendido (Extensible HyperText Markup Language – XHTML). Pela definição da W3C (WORLD WIDE WEB CONSORTIUM, 2009a)

“XHTML é um variante do HTML que usa a sintaxe do XML, a Linguagem de Marcação Estendida. XHTML tem os mesmos elementos (para parágrafos, etc.) que o HTML, mas a sintaxe é levemente diferente. Devido a XHTML ser uma aplicação XML, pode-se usar outras ferramentas XML com ela [...]”.

Entretanto, em 2004, Apple, Mozilla e Opera criam o Web Hypertext Application Technology Working Group (WHATWG), pois estavam descontentes com algumas propostas do W3C, especialmente a falta de preocupação com as necessidades dos desenvolvedores Web. A WHATWG começou a trabalhar no HTML 5, com a intenção de adequar o HTML às necessidades reais dos desenvolvedores. Eventualmente a W3C aceitou discutir o HTML 5 e postergou o desenvolvimento da segunda versão do XHTML. Em Janeiro de 2009 a W3C publicou a primeira especificação da quinta versão do HTML, ainda em versão de testes. Segundo a especificação

(WORLD WIDE WEB CONSORTIUM, 2009b) o HTML 5 substituirá tanto o HTML 4 quanto o XHTML 1.

3.2.2 Folhas de Estilo

As folhas de estilo em cascata (Cascade Style Sheet - CSS), servem para “descrever a apresentação dos documentos Web, incluindo cores, leiaute, e informações da fonte, assim como alterar a apresentação em diferentes dispositivos como aqueles com grandes telas, pequenas telas ou impressoras” (WORLD WIDE WEB CONSORTIUM, 2009a). Uma folha de estilo pode ser escritas dentro do documento HTML ou em um documento externo referenciado pelo documento HTML. A grande inovação das folhas de estilo é que permitem definir a aparência de diversas páginas em um só documento.

Exemplo de documento de folha de estilo:

```
01 @CHARSET "UTF-8";
02
03 body {
04  /* Comentário... */
05  background-color: black;
06  color: white;
07  font-size: 20px;
08  text-align: center;
09 }
```

Figura 9: Exemplo de código CSS.

Silva (2008, p. 418) apresenta algumas vantagens obtidas com o uso de folhas de estilo como o controle total sobre a formatação de um site em um único arquivo central, a agilização da manutenção e reestruturação de um site, e a redução do tempo de carga dos documentos Web.

Para demonstrar a versatilidade das folhas de estilo, Dave Shea criou em 2003 o ZenGarden. Este site é composto de apenas um documento HTML com diversas folhas de estilo que modificam totalmente a aparência da página. Esta página pode ser conferida em (SHEA, 2009).

3.2.3 JavaScript

Para possibilitar maiores recursos para os documentos HTML, foi criado a linguagem de script de programação JavaScript. Assim como as folhas de estilo, o JavaScript pode ser escrito dentro do documento HTML ou em um documento externo referenciado pelo documento HTML.

Exemplo de documento de JavaScript:

```
01 var x = 7;
02 var y = x * x;
03
04 function uma_funcao(z) {
05  // Comentário...
06  variavel = (x + y) / z;
07  if (variavel > 1) {
08    alert("exemplo...");
09  }
10  document.write("Variavel = " + variavel);
11 }
```

Figura 10: Exemplo de código JavaScript.

O JavaScript foi criado numa parceria entre a Netscape e a Sun Microsystems em 1995, tendo sintaxe semelhante a linguagem de programação Java. Em conjunto com as folhas de estilo permitem criar páginas HTML dinâmicas, podendo, através de uma ação do usuário, alterar a apresentação da página, validar dados, executar operações matemáticas, ou mesmo carregar toda uma nova página.

4 THREADS

Oliveira, Carissimi e Toscani (2004, p. 65) definem thread como "um fluxo de execução" de um processo. Já o processo é definido por Tanenbaum e Steen (2007, p.42) como "um programa em execução".

Um sistema Multithreading "[...] separa um processo em várias threads. Cada uma dessas threads é executada independentemente." (SUN MICROSYSTEMS, 2008, p. 17, tradução minha). Isso acarreta em vantagens e desvantagens em comparação a um sistema monothread.

Tanenbaum e Steen (2007, p. 43) citam como principal vantagem o "[...] fato de que, em um processo monothread, sempre que for executada uma chamada bloqueadora de sistema, o processo é bloqueado como um todo", enquanto que um processo multithread teria apenas uma de suas threads bloqueadas.

