

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**

**Um Estudo do Desempenho de Aplicações Web**

**Victor Samuel Nicanor Alves**

Florianópolis – SC

2012|1

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA  
CURSO DE CIÊNCIAS DA COMPUTAÇÃO**

**Um Estudo do Desempenho de Aplicações Web**

**Victor Samuel Nicanor Alves**

Trabalho de conclusão de curso  
apresentado como parte dos  
requisitos para obtenção do grau  
de Bacharel em Ciências  
da Computação

Florianópolis – SC

2012|1

**VICTOR SAMUEL NICANOR ALVES**

**UM ESTUDO DO DESEMPENHO DE APLICAÇÕES WEB**

Trabalho de conclusão de curso apresentado como parte dos requisitos para  
obtenção do grau de Bacharel em Ciências da Computação

Orientador:

---

**Prof. Dr. Mário Antônio Ribeiro Dantas**

Banca examinadora:

---

**Prof. Dr. Fernando Ostuni Gauthier**

---

**Prof. Dr. Roberto Willrich**

## **Agradecimentos**

Gostaria de demonstrar a minha gratidão por diversas pessoas que me auxiliaram e acompanharam desde a preparação para o ingresso neste curso até a realização desse projeto.

Agradeço aos meus pais, Jair e Catarina, aos meus irmãos, Leonardo e Giuliana, por apoiarem minhas decisões e me auxiliarem em todos os acontecimentos de minha vida.

O suporte proporcionado pelos meus tios Luis e Walkyria e primos Ricardo, Vivian e Soraia, foi fundamental para a realização de meus estudos. Sou infinitamente grato a vocês e toda a nossa família.

Sem a minha amada namorada Juliana, não teria concluído esse curso. Agradeço eternamente por todo o esforço e companhia, fundamentais nesse momento de minha vida. Com você tudo ficou mais fácil.

Aos meus amigos da república d'usKra e seus "agregados", por todas as risadas, amigos do CALICO pelos bons momentos jogando sinuca e conversando até tarde e por muitos outros amigos e amigas que conheci pela universidade.

Meus amigos messiânicos aos quais tenho eterna gratidão.

Agradeço ao meu orientador professor Mario Dantas, por me transmitir conteúdo valioso para a realização desse projeto, e aos membros da banca avaliadora, professores Roberto Willrich e Fernando Gauthier, por suas contribuições para o trabalho. E também ao apoio técnico da equipe da SETIC.

Agradeço à Korin Agricultura Natural e à Aproxima Comunicação por possibilitar a execução de testes em seus websites, os quais foram muito importantes para a realização desse trabalho.

Graças ao apoio dessas pessoas e de diversas outras não listadas, pude completar mais essa etapa de minha vida. A todos, meus mais sinceros agradecimentos.

*“Costumamos referir-nos ao estudo como se só houvesse uma modalidade. Entretanto, existe o estudo vivo e o estudo morto. Aprender por aprender é estudo morto, enquanto aprender algo para ser utilizado na sociedade é estudo vivo”. - Meishu-Sama*

# Sumário

|   |    |
|---|----|
| Resumo.....                                     | 9  |
| Lista de Figuras .....                          | 10 |
| Lista de Tabelas.....                           | 12 |
| Lista de Acrônimos .....                        | 13 |
| 1. Introdução .....                             | 14 |
| 1.1. Objetivos Gerais .....                     | 14 |
| 1.3. Objetivos Específicos .....                | 15 |
| 1.4. Limitações .....                           | 15 |
| 1.5. Organização do Trabalho .....              | 15 |
| 2. Internet e World Wide Web .....              | 15 |
| 2.1. Introdução.....                            | 16 |
| 2.2. A Internet.....                            | 16 |
| 2.2.1. Histórico .....                          | 16 |
| 2.2.2. Organização.....                         | 17 |
| 2.2.3. Internet Protocol.....                   | 21 |
| 2.2.4. Protocolos que utilizam a Internet ..... | 24 |
| 2.2.5. Transmission Control Protocol .....      | 25 |
| 2.2.6. User Datagram Protocol.....              | 25 |
| 2.3. A World Wide Web .....                     | 25 |
| 2.3.1. Hipertexto.....                          | 26 |
| 2.3.2. Histórico .....                          | 26 |
| 2.3.3. Web 2.0.....                             | 27 |
| 2.3.4. Web 3.0.....                             | 28 |
| 2.4. Considerações Finais .....                 | 28 |

|         |   |    |
|---------|---|----|
| 3.      | Parâmetros Relativos ao Desempenho.....                   | 29 |
| 3.1.    | Introdução.....   | 29 |
| 3.2.    | Atraso .....  | 29 |
| 3.3.    | Perda de Pacotes .....                                    | 30 |
| 3.5.    | Largura de Banda .....                                    | 31 |
| 3.6.    | Vazão .....   | 31 |
| 3.7.    | Metro Ethernet.....                                       | 31 |
| 3.8.    | Considerações Finais .....                                | 32 |
| 4.      | Proposta: Um Estudo do Desempenho de Aplicações Web ..... | 33 |
| 4.1.    | Introdução.....   | 33 |
| 4.2.    | Objetivo Geral.....                                       | 33 |
| 4.3.    | O Problema .....  | 33 |
| 4.4.    | Web Performance Optimization .....                        | 36 |
| 4.5.    | Psicologia da Velocidade.....                             | 38 |
| 4.6.    | Técnicas de Otimização da Web .....                       | 41 |
| 4.6.1.  | Renderização Progressiva .....                            | 44 |
| 4.6.2.  | Paralelização com Múltiplos Domínios.....                 | 46 |
| 4.6.3.  | Folhas de Estilo no Topo.....                             | 48 |
| 4.6.4.  | CSS Spriting.....   | 49 |
| 4.6.5.  | DATA: URI e MHTML.....                                    | 51 |
| 4.6.6.  | Minificação e Arquivos .....                              | 52 |
| 4.6.7.  | Posicionamento de JavaScript .....                        | 52 |
| 4.6.8.  | Compactação .....   | 55 |
| 4.6.9.  | Gerenciamento de Cookies.....                             | 56 |
| 4.6.10. | Redes de Fornecimento de Conteúdo .....                   | 57 |
| 4.8.    | Considerações Finais .....                                | 58 |

|        |  |    |
|--------|--|----|
| 5.     | Ambiente e Resultados Experimentais.....   | 59 |
| 5.1.   | Conceitos.....                             | 59 |
| 5.3.   | Ambiente Experimental.....                 | 61 |
| 5.5.   | Estudo de Caso 1: ufsc.br .....            | 63 |
| 5.5.1. | Análise do Site .....                      | 63 |
| 5.5.2. | Procedimentos Utilizados.....              | 64 |
| 5.5.4. | Resultados Obtidos .....                   | 65 |
| 5.6.   | Estudo de Caso 2: www.korin.com.br.....    | 66 |
| 5.6.1. | Análise do Site .....                      | 66 |
| 5.6.2. | Procedimentos Utilizados.....              | 67 |
| 5.6.3. | Resultados Obtidos .....                   | 67 |
| 5.7.   | Estudo de Caso 3: www.aproxima.com.br..... | 68 |
| 5.7.2. | Análise do Site .....                      | 69 |
| 5.7.3. | Procedimentos Utilizados.....              | 69 |
| 5.7.4. | Resultados Obtidos .....                   | 70 |
| 5.8.   | Resultados Experimentais .....             | 71 |
| 6.     | Conclusão e Trabalhos Futuros .....        | 72 |
|        | Referências Bibliográficas.....            | 73 |
|        | Anexos .....                               | 83 |

## Resumo

O desenvolvimento de tecnologias utilizadas em aplicações Web acarretou em maiores níveis de complexidade e, conseqüentemente, certa ineficiência de execução. O tempo de carregamento das páginas é considerado o ponto-chave no quesito nível de satisfação do usuário, resultando diretamente no sucesso ou insucesso da aplicação.

Por muito tempo, a ineficiência de páginas Web foi relacionada somente ao poder de processamento de servidores e a baixa capacidade de enlaces de rede. Entretanto, novas linhas de pesquisa indicam que os maiores obstáculos se apresentam depois desses fatores, no navegador do usuário.

Tendo em mente essa realidade, torna-se necessário o estudo de técnicas que identifiquem os gargalos de desempenho e que ofereçam soluções factíveis para os problemas apresentados.

Esse trabalho apresenta uma pesquisa de dez técnicas úteis no desenvolvimento de aplicações Web, em adição, e são apresentados estudos de casos empíricos em ambientes reais que comprovam a melhoria ao se aplicar tais técnicas.

Palavras-chave: Desempenho na Web, Web Performance Optimization (WPO), Front-end Optimization (FEO)

## Lista de Figuras

|   |    |
|---|----|
| Figura 2.1 Organização da Internet Mundial (STEWART, 2000).....   | 18 |
| Figura 2.2 Organização da Internet no Brasil (NIC.BR, 2005).....  | 20 |
| Figura 2.3 Comparativo entre países IPv6 (RIPE-NCC, 2012).....  | 24 |
| Figura 2.4 Pilha de Protocolos da Internet .....  | 25 |
| Figura 2.5 A Evolução da Web (SPIVACK, 2008) .....  | 27 |
| Figura 4.1 Comparativo entre velocidade e funcionalidade (GOMEZ, 2010) ...                                | 34 |
| Figura 4.2 Impacto de um segundo no carregamento de páginas (ABERDEEN, 2008) .....                        | 35 |
| Figura 4.3 Impacto do atraso em mecanismos de busca (SCHURMAN e BRUTLAG, 2009) .....                      | 36 |
| Figura 4.4 <i>Back-end</i> vs. <i>Front-end</i> .....   | 38 |
| Figura 4.5 Quadrados A e B com mesma cor (STEFANOV, 2010).....  | 39 |
| Figura 4.6 Diversas Percepções do Tempo (STEFANOV, 2010).....   | 40 |
| Figura 4.7 Fatores que envolvem uma página Web (HTTPARCHIVE, 2012) ..                                     | 41 |
| Figura 4.8 Evolução do tamanho de páginas Web (HTTPARCHIVE, 2012) ....                                    | 42 |
| Figura 4.9 Diagrama Cascata de Carregamento de Página.....  | 43 |
| Figura 4.10 Carregamento sem renderização progressiva .....   | 44 |
| Figura 4.11 Carregamento com renderização progressiva .....   | 45 |
| Figura 4.12 Renderização Progressiva de uma busca no Google (STEFANOV, 2010) .....                        | 46 |
| Figura 4.13 White Blank Screen of Death .....   | 49 |
| Figura 4.14 Princípio de CSS Spriting.....  | 50 |
| Figura 4.15 Sprites utilizados em Websites e games conhecidos (FACEBOOK, 2012) (SPRITESTITCH, 2008) ..... | 50 |
| Figura 4.16 Bloqueio devido a código JavaScript.....  | 53 |
| Figura 4.17 Javascript movido para o fim.....   | 53 |
| Figura 4.18 Carregamento de Google Calendar. Apenas JavaScript crucial carregado no topo .....            | 54 |
| Figura 4.19 Rede de Fornecimento de Conteúdo (CSLONGWANG, 2012).....                                      | 57 |
| Figura 5.1 Diagrama Cascata (BEHESHTI, 2011) .....  | 59 |

|  |    |
|--|----|
| Figura 5.2 Estágios de carregamento de um elemento de página .....                     | 60 |
| Figura 5.3 Página Inicial da Universidade Federal de Santa Catarina (UFSC.BR, ?) ..... | 63 |
| Figura 5.4 Resultados de Intervenções em UFSC.br .....                                 | 65 |
| Figura 5.5 Sprite criado para UFSC.br .....  | 65 |
| Figura 5.6 Página Inicial da Korin Agricultura Natural (KORIN, ?) .....                | 66 |
| Figura 5.7 Resultados de Intervenções em www.korin.com.br.....                         | 68 |
| Figura 5.8 Página Inicial da Aproxima Comunicação (APROXIMA, ?).....                   | 68 |
| Figura 5.9 Sprite Aproxima.....  | 70 |
| Figura 5.10 Resultados de Intervenções em www.aproxima.com.br.....                     | 70 |
| Figura 5.11 Média de melhorias dos casos.....  | 71 |

## Lista de Tabelas

|  |    |
|--|----|
| Tabela 2.1 Exemplo de Endereço IP .....  | 22 |
| Tabela 4.1 Influência do tempo sobre atenção (NIELSEN, 2010) .....                     | 40 |
| Tabela 4.2 Subdomínios de Websites (SOUDERS, 2009) .....                               | 47 |
| Tabela 4.3 Demonstração de Data: URI .....   | 51 |
| Tabela 4.4 Redução de tamanho utilizando JSMIn e Dojo Compressor (SOUDERS, 2008) ..... | 52 |
| Tabela 4.5 Tipos de compressão .....   | 56 |
| Tabela 5.1 DATA:URI aproxima.com.br .....  | 70 |

## **Lista de Acrônimos**

CDN: Content Delivery Network

CSS: Cascading Style Sheets

FEO: Front-end Optimization

HTML: Hypertext Markup Language

JS: JavaScript

TTFB: Time to First Byte

UFSC – Universidade Federal de Santa Catarina

WBSD: White Blank Screen of Death

WPO: Web Performance Optimization

# 1. Introdução

As tecnologias que integram a World Wide Web estão em constante evolução e as aplicações que as utilizam tornam-se mais poderosas. As mesmas tendem a ser mais complexas, tornando-as, muitas vezes, pouco eficientes no âmbito de desempenho.

O desempenho passou a ser considerado o fator mais importante para o sucesso de aplicações Web decorrente ao fato dos usuários passarem a não aturar longos carregamentos de páginas. (WILSON, 2010)

Dado o presente desafio, surgem técnicas que atuam nos métodos de construção de páginas Web buscando extrair máxima eficiência das tecnologias utilizadas, sem a necessidade de intervenções na infraestrutura do sistema.

Essas técnicas foram reunidas nesse trabalho com a finalidade de fornecer uma base de consulta para desenvolvedores de aplicações Web, otimizando assim o desempenho do ambiente.

Dentro da proposta de tornar ambientes Web mais eficientes, as técnicas abordadas neste documento foram aplicadas a ambientes reais, sendo ilustrado o ganho considerável que elas proporcionam.

## 1.1. Objetivos Gerais

O objetivo deste trabalho consiste em uma *survey* de técnicas para melhoria do desempenho de aplicações Web.

Para tal foi necessário um levantamento teórico na área de Internet, Web e os conceitos de redes que as envolvem. Com isso, se adaptou aplicações com as técnicas estudadas, a fim de apresentar os ganhos concretos ao utilizá-las.

### **1.3. Objetivos Específicos**

- Realizar um levantamento de técnicas que resultam em maior eficiência de aplicações Web;
- Efetuar estudos de casos empíricos.

### **1.4. Limitações**

- Não faz parte desse trabalho demonstrações formais das técnicas estudadas.

### **1.5. Organização do Trabalho**

Este trabalho de conclusão de curso é composto por seis capítulos. No capítulo 2 foram estudadas a Internet e a World Wide Web quanto a seus históricos, como são organizadas e as tecnologias que as compõem.

Em seguida, no capítulo 3, são levantados parâmetros relativos a desempenho de ambientes Web e as tecnologias que oferecem maior aproveitamento da rede.

O capítulo 4 é composto pela proposta desse trabalho, que consiste em agrupar técnicas que resultam em maior desempenho para aplicações Web. No mesmo capítulo são abordados diversos fatos que confirmam a importância em se buscar maior eficiência de aplicações Web.

O ambiente e os resultados experimentais são apresentados no capítulo 5 deste documento, onde são levantados conceitos relativos a desempenho de páginas Web e uma lista de passos a ser seguida para extrair maior eficiência no desenvolvimento de Websites. O capítulo é finalizado com três estudos de casos empíricos e seus resultados.

O capítulo 6 consta as conclusões e sugestões para trabalhos futuros.

## **2. Internet e World Wide Web**

## **2.1. Introdução**

Neste capítulo serão abordados os conceitos teóricos sobre a Internet e a World Wide Web, com o objetivo de conceituar esses termos. Primeiramente será abordada a Internet: seu histórico, organização e um breve estudo das tecnologias utilizadas como base de seu funcionamento. Posteriormente será feito um estudo sobre a World Wide Web e seu histórico. Por fim, as considerações finais sobre a Internet e Web configuram o fato, de como esses termos são empregados erroneamente pelo usuário.

## **2.2. A Internet**

A Internet é uma rede de computadores com alcance global por onde dispositivos eletrônicos trocam informações (KUROSE e ROSS, 2010). É um ambiente complexo, que envolve diversos tipos de componentes necessários para a transmissão de sinais digitais ao redor do mundo, endereçando cada mensagem a seu destino.

Comunicação é um processo social fundamental. É a atividade chave na interação interpessoal e desde os tempos remotos define os rumos da sociedade. A humanidade busca novas formas de comunicação durante toda sua existência e vem alcançando grandes resultados nas últimas décadas, consequência do desenvolvimento tecnológico e a eficiência com que nos comunicamos atualmente. É evidente que os avanços econômicos, culturais e sociais são resultados do processo de globalização e o encurtamento de distâncias.

A Internet exerce nesse cenário um papel crucial. Ela envolve diversas áreas de conhecimento e atividade humanas. Pode ser considerada indispensável para um bom andamento dos processos atuais. E, apesar de décadas de existência, ainda há muito espaço para a Internet se desenvolver e alcançar novos níveis de eficiência e alcance.

### **2.2.1. Histórico**

Essencial para o desenvolvimento da cultura moderna, a Internet teve seu início na disputa entre os Estados Unidos e União Soviética durante as décadas da Guerra Fria. Enquanto os russos enviavam seus primeiros satélites ao espaço, centros de pesquisa norte americanos desenvolviam a base da tecnologia que utilizamos hoje para nos comunicarmos mundialmente (STEWART, 2000).

Em 1961, Leonard Kleinrock, professor da Universidade da Califórnia em Los Angeles, contribuiu em sua pesquisa de PhD com a teoria matemática por trás da rede de troca de pacotes. Ele foi o primeiro a estabelecer uma conexão entre dois computadores através desse novo modelo de comunicação no mesmo ano. (ISOC) (KLEINROCK, 1961)

Evidentemente, Kleinrock e seu grupo de pesquisa desenvolveram apenas um protótipo, o qual posteriormente deu origem a ARPANET: um sistema que chegou a englobar grande parte do território norte-americano, conectou diversos centros de pesquisa e governamentais. Mais tarde, o sistema foi expandido para a rede mundial de computadores (DARPA).

Neste contexto, pode se definir como rede (do Inglês *net*): um grupo de computadores que se comunicam entre si, seguindo um padrão de comunicação pré-estabelecido. E o termo Internet (do inglês *internetworking*): a interligação de múltiplas redes de computadores, decorrente ao crescimento da primeira grande rede, que se expandiu para diversas nações, alcançando proporções mundiais. (RFC1296, 1992)

### **2.2.2. Organização**

Com base no grande crescimento da Internet, ainda na década de 1980, foi necessário estabelecer padrões organizacionais para manter a rede sobre controle, sendo que, inicialmente, o governo norte-americano era responsável de tal organização (STEWART, 2000).

Foram então criadas instituições internacionais, responsáveis por garantir a integridade da rede (STEWART, 2000).

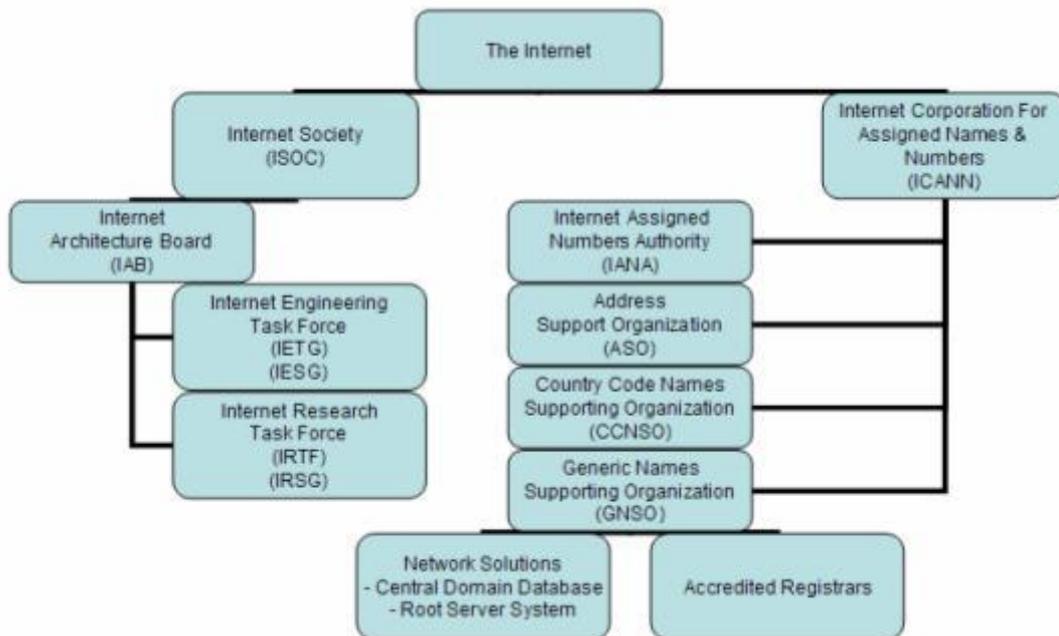


Figura 2.1 Organização da Internet Mundial (STEWART, 2000)

A seguir, uma breve descrição sobre cada elemento da organização mundial da Internet:

- **Internet Society (ISOC)**

A *Internet Society* foi fundada em 1992 com objetivo de ser a principal entidade responsável pela Internet. Compartilha a visão “A Internet para Todos” e tem como missão promover o desenvolvimento, evolução e o uso da Internet mundialmente. (ISOC, 1992)

- **Internet Architecture Board (IAB)**

Composta por 13 membros nomeados pela IETF, a IAB tem como missão planejar o futuro da Internet, organizar RFCs, dentre outras responsabilidades junto a outras entidades, como IETF, IESG e IANA (RFC2850, 2012).

- **Internet Engineering Task Force (IETF) e Internet Engineering Steering Group (IESG)**

A IETF é uma associação aberta para indivíduos preocupados com o futuro da Internet e da garantia de sua operação. É responsável por estudar

problemas técnicos e operacionais, e indicar as melhores tecnologias ao IESG, o qual a gerencia de acordo com as regras estabelecidas pela ISOC (RFC3935, 2004).

- ***Internet Research Task Force (IRTF) e Internet Research Steering Group (IRSG)***

Trabalha juntamente com a IETF promovendo as pesquisas importantes para a evolução da Internet. O IRSG define o rumo das pesquisas da instituição e, conseqüentemente, o futuro da rede (RFC5743, 2009).

- ***Internet Corporation For Assigned Names And Numbers(ICANN)***

Entidade responsável pela organização dos endereços da Internet e o sistema de nomes de domínios global, garantindo que não haja endereços duplicados e que os nomes de domínio sejam sempre resolvidos. Formada em 1998 assumiu a posição do governo norte-americano, incumbido até então de tal tarefa.

Incorpora em sua organização a ASO (Address Supporting Organization), CCNSO (Country Code Names Support Organization) e GNSO (Generic Names Support Organization), cada uma responsável por uma área específica (ICANN).

- ***Network Solutions (NSI)***

Primeiro provedor público de nomes de domínios. Hoje é responsável pelo diretório raiz de domínios (NSI, 1979).

- ***Accredited Domain Name Registrars***

São organizações ou empresas responsáveis de distribuir os domínios a usuários finais. Devem ser credenciados para utilizar domínios primários genéricos (.com, .net, .biz, .info) ou de nível de país (.br, .uk, .jp) (DOTANDCO, 2011).

### **2.2.2.1. Organização no Brasil**

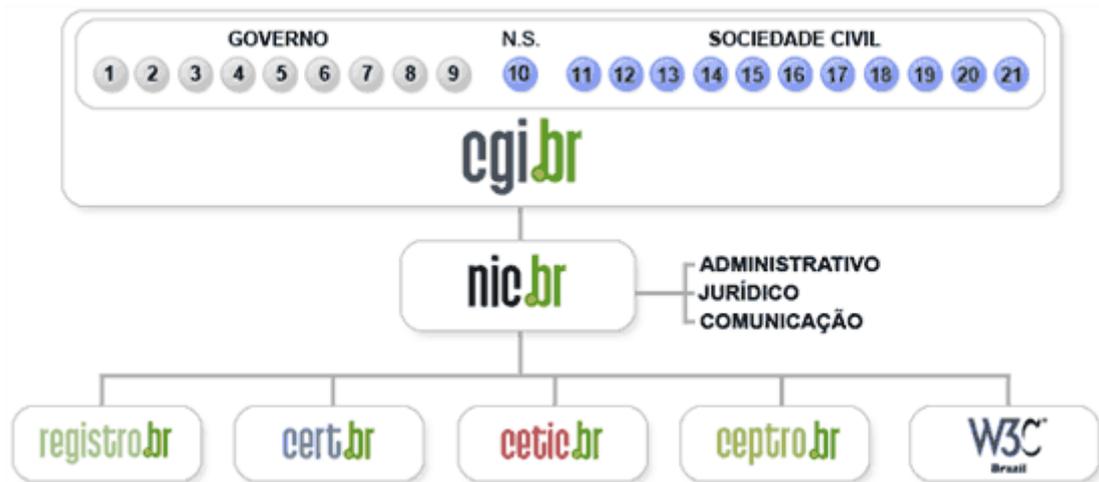


Figura 2.2 Organização da Internet no Brasil (NIC.BR, 2005)

- **Comitê Gestor da Internet no Brasil: (CGI.br)**

Comitê responsável por gerenciar e definir o futuro da Internet no Brasil, composta por representantes do governo, como os Ministérios da Ciência e Tecnologia, das Comunicações e da sociedade civil, como empresas e academia. Criado através do Decreto N°4.829, de três de setembro de 2003. (CGI.br, 2003)

- **Núcleo de Informação e Coordenação do Ponto Br: (NIC.br)**

Executa as decisões do Comitê Gestor através de centros de estudo CETIC, CERT e CEPTR0 e suporta o LACNIC, Registro de Endereços da Internet para a América Latina e Caribe (NIC.BR, 2005).

- **Centro de Estudos Sobre as Tecnologias da Informação e da Comunicação: (CETIC.br)**

Responsável de apresentar pesquisas na área de tecnologia da informação e da comunicação no espaço brasileiro. Disponibiliza publicações valiosas indicando como a Internet está evoluindo no país (CETIC.BR, 2005).

- **Centro de Estudos, Respostas e Tratamento de Incidentes de Segurança no Brasil: (CERT.br)**

Organização responsável por tratar incidentes de segurança relativas a Internet brasileira e conscientizar a população sobre os problemas de segurança do meio cibernético (CERT.BR, 2005).

- **Centro de Estudos e Pesquisas em Tecnologias de Redes e Operações: (CEPTRO.br)**

Responsável em desenvolver soluções para a infraestrutura da Internet brasileira, relativo tanto a *hardware* como *software* (CEPTRO.BR, 2005).

- **W3C Brasil**

Braço brasileiro do consórcio internacional da *World Wide Wide* (W3C), que rege padrões garantindo o desenvolvimento da Web (W3C.BR, 2007).

### **2.2.3. Internet Protocol**

Desde sua criação, a Internet foi pensada para agregar um número grande de máquinas, integrando uma grandiosa rede distribuída geograficamente. Um problema inerente a tal arquitetura é a forma como cada integrante da rede é referenciado. Para solucionar tal dificuldade, foi criado o sistema de endereçamento através do *Internet Protocol*, ou IP, um protocolo encarregado de referenciar cada máquina conectada a rede (RFC791, 1981).

#### **2.2.3.1. Rede de Troca de Pacotes**

Em meados da década de 1960, Paul Baran contribuiu com uma proposta que mudaria a forma como computadores se comunicavam (BARAN, 1962). Agora as mensagens seriam quebradas em pacotes e dividiriam o mesmo meio de maneira sequencial, um a um. Ao chegar a seu destino, os pacotes são reagrupados e a mensagem pode ser finalmente entregue ao destinatário.

Esse ambiente segue o modelo de serviço *best-effort*, onde a rede procura entregar a mensagem ao destino o mais rápido possível (DANTAS, 2010). Contudo, todos os diversos pacotes que trafegam pela rede seguem com a mesma prioridade. No momento em que há um grande número de mensagens pela rede, o desempenho da rede é prejudicado, ela não oferece mais as condições ideais para entregar as mensagens no menor tempo possível.

#### **2.2.3.2. IPv4**

O IP versão 4, comumente referenciado como IPv4, é a versão atual e mais usada desse protocolo. Embora pareça ser sua quarta implementação, esta é a primeira versão utilizada, sendo derivado do *Transmission Control Protocol* (TCP) ou Protocolo de Controle de Transmissão que será detalhado a seguir (RFC791, 1981) (RFC793, 1981).

O cabeçalho do datagrama IP é de no mínimo 20 Bytes. Os valores principais estão nos campos IP origem, IP destino e tamanho da mensagem. São encontrados também campos referentes a identificação dos fragmentos da mensagem e checagem de erro (KUROSE e ROSS, 2010).

- **Endereçamento**

Solucionando o problema de como localizar as máquinas conectadas a rede, o endereçamento IP adota uma sequência de quatro números de 0 a 255, com a capacidade de representar mais do que quatro bilhões de endereços:  $255^4 = 4,294,967,296$ , ilustrado pela Tabela 2.1 (KUROSE e ROSS, 2010).

| Exemplo de endereço IP |                                     |
|------------------------|-------------------------------------|
| Decimal:               | 238 . 17 . 159 . 4                  |
| Binary:                | 11101110 00010001 10011111 00000100 |

Tabela 2.1 Exemplo de Endereço IP

Os quatro Bytes pareciam mais do que suficientes para endereçar todas as máquinas conectadas a rede. Dificilmente os engenheiros previram que mais do que quatro bilhões de computadores seriam conectados a mesma rede. No entanto, hoje esse número não é suficiente para abrigar todos os diversos dispositivos que se conectam a rede mundial (NRO, 2011) (MOREIRAS, 2009).

Dessa, dentre outras necessidades, surge o projeto de renovação do IPv4. Mais aceito e que está em processo de transição há mais de uma década. Essa nova versão será detalhada a seguir.

### 2.2.3.3. IPv6

Com o intuito de superar as restrições da versão 4 do IP, a IETF publicou a especificação do protocolo da internet versão 6 (RFC2460, 1998; RFC2616,

1999). As características principais dessa nova versão são referentes a endereçamento, extensibilidade, controle de fluxo e simplificação de seu cabeçalho. Essa nova versão é referenciada principalmente como IPv6 ou IPng (*Internet Protocol Next Generation*)

- **Cabeçalho simplificado**

Em comparação com a versão anterior, o cabeçalho da versão 6 é menor, reduzindo o *overhead* causado por um cabeçalho extenso (RFC1726, 1994).

- **Endereçamento**

IPv6 utiliza endereçamento de 128 bits com a capacidade de representar 340 undecilhões (na ordem de  $10^{38}$ ) de equipamentos eletrônicos conectados a rede. (FLORENTINO, 2011).

- **Extensibilidade**

Suporte a opções que melhoram e especializam o protocolo, tornando-o mais flexível. Agora as opções são incluídas após o cabeçalho, fazendo parte dos dados e sendo somadas no campo *Payload Length* (DAS, 2008).

- **Controle de Fluxo de Tráfego**

O protocolo traz recursos de qualidade de serviço (QoS) presentes no seu cabeçalho, informando roteadores em como diferenciar pacotes de acordo com a importância de seus dados (DANTAS, 2010).

- **Transição para IPv6**

O processo de implantação do IP versão 6 mostrou-se complicado e de difícil aceitação. Devido ao fato de englobar todos os níveis que envolvem a internet, desde fabricantes de equipamentos de rede, provedores de acesso a aplicações.

Como solução, foram apresentadas diversas técnicas para a transição de IPv4 para IPv6 (SELLERS, 2009). As técnicas foram classificadas entre Pilha Dupla (RFC4213, 2005), Tunelamento (RFC2893, 2000) e Tradução (RFC2765, 2000) (RFC3142, 2001).

A Internet Society busca conscientizar todos os envolvidos com a Internet para investir nessa mudança. Está marcado para o dia 6 de junho de 2012 um

limite para empresas do mundo todo se prepararem para rodar IPv6 por 24 horas, incluindo grandes figuras do mercado (WORLDIPV6DAY, 2012). A participação é voluntária, porém já envolve Google, Facebook, Akamai, entre outros. Durante a transição, os provedores de serviço deverão suportar ambos os protocolos em paralelo.

Ian Flint relatou a experiência que tiveram ao tentar implantar IPv6 por poucas horas no site principal da Yahoo! como preparação para o World IPv6 Day (FLINT, 2011). Através de seus testes puderam concluir que a transação será muito complexa por dois motivos: 1) porque o sistema que eles controlam é muito complexo; 2) poucos usuários já estão aptos a usar a nova tecnologia.

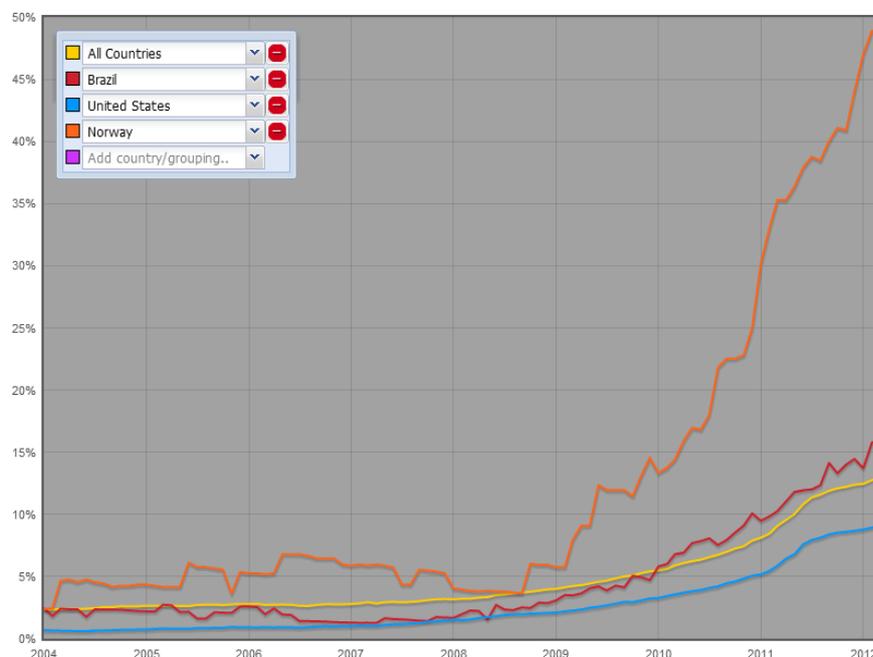


Figura 2.3 Comparativo entre países IPv6 (RIPE-NCC, 2012)

A Figura 2.3 representa o gráfico do percentual de adoção de IPv6 por país (RIPE-NCC, 2012). O Brasil está se preparando, atualmente acima da média mundial, com aproximadamente 15%, porém bem atrás da Noruega.

#### 2.2.4. Protocolos que utilizam a Internet

Há uma vasta gama de protocolos que utilizam IP, dentre eles os mais importantes são TCP (RFC793, 1981) e UDP (RFC768, 1980). A Figura 2.4

lista alguns dos protocolos das camadas de transporte e aplicação do modelo OSI de cinco camadas, trabalhando sobre a camada de rede (Internet), onde o IP se encaixa (ZIMMERMANN, 1980) (PETERSON e DAVIE, 2007).

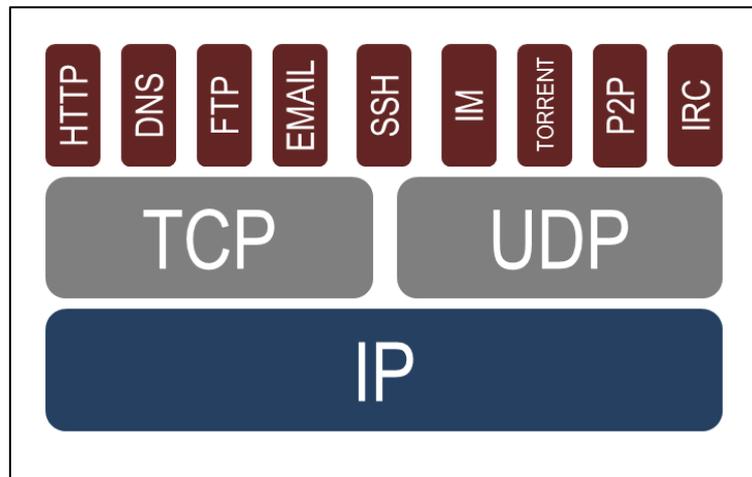


Figura 2.4 Pilha de Protocolos da Internet

### 2.2.5. Transmission Control Protocol

O Protocolo de Controle de Transmissão (TCP) opera em conjunto com o IP (RFC793, 1981). Apresenta o recurso TCP, que garante que a mensagem é entregue corretamente, sem estar corrompida. Incorpora também recursos para manter o fluxo da rede saudável e não congestioná-la.

### 2.2.6. User Datagram Protocol

O Protocolo de Datagramas de Usuário é paralelo ao TCP e funciona em conjunto com IP (RFC768, 1980). É indicado para aplicações sensíveis ao tempo, onde a mensagem deve ser entregue o mais rápido possível, independente se estão corrompidas ou não. É utilizado geralmente por aplicações multimídia.

## 2.3. A World Wide Web

A World Wide Web, ou somente Web é a principal aplicação que utiliza o protocolo HTTP (RFC2616, 1999) em conjunto de protocolos TCP e IP. É um imenso sistema distribuído multimídia, interativo e não linear, onde são disponibilizados diversos tipos de mídia como: textos, sons, vídeos, imagens e animações. É uma extensa base de dados disponível a todos que têm acesso a grande rede de computadores (KUROSE e ROSS, 2010).

### **2.3.1. Hypertexto**

Desenvolvido por Ted Nelson, o termo hypertexto é uma forma de representar informação de maneira não linear. Diferente de um livro ou televisão, o hypertexto tem um fluxo não contínuo e único, de acordo com as interações do usuário a medida do tempo. (NELSON, 1981)

Sendo assim, documentos hypertexto são interligados entre si formando uma relação lógica entre seu conteúdo. Essa rede interligada de informação compõe uma grande base de dados de fácil interação e que disponibiliza uma nova forma de busca de conhecimento. (BEVILACQUA, 1989)

### **2.3.2. Histórico**

Ted Nelson buscava criar um sistema de armazenamento e indexação acessada eletronicamente onde qualquer um conseguiria adicionar documentos e formar seu próprio estilo de indexação. (STEWART, 2000)

Inspirando-se nessa ideia, Tim Berners-Lee em conjunto com Robert Cailliau propôs um sistema global de hypertexto, com o objetivo de tornar o processo de autoria mais simples e abrangente. (BERNERS-LEE, 1990)

Primeiramente o sistema World Wide Web seria disponibilizado apenas para acadêmicos compartilhar seu conhecimento, contribuindo para a ciência. Berners-Lee não previu o crescimento da Web a tal ponto que alcançasse o mundo todo em diversas áreas e não só acadêmica (CERN, 2008).

### 2.3.3. Web 2.0

A Figura 2.5 representa a evolução da Web, apresentando em linha do tempo a adoção de novas tecnologias a esse ambiente.

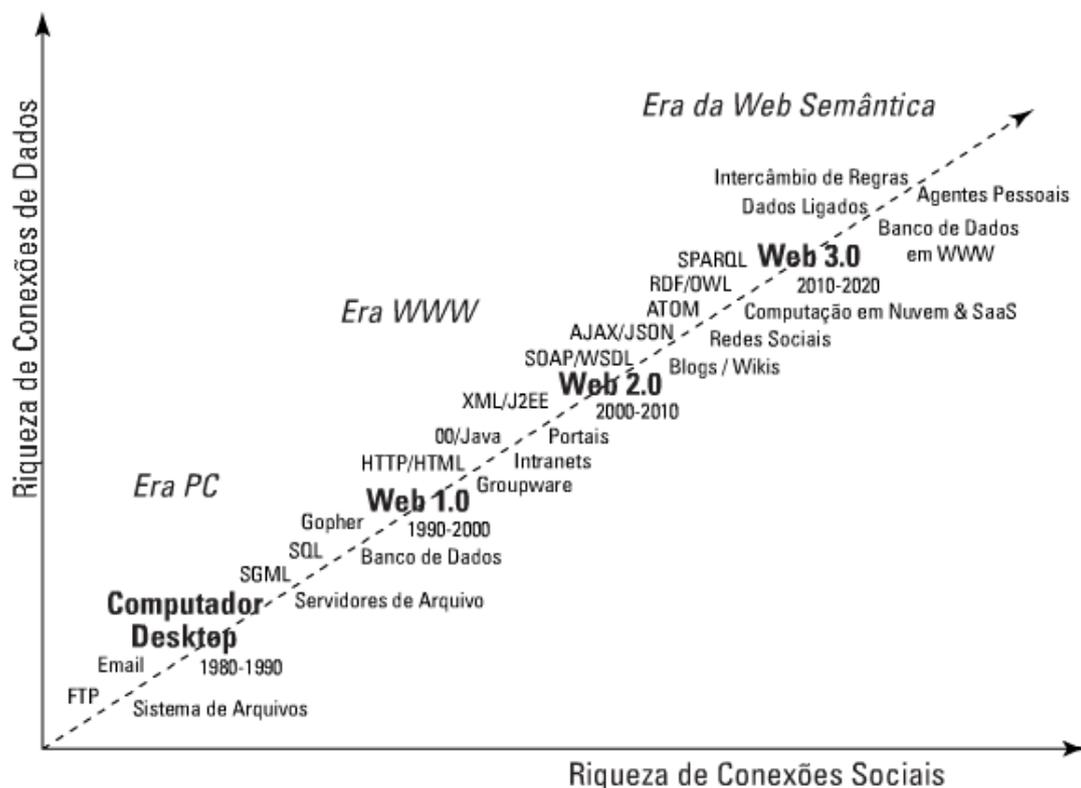


Figura 2.5 A Evolução da Web (SPIVACK, 2008)

O termo Web 2.0 foi empregado para definir uma nova geração da forma como sites são construídos e as tecnologias que os compõe, tornando o ambiente antes estático e dependente de seus autores para colaborativo. O termo chave para essa nova etapa é interatividade. Nesse momento, o usuário dá um passo a frente e começa a contribuir com o conteúdo disponibilizado na Web. (O'REILLY, 2005)

O primeiro avanço da Web 2.0 foi criação de sistemas de blogs, onde cada indivíduo pode criar sua própria página e contribuir com suas ideias, somando com o conteúdo da Web. (PEMBERTON, 2006)

Em sequência, surgiu JavaScript e XML. Assíncronos conhecido como Ajax (do inglês *Asynchronous JavaScript and XML*), tecnologia que melhorou muito

a experiência do usuário em páginas Web, tornando-as muito mais interativas. (GARRETT, 2005)

#### **2.3.4. Web 3.0**

O contínuo avanço das tecnologias que envolvem a Web abriu espaço para outros campos do conhecimento atuarem, contribuindo cada vez mais para um futuro inovador. (AGARWAL, 2009). Agarwal adiciona que a Web tende a ser mais personalizável e adaptada a dispositivos móveis. O foco é na usabilidade e na inteligência virtual.