Por exemplo, em um aplicativo de planilhas, se este aplicativo for monothread quando o usuário entra com alguma alteração num campo que acarretará em cálculos em outros campos o usuário terá que esperar pela conclusão dos cálculos para poder realizar outra atividade. Já se o aplicativo for multithread enquanto uma thread se encarrega de realizar os cálculos, outra poderia permitir que o usuário realizasse uma outra. atividade sem a necessidade de esperar pela conclusão da primeira thread.

A figura abaixo, mostra como é representado um processo monothread, e um processo multithread.

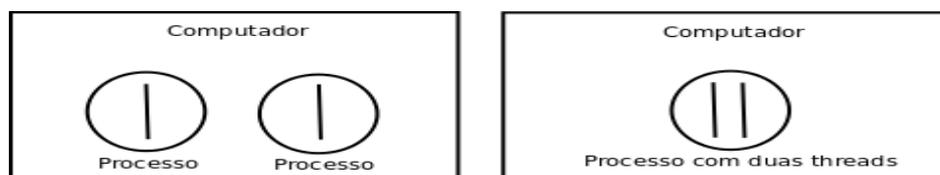


Figura 11: Exemplo de processo monothread e multithread.

5 DESENVOLVIMENTO

O navegador foi desenvolvido utilizando-se a linguagem de programação Java e sua biblioteca de classes para implementar algumas das funcionalidades do navegador.

O ambiente montado para o desenvolvimento deste software consiste em:

- Sistema Operacional Ubuntu Hardy Heron (8.04)

- Ambiente de Desenvolvimento Integrado Eclipse Europa (3.3)
- Java Development Kit 6

Primeiramente foi construído um protótipo da interface do programa, utilizando-se dos componentes da biblioteca SWING. Na figura abaixo está a estrutura usada para criar o protótipo.

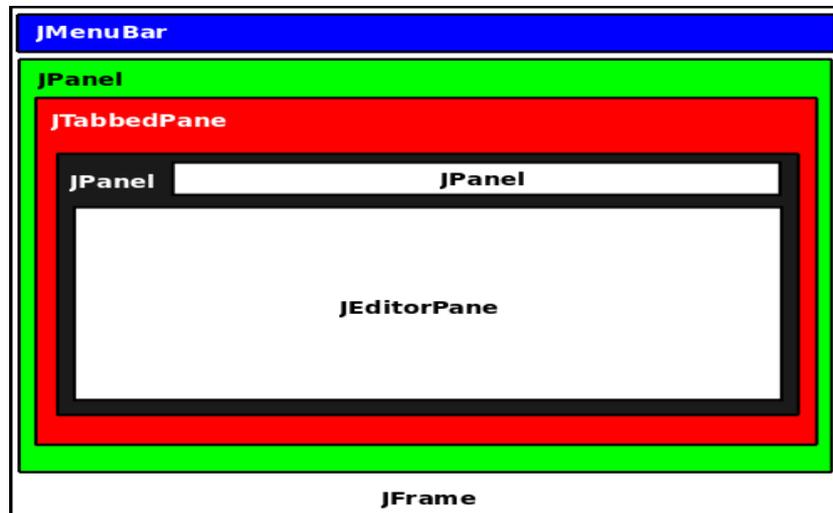


Figura 12: Estrutura da interface do navegador.

Com base nessa estrutura foram criadas as classes que compõem o navegador. O diagrama de classes abaixo demonstra as classes implementadas.

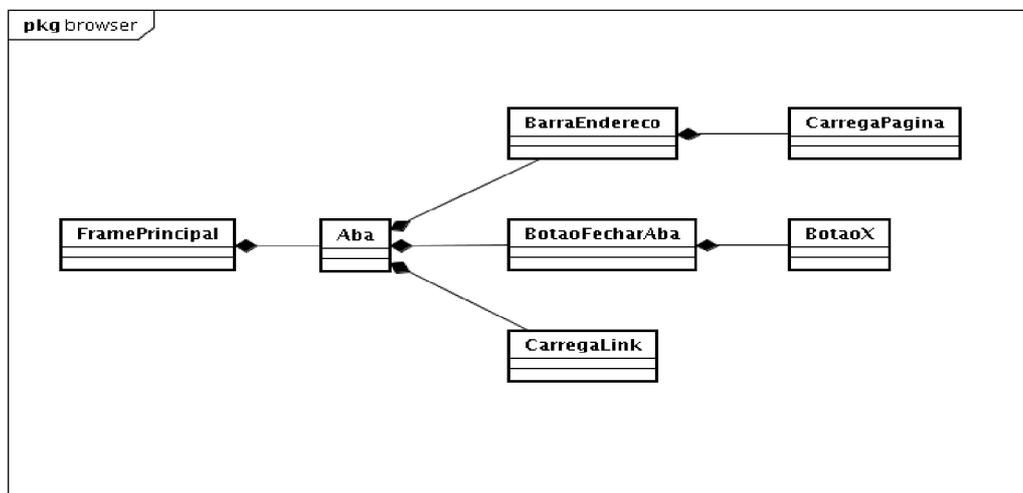


Figura 13: Diagrama de classes simplificado.