## **2.4. Considerações Finais**

Nesse capítulo abordou-se os conceitos básicos sobre a Internet e a World Wide Web.. Foram descritos breves históricos sobre Internet e Web e a diferença entre ambos.

Atualmente, principalmente no nosso país, o termo Internet é erroneamente confundido com a World Wide Web. A Web é uma entre diversas outras aplicações que utilizam a Internet como base e, em paralelo a mesma, podemos citar aplicações de uso comum como correio eletrônico, mensagens instantâneas, streaming multimídia, IRC, torrent, etc. No espectro de 65 mil portas utilizadas nos protocolos de transporte, a Web representa somente uma delas.

Embora tecnicamente a Web seja somente uma dentre tantas aplicações que utilizam a Internet, ela representa grande percentual de uso da infraestrutura da Internet e mobiliza a maioria dos recursos disponíveis.

## **3. Parâmetros Relativos ao Desempenho**

### **3.1. Introdução**

Em continuidade ao trabalho de pesquisa sobre o desempenho de aplicações Web, reconheceu-se a necessidade em consolidar o conhecimento sobre os aspectos técnicos do ambiente. Esse capítulo apresenta um estudo realizado sobre os principais fatores relativos a redes de computadores que influenciam diretamente no uso da Internet, e as deficiências que a estrutura oferece em termos de transmissão de dados. Dentre diversas variáveis, serão abordados termos como atraso, perda de pacotes, largura de banda e vazão.

### **3.2. Atraso**

A função de qualquer rede de computadores é a troca de informação entre dispositivos. Entretanto, qualquer circuito eletrônico possui limitações físicas onde sinais elétricos vagam com velocidade extremamente alta, porém um atraso é perceptível quando o trajeto a ser percorrido é longo (DANTAS, 2010).

Por ser um sistema altamente complexo, envolvendo inúmeros componentes de rede, a Internet apresenta maiores dificuldades quanto a transmissão de informação, resultando em maior impacto no tempo de transmissão (KUROSE e ROSS, 2010) (CHESHIRE, 1996).

O tempo gasto por um pacote ao trafegar a Internet desde sua origem ao seu destino é chamado atraso. Consiste na soma de pequenos tempos gastos por cada elemento da rede, resultando em um impacto considerável e perceptível.

O atraso pode ser dividido em:

- Atraso de processamento:

Tempo gasto por um elemento da rede ao processar a informação, tanto na criação, no roteamento e na recepção da mensagem.

- Atraso de transmissão:

Tempo gasto pela interface de rede, para transmitir a mensagem por completo ao meio em que a mensagem irá trafegar até seu destino.

- Atraso de propagação

Inerente ao meio, a propagação dos dados pelos fios, fibras ou ar, resultam num atraso considerável, dependendo da distância ponto a ponto. Os sinais elétricos trafegam próximos a velocidade da luz, mas mensagens intercontinentais demoram centenas de milissegundos, atraso que, acumulado, é perceptível.

- Atraso de fila

Em redes de comutação de pacotes, os nodos internos da rede chamados de roteadores são obrigados a adotar uma fila caso receba mais pacotes do que consegue transmitir. O tempo em que o pacote aguarda na fila até ser transmitido pode exercer uma grande parcela no tempo total de tráfego, dependendo das condições da rede (KUROSE e ROSS, 2010) (KLEINROCK, 1961).

### **3.3. Perda de Pacotes**

Redes de comutação de pacotes estão vulneráveis a perda de pacotes. A principal causa está vinculada ao atraso de fila, onde um número finito de pacotes podem aguardar processamento, isto é, a fila tem um tamanho finito. Caso um roteador receba número muito grande de pacotes, maior do que a sua fila de entrada, pacotes serão descartados e a informação será perdida (KUROSE e ROSS, 2010).

### **3.5. Largura de Banda**

No campo de redes de computadores, o termo largura de banda está relacionado com a capacidade de bits por segundo que um enlace de rede pode transportar. Analogamente, relaciona-se com o diâmetro de um cano onde quanto maior o diâmetro, maior o potencial do canal de transporte.

Entretanto, esse termo dá margens a falsa interpretação se relacionado a transmissão de sinais eletrônicos, onde a unidade de largura de banda é Hertz, relativo a faixa de frequência de um sinal. O mais indicado seria utilizar taxa de transferência ou capacidade de transferência (SIEMON, 1997) (DANTAS, 2010).

### **3.6. Vazão**

O termo vazão ou *throughput* está diretamente vinculado a taxa de transferência. Entretanto, vazão está relacionada a quantidade efetiva de bits transferidos por unidade de tempo, sendo esta igual ou inferior a capacidade de transferência do enlace (KUROSE e ROSS, 2010).

Dada a limitação física presente em redes de computadores quanto a eficiência durante a transmissão de dados, surgem esforços como OSCARS e OpenFlow que buscam remodelar a forma como a informação trafega pela rede.

### **3.7. Metro Ethernet**

A Metro Ethernet é uma rede de alcance metropolitano (MAN) que utiliza o padrão Ethernet (IEEE802.3). Por ter área de cobertura reduzida, apresenta-se consideravelmente mais eficiente que redes geograficamente distribuídas (WAN), alcançando taxas de transmissão muito elevadas, solucionando as deficiências inerentes a redes de computadores, apresentadas anteriormente neste capítulo.

Entretanto, para usufruir de tal eficiência plena, é necessário que os computadores que estejam conectados sobre a mesma rede, sem utilizar nenhum outro meio externo. Sendo assim, isso limita a aplicabilidade dessa arquitetura a sistemas localizados, como Intranets, por exemplo.

No Brasil, um grande esforço está sendo realizado pela CEMIG, no estado de Minas Gerais, implantando redes de até um Gbps em dezenas de cidades (CEMIG, 2007).

### **3.8. Considerações Finais**

Nesse capítulo foram abordados conceitos básicos sobre redes de computadores, tais como atraso, perda de pacotes, largura de banda e vazão. Tais termos são fundamentais para bom entendimento do funcionamento de um ambiente distribuído e as variáveis que o envolvem.

Vale ressaltar que a latência é um fator inerente a sistemas eletrônicos, tal qual a Internet. Esta, por deveras complexa, envolve diversos fatores que contribuem para o atraso em troca de mensagens. Essa degradação pode ser atenuada com a adoção de equipamentos mais poderosos, porém a limitação física, principalmente relativo ao atraso de propagação, estará presente, independente do meio que se use para transmissão de dados.

Muitas das terminologias de redes estão presentes no vocabulário de usuários de Internet e nem sempre são utilizadas de maneira precisa. Termos como largura de banda são comumente referenciados como taxa de transferência, o que dá margens a certo desentendimento.

## **4. Proposta: Um Estudo do Desempenho de Aplicações Web**

### **4.1. Introdução**

Tendo em vista os cenários apresentados nos capítulos 2 e 3, verificou-se a necessidade de realizar um estudo sobre a World Wide Web e os fatores que contribuem para sua utilização, dados que serão abordados nesse capítulo. Mais especificamente, o estudo será focado em variáveis relativas ao desempenho de páginas Web quanto ao tempo de carregamento das mesmas, sendo que o nível de satisfação do usuário está diretamente relacionado ao mesmo.

Primeiramente será descrito o objetivo desse trabalho, seguido pelo detalhamento do problema e a motivação da pesquisa.

Posteriormente serão detalhadas técnicas relevantes ao tema que, se adotadas por aplicações Web, resultarão em maior eficiência.

Esse capítulo dará o embasamento teórico necessário para os experimentos práticos presentes no capítulo seguinte, todos com o intuito de ilustrar o conteúdo aqui reunido.

### **4.2. Objetivo Geral**

Esse trabalho consistirá na criação de uma *survey* de técnicas que buscam maior eficiência de sítios eletrônicos da World Wide Web, principalmente relacionado ao tempo necessário para seu carregamento e visualização.

### **4.3. O Problema**

O avanço das tecnologias que suportam a Internet, desde enlace a processamento, criou a necessidade de tornar páginas Web mais eficientes.

Resultados apresentados por grandes empresas como Amazon (AMAZON, 2006), demonstram como milissegundos no tempo de carregamento de páginas influenciam de maneira crucial nos negócios online.

Diferenciando-se da busca por máquinas cada vez mais poderosas, o objetivo agora é otimizar páginas, analisando de maneira inteligente e detalhada como cada elemento reage e sua importância durante o carregamento. Através do trabalho realizado nesses elementos consegue-se o maior ganho possível no tempo de carregamento.

Fred Wilson, investidor em aplicações para a Web, palestrou na conferência *Future of Web Apps* com o título *The 10 Golden Principles of Successful Web Apps* (WILSON, 2010). Dentre diversos tópicos, o fator número um contribuinte para o sucesso de aplicações Web é a velocidade. Ele adiciona que velocidade não é somente uma funcionalidade adicional do sistema, e sim, a mais importante das funcionalidades, um requisito.

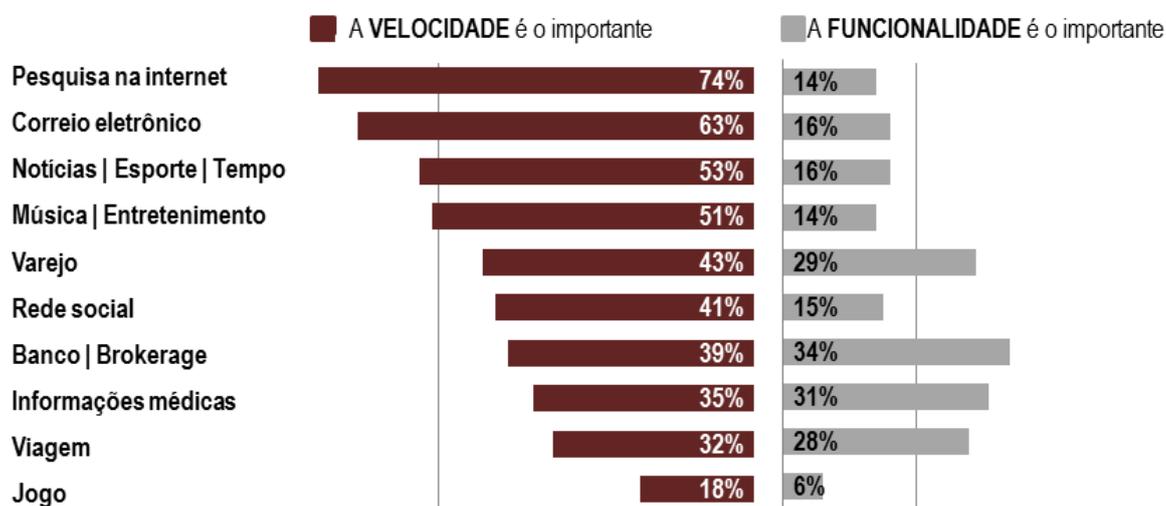


Figura 4.1 Comparativo entre velocidade e funcionalidade (GOMEZ, 2010)

No ponto de vista de satisfação do usuário, a Figura 4.1 representa a comparação entre velocidade e funcionalidade de aplicações Web. Velocidade impera com maior importância sobre todas as áreas pesquisadas.

Os resultados de uma pesquisa realizada por Akamai demonstraram que a maioria dos usuários espera o carregamento das páginas em menos de 2 segundos (AKAMAI, 2010). Um terço dos visitantes que presenciam um problema tende a não voltar. Metade desses irá compartilhar um a imagem negativa do site a outros. Após 3 segundos, 40% abandonarão o site. (GOMEZ, 2010)

Pesquisas resultam em dados interessantes como os listados pela Figura 4.2, estimando o impacto de apenas um segundo no tempo de carregamento de páginas. (ABERDEEN, 2008)

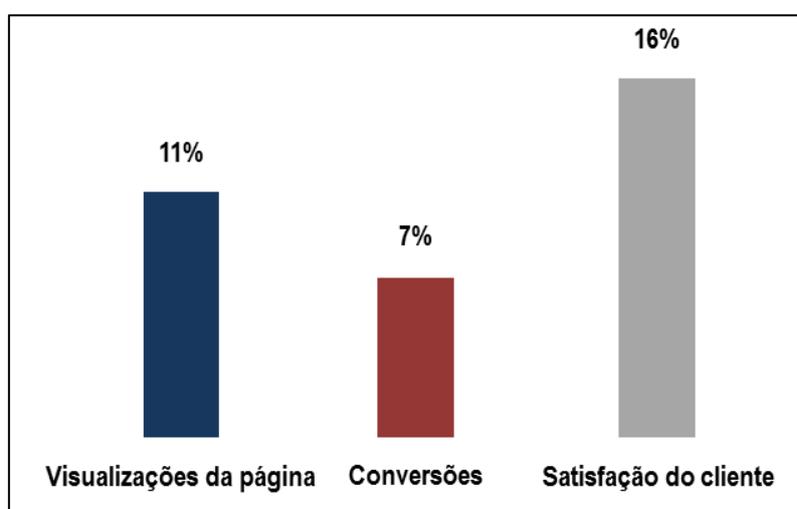


Figura 4.2 Impacto de um segundo no carregamento de páginas (ABERDEEN, 2008)

Visando solidificar tal teoria, grandes empresas, como Google, Microsoft e Amazon, realizaram testes sobre uma fração de seus usuários, medindo o impacto da velocidade sobre seus negócios.

Por exemplo, uma pesquisa realizada pela Microsoft em sua aplicação de busca Bing. Propositadamente, o tempo de carregamento de páginas foi atrasado em 50 a 2000 ms. Com isso, concluíram que, se suas páginas levassem apenas 2 segundos a mais para carregar, eles teriam uma redução de 4.3% de lucro por usuário (SCHURMAN e BRUTLAG, 2009).

|        | Distinct Queries/User | Query Refinement | Revenue/User | Any Clicks | Satisfaction | Time to Click (increase in ms) |
|--------|-----------------------|------------------|--------------|------------|--------------|--------------------------------|
| 50ms   | -                     | -                | -            | -          | -            | -                              |
| 200ms  | -                     | -                | -            | -0.3%      | -0.4%        | 500                            |
| 500ms  | -                     | -0.6%            | -1.2%        | -1.0%      | -0.9%        | 1200                           |
| 1000ms | -0.7%                 | -0.9%            | -2.8%        | -1.9%      | -1.6%        | 1900                           |
| 2000ms | -1.8%                 | -2.1%            | -4.3%        | -4.4%      | -3.8%        | 3100                           |

- Means no statistically significant change

Figura 4.3 Impacto do atraso em mecanismos de busca (SCHURMAN e BRUTLAG, 2009)

Outro estudo de caso realizado pela Amazon alcançou resultados ainda mais consideráveis. Em sua pesquisa puderam concluir que apenas 100 ms adicionais no tempo de carregamento de página são o bastante para reduzir vendas em 1% (AMAZON, 2006).

A organização Mozilla aumentou a taxa de *download* diário de seu navegador em 15.4% reduzindo somente 2.2 segundos de carregamento da página principal de seu site, que antes era de 7.7 segundos (MOZZILA, 2010).

A cadeia de comércio eletrônico Shopzilla reformulou sua plataforma por completo, reduzindo de 7 para 1 segundo o tempo de carregamento de páginas. Com essa melhora, as vendas aumentaram de 7 a 12% (DIXON, 2009).

Pesquisas nessa área ainda são muito recentes e lentamente estão alcançando a mídia, como por exemplo, matérias em jornais de grande circulação, como o New York Times (LOHR, 2012).

#### 4.4. Web Performance Optimization

Steve Souders, entusiasta de performance na Web e atualmente engenheiro da Google, criou o termo *Web Performance Optimization* (WPO)

para representar os esforços investidos nessa área, que busca o crescimento contínuo da Web no que se diz respeito a desempenho. (SOUDERS, 2010)

WPO consiste em uma área focada em intervir nas atuais práticas e padrões, evangelizando tanto desenvolvedores como usuários, para uma nova era da World Wide Web.

Jakob Nielsen, guru em usabilidade, iniciou pesquisas na área de desempenho para a Web ainda na década de 1990. Em seu artigo, ele menciona que há quatro pontos contribuintes para o tempo de carregamento de páginas: (NIELSEN, 1997)

- Processamento do servidor
- Conexão do servidor
- Transmissão pela Internet
- Conexão do usuário
- Velocidade do navegador

Seguindo-se essa teoria, veio-se tratando o problema de velocidade de carregamento de websites como um problema de infraestrutura, o chamado *back-end*. Era consenso relacionar lentidão ao poder de processamento, algoritmos e taxa de transmissão. Entretanto, Tenni Theurer, da Yahoo!, concluiu que o tempo gasto no servidor é relativamente pequeno e consiste em apenas 20% aproximadamente do tempo de carregamento das páginas. Os outros 80% são visíveis ao cliente, e consiste no tempo gasto pela transmissão de arquivos e pelo processamento do navegador (THEURER, 2006).

A seguir, a Figura 4.4 apresenta um diagrama cascata que demonstra o carregamento de uma página Web, indicando a sequência e o tempo gasto para recebimento de cada elemento que compõe a página.

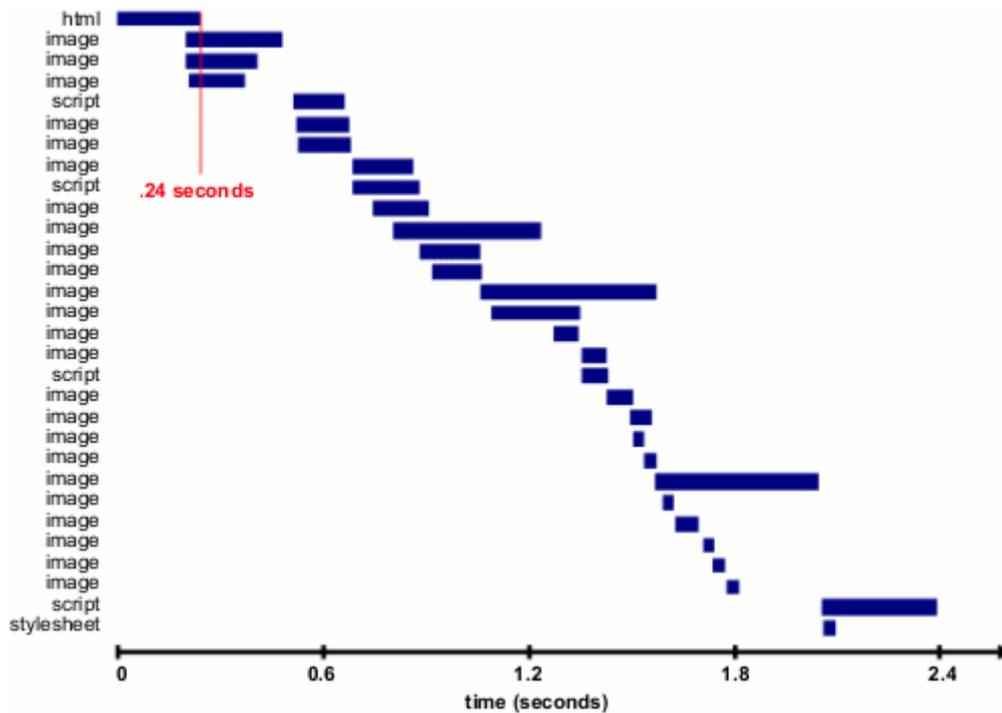


Figura 4.4 *Back-end vs. Front-end*

Percebe-se que a primeira linha representa o arquivo HTML, computado pelo servidor da aplicação em resposta a requisição do cliente. Esse é basicamente todo o processamento feito pelo servidor durante todo o carregamento, sendo que os outros 2.2 segundos (90%) correspondem ao tempo de transferência e processamento do navegador.

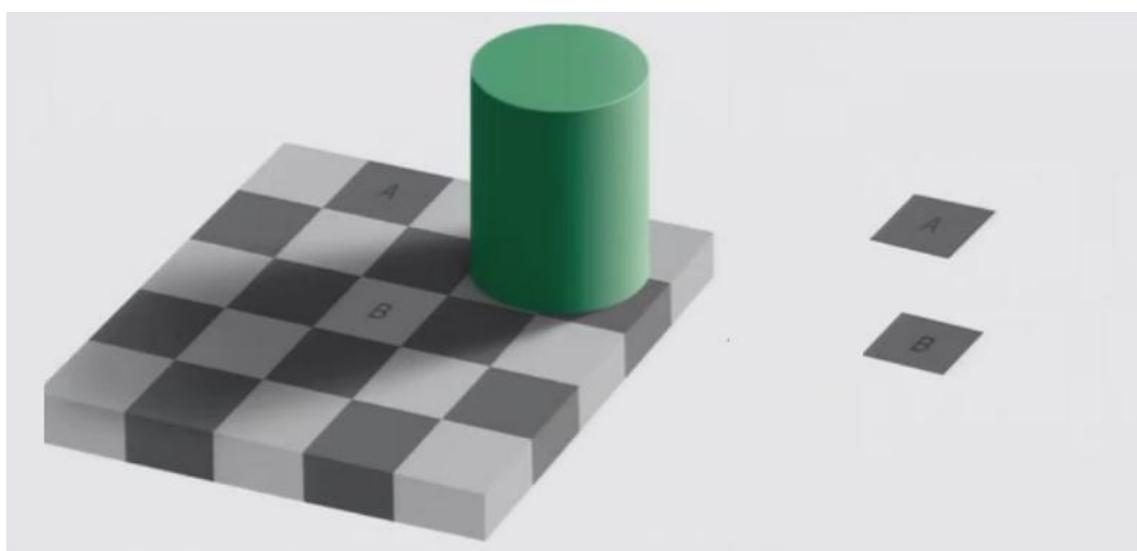
Souders adicionou a mesma teoria como “*The Performance Golden Rule*”, ou “A Regra de Ouro do Desempenho”. Isso o motiva a focar esforços em desenvolvimento de técnicas e ferramentas para otimizar o tempo de carregamento em *front-end*, o tempo gasto no navegador. (SOUDERS, 2012)

## 4.5. Psicologia da Velocidade

Tendo em mente a importância da velocidade para aplicações computacionais, convém buscar mais informações sobre o impacto da velocidade no usuário, a fim de legitimizar os esforços e investimentos na área, e assim, encontrar o ponto focal do problema estabelecendo um objetivo claro a ser alcançado.

A percepção humana não é sempre precisa e erros de interpretação são muito comuns. Somos facilmente enganados pelos nossos sentidos e dificilmente identificamos quando podemos estar errados.

Por exemplo, a Figura 4.5 apresenta dois quadrados A e B, ambos com a mesma coloração em escala de cinza. Enquanto situados dentro de um cenário com outras tonalidades, nosso cérebro interpreta A com cor preta e B com branca. Entretanto, ao remover o cenário e comparar somente os quadrados, pode-se claramente perceber que ambos quadrados possuem exatamente a mesma cor.



**Figura 4.5 Quadrados A e B com mesma cor (STEFANOV, 2010)**

Baseando-se nesse princípio onde a imprecisão dos sentidos humanos é preponderante, Stoyan Stefanov consolida sua pesquisa, demonstrando que apresentamos a mesma dificuldade quanto a percepção do tempo. (STEFANOV, 2010)

Stefanov menciona que a percepção do tempo é influenciada por muitos fatores, como idade, geografia, cultura, clima, hora do dia, dia da semana e até temperatura corporal.



Figura 4.6 Diversas Percepções do Tempo (STEFANOV, 2010)

Na Figura 4.6, Stefanov compara o tempo real gasto por uma aplicação executar uma tarefa com os tempos esperado, percebido e lembrado pelo usuário. Nota-se uma grande diferença, principalmente entre o tempo real e o que o usuário de fato se lembra. Caso o resultado ocorrido não seja do agrado do usuário, em apenas dois dias após o ocorrido, a lembrança pode diferenciar cinco vezes do tempo real.

O conteúdo da aplicação também influencia na percepção do usuário, sendo que ao deparar-se com conteúdo desagradável, desconhecido, entediante ou informações excessivas, o impacto é ainda maior.

| Tempo       | Percepção                          |
|-------------|------------------------------------|
| 0.1s        | Ilusão de resposta instantânea     |
| 1.0s        | Fluxo de pensamento mantido        |
| 10.0s       | Limite de atenção                  |
| Mais de 10s | Usuário facilmente se desconcentra |

Tabela 4.1 Influência do tempo sobre atenção (NIELSEN, 2010)

A Tabela 4.1 Influência do tempo sobre atenção resume o trabalho de Jakob Nielsen, onde compara a influência do tempo e o nível de atenção mantida durante uso de uma aplicação. Esses dados comprovam a necessidade de se buscar aplicações eficientes que se aproximam de ter respostas instantâneas. (NIELSEN, 2010)

Enquanto não alcançamos o sistema ideal, com respostas abaixo de 0,1 segundo, podemos buscar formas de suavizar o impacto da espera do usuário durante o tempo de processamento da tarefa. Como exemplo, há estudos sobre uso correto de barras de progresso (HARRISON, AMENTO, *et al.*, 2007) e diversas outras técnicas (SEOW, 2008) para manter a experiência do usuário num nível satisfatório. Essas pesquisas demonstram a importância de prover o usuário com informações precisas quanto ao progresso da execução e possivelmente o tempo esperado para conclusão, não importando de fato se a aplicação é mais lenta.

Joshua Bixby, da Strangeloop Networks, descreve a mesma teoria e menciona que existe um bloqueio natural do ser humano e o que cabe a nós é buscar uma Web mais rápida, já que não podemos acelerar a evolução natural do ser humano (BIXBY, 2012).

## 4.6. Técnicas de Otimização da Web

Páginas Web são documentos HTML compostos por dezenas de outros elementos, como imagens, folhas de estilos, scripts JavaScript dentre outros. A figura 4.7 apresenta o peso de cada recurso exerce sobre o tamanho total de uma página.

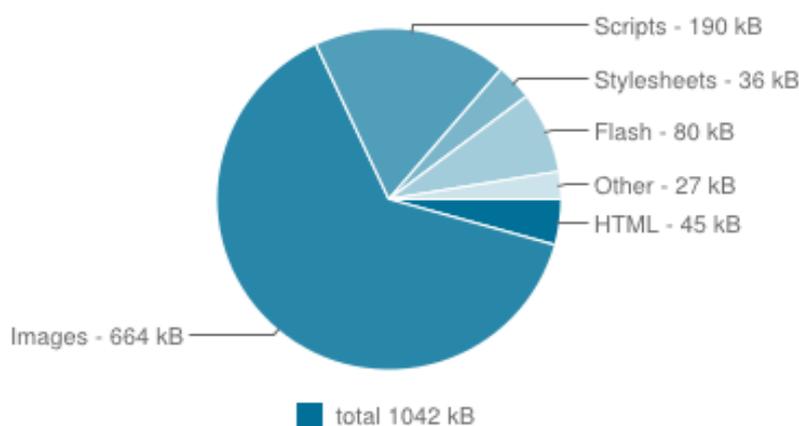
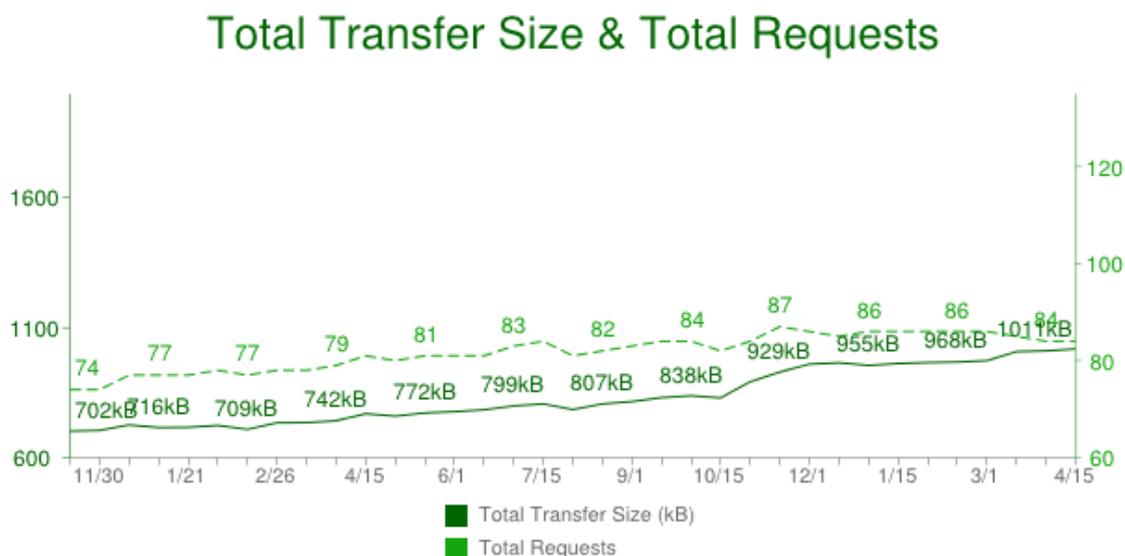


Figura 4.7 Fatores que envolvem uma página Web (HTTPARCHIVE, 2012)

A Figura 4.7 Fatores que envolvem uma página Web ilustra a evolução no tamanho médio de páginas Web, o qual ultrapassa a marca de 1 MB

contabilizando o documento HTML e suas dezenas de elementos. O custo para transferir tal página é considerável tendo em mente os diversos fatores estudados no capítulo 3.



**Figura 4.8** Evolução do tamanho de páginas Web (HTTPARCHIVE, 2012)

O Brasil alcança níveis cada vez maiores de qualidade a acesso a Internet (CGI.BR, 2010). Assumindo acesso de 2 Mbps, teoricamente uma página de 1MB deveria ser transferida em 4 segundos. Entretanto ao requisitar um documento HTML, o navegador segue uma série de procedimentos e restrições fazendo com que o tempo de acesso a uma página seja superior ao esperado.

A requisição de cada elemento da página Web respeita a ordem em que eles são declarados no documento HTML. Sendo assim, primeiramente o navegador recebe o HTML para então iniciar as requisições subsequentes, recebendo então os elementos que compõem a página.

A melhor maneira de demonstrar o progresso de carregamento de uma página é através de um gráfico cascata e está representado pela Figura 4.9 Diagrama Cascata de Carregamento de Página. O eixo horizontal representa o tempo e o eixo vertical, de cima para baixo, é possível acompanhar a sequência de requisições e o tempo gasto por cada elemento, representado pelos retângulos.



**Figura 4.9 Diagrama Cascata de Carregamento de Página**

Como é possível observar, o carregamento de páginas Web envolvem diversos elementos e variáveis. Reconhecendo essa dificuldade, muitas técnicas foram desenvolvidas buscando extrair o máximo de eficiência desse sistema e assim reduzir o tempo de carregamento.

A seguir serão detalhadas algumas técnicas úteis no desenvolvimento de páginas Web que buscam alcançar menor tempo de carregamento e, conseqüentemente, maior nível de satisfação do usuário.

#### 4.6.1. Renderização Progressiva

Jacob Nielsen menciona em seus trabalhos a importância de dar ao usuário um *feedback* enquanto um sistema computacional processa uma tarefa complexa. (NIELSEN, 1997) Tal *feedback* pode ser expresso em diversas formas, mais ou menos eficientes quanto ao manter o usuário num nível de satisfação alto. Exemplos de mensagens seriam:

- Uma mensagem informando o tempo esperado para conclusão;
- Barra de progresso;
- Ícones indicadores de processamento, como uma ampulheta ou *spinner*.

Páginas Web 2.0 são complexas e necessitam de muito processamento ao serem requisitadas, fazendo com que o tempo de carregamento seja muito alto. Entretanto, não somente convém reduzir o tempo de processamento no servidor. É possível explorar formas de transferência e renderização em partes, a medida que a página é calculada e transferida. As Figura 4.10 e Figura 4.11 apresentam um comparativo da adoção dessa técnica, onde os elementos que compõem a página são requisitados antes do recebimento completo da página (SOUDERS, 2009).

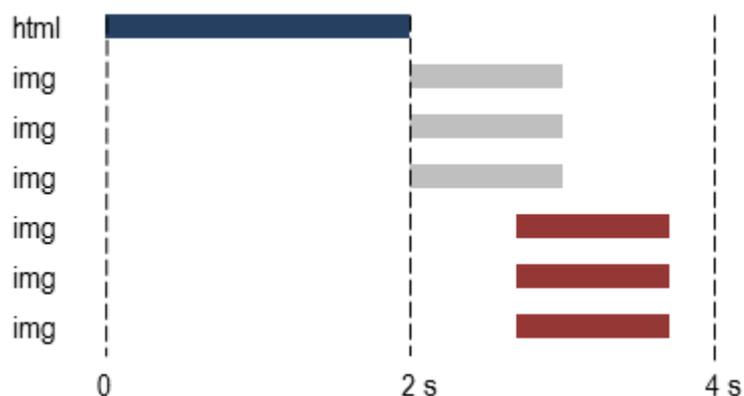
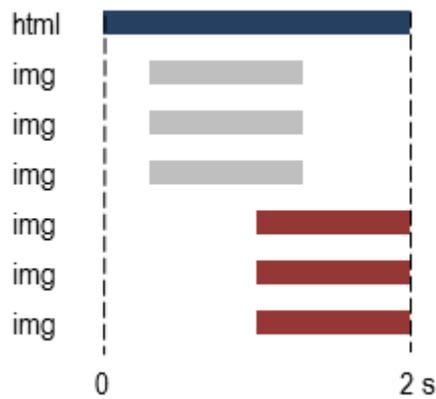


Figura 4.10 Carregamento sem renderização progressiva



**Figura 4.11 Carregamento com renderização progressiva**

A renderização progressiva de páginas Web consiste em oferecer ao usuário uma sensação de progresso, ao mesmo tempo em que transmite uma resposta consideravelmente rápida a requisição do usuário. O próprio carregamento da página já é um indicativo de que a tarefa está sendo processada, sendo possível uma estimativa de quando estará completa (SEOW, 2008).

Sendo assim, o desenvolvedor deve transmitir ao usuário o que é mais importante e que não demande muito processamento, conseguindo assim o menor tempo de resposta possível. As áreas mais importantes consistem no logotipo da página e áreas estáticas. Já em segundo e terceiro plano ficam os conteúdos calculados dinamicamente, que também são importantes, mas requerem certo processamento para serem retornados (STEFANOV, 2010).

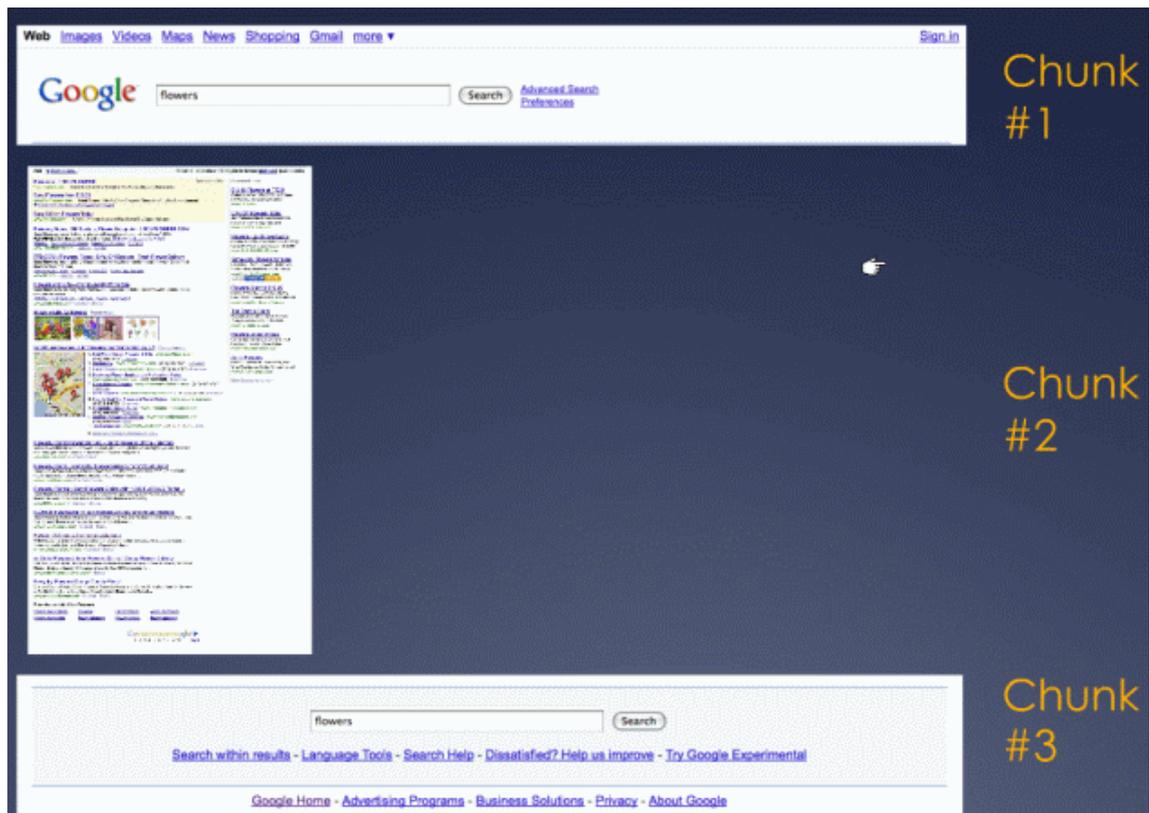


Figura 4.12 Renderização Progressiva de uma busca no Google (STEFANOV, 2010)

Como exemplo, a Figura 4.12 ilustra como a ferramenta Google.com transfere o conteúdo de uma busca em blocos, também chamados de *chunks*. Em um primeiro momento, o servidor retorna apenas o cabeçalho da página, ação que requer pouco processamento. Em seguida, é retornado o corpo da página, contendo o resultado da busca, o que demanda processamento considerável. Finalizando, o rodapé da página é carregado, o qual muitas vezes não é visualizado pelo usuário (SOUDERS, 2009) (STEFANOV, 2010).

#### 4.6.2. Paralelização com Múltiplos Domínios

A técnica que utiliza múltiplos domínios procura superar o limite de conexões que o navegador pode estabelecer com o servidor durante o carregamento da página. A especificação de HTTP/1.1 (RFC2616, 1999) limita em somente duas conexões simultâneas por servidor, porém os desenvolvedores foram além desse limite, sendo que os navegadores atuais alcançam até seis conexões por servidor.

O ganho no tempo de carregamento ao paralelizar a transferência de elementos é significativo. Entretanto, essa técnica deve ser utilizada com cuidado devido ao *overhead* causado por cada resolução de nome de domínio necessária para cada novo domínio ou subdomínio incluídos.

| Web site  | Images | Scripts | Stylesheets | Number of domains |
|---|--------|---------|-------------|-------------------|
| <a href="http://www.aol.com/">http://www.aol.com/</a>   | 59     | 6       | 2           | 3                 |
| <a href="http://www.ebay.com/">http://www.ebay.com/</a>   | 33     | 5       | 2           | 3                 |
| <a href="http://www.facebook.com/">http://www.facebook.com/</a>   | 96     | 14      | 14          | 10                |
| <a href="http://www.google.com/search?q=flowers">http://www.google.com/search?q=flowers</a>               | 3      | 1       | 0           | N/A               |
| <a href="http://search.live.com/results.aspx?q=flowers">http://search.live.com/results.aspx?q=flowers</a> | 6      | 1       | 4           | 5                 |
| <a href="http://www.msn.com/">http://www.msn.com/</a>   | 45     | 7       | 3           | 3                 |
| <a href="http://www.myspace.com/">http://www.myspace.com/</a>   | 16     | 14      | 2           | 3                 |
| <a href="http://en.wikipedia.org/wiki/Flowers">http://en.wikipedia.org/wiki/Flowers</a>                   | 33     | 6       | 9           | 2                 |
| <a href="http://www.yahoo.com/">http://www.yahoo.com/</a>   | 28     | 4       | 1           | 1                 |
| <a href="http://www.youtube.com/">http://www.youtube.com/</a>   | 23     | 7       | 1           | 5                 |

**Tabela 4.2 Subdomínios de Websites (SOUDERS, 2009)**

O limite de conexões em paralelo é dado por domínio e não por IP. Sendo assim, todos os recursos podem ser armazenados na mesma máquina sobre mesmo IP, porém utilizando domínios diferentes. Como solução, basta configurar os domínios secundários com um registro DNS CNAME apontando para o IP principal (SOUDERS, 2009).

De acordo com Yahoo!, o número ideal de subdomínios é específico para cada aplicação e depende da quantidade total de elementos, seu tamanho e complexidade da página (THEURER, 2007).

### 4.6.3. Folhas de Estilo no Topo

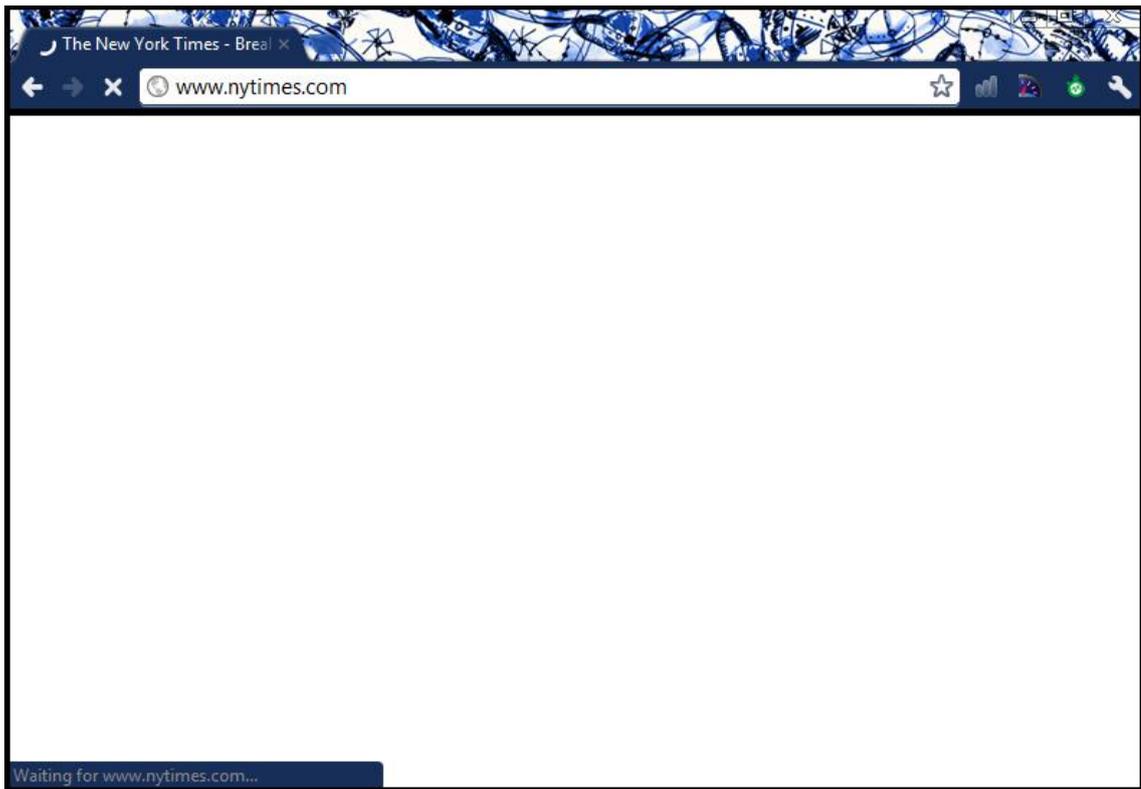
Folhas de estilo são também chamadas CSS (do inglês *Cascating Style Sheets*) e são responsáveis por definir o estilo utilizado na página, quanto a cores, fontes e imagens utilizadas (RFC2318, 1998).