A classe *FramePrincipal* quando instanciada cria o menu, o painel que conterá as abas e uma aba inicial. Através do método *novaAba*, esta classe gera uma nova aba e a adiciona ao painel sempre que solicitado. Uma aba é representada pela classe *Aba* que contém a estrutura mínima para ser possível a abertura de uma página da Internet.

Todo o trabalho de comunicação entre o navegador e os servidores da Internet é feita pela classe *JEditorPane*, pertencente a biblioteca SWING. Essa mesma classe se encarrega de

interpretação das página HTML. Foi criada duas classes que utilizam os métodos da classe *JEditorPane*, *CarregaPagina* e *CarregaLink*. Tais classes utilizam um controle de eventos para saber quando é feito a solicitação de uma página por parte do usuário.

A classe *Aba* é filha da classe *Thread*, sendo assim, cada aba criada representa uma thread do navegador. Isso torna cada aba passível do chaveamento de threads dentro do processo do navegador. Quando a aba é fechada a thread é destruída. Cada aba tem sua barra de endereços para receber uma url de destino, e uma tela para mostrar a página HTML, esta tela sendo um *JEditorPane*.

Abaixo, a figura 14 mostra o navegador funcionando, com duas abas abertas, sendo que a aba selecionada mostra uma página de exemplo criada para executar testes com o navegador. Esta página HTML de exemplo pode ser consultada no apêndice B.



Figura 14: Tela do Navegador.

5.1 TESTES COMPARATIVOS

5.1.1 Comparação entre navegador usando threads e sem usa-las

Foram realizados testes em dois computadores, um com processador de dois núcleos e outro com processador de um núcleo. Foi comparado o tempo de criação de uma aba entre o navegador implementado e uma adaptação do mesmo que não utiliza threads. Sabe-se que processadores com mais de um núcleo tendem a aproveitar melhor um aplicativo multithread, então é esperado que o navegador com threads leve vantagem no processador de dois núcleos. Os testes realizados comprovaram isso, mostrando que na média a criação e execução de uma nova aba tende a ser mais rápida no navegador multithread quando está trabalhando com o processador de dois núcleos. Já no processador de um núcleo, o desempenho foi semelhante ao navegador sem múltiplas threads.

5.1.2 Comparação entre o navegador e um navegador monothread comercial

Foram realizados testes para comparar o desempenho do navegador com um navegador monothread comercial. Infelizmente o único navegador asseguradamente monothread encontrado foi o Internet Explorer 6. Infelizmente porque este navegador, sendo nativo do sistema operacional Windows, utiliza artefatos do próprio sistema para ter uma execução otimizada, sendo assim não é um navegador adequado para comparar com o navegador criado, já que este não pode utilizar os mesmos artefatos.

Nestes testes notou-se que quando abria-se uma nova janela do navegador monothread, a memória consumida aumenta em 1000k na média, já o navegador multithread consumiu 5000k cada vez que uma nova aba foi utilizada com a mesma url. É possível que isto se deva ao fato do navegador Internet Explorer guardar em cache os dados da página acessada, enquanto que o navegador multithread não implementa esta funcionalidade e necessita buscar do servidor os dados a cada solicitação. Apesar do consumo de memória ser maior em comparação com o navegador monothread, o uso da Web foi melhor no navegador multithread, sendo mais rápido a abertura de uma nova janela.

5.1.2 Comparação de uso com outros navegadores

Comparando a utilização do navegador desenvolvido com outros navegadores comerciais, notou-se um desempenho semelhante. As principais diferenças notadas são devidas a outros recursos incluídos nos navegadores comerciais. Por ser escrito em java, o navegador aqui desenvolvido tem a vantagem de funcionar em qualquer computador com uma máquina virtual java.

6 CONCLUSÃO

Com o presente trabalho desenvolveu-se um navegador que consegue solicitar documentos da Web de servidores da Internet e que consegue interpretar satisfatoriamente o seu HTML, mostrando a página aos usuário. Também se introduziu um sistema de abas no navegador e Threads neste sistema. Com isso conseguiu-se obter as vantagens que um sistema de múltiplas Threads trás.

A maior vantagem obtida com o uso de multithreads no sistema de abas é que não é necessário esperar a conclusão de uma solicitação de uma página HTML por parte do navegador para fazer outra solicitação, ou visualizar o resultado da interpretação do código de outra aba. Com o uso de múltiplas threads é possível que diversas abas solicitem páginas da Web sem bloquear o uso do navegador.