Por terem importante papel na apresentação do conteúdo da página, os navegadores atrasam a renderização do documento HTML até que todos os arquivos CSS estejam transferidos e processados. Desenhar os elementos na tela requer muito processamento e, ao atrasar essa ação, o navegador consegue evitar subseqüentes renderizações.

Os componentes de uma página Web são requisitados de acordo com a seqüência que são referenciados no documento HTML. Sendo assim, a boa prática é referenciar os elementos CSS no início, no cabeçalho do documento, entre tags <head></head>. Do contrário, adicionar referências para arquivos CSS no final do documento atrasa o início da renderização até que todos os elementos anteriores estejam processados.

Essa técnica complementa a prática de renderização progressiva citada previamente, fazendo com que o carregamento gradual da página forneça ao usuário uma noção de progresso, indicando que o trabalho requisitado está sendo realizado.

O objetivo principal dessa técnica é evitar a indesejada tela branca (*White Blank Screen of Death*), representada pela Figura 4.13. Ela indica que o navegador está processando e recebendo as informações da página e aguardando o tempo certo para mostrá-la ao usuário.



**Figura 4.13 White Blank Screen of Death**

Portanto, convém ressaltar a importância de referenciar arquivos CSS no topo do documento HTML, para que os mesmos sejam requisitados e analisados o quanto antes, habilitando o navegador a desenhar a página o quanto antes.

#### **4.6.4. CSS Spriting**

Reconhecendo-se o impacto que o retardo de redes de computadores exerce sobre aplicações distribuídas (DANTAS, 2010), observa-se que o estabelecimento de conexões TCP para cada requisição de elemento de uma página Web é custoso. Sendo assim, foram criadas técnicas que reduzem a necessidade de novas conexões, agrupando múltiplos elementos.

Seguindo esse princípio, a técnica de Spriting consiste em agrupar elementos gráficos a fim de reduzir o número de conexões TCP. Essa técnica foi e ainda é muito usada para jogos eletrônicos convencionais, principalmente

quando a memória disponível era limitada e o *overhead* para transferir ou manter muitas imagens em memória era grande demais (SHEA, 2004).

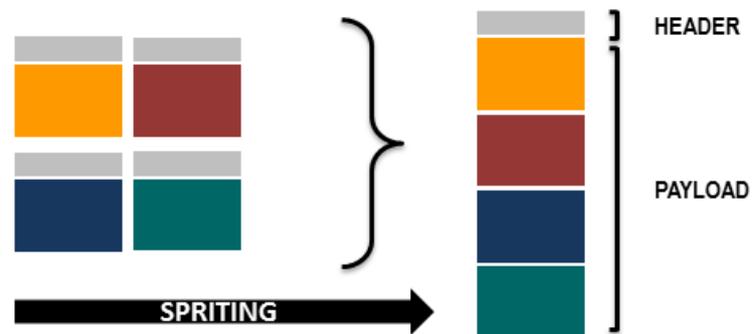


Figura 4.14 Princípio de CSS Spriting

Em páginas Web outros fatores tornam o uso de sprites interessante e acarretam ganho no tempo de carregamento. Inicialmente, ao agrupar, por exemplo, dez imagens em uma, o número de conexões resultantes será apenas um, o que implicará consideravelmente no desempenho da página (SOUDERS, 2008).

Outra vantagem não tão fácil de ser notada é o *overhead* implícito em cada imagem, pertinentes a cada codificação utilizada. Citando o exemplo anterior, ao agrupar as dez imagens, os bytes de nove cabeçalhos não serão mais necessários (BOUTELL, 1996).

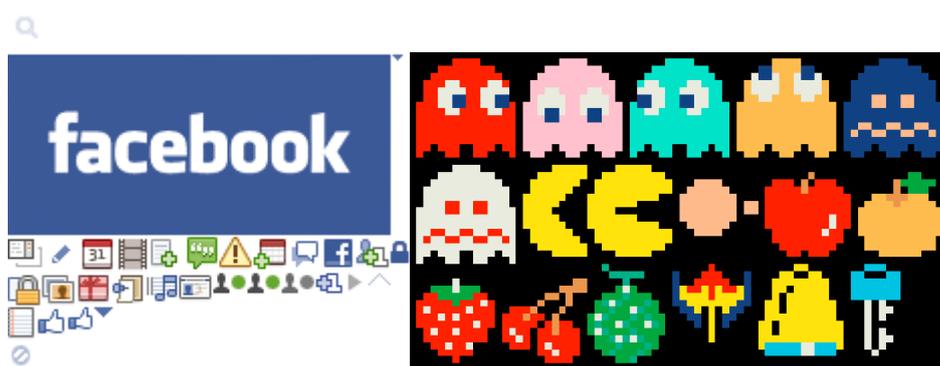


Figura 4.15 Sprites utilizados em Websites e games conhecidos (FACEBOOK, 2012)  
(SPRITESTITCH, 2008)

A forma de posicionar sprites numa página Web consiste em utilizar propriedades CSS de posicionamento de plano de fundo. Define-se o sprite

como imagem de plano de fundo de todos os elementos ilustrados por ele e, utilizando o atributo CSS *background-position*, é possível reposicionar a imagem corretamente para cada elemento da página (COYIER, 2009).

#### 4.6.5. DATA: URI e MHTML

Seguindo o mesmo princípio apresentado na seção 4.6.1, é possível incorporar ao documento pequenas imagens, codificadas em base 64. Essa técnica utiliza os padrões Data: URI (RFC2397, 1998) e MHTML (RFC2557, 1999).

A Tabela 4.3 demonstra como uma imagem pode ser codificada em base 64 e adicionada num documento HTML. O tamanho e complexidade da imagem resultam proporcionalmente no tamanho da sequência codificada.

| Data: URI   | Imagem  |
|---|---|
| <pre>&lt;img<br/>src="data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAAA8AAAAPCAIAAAC0tAldAA<br/>AAAXNSR0IArs4c6QAAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQA<br/>AAAcSURBVChTY2BY9IIERIJSolMjqjEDdzRMKAwTAP/DWlqctmtUAAAAEIFTkSuQmCC"&gt;</pre> |  |

Tabela 4.3 Demonstração de Data: URI

Como consequência, não há necessidade de criar requisição para tal imagem, já que no momento que o arquivo HTML é requisitado, a imagem já o acompanha. Porém algumas dificuldades acompanham seu uso, como a impossibilidade de armazenar em cache e a manutenção das imagens.

#### 4.6.6. Minificação e Arquivos

A técnica de minificação tem o propósito de reduzir o tamanho de código de páginas Web, removendo conteúdo desnecessário para a execução do arquivo, tal qual espaços em branco, tabulações, comentários, etc. (YAHOO!, 2012).

A Tabela 4.4 apresenta os resultados de duas ferramentas, JSMin (CROCKFORD, 2003) e Dojo Compressor (DOJO, 2012). Observa-se que o ganho médio em tamanho alcança os 25%, resultando em uma técnica considerável.

| Web site                  | Tamanho original | JSMin     | Dojo Compressor |
|---------------------------|------------------|-----------|-----------------|
| http://www.amazon.com/    | 204K             | 31K (15%) | 48K (24%)       |
| http://www.aol.com/       | 44K              | 4K (10%)  | 4K (10%)        |
| http://www.cnn.com/       | 98K              | 19K (20%) | 24K (25%)       |
| http://www.myspace.com/   | 88K              | 23K (27%) | 24K (28%)       |
| http://www.wikipedia.org/ | 42K              | 14K (34%) | 16K (38%)       |
| http://www.youtube.com/   | 34K              | 8K (22%)  | 10K (29%)       |
| Média                     | 85K              | 17K (21%) | 21K (25%)       |

Tabela 4.4 Redução de tamanho utilizando JSMin e Dojo Compressor (SOUDERS, 2008)

Por conseguinte, a técnica de minificação de código apresenta um bom resultado e economia considerável no tempo de transferência de arquivos que compõem uma página Web.

#### 4.6.7. Posicionamento de JavaScript

JavaScript tornou possível o avanço de páginas Web para a chamada Web 2.0, graças ao dinamismo e interatividade que proporciona. A adoção de scripts em páginas HTML torna-as mais ricas e complexas acarretando também em maior dificuldade quanto a transferência da página para o navegador (GARRETT, 2005).

Atualmente tornou-se comum o uso de frameworks JavaScript como jQuery e MooTools e muitos plugins que facilitam o desenvolvimento, porém resultam em peso extra de processamento e transferência (JQUERY, 2006) (MOOTOOLS, 2006).

Durante o carregamento de arquivos JavaScript externos ao documento HTML, os navegadores costumam bloquear as transferências subsequentes até que o *script* esteja completamente baixado, analisado e executado, o que atrasa muito a visualização da página. Sendo assim, é interessante adicionar scripts JavaScript de maneira mais inteligente possível (SOUDERS, 2008).

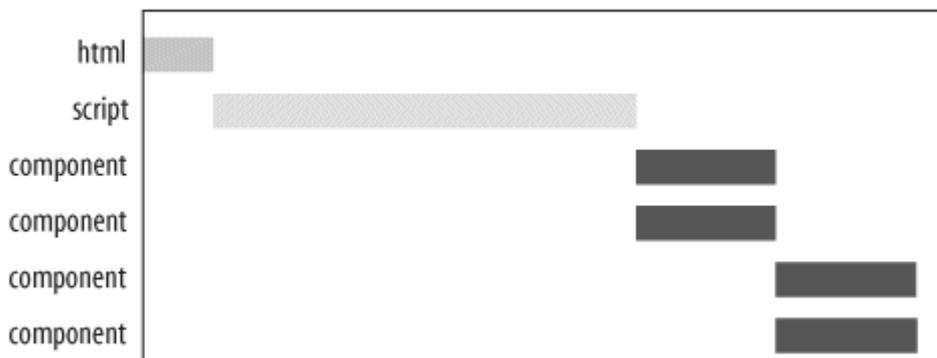


Figura 4.16 Bloqueio devido a código JavaScript

Como é possível observar na Figura 4.16 scripts adicionados no topo do documento HTML bloqueiam o acesso a elementos subsequentes tornando o carregamento muito mais demorado.

O ideal seria adicionar scripts no final do documento HTML, resultando em menor impacto final no carregamento, sendo que todos os elementos da página já foram transferidos para o cliente. Esse cenário é apresentado pela Figura 4.17. Observe que os componentes que compõem a página foram carregados antes do código JavaScript, e possivelmente renderizando a página mais cedo.

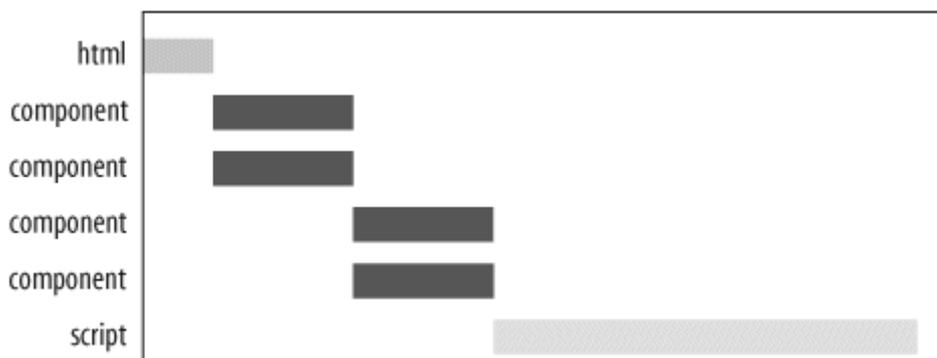


Figura 4.17 Javascript movido para o fim

Entretanto, essa opção não é aceitável em muitas aplicações. Isto porque o script JavaScript pode ser requisitado durante o carregamento dos outros elementos e não pode ser movido para o final. Essa é uma difícil decisão de projeto, porém investimento de tempo nessa técnica resulta em ganho substancial em desempenho (SOUDERS, 2009).

Como saída, muitos Websites complexos como Facebook e Google Calendar buscam dividir o script deixando somente o mínimo necessário no topo do documento, atrasando ao máximo o carregamento do restante do script (SOUDERS, 2009).

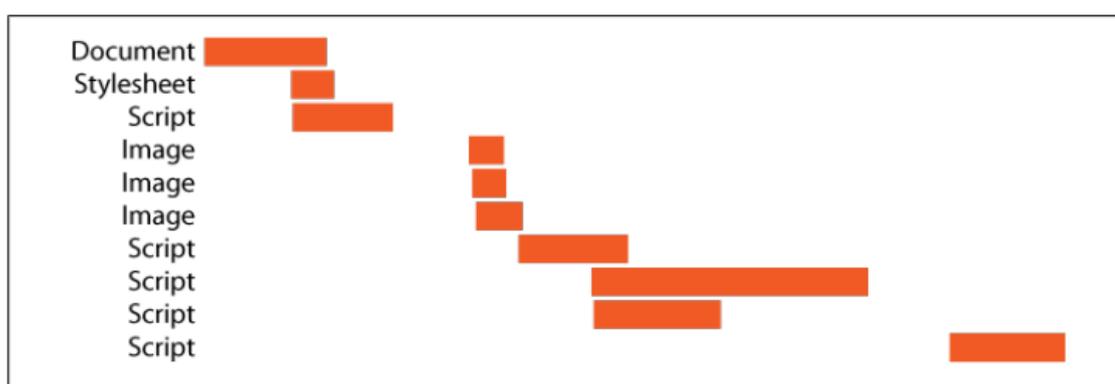


Figura 4.18 Carregamento de Google Calendar. Apenas JavaScript crucial carregado no topo

Existem ferramentas como Microsoft Doloto (LIVSHITS e KICIMAN, 2008) e ControlJS (SOUDERS, 2010) que auxiliam nessa divisão de *payload*, posicionando no início somente o código realmente importante durante o carregamento da página.

As soluções mais comuns para esse problema são:

1. Posicionar scripts externos no final do documento HTML.
2. Carregar scripts de maneira assíncrona
3. Solicitar scripts adicionais o mais tardar possível, sobre demanda.

A escolha de qual solução utilizar dependerá da complexidade da aplicação e o quão crítico são os scripts para o carregamento da página (SOUDERS, 2009).

#### **4.6.8. Compactação**

Tendo em vista a grande complexidade de páginas Web atuais, as quais necessitam de diversos recursos, cada vez maiores e mais difíceis de transferir para o cliente (HTTPARCHIVE, 2012), convém analisar a estrutura de cada arquivo e a possibilidade de compactar seus dados utilizando algoritmos de compactação padrão.

De acordo com Radhakrishnan (SRINIVASAN, 2003) e especificado por (RFC2616, 1999), a compactação HTTP pode ser feita nos níveis de conteúdo e de transferência.

Compactação de conteúdo, ou compactação estática, consiste em manter o arquivo compactado enquanto armazenado, sem necessidade dos dados serem processados pelo servidor. A medida que o conteúdo for requisitado pelo cliente, não há necessidade de compactá-lo ou descompactá-lo. Como exemplo podemos citar imagens, vídeos, flash, etc.

Compactação por transferência é aplicada a medida que o dado é requisitado pelo cliente, principalmente a conteúdo dinâmico, criado no momento da requisição. Enquadram-se nessa categoria arquivos HTML, CSS, JavaScript, XML, etc. Diferente do método de compactação estático, esse método requer explícito ajuste no que se diz respeito a configurações do servidor HTTP, o que infelizmente ainda é pouco utilizado (YAHOO!, 2009).

De acordo com a especificação HTTP/1.1, os algoritmos utilizados para compactação de transferência são GZIP (RFC1952, 1996) e DEFLATE (RFC1951, 1996). GZIP é baseado em DEFLATE, o qual é uma combinação do algoritmo LZ77 e com a Codificação de Huffman.

Tony Gentilcore (GENTILCORE, 2009) comparou diferentes tipos de otimização de conteúdo não comprimido com resultados de compressão gzip. Analisando o top 10 Websites do ranking Alexa, conseguiu interessantes resultados, representados pela Tabela 4.5.

| Técnica de Otimização                   | Redução de tamanho |
|---|--------------------|
| Delegação de eventos JavaScript         | 5%                 |
| URLs relativas                          | 3%                 |
| Remover espaços em branco e comentários | 4%                 |
| Estilos inline                          | 1%                 |
| Compressão gzip                         | 70%                |

Tabela 4.5 Tipos de compressão

O resultado comprova que compressão GZIP impera sobre qualquer outra técnica de otimização de conteúdo, reduzindo a quantidade de Bytes transferidos em até 70%.

GZIP compacta dados dinâmicos calculados durante a requisição através de módulos instalados no servidor, como mod\_gzip e mod\_deflate para apache (KOLICH, 2009).

#### 4.6.9. Gerenciamento de Cookies

Cookie HTTP é um pequeno bloco de informação armazenado no navegador para melhorar a experiência do usuário ao navegar em um Website. Os dados armazenados são variados e podem conter um identificador único de sessão, ou oferecer a capacidade de o servidor reconhecer o navegador em um segundo acesso ao Website (RFC6265, 2011).

Para desempenho, o principal atributo de um cookie é o domínio em que ele está associado. A cada requisição feita a esse domínio, o navegador transfere o valor de todos os cookies para o servidor, com o objetivo de identificar-se. Como exemplo, pode-se vincular cookies a um domínio “exemplo.com.br”. Qualquer requisição de conteúdo a esse domínio, incluindo seus subdomínios forçará o navegador a adicionar aos cabeçalhos HTTP os cookies que detém (SLOUDERS, 2009).

Sendo assim, qualquer requisição de componentes de uma página, incluindo o documento HTML, imagens e arquivos texto resultam em demasiadas transferências de cookies entre navegador e servidor, o que pode ser facilmente contornado ao hospedar elementos estáticos em domínios diferentes do associado ao cookie (YAHOO!, 2012).

A prática mais indicada é associar cookies ao domínio “www.exemplo.com” e hospedar somente o documento HTML a esse domínio. Qualquer outro elemento deve ser hospedado em um subdomínio, como “estatico.exemplo.com”. Sendo assim, cookies serão transmitidos somente uma única vez (THEURER, 2007).

#### 4.6.10. Redes de Fornecimento de Conteúdo

Redes de Fornecimento de Conteúdo, mais comumente referenciadas como *Content Delivery Networks* (CDN), distribuem geograficamente o conteúdo estático de um Website, replicando o servidor principal, com o objetivo de reduzir a distância e assim, diminuir a latência inerente de rede.

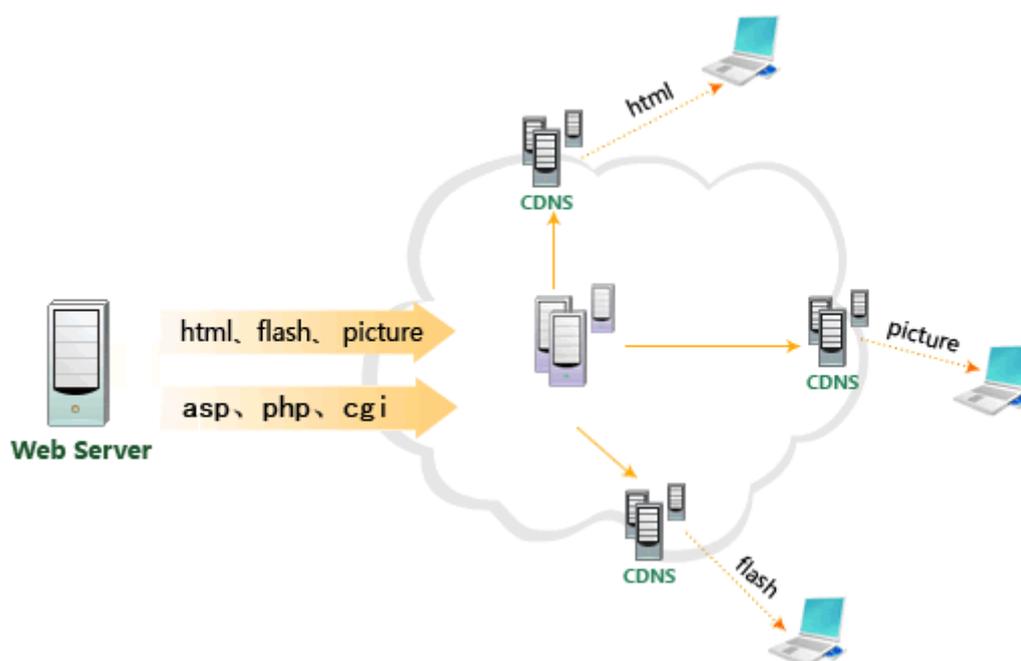


Figura 4.19 Rede de Fornecimento de Conteúdo (CSLONGWANG, 2012)

No Brasil esse serviço ainda é pouco explorado. Sites Brasileiros de grande porte detém sua própria infraestrutura de CDN e empresas como Azion oferecem o serviço no mercado brasileiro (AZION, 2012) (PAUKA, 2009).

## **4.8. Considerações Finais**

Nesse capítulo foi apresentada a proposta desse trabalho, a qual consiste em uma survey das técnicas de melhoria do desempenho de páginas Web. Primeiramente foram demonstrados fatos que motivaram esforços nesse foco de pesquisa. Em segundo lugar, aspectos relativos a percepção humana são explicados e como nos influenciam negativamente. Para finalizar, 10 técnicas que acarretam melhoria de desempenho de páginas foram detalhadas, criando um arcabouço de conhecimento necessário para execução dos experimentos relatados no capítulo 5 desse trabalho.

## 5. Ambiente e Resultados Experimentais

Os capítulos 2, 3 e 4 constituíram um levantamento de conceitos e técnicas relevantes para se obter melhorias no desempenho de páginas Web.

Neste capítulo, primeiramente serão estudados conceitos técnicos específicos do tema proposto, necessários para análise minuciosa do desempenho de páginas Web. Em seguida, serão ilustrados três estudos de casos no processo de adoção das técnicas aqui reunidas e consecutivamente os resultados obtidos.

### 5.1. Conceitos

A seguir serão abordados alguns conceitos relativos necessários para o entendimento do carregamento de páginas Web. Para facilitar o estudo, será utilizado o diagrama de cascata, o qual é a forma mais indicada para representar o carregamento de páginas (BEHESHTI, 2011).

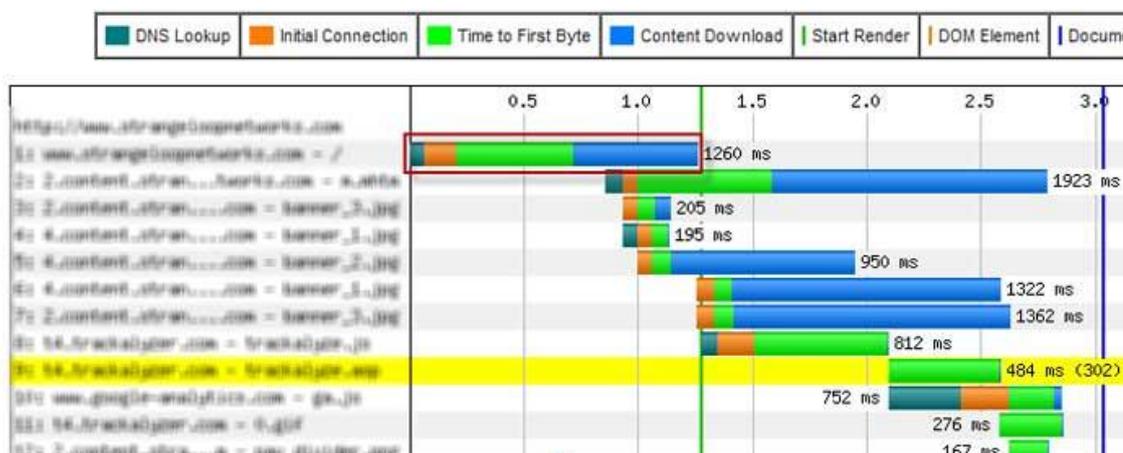


Figura 5.1 Diagrama Cascata (BEHESHTI, 2011)

A Figura 5.1 apresenta um diagrama cascata descrevendo o carregamento de uma página Web. Cada barra horizontal representa um componente da página, sendo que o eixo vertical representa a ordem em que são processadas e o eixo horizontal, o tempo. O código de cores apresenta diversos estágios durante o carregamento da página e de cada elemento que a compõe. A Figura

5.2 apresenta o detalhe de como os quatro estágios de carregamento de um elemento da página Web são representados no diagrama cascata.



Figura 5.2 Estágios de carregamento de um elemento de página

Os estágios são em sequência:

- **Resolução de Nome de Domínio:**

Tempo necessário para tradução do domínio em endereços IP (KUROSE e ROSS, 2010).

- **Estabelecimento de Conexão TCP:**

Processo necessário para início de transferência de dados com o protocolo HTTP (RFC793, 1981).

- **Tempo para Recebimento de Primeiro Byte (TTFB):**

Do inglês *Time to First Byte*, é o tempo decorrido desde a requisição ao servidor até o recebimento do primeiro byte pelo cliente.

- **Tempo de Download:**

Tempo necessário para transferência completa do arquivo entre o servidor e cliente.

Tendo em vista as técnicas e conceitos aqui estudados, surge a necessidade em utilizar o conteúdo até então reunido, a fim de comprovar a eficácia em empregá-los em aplicações Web, consolidando o conhecimento e alcançando os objetivos estabelecidos. Em seguida, serão descritos o ambiente experimental utilizado, composto por diversas técnicas e métodos que buscam o resultado esperado: a melhora do desempenho de páginas Web.

Finalizando o trabalho, estão reunidos três estudos de caso reais que demonstrarão a eficácia do conteúdo aqui estudado.

### 5.3. Ambiente Experimental

Com o intuito de dar legitimidade ao estudo realizado, esse capítulo irá descrever os testes elaborados, todos visando contrastar o impacto acarretado pelo uso das técnicas presentes neste documento.

Serão avaliados três Websites reais, quanto ao tempo de carregamento e arquitetura, podendo assim levantar os pontos de maior impacto no desempenho de cada aplicação.

As ferramentas que serão utilizadas para análise de desempenho e otimização das páginas são:

- **WebPagetest.org:** Ferramenta online gratuita que analisa minuciosamente o processo de carregamento de páginas Web (MEENAN, 2008).
- **Firebug:** Extensão do navegador Mozilla Firefox (FIREBUG, ?).
- **Pingdom Tools Full Page Test:** Ferramenta online similar a WebPagetest.org (PINGDOM, ?).
- **CSS Compressor:** Compressor online de arquivos CSS (MINIFYCSS, ?).
- **JavaScript Compressor:** Compressor online para arquivos JavaScript (MINIFYJAVASCRIPT, ?).
- **SpriteMe:** ferramenta de criação automática de CSS Sprites (SPRITEME, ?).
- **Smush.it:** Otimizador online de imagens (SMUSH.IT, ?).
- **Yslow:** Ferramenta que analisa páginas Web quanto a adoção de boas práticas de desempenho (YSLOW, ?).

Com a união das ferramentas apresentadas, será possível uma análise minuciosa de cada Website seguindo 10 passos:

1. Remover referências inválidas (erro 404)
2. Agrupar imagens criando CSS Sprites
3. Substituir arquivos de imagens por DATA:URI
4. Unificar scripts e folhas de estilo

5. Minificar scripts e folhas de estilo
6. Otimizar conteúdo de imagens
7. Ordenar scripts e folhas de estilo
8. Adicionar subdomínios
9. Habilitar conexão persistente
10. Ativar compactação gzip

Com o objetivo de avaliar os resultados obtidos, foram escolhidos cinco indicadores de desempenho:

- **Requisições:** número de requisições HTTP necessárias para carregar a página;
- **Tamanho:** soma do tamanho de todos os recursos que compõem a página;
- **Início renderização:** tempo em que o navegador inicia o desenho da tela;
- **Carregamento concluído:** evento disparado pelo navegador quando a página está completa;
- **Nota YSlow:** Resultado obtido da análise feita pelo serviço YSlow quanto ao desempenho da página (YSLOW, ?).

Ao reconhecer a lista de 10 passos e tendo em mente os conceitos listados na seção 5.1 desse documento, será possível levantar os problemas presentes nas páginas Web que serão avaliadas.

Em primeiro lugar será analisada a página principal da Universidade Federal de Santa Catarina. Em seguida, serão demonstradas as intervenções aplicadas a página principal da Korin Agricultura Natural, empresa brasileira de agricultura orgânica. Finalizando o estudo, a análise da página principal do site da Aproxima, uma agência de comunicação com sede na cidade de São Paulo.

## 5.5. Estudo de Caso 1: ufsc.br

O primeiro estudo de caso desse trabalho abordará a análise da página principal da Universidade Federal de Santa Catarina, localizada com o endereço eletrônico <http://ufsc.br>.



Figura 5.3 Página Inicial da Universidade Federal de Santa Catarina (UFSC.BR, ?)

### 5.5.1. Análise do Site

Em conjunto com os técnicos da Superintendência de Governança Eletrônica e Tecnologia da Informação e Comunicação da UFSC (SeTIC), foi possível analisar a estrutura da página inicial da universidade e discutir as possíveis intervenções e dificuldades presentes nesse projeto.

A seguir as características presentes na página passíveis de intervenção:

- Imagens não otimizadas;
- Demasiado número de arquivos externos de estilo e script;
- Diversas imagens passíveis de agrupamento;
- Arquivos não minificados;
- Bibliotecas desatualizadas;
- Cookies HTTP vinculados a ufsc.br.

Como base de sua estrutura, a UFSC utiliza a plataforma Wordpress para gerência de conteúdo de seu Website principal (WORDPRESS, 2003). Ao optar pelo uso de um framework como esse, o gerenciamento do site torna-se mais fácil e seguro. Entretanto, acopla o desenvolvedor a uma arquitetura muitas vezes demasiada complexa para a necessidade da página. Todas as

características que acompanham um framework devem ser cuidadosamente avaliadas quanto a sua real aplicabilidade no desenvolvimento do site.

Após identificar os pontos críticos passíveis de intervenção, segue-se utilizando as ferramentas listadas na seção 5.3 deste documento, buscando otimizar o tempo de carregamento da página.

### **5.5.2. Procedimentos Utilizados**

Com o objetivo de extrair a máxima eficiência da página da universidade, foram aplicadas as seguintes técnicas:

- (2). Agrupar imagens criando CSS Sprites
- (3). Substituir arquivos de imagens por DATA:URI
- (4). Unificar scripts e folhas de estilo
- (5). Minificar scripts e folhas de estilo
- (6). Otimizar conteúdo de imagens
- (7). Ordenar scripts e folhas de estilo
- (8). Adicionar subdomínios

Dos 10 passos apresentados, apenas as ações 1 (Remover 404), 9 (keep-alive) e 10 (gzip) não foram aplicadas. Isto porque o site não apresentou nenhuma referência inválida e keep-alive e gzip já estavam habilitados.

Vale ressaltar que, dada a impossibilidade de acessar o código fonte da página da UFSC, os resultados referentes a tempo de carregamento aqui obtidos foram estimados, baseando-se na proporção de redução de tamanho e redução de conexões necessárias.

A seguir serão apresentados os resultados obtidos.

#### 5.5.4. Resultados Obtidos

De uma maneira geral, a intervenção da página da UFSC resultou um ganho em todos os indicadores, com destaque para a redução em 68% de requisições e 48% do tempo total de carregamento. O tempo de início de renderização foi reduzido em 36,9% e o tamanho da página tornou-se 17% menor. A avaliação de YSlow subiu de 80 para 85, o que é considerável. A Figura 5.4 ilustra os resultados obtidos.

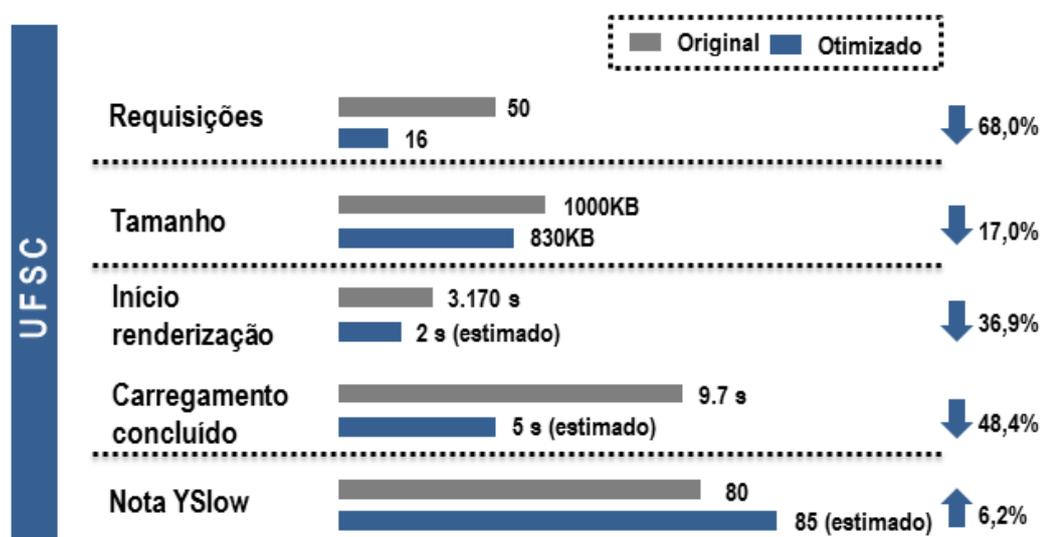


Figura 5.4 Resultados de Intervenções em UFSC.br

A Figura 5.5 apresenta o *sprite* criado para utilizar no site da universidade. Com ele foi possível reduzir o número de requisições HTTP de 22 para apenas uma.



Figura 5.5 Sprite criado para UFSC.br

Embora o Website da UFSC utilize o framework WordPress como base e o mesmo ofereça certa inflexibilidade quanto a customizações, é possível alcançar certo ganho sem comprometer a estrutura padrão. Por ser um

framework baseado em extensões, WordPress oferece outras facilidades, como o plugin W3 Total Cache, que promete otimizar facilmente o desempenho de aplicações WordPress.

## 5.6. Estudo de Caso 2: [www.korin.com.br](http://www.korin.com.br)

Nessa seção será abordada a análise realizada sobre o Website institucional da Korin Agricultura Natural. A Korin é uma empresa especializada em agricultura orgânica situada em São Paulo, e seu site utiliza a plataforma Microsoft .NET (MICROSOFT, ?).



Figura 5.6 Página Inicial da Korin Agricultura Natural (KORIN, ?)

### 5.6.1. Análise do Site

A página principal do site da Korin apresenta um design rico e um rotacionador automático de imagens que utiliza o framework jQuery (JQUERY, 2006) como base. Após a análise completa do site, os erros encontrados que implicam em perda de desempenho foram:

- Imagens não otimizadas;
- Arquivos não minificados;
- Bibliotecas duplicadas e desatualizadas;
- Referências inválidas (erro 404);
- Scripts e códigos mal posicionados;
- Conexão persistente desabilitada;
- Compactação gzip desativada.

### **5.6.2. Procedimentos Utilizados**

Das 10 técnicas apresentadas, oito foram aplicáveis no site da Korin, sendo elas:

- (1). Remover referências inválidas (erro 404)
- (4). Unificar scripts e folhas de estilo
- (5). Minificar scripts e folhas de estilo
- (6). Otimizar conteúdo de imagens
- (7). Ordenar scripts e folhas de estilo
- (8). Adicionar subdomínios
- (9). Habilitar conexão persistente
- (10). Ativar compactação gzip

Pela alta complexidade das imagens utilizadas no design do site, não foi possível criar CSS sprites ou DATA:URI's. É importante observar que os passos 9 e 10 não estavam efetivamente ativados nesse caso. Ao habilitar ambas as opções a economia foi substancial.

### **5.6.3. Resultados Obtidos**

Por ser um site repleto de imagens, os passos referentes a compactação exerceram maior impacto, resultando em 67% de redução no tamanho total da página. Outro fator que contribuiu para esse resultado foi a remoção de bibliotecas jQuery duplicadas e não minificadas, visto que seu tamanho é considerável.

Outro valor que se destaca é o tempo de início de renderização, que apresentou ganho de 27%, reduzindo de 3 para 2.2 s. A Figura 5.7 demonstra os resultados obtidos.

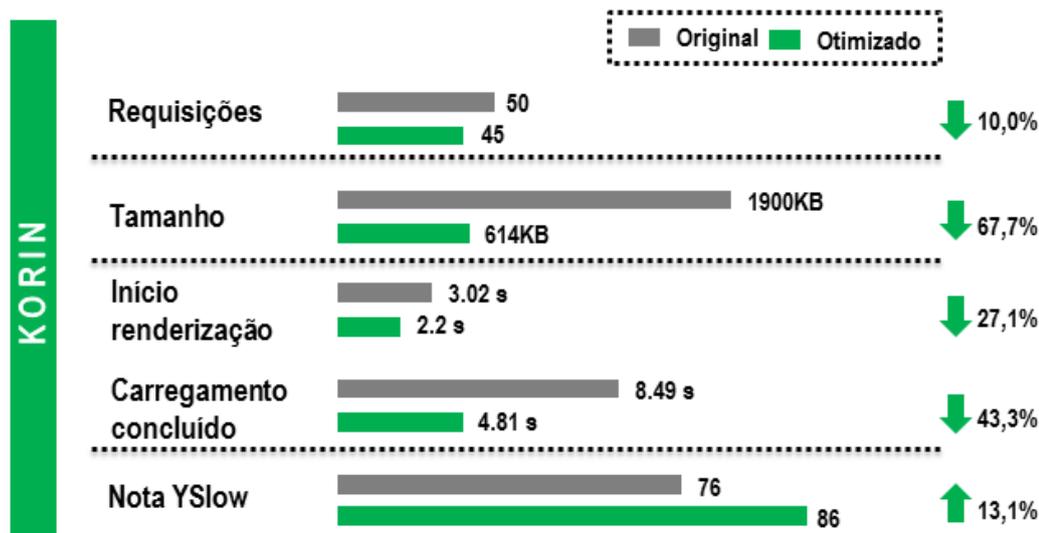


Figura 5.7 Resultados de Intervenções em [www.korin.com.br](http://www.korin.com.br)

O tempo de carregamento total da página caiu 43%, de 8.49s para 4.81, o que demonstra grande melhora no desempenho. Para finalizar, a avaliação YSlow subiu 10 pontos, alcançando a marca dos 86 pontos.

## 5.7. Estudo de Caso 3: [www.aproxima.com.br](http://www.aproxima.com.br)

Aproxima é uma agência de comunicação localizada na cidade de São Paulo. Seu site funciona com WordPress e a plataforma é Microsoft .NET. Em seguida será retratada a análise realizada no site <http://www.aproxima.com.br>.



Figura 5.8 Página Inicial da Aproxima Comunicação (APROXIMA, ?)

### **5.7.2. Análise do Site**

O Website da Aproxima tem aspecto de blog e apresenta um design limpo com poucas imagens, porém todas com tamanho significativo. A seguir os pontos passíveis de otimização presentes no site:

- Imagens não otimizadas;
- Arquivos não minificados;
- Bibliotecas duplicadas e desatualizadas;
- Referências inválidas (erro 404);
- Scripts e códigos mal posicionados;
- Compactação gzip desativada.

### **5.7.3. Procedimentos Utilizados**

Buscando solucionar os fatores que comprometem o desempenho da página Web, foram aplicados os seguintes passos:

- (1). Remover referências inválidas (erro 404)
- (2). Agrupar imagens criando CSS Sprites
- (3). Substituir arquivos de imagens por DATA:URI
- (4). Unificar scripts e folhas de estilo
- (5). Minificar scripts e folhas de estilo
- (6). Otimizar conteúdo de imagens
- (7). Ordenar scripts e folhas de estilo
- (10). Ativar compactação gzip

O passo 8 (Subdomínios) não foi aplicado pois resultou em perda de desempenho de carregamento da página. A conexão persistente coberta pelo passo 9 já estava habilitada.

Os passos 2 e 3 foram utilizados em comunhão. Primeiramente foi criado um sprite contendo três imagens, representado pela Figura 5.9.



Figura 5.9 Sprite Aproxima

Só então foi possível a criação da tag DATA:URI, baseando-se no sprite:

```

```

Tabela 5.1 DATA:URI aproxima.com.br

Como resultado, reduziu-se o número de conexões necessárias de três para zero e o tamanho em 16%, de 4.8 KB para 4KB.

#### 5.7.4. Resultados Obtidos

A intervenção no site da Aproxima foi sem dúvida a mais significativa desse trabalho, com ganhos como 57% na redução do tamanho da página e 78% menos requisições realizadas. A Figura 5.10 apresenta os resultados compilados.

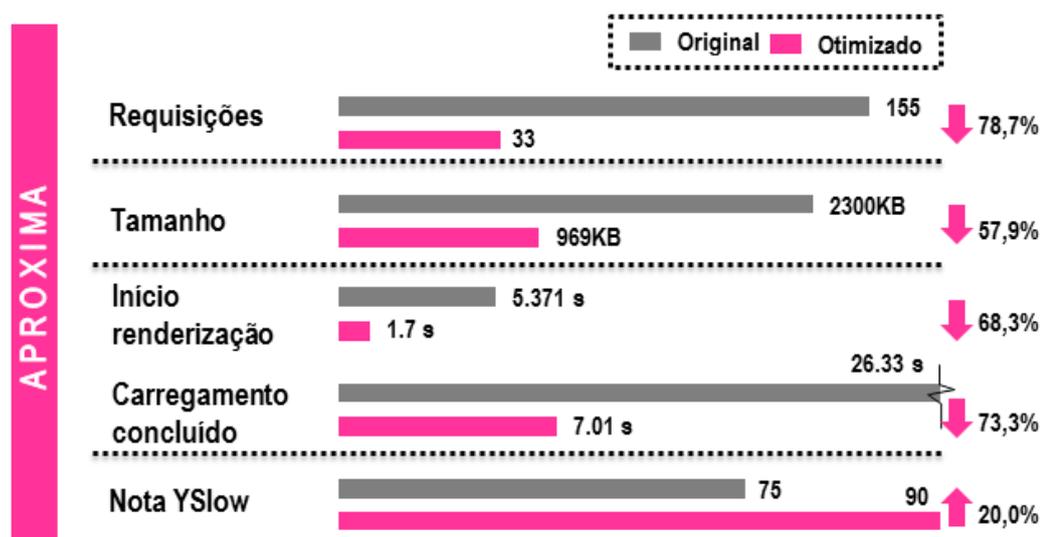


Figura 5.10 Resultados de Intervenções em www.aproxima.com.br

Por apresentar um design limpo, o Website da Aproxima tende a ter desempenho elevado. Entretanto, apresentava problemas como imagens não otimizadas e conexões massivas com elementos de redes sociais. Tais otimizações renderam 20 pontos adicionais na escala YSlow o que comprova o grande resultado obtido.

## 5.8. Resultados Experimentais

De uma maneira geral, os três Websites aqui analisados foram passíveis de melhoria e apresentaram substanciais ganhos de desempenho. A Figura 5.11 apresenta a média dos ganhos em cada fator indicativo.