Entretanto, poderia ser ainda melhor o aproveitamento das threads pelo navegador, explorando ainda mais suas vantagens, como a implementação de um interpretador HTML e CSS multithread, um interpretador JavaScript multithread e outras funcionalidades que possam tirar proveito das múltiplas threads.

REFERÊNCIAS

BERNERS-LEE, Tim; CAILLIAU, Robert. **WorldWideWeb**: Proposal for a HyperText Project.
Disponível em: <<http://www.w3.org/Proposal.html>>. Acesso em: 02 out. 2009.

INTERNATIONAL TELECOMMUNICATION UNION. **Trends in Telecommunication Reform**

2008: Six Degrees of Sharing. Disponível em:

<http://www.itu.int/newsroom/press_releases/2008/35.html>. Acesso em: 02 out. 2009.

BERNERS-LEE, Tim. **The WorldWideWeb browser.** Disponível em:

<<http://www.w3.org/People/Berners-Lee/WorldWideWeb.html>>. Acesso em: 02 out. 2009.

WIKIPEDIA. **Wikipedia.** Fonte das imagens dos navegadores Netscape, Internet Explorer e Firefox. Disponível em: <<http://www.wikipedia.org/>>. Acesso em: 13 out. 2009.

NET APPLICATIONS. **Browser Market Share.** Disponível em:

<<http://marketshare.hitslink.com/report.aspx?qprid=0>>. Acesso em: 13 out. 2009.

EXCELL GROUP. **Internet Service Provider.** Fonte da imagem ISP. Disponível em:

<<http://www.excellgroup.com/generated/0001/000151/v003/transform10/cache186/ISP.jpg>>. Acesso em: 13 out. 2009.

TORRES, Gabriel; LIMA, Cássio. **Comparação entre TCP/IP e OSI.** Fonte da imagem de comparação entre as camadas do modelo TCP/IP e do modelo OSI. Disponível em:

<<http://www.clubedohardware.com.br/imageview.php?image=17520>>. Acesso em: 13 out. 2009.

TANENBAUM, Andrew Stuart. **Redes de Computadores.** 4. ed. Rio de Janeiro: Elsevier, 2003. 945 p.

DARPA (Virginia, Estados Unidos) (Org.). **RFC 791:** Internet Protocol, 1981. Disponível em:

<<http://tools.ietf.org/html/rfc791>>. Acesso em: 07 out. 2009.

DARPA (Virginia, Estados Unidos) (Org.). **RFC 793:** Transmission Control Protocol, 1981.

Disponível em: <<http://www.ietf.org/rfc/rfc793.txt>>. Acesso em: 07 out. 2009.

BERNERS-LEE, Tim; FIELDING, Roy T.; NIELSEN, Henrik Frystyk. **RFC 1945:** Hypertext

Transfer Protocol, 1996. Disponível em: <<ftp://ftp.rfc-editor.org/in-notes/rfc1945.txt>>. Acesso em: 07 out. 2009.

WORLD WIDE WEB CONSORTIUM (Org.). **HTML & CSS.** Disponível em:

<<http://www.w3.org/standards/webdesign/htmlcss>>. Acesso em: 17 out. 2009.

WORLD WIDE WEB CONSORTIUM (Org.). **HTML 5 Draft.** Disponível em:

<<http://www.w3.org/TR/2008/WD-html5-20080122/>>. Acesso em: 20 out. 2009.

SILVA, Maurício Samy. **Construindo Sites com CSS e (X)HTML:** Sites Controlados por Folhas de Estilo em Cascata. São Paulo: Novatec, 2008. 446 p.

SHEA, Dave. **Zen Garden:** The Beauty of CSS Design. Disponível em:

<<http://www.csszengarden.com/>>. Acesso em: 16 out. 2009.

OLIVEIRA, Rômulo Silva de; CARISSIMI, Alexandre da Silva; TOSCANI, Simão Sirineo.

Sistemas Operacionais. 3. ed. Porto Alegre: Sagra Luzzatto, 2004. 274 p.

TANENBAUM, Andrew Stuart e STEEN, Maarten Van. **Sistemas Distribuídos:** Princípios e paradigmas. 2. ed. São Paulo: Pearson Prentice Hall, 2007. 402p.

SUN MICROSYSTEMS (California, Estados Unidos) (Org.). **Multithreaded Programming Guide.** Santa Clara: Sun, 2008. 270 p. Disponível em:

<<http://dlc.sun.com/pdf/816-5137/816-5137.pdf>>. Acesso em: 08 out. 2009.