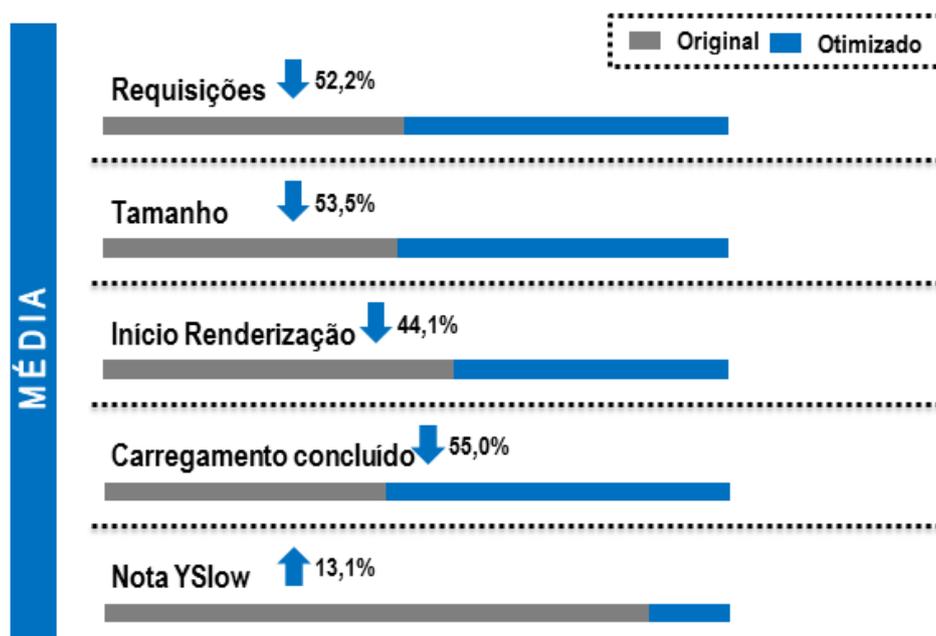


Figura 5.11 Média de melhorias dos casos

Em média, a quantidade de requisições reduziu 52%, as páginas ficaram 53% menores e houve 13% de crescimento na qualificação YSlow. Quanto aos tempos de carregamento, os navegadores iniciaram a renderização da tela 44% mais rápido e levou 55% menos tempo para carregar páginas por completo.

## 6. Conclusão e Trabalhos Futuros

Esse trabalho consistiu em uma *survey* para melhoria de desempenho de aplicações Web, incluindo 10 técnicas úteis no desenvolvimento de qualquer aplicação Web.

Diferente de outros fluxos de pesquisas voltados a otimização algorítmica e banco de dados, esse documento foca no tempo consumido no navegador do usuário, o que corresponde a até 80% do tempo de carregamento de páginas.

Com o intuito de demonstrar a eficiência das técnicas aqui abordadas, foram realizadas três intervenções em Websites reais, diferentes por natureza quanto a arquitetura e plataforma.

Os resultados apresentaram, em média, redução de 55% no tempo de carregamento, chegando em 73% em um dos testes. O tráfego necessário reduziu em média 53%, o que é um bom indicativo de eficiência.

Embora existam inúmeras combinações de plataformas e frameworks para desenvolvimento Web, provou-se que a adoção de tais técnicas é factível a qualquer aplicação Web.

Como trabalhos futuros, propõe-se uma abordagem mais formal visando sistematizar a pesquisa, a criação de um guia similar a um guia de estilos, com boas práticas de desempenho na Web e a realização de estudos mais aprofundados cobrindo técnicas mais avançadas e específicas.

## Referências Bibliográficas

- ABERDEEN, B. S. The Performance of Web Applications. Customers Are Won or Lost in One Second. Aberdeen. [S.l.]. 2008.
- AGARWAL, A. Web 3.0 Concepts Explained in Plain English. **Digital Inspiration**, 2009. Disponível em: <<http://www.labnol.org/internet/Web-3-concepts-explained/8908/>>. Acesso em: 31 mar. 2012.
- AKAMAI, C. R. Consumer Response to Travel Site Performance. Akamai. [S.l.]. 2010.
- AMAZON. Make Data Useful. Stanford Data Mining. [S.l.]: [s.n.]. 2006.
- AZION. Azion Content Delivery Network (CDN) – Mapa de Presença. Azion Technologies, 2012. Disponível em: <<http://www.azion.com.br/network-br/>>. Acesso em: 10 maio 2012.
- BARAN, P. On Distributed Communications Networks. RAND Corporation. [S.l.]. 1962.
- BEHESHTI, H. Waterfalls 101: How to Read a Waterfall Chart. Strangeloop Networks Blog, 2011. Disponível em: <<http://www.strangeloopnetworks.com/resources/Webinars/waterfalls-101-how-to-read-a-waterfall-chart/>>. Acesso em: 10 maio 2012.
- BERNERS-LEE, T. WorldWideWeb: Proposal for a HyperText Project. W3C, 1990. Disponível em: <<http://www.w3.org/Proposal.html>>. Acesso em: 31 mar. 2012.
- BEVILACQUA, A. F. Hypertext: Behind the Hype. ERIC Digest, 1989. Disponível em: <<http://www.ericdigests.org/pre-9212/hype.htm>>. Acesso em: 10 maio 2012.
- BIXBY, J. Our need for Web speed: It's about neuroscience, not entitlement. Web Performance Today, 2012. Disponível em: <<http://www.Webperformancetoday.com/2012/03/21/neuroscience-page-speed-Web-performance/>>. Acesso em: 21 abr. 2012.
- BOUTELL, T. PNG (Portable Network Graphics) Specification. World Wide Web Consortium, 1996. Disponível em: <<http://www.w3.org/TR/REC-png-multi.html>>. Acesso em: 10 maio 2012.

BROWERSCOPE. Network. Browserscope, 2012. Disponível em: <<http://www.browserscope.org>>. Acesso em: 11 abr. 2012.

CEPTRO.BR. Quem Somos. Centro de Estudos e Pesquisas em Tecnologia de Redes e Operações, 2005. Disponível em: <<http://www.ceptro.br/CEPTRO/QuemSomos>>. Acesso em: 10 maio 2012.

CERN. How the Web Began. European Organization for Nuclear Research, 2008. Disponível em: <<http://public.Web.cern.ch/public/en/About/WebStory-en.html>>. Acesso em: 10 maio 2012.

CERT.BR. Sobre o CERT.br. Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil, 2005. Disponível em: <<http://www.cert.br/sobre/>>. Acesso em: 10 maio 2012.

CETIC.BR. Sobre o CETIC.br. Centro de Estudos sobre as Tecnologias da Informação e da Comunicação, 2005. Disponível em: <<http://cetic.br/sobre-ceticbr/>>. Acesso em: 10 maio 2012.

CGI.BR. Decretos, 2003. Disponível em: <<http://cgi.br/regulamentacao/decr4829.htm>>. Acesso em: 31 mar. 2012.

CGI.BR. TIC Domicílios e Empresas. CGI.br. [S.l.]. 2010.

CHESHIRE, S. It's the Latency, Stupid. [S.l.]. 1996.

COYIER, C. CSS Sprites: What They Are, Why They're Cool, and How To Use Them. CSS-Tricks, 2009. Disponível em: <<http://css-tricks.com/css-sprites/>>. Acesso em: 10 maio 2012.

CROCKFORD, D. JSMIn, The JavaScript Minifier. Wrrrld Wide Web, 2003. Disponível em: <<http://www.crockford.com/javascript/jsmin.html>>. Acesso em: 10 maio 2012.

CSLONGWANG. CS Long Wang CDN. CS Long Wang Hosting, 2012. Disponível em: <<http://www.cslongwang.com/other/HostCdn.asp>>. Acesso em: 10 maio 2012.

DANTAS, M. Redes de Comunicação e Computadores: Abordagem Quantitativa. 1a. ed. Florianópolis: Visual Books, 2010.

DARPA. Partial Bibliography of the Internet/ARPANET. Defense Advanced Research Projects Agency. Disponível em: <[http://www.darpa.mil/About/History/PARTIAL\\_BIBLIOGRAPHY\\_OF\\_THE\\_INTERNET/ARPANET.aspx](http://www.darpa.mil/About/History/PARTIAL_BIBLIOGRAPHY_OF_THE_INTERNET/ARPANET.aspx)>. Acesso em: 10 maio 2012.

DAS, K. IPv6 Header Deconstructed. IPv6.com, 2008. Disponível em: <<http://ipv6.com/articles/general/IPv6-Header.htm>>. Acesso em: 10 maio 2012.

DIXON, P. Shopzilla's Site Redo - You Get What you Measure. Velocity. [S.l.]: [s.n.]. 2009.

DOJO. Dojo Toolkit. Dojo Toolkit, 2012. Disponível em: <<http://dojotoolkit.org>>. Acesso em: 10 maio 2012.

DOTANDCO. ICANN Registrars. News on Domain Names by Dot and Co, 2011. Disponível em: <[http://www.dotandco.net/ressources/icann\\_registrars/index.en](http://www.dotandco.net/ressources/icann_registrars/index.en)>. Acesso em: 10 maio 2012.

FACEBOOK. Facebook. Facebook, 2012. Disponível em: <<http://www.facebook.com>>. Acesso em: 10 maio 2012.

FLINT, I. World IPv6 Day: What We Learned. Velocity. [S.l.]: [s.n.]. 2011.

FLORENTINO, A. Endereçamento IPv6. IPv6.br Blog, 2011. Disponível em: <<http://ipv6.br/blog/?p=23>>. Acesso em: 31 mar. 2012.

GARRETT, J. J. Ajax: A New Approach to Web Applications. Adaptive Path, 2005. Disponível em: <<http://www.adaptivepath.com/ideas/ajax-new-approach-Web-applications>>. Acesso em: 10 maio 2012.

GENTILCORE, T. Beyond Gzipping. Velocity. San Jose: [s.n.]. 2009.

GOMEZ. When Seconds Count Survey. Compuware. [S.l.]. 2010.

GOMEZ. Why Web Performance Matters: Is Your Site Driving Customers Away? GOMEZ. [S.l.]. 2010.

HARRISON, C. et al. **Rethinking the progress bar**. 20th Annual ACM Symposium on User interface Software and Technology. Newport, Rhode Island, USA, October 07 - 10, 2007: UIST '07. ACM. 2007. p. 115-118.

HTTPARCHIVE. Trends and Stats. **HTTP Archive**, 2012. Disponível em: <<http://httparchive.org>>. Acesso em: 10 abr. 2012.

ICANN. Welcome to ICANN! **Internet Corporation for Assigned Names and Numbers**. Disponível em: <<http://www.icann.org/en/about/welcome>>. Acesso em: 10 maio 2012.

IEEE. IEE 802.3 Ethernet Working Group. IEEE. [S.l.].

ISOC. Mission. **Internet Society**, 1992. Disponível em: <<http://www.internetsociety.org/who-we-are/mission>>. Acesso em: 28 mar. 2012.

ISOC. History of the Internet. **Internet Society**. Disponível em: <<http://www.internetsociety.org/internet/internet-51/history-internet>>. Acesso em: 10 maio 2012.

JQUERY. **jQuery. Write Less, Do More**, 2006. Disponível em: <<http://jquery.com/>>. Acesso em: 10 maio 2012.

JUPITER, R. **Retail Web Site Performance**. Akamai. [S.l.]. 2006.

KEYNOTE. **Performance Indices**. Keynote. [S.l.]. 2012.

KLEINROCK, L. **Information Flow in Large Communication Nets**. Massachusetts Institute of Technology. [S.l.]. 1961.

KOLICH, M. S. HOWTO: Use Apache mod\_deflate To Compress Web Content (Accept-Encoding: gzip), 2009. Disponível em: <<http://mark.koli.ch/2009/04/howto-use-apache-mod-deflate-to-compress-Web-content-obsessed-with-speed-of-kolichcommobi.html>>. Acesso em: 10 maio 2012.

KUROSE, J. F.; ROSS, W. K. **Redes de Computadores e a Internet uma Abordagem Top-Down**. 5a. ed. São Paulo: Pearson Education, 2010.

LIVSHITS , B.; KICIMAN, E. Doloto. **Microsoft Research**, 2008. Disponível em: <<http://research.microsoft.com/en-us/projects/doloto/>>. Acesso em: 10 maio 2012.

LOHR, S. For Impatient Web Users, an Eye Blink Is Just Too Long to Wait. **The New York Times**, New York, p. 29, fev. 2012.

MOOTOOLS. **MooTools, a Compact JavaScript Framework**, 2006. Disponível em: <<http://mootools.net/>>. Acesso em: 10 maio 2012.

MOREIRAS, A. M. Entenda o Esgotamento do IPv4. **IPv6.br**, 2009. Disponível em: <<http://www.ipv6.br/IPV6/ArtigoEsgotamentoIPv4>>. Acesso em: 10 maio 2012.

MOZZILA, B. C. Firefox & Page Load Speed - Part II. **Blog of Metrics**, 2010. Disponível em: <<http://blog.mozilla.org/metrics/2010/04/05/firefox-page-load-speed-%E2%80%93-part-ii/>>. Acesso em: 14 abr. 2012.

NELSON, T. **Literary Machines**: The report on, and of, Project Xanadu concerning word processing, electronic publishing, hypertext, thinkertoys, tomorrow's intellectual revolution, and certain other topics including knowledge, education and freedom. Sausalito, California: Mindful Press, 1981.

NIC.BR. Quem Somos. **Núcleo de Informação e Coordenação do Ponto BR**, 2005. Disponível em: <<http://www.nic.br/sobre-nic/nicbr.htm>>. Acesso em: 10 maio 2012.

NIELSEN, J. **The Need for Speed**. useit.com. [S.l.]. 1997.

NIELSEN, J. Website Response Times. **Userit.com**, 2010. Disponível em: <<http://www.useit.com/alertbox/response-times.html>>. Acesso em: 21 abr. 2012.

NRO. Fee Pool of IPv4 Address Space Depleted. **Number Resource Organization**, 2011. Disponível em: <<http://www.nro.net/news/ipv4-free-pool-depleted>>. Acesso em: 10 maio 2012.

NSI. Network Solutions is a Remarkable Company. **Network Solutions**, 1979. Disponível em: <<http://about.networksolutions.com/site/network-solutions-is-a-remarkable-company/>>. Acesso em: 10 maio 2012.

O'REILLY, T. What Is Web 2.0. **O'Reilly**, 2005. Disponível em: <<http://oreilly.com/Web2/archive/what-is-Web-20.html>>. Acesso em: 31 mar. 2012.

PAUKA, D. Opções de CDN no Brasil. **Developer\*.com**, 2009. Disponível em: <<http://www.dirceupauka.com/opcoes-de-cdn-no-brasil>>. Acesso em: 10 maio 2012.

PEMBERTON, S. **Web 4.0: Start Planning Now. The Web and Beyond**, 10th SIGCHI.NL Conference. Amsterdam: [s.n.]. 2006.

PETERSON, L. L.; DAVIE, B. S. **Computer Networks: A System Approach**. 4a. ed. [S.I.]: Morgan Kaufmann, 2007.

PORT80. Port80's 2010 HTTP Compression Survey on the Top 1000 Corporations' Web Sites. **Port80**, 2010. Disponível em: <<http://www.port80software.com/surveys/top1000compression/>>. Acesso em: 06 abr. 2012.

RFC1296. **Internet Growth (1981-1991)**. Internet Engineering Task Force. [S.I.]. 1992.

RFC1726. Technical Criteria for Choosing IP The Next Generation (IPng). Internet Engineering Task Force. [S.I.]. 1994.

RFC1951. DEFLATE Compressed Data Format Specification version 1.3. IETF. [S.I.]. 1996.

RFC1952. GZIP file format specification version 4.3. IETF. [S.I.]. 1996.

RFC2318. **The text/css Media Type**. Internet Engineering Task Force. [S.I.]. 1998.

RFC2397. **The "data" URL scheme**. IETF. [S.I.]. 1998.

RFC2460. Internet Protocol, Version 6 (IPv6). IETF. [S.I.]. 1998.

RFC2557. MIME Encapsulation of Aggregate Documents, such as HTML (MHTML). IETF. [S.I.]. 1999.

RFC2616. Hypertext Transfer Protocol -- HTTP/1.1. IETF. [S.I.]. 1999.

RFC2765. **Stateless IP/ICMP Translation Algorithm (SIIT)**. Internet Engineering Task Force. [S.I.]. 2000.

RFC2850. **Charter of the Internet Architecture Board (IAB)**. Internet Engineering Task Force. [S.I.]. 2012.

RFC2893. **Transition Mechanisms for IPv6 Hosts and Routers**. Internet Engineering Task Force. [S.I.]. 2000.

RFC3142. **An IPv6-toIPv4 Transport Relay Translator**. Internet Engineering Task Force. [S.I.]. 2001.

RFC3935. **A Mission Statement for the IETF**. Internet Engineering Task Force. [S.I.]. 2004.

RFC4213. Basic Transition Mechanisms for IPv6 Hosts and Routers. **Internet Engineering Task Force**, 2005. Disponível em: <<http://tools.ietf.org/html/rfc4213>>. Acesso em: 10 maio 2012.

RFC5743. Definition of an Internet Research Task Force (IRTF) Document Stream. Internet Engineering Task Force. [S.I.]. 2009.

RFC6265. **HTTP State Management Mechanism**. Internet Engineering Task Force. [S.I.]. 2011.

RFC768. **User Datagram Protocol**. Internet Engineering Task Force. [S.I.]. 1980.

RFC791. **Internet Protocol**. Internet Engineering Task Force. [S.I.]. 1981.

RFC793. **Transmission Control Protocol**. Internet Engineering Task Force. [S.I.]. 1981.

RIPE-NCC. IPv6 Enabled Networks. **RIPE NCC**, 2012. Disponível em: <<http://v6asns.ripe.net/v/6>>. Acesso em: 22 abr. 2012.

ROSS, J.-M. Why Speed Matters. **O'Reilly Radar**, 2011. Disponível em: <<http://radar.oreilly.com/2011/04/why-speed-matters.html>>. Acesso em: 14 abr. 2012.

SCHURMAN, E.; BRUTLAG, J. The User and Business Impact of Server Delays, Additional Bytes, and HTTP Chunking in Web Search. O'Reilly's Velocity. San Jose: [s.n.]. 2009.

SELLERS, C. **IPv6 Transition Mechanisms and Strategies**. IPv6 Technology Conference. Rocky Mountain: [s.n.]. 2009.

SEOW, S. C. **Designing and Engineering Time: The Psychology of Time Perception in Software**. [S.I.]: Addison-Wesley Professional, 2008.

SHEA, D. CSS Sprites: Image Slicing's Kiss of Death. **A List Apart**, 2004. Disponível em: <<http://www.alistapart.com/articles/sprites>>. Acesso em: 22 abr. 2012.

SIEMON. Megahertz, Megabits, MegaConfusion. Siemon. [S.I.]. 1997.

SOULDERS, S. **High Performance Web Sites**. 1a. ed. [S.I.]: O'Reilly, 2008.

SOULDERS, S. **Even Faster Web Sites**. 1a. ed. [S.I.]: O'Reilly, 2009.

SOULDERS, S. Control JS, 2010. Disponível em: <<http://stevesouders.com/controljs/>>. Acesso em: 10 maio 2012.

SOULDERS, S. WPO - Web Performance Optimization. **High Performance Web Sites**, 2010. Disponível em: <<http://www.stevesouders.com/blog/2010/05/07/wpo-Web-performance-optimization/>>. Acesso em: 22 abr. 2012.

SOULDERS, S. the Performance Golden Rule. **High Performance Websites**, 2012. Disponível em: <<http://www.stevesouders.com/blog/2012/02/10/the-performance-golden-rule/>>. Acesso em: 22 abr. 2012.

SPIVACK, N. Semantic Web Talk. **Nova Spivack**, 2008. Disponível em: <[http://novaspivack.typepad.com/nova\\_spivacks\\_Weblog/2008/02/video-of-my-sem.html](http://novaspivack.typepad.com/nova_spivacks_Weblog/2008/02/video-of-my-sem.html)>. Acesso em: 10 maio 2012.

SPRITESTITCH. Pacman Cross Stic and Needlepoint Patterns. **Sprite Stitch**, 2008. Disponível em: <<http://www.spritestitch.com/?p=89>>. Acesso em: 10 maio 2012.

SRINIVASAN, R. Speed Web delivery with HTTP compression - A look at the page-delivery effects of data compression in HTTP 1.1. TATA Consultancy. [S.l.]. 2003.

STEFANOV, S. Progressive Rendering via Multiple Flushes. **phpied.com**, 2010. Disponível em: <<http://www.phpied.com/progressive-rendering-via-multiple-flushes/>>. Acesso em: 10 maio 2012.

STEFANOV, S. **Psychology of Performance**. Velocity. [S.l.]: [s.n.]. 2010.

STEWART, W. Internet Management. **Living Internet**, 2000. Disponível em: <[http://www.livinginternet.com/i/iw\\_mgmt.htm](http://www.livinginternet.com/i/iw_mgmt.htm)>. Acesso em: 10 maio 2012.

STEWART, W. The Internet. **Living Internet**, 2000. Disponível em: <<http://www.livinginternet.com/>>. Acesso em: 10 maio 2012.

STEWART, W. World Wide Web (WWW) History. **Living Internet**, 2000. Disponível em: <<http://www.livinginternet.com/w/wi.htm>>. Acesso em: 10 maio 2012.

TAGMAN. Just One Second Delay In Page-Load Can Cause 7% Loss In Customer Conversions. **TagMan**, 2012. Disponível em: <<http://blog.tagman.com/2012/03/just-one-second-delay-in-page-load-can-cause-7-loss-in-customer-conversions/>>. Acesso em: 13 abr. 2012.

THEURER, T. Performance Research, Part 1: What the 80/20 Rule Tells Us about Reducing HTTP Requests. **Yahoo! User Interface Blog**, 2006. Disponível em: <<http://www.yuiblog.com/blog/2006/11/28/performance-research-part-1/>>. Acesso em: 22 abr. 2012.

THEURER, T. Performance Research, Part 3: When the Cookie Crumbles. **Yahoo! User Interface Blog**, 2007. Disponível em: <<http://yuiblog.com/blog/2007/03/01/performance-research-part-3/>>. Acesso em: 10 maio 2012.

THEURER, T. Performance Research, Part 4: Maximizing Parallel Downloads in the Carpool Lane. **Yahoo! User Interface Blog**, 2007. Disponível

em: <<http://yuiblog.com/blog/2007/04/11/performance-research-part-4/>>.  
Acesso em: 10 maio 2012.

W3C.BR. Sobre o W3C. **Consórcio World Wide Web**, 2007. Disponível em: <<http://www.w3c.br/Sobre>>. Acesso em: 10 maio 2012.

Web Server Survey. NETCRAFT. [S.l.]. 2012.

WEI, D.; JIAN, C. **Frontend Performance Engineering in Facebook**. Velocity. San Jose: [s.n.]. 2009.

WILSON, F. **The 10 Golden Principles of Successful Web Apps**. Future of Web Apps. [S.l.]: [s.n.]. 2010.

WORLDIPV6DAY. **WorldIPv6Day**, 2012. Disponível em: <<http://www.worldipv6day.org/>>. Acesso em: 22 abr. 2012.

YAHOO! YUI Compressor. **Yahoo! Developer Network**, 2009. Disponível em: <<http://developer.yahoo.com/yui/compressor/>>. Acesso em: 10 maio 2012.

YAHOO! Yahoo! Performance Rules. **Yahoo! Developers**, 2012. Disponível em: <<http://developer.yahoo.com/performance/>>. Acesso em: 10 maio 2012.

ZIMMERMANN, H. OSI Reference Model - The OSI Model of Architecture for Open Systems Interconnection. **IEEE Transactions on Communications**, 04 abr. 1980. 28.

## Anexos

### UM ESTUDO DE DESEMPENHO DE APLICAÇÕES WEB

Victor Samuel Nicanor Alves<sup>1</sup>

<sup>1</sup>Departamento de Informática e estatística – Universidade Federal de Santa Catarina (UFSC)  
Caixa Postal 476 – 88.040-970 – Florianópolis – SC – Brazil

victors@inf.ufsc.br

**Abstract.** *The development of technologies of Web Applications drove the Web to higher levels of complexity and consecutively, brought certain inefficiency of performance. The load time of pages is considered the key-point of success. With that in mind, it becomes necessary the study of techniques that identify the bottlenecks of performance providing feasible solutions for such problem. This survey includes ten useful techniques for web applications development and, in addition, it presents empiric case uses in real environments to proof the good results driven by such techniques.*

**Resumo.** *O desenvolvimento de tecnologias utilizadas em aplicações Web acarretou em maiores níveis de complexidade e, conseqüentemente, certa ineficiência de execução. O tempo de carregamento de páginas é considerado o ponto-chave relacionado com o nível de satisfação do usuário, resultando diretamente no sucesso de aplicações. Torna-se necessário o estudo de técnicas que identifiquem os gargalos de desempenho e que ofereçam soluções factíveis para os problemas apresentados. Esse artigo apresenta uma pesquisa de dez técnicas úteis no desenvolvimento de aplicações Web, em adição, e são apresentados estudos de casos empíricos em ambientes reais que comprovam a melhoria ao se aplicar tais técnicas.*

## 1. Introdução

As tecnologias que integram a World Wide Web estão em constante evolução e as aplicações que as utilizam tornam-se mais poderosas. As mesmas tendem a ser mais complexas, tornando-as, muitas vezes, pouco eficientes no âmbito de desempenho.

O desempenho passou a ser considerado o fator mais importante para o sucesso de aplicações Web (WILSON, 2010), pois os usuários passaram a não mais aturar longos tempos de carregamento de páginas. Dado o presente desafio, surgem técnicas que atuam nos métodos de construção de páginas Web buscando extrair máxima eficiência das tecnologias utilizadas, sem a necessidade de intervenções na infraestrutura do sistema. Tais técnicas foram reunidas nesse artigo com a finalidade de fornecer uma base de consulta para desenvolvedores de aplicações Web, para que assim o desempenho do ambiente Web seja otimizado.

Dentro da proposta de tornar ambientes Webs mais eficientes, as técnicas abordadas neste documento foram aplicadas a ambientes reais, ilustrando assim o ganho considerável que elas proporcionam.

## 2. Proposta

O objetivo deste artigo consiste em uma *survey* de técnicas para melhoria do desempenho de aplicações Web. Para tal foi necessário um levantamento teórico na área de Internet, Web e os conceitos de redes que as envolvem, para então poder adaptar aplicações, possivelmente reais, com tais técnicas estudadas, a fim de apresentar os ganhos concretos ao utilizá-las.

## 3. O Problema

O avanço das tecnologias que suportam a Internet, desde enlace a processamento, criou a necessidade de tornar páginas Web mais eficientes. Diferenciando-se da busca por máquinas cada vez mais poderosas, o objetivo agora é otimizar páginas, analisando-as de maneira inteligente e detalhada como cada elemento reage e sua importância durante o carregamento, trabalhando-os para conseguir o maior ganho possível no tempo de carregamento.

Fred Wilson, investidor em aplicações para a Web, palestrou na conferência *Future of Web Apps* com o título *The 10 Golden Principles of Successful Web Apps* (WILSON, 2010). Dentre diversos tópicos, o fator número um contribuinte para o sucesso de aplicações Web é velocidade. Ele adiciona que velocidade não é somente uma funcionalidade adicional do sistema, é a mais importante das funcionalidades, é um requisito.

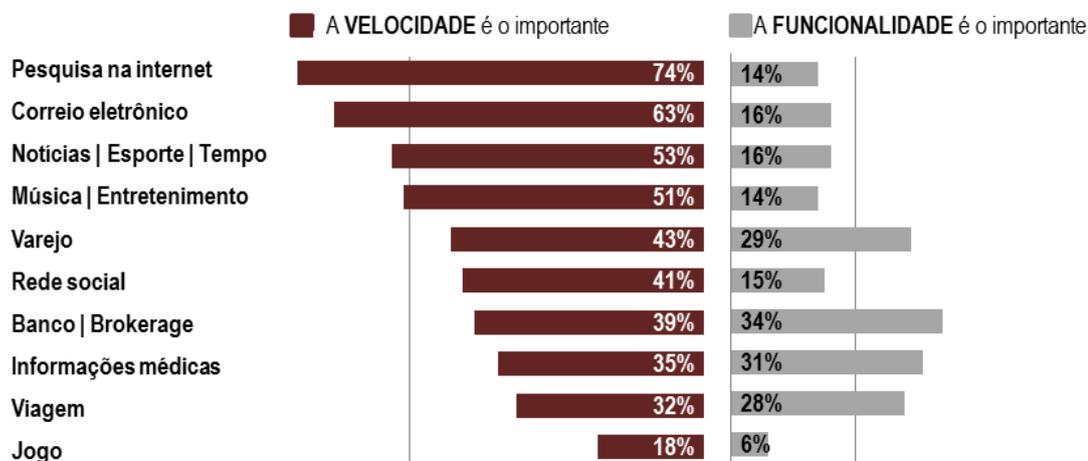


Figura 12: Comparativo entre velocidade funcionalidade (GOMEZ, 2010)

No ponto de vista de satisfação do usuário, a Figura 1 representa a comparação entre velocidade e funcionalidade de aplicações Web. Velocidade impera com maior importância sobre todas as áreas pesquisadas.

Uma pesquisa realizada pela Microsoft em sua aplicação de busca Bing. Em parte do seu tráfego, eles adicionaram atrasos adicionais, de 50 a 2000ms e o resultado foi impressionante. Se suas páginas levassem apenas 2 segundos a mais para carregar, a Microsoft teria uma redução de 4.3% de lucro por usuário (SCHURMAN e BRUTLAG,

2009). Outro estudo de caso realizado pela Amazon alcançou resultados ainda mais impressionantes. Em sua pesquisa eles concluíram que apenas 100 ms adicionais no tempo de carregamento de página são o bastante para reduzir vendas em 1% (AMAZON, 2006). A cadeia de comércio eletrônico Shopzilla reformulou sua plataforma por completo, reduzindo de 7 segundos para 1 segundo de carregamento de páginas. Com essa melhora, as vendas aumentaram de 7 a 12% (DIXON, 2009).

## 4. Técnicas de Otimização da Web

A requisição de cada elemento da página Web respeita a ordem em que são declarados no documento HTML. Sendo assim, primeiramente o navegador recebe o HTML para então iniciar as requisições subsequentes, recebendo então os elementos que compõem a página.

Como é possível observar, o carregamento de páginas Web envolvem diversos elementos e variáveis. Reconhecendo essa dificuldade, muitas técnicas foram desenvolvidas buscando extrair o máximo de eficiência desse sistema e assim reduzindo o tempo de carregamento.

A seguir serão detalhadas algumas técnicas úteis no desenvolvimento de páginas Web e buscam alcançar menor tempo de carregamento e conseqüentemente maior nível de satisfação do usuário.

### 4.1. Renderização Progressiva

A renderização progressiva de páginas Web consiste em oferecer ao usuário uma sensação de progresso, ao mesmo tempo que transmite a uma resposta consideravelmente rápida a requisição do usuário. O próprio carregamento da página já é um indicativo de que a tarefa está sendo processada sendo possível uma estimativa de quando estará completa (SEOW, 2008).

Sendo assim, o desenvolvedor deve transmitir ao usuário o que é mais importante e que não demande muito processamento para conseguir o menor tempo de resposta possível. As áreas mais importantes consistem no logotipo da página e áreas estáticas. Já em segundo e terceiro plano ficam os conteúdos calculados dinamicamente, que também são importantes, mas requerem certo processamento para serem retornados (STEFANOV, 2010).

### 4.2. Múltiplos Domínios

A técnica que utiliza múltiplos domínios procura superar o limite de conexões que o navegador pode estabelecer com o servidor durante o carregamento da página. A especificação de HTTP/1.1 (RFC2616, 1999) limita em somente duas conexões simultâneas por servidor, porém os desenvolvedores foram além desse limite, sendo que os navegadores atuais alcançam até seis conexões por servidor.

O ganho no tempo de carregamento ao paralelizar a transferência de elementos é significativo. Entretanto, essa técnica deve ser utilizada com cuidado devido ao *overhead* causado por cada resolução de nome de domínio necessária para cada novo domínio ou subdomínio incluídos.

O limite de conexões em paralelo é por domínio e não por IP. Sendo assim, todos os recursos podem ser armazenados na mesma máquina sobre mesmo IP, porém utilizando domínios diferentes. Como solução, basta configurar os domínios secundários com um registro DNS CNAME apontando para o IP principal (SOUDERS, 2009).

De acordo com Yahoo, o número ideal de subdomínios é específico para cada aplicação e depende da quantidade total de elementos, seu tamanho e complexidade da página (THEURER, 2007).

### 4.3. Folhas de Estilo no Topo

Por terem importante papel na apresentação do conteúdo da página, os navegadores atrasam a renderização documento HTML até que todos os arquivos CSS estejam baixados e processados. Isto porque desenhar todos os elementos na tela querem muito processamento evitando assim subsequentes renderizações.

Os componentes de uma página Web são requisitados de acordo com a sequência que são referenciados no documento HTML. Sendo assim, a boa prática é referenciar os elementos CSS no início, no cabeçalho do documento, entre tags <head></head>. Do contrário, adicionar referências para arquivos CSS no final do documento atrasa o início da renderização até que todos os elementos anteriores estejam processados. Essa técnica complementa a prática de renderização progressiva citada previamente, fazendo com que o carregamento gradual da página forneça ao usuário uma noção de progresso, indicando que o trabalho requisitado está sendo realizado.

### 4.4. CSS Spriting

A técnica de Spriting consiste em agrupar elementos gráficos a fim de reduzir o número de conexões TCP. Essa técnica foi e ainda é muito usada para jogos eletrônicos convencionais, principalmente quando a memória disponível era limitada e o *overhead* para transferir ou manter muitas imagens em memória era grande demais (SHEA, 2004).

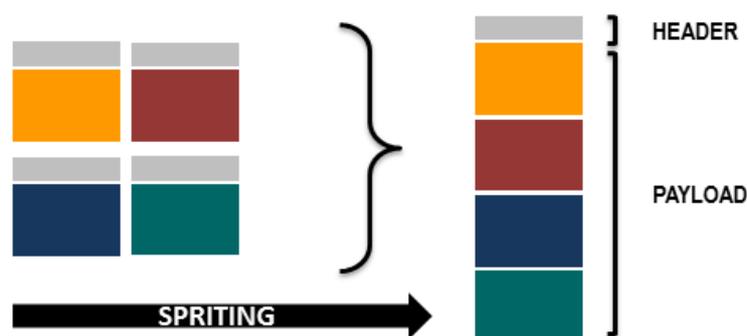


Figura 2: Princípio de CSS Spriting

Em páginas Web outros fatores tornam o uso de sprites interessante e encorajam ganho no tempo de carregamento. Inicialmente, ao agrupar, por exemplo, dez imagens em uma, o número de conexões resultantes será apenas um, o que implicará consideravelmente no desempenho da página (SOUDERS, 2008). Outra vantagem não

tão fácil de ser notada é o *overhead* implícito em cada imagem, pertinentes a cada codificação utilizada. Citando o exemplo anterior, ao agrupar as dez imagens, os bytes de 9 cabeçalhos não serão mais necessários (BOUTELL, 1996).

A forma de posicionar sprites numa página Web consiste em utilizar propriedades CSS de posicionamento de plano de fundo. Define-se o sprite como imagem de plano de fundo de todos os elementos ilustrados por ele e, utilizando o atributo CSS background-position, é possível reposicionar a imagem corretamente para cada elemento da página (COYIER, 2009).

## 4.5. DATA: URI e MHTML

É possível incorporar ao documento pequenas imagens, codificadas em base 64. Essa técnica utiliza os padrões Data: URI (RFC2397, 1998) e MHTML (RFC2557, 1999).

A tabela 1 demonstra como uma imagem pode ser codificada em base 64 e adicionada num documento HTML. O tamanho e complexidade da imagem resultam proporcionalmente no tamanho da sequência codificada.

| Data: URI  | Imagem  |
|--|---|
| <pre>&lt;img src="data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAA8AAAAPCAIAAAC0tAIdAA AAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQA AAAcSURBVChTY2BY9IIERIJSolMjqjEDdzRMKAwTAP/DWlqctmtUAAAAEIFTkSuQmCC"&gt;</pre> |  |

Tabela 1 Demonstração de Data: URI

Como consequência, não há necessidade de criar requisição para tal imagem, já que no momento que o arquivo HTML é requisitado, a imagem já o acompanha. Porém algumas dificuldades acompanham seu uso, como a impossibilidade de armazenar em cache e a manutenção das imagens.

## 4.6. Minificação e Arquivos

A técnica de minificação tem o propósito de reduzir o tamanho de código de páginas Web, removendo conteúdo desnecessário para a execução do arquivo, tal qual espaços em branco, tabulações, comentários, etc. (YAHOO!, 2012).

## 4.7. JavaScript

Durante o carregamento de arquivos JavaScript externos ao documento HTML, os navegadores costumam não iniciar nenhum *download* até que o script esteja completamente baixado, analisado e executado, o que atrasa muito a visualização da página. Sendo assim, é interessante adicionar scripts JavaScript de maneira mais inteligente possível (SOUDERS, 2008).

O ideal seria adicionar scripts no final do documento HTML, resultando em menor impacto final no carregamento, sendo que todos os elementos da página já foram transferidos para o cliente. Dessa maneira, os componentes que compõem a página serão carregados antes do código javascript, e possivelmente renderizando a página mais cedo.

Entretanto, essa opção não é aceitável em muitas aplicações. Isto porque o script JavaScript pode ser requisitado durante o carregamento dos outros elementos e não pode ser movido para o final. Essa é uma difícil decisão de projeto porém investimento de tempo nessa técnica resulta em ganho substancial em desempenho (SOUDERS, 2009).

Como saída, muitos Websites complexos como Facebook e Google Calendar buscam dividir o script deixando somente o mínimo necessário no topo do documento, atrasando ao máximo o carregamento do restante do script (SOUDERS, 2009).

Existe ferramentas como Microsoft Doloto (LIVSHITS e KICIMAN, 2008) e ControlJS (SOUDERS, 2010) que auxiliam nessa divisão de *payload*, posicionando no início somente o código realmente importante durante o carregamento da página.

As soluções mais comuns para esse problema são:

Posicionar scripts externos no final do documento HTML.

Carregar scripts de maneira assíncrona

Solicitar scripts adicionais o mais tardar possível, sobre demanda.

A escolha de qual solução utilizar dependerá da complexidade da aplicação e o qual crítico são os scripts para o carregamento da página (SOUDERS, 2009).

## 4.8. Compactação

De acordo com Radhakrishnan (SRINIVASAN, 2003) e especificado por (RFC2616, 1999), a compactação HTTP pode ser feita nos níveis de conteúdo e de transferência. Compactação de conteúdo, ou compactação estática, consiste em manter o arquivo compactado enquanto armazenado, sem necessidade dos dados serem processados pelo servidor. A medida que o conteúdo for requisitado pelo cliente, não há necessidade de compactá-lo ou descompactá-lo. Como exemplo podemos citar imagens, vídeos, flash, etc.

Compactação por transferência é aplicada a medida que o dado é requisitado pelo cliente, principalmente a conteúdo dinâmico, criado no momento da requisição. Se enquadram nessa categoria arquivos HTML, CSS, JavaScript, XML, etc. Diferente do método de compactação estático, esse método requer explícito ajuste no que se diz respeito a configurações do servidor HTTP, o que infelizmente ainda é pouco utilizado (YAHOO!, 2009).

De acordo com a especificação HTTP/1.1, os algoritmos utilizados para compactação de transferência são GZIP (RFC1952, 1996) e DEFLATE (RFC1951, 1996). GZIP é baseado em DEFLATE, o qual é uma combinação do algoritmo LZ77 e com a Codificação de Huffman.

## 4.9. Gerenciamento de Cookies

Para desempenho, o principal atributo de um cookie é o domínio em que ele está associado. A cada requisição feita a esse domínio, o navegador transfere o valor de todos os cookies para o servidor, com o objetivo de identificar-se. Como exemplo, pode-se vincular cookies a um domínio “exemplo.com.br”. Qualquer requisição de conteúdo a esse domínio, incluindo seus subdomínios forçará o navegador a adicionar aos cabeçalhos HTTP os cookies que detém (SOUDERS, 2009).

Sendo assim, qualquer requisição de componentes de uma página, incluindo o documento HTML, imagens e arquivos texto resultam em demasiadas transferências de cookies entre navegador e servidor, o que pode ser facilmente contornado ao hospedar elementos estáticos em domínios diferentes do associado ao cookie (YAHOO!, 2012).

A prática mais indicada é associar cookies ao domínio “www.exemplo.com” e hospedar somente o documento HTML a esse domínio. Qualquer outro elemento deve ser hospedado em um subdomínio, como “estatico.exemplo.com”. Sendo assim, cookies serão transmitidos somente uma única vez (THEURER, 2007).

## 4.10. Redes de Fornecimento de Conteúdo

Redes de Fornecimento de Conteúdo, mais comumente referenciadas como *Content Delivery Networks* (CDN), distribuem geograficamente o conteúdo estático de um Website, replicando o servidor principal, com o objetivo de reduzir a distância e assim, diminuir a latência inerente de rede.

No Brasil esse serviço ainda é pouco explorado. Sites Brasileiros de grande porte detém sua própria infraestrutura de CDN e empresas como Azion oferecem o serviço no mercado brasileiro (AZION, 2012) (PAUKA, 2009).

## 5. Ambiente e Resultados Experimentais

Foram avaliados três Websites reais, quanto ao tempo de carregamento e arquitetura, podendo assim levantar os pontos de maior impacto no desempenho de cada aplicação.

As ferramentas que serão utilizadas para análise de desempenho e otimização das páginas são:

**WebPagetest.org:** Ferramenta online gratuita que analisa minuciosamente o processo de carregamento de páginas Web (MEENAN, 2008).

**Firebug:** Extensão do navegador Mozilla Firefox (FIREBUG, ?).

**Pingdom Tools Full Page Test:** Ferramenta online similar a WebPagetest.org (PINGDOM, ?).

**CSS Compressor:** Compressor online de arquivos CSS (MINIFYCSS, ?).

**JavaScript Compressor:** Compressor online para arquivos JavaScript (MINIFYJAVASCRIPT, ?).

**SpriteMe:** ferramenta de criação automática de CSS Sprites (SPRITEME, ?).

**Smush.it:** Otimizador online de imagens (SMUSH.IT, ?).

**Yslow:** Ferramenta que analisa páginas Web quanto a adoção de boas práticas de desempenho (YSLOW, ?).

Com a união das ferramentas apresentadas, será possível uma análise minuciosa de cada Website seguindo 10 passos:

1. Remover referências inválidas (erro 404)
2. Agrupar imagens criando CSS Sprites
3. Substituir arquivos de imagens por DATA:URI

4. Unificar scripts e folhas de estilo
5. Minificar scripts e folhas de estilo
6. Otimizar conteúdo de imagens
7. Ordenar scripts e folhas de estilo
8. Adicionar subdomínios
9. Habilitar conexão persistente
10. Ativar compactação gzip

Com o objetivo de avaliar os resultados obtidos, foram escolhidos cinco indicativos de desempenho:

**Requisições:** número de requisições HTTP necessárias para carregar a página;

**Tamanho:** soma do tamanho de todos os recursos que compõem a página;

**Início renderização:** tempo em que o navegador inicia o desenho da tela;

**Carregamento concluído:** evento disparado pelo navegador quando a página está completa;

**Nota YSlow:** Resultado obtido da análise feita pelo serviço YSlow quanto ao desempenho da página (YSLOW, ?).

## 5.1. Estudo de Caso 1: ufsc.br

O primeiro estudo de caso desse trabalho abordará a análise da página principal da Universidade Federal de Santa Catarina, localizada com o endereço eletrônico <http://ufsc.br>. Em conjunto com o pessoal da Superintendência de Governança Eletrônica e Tecnologia da Informação e Comunicação da UFSC (SeTIC), foi possível analisar a estrutura da página inicial da universidade e discutir as possíveis intervenções e dificuldades presentes nesse projeto. Como base de sua estrutura, a UFSC utiliza a plataforma Wordpress para gerência de conteúdo de seu Website principal (WORDPRESS, 2003). Ao optar pelo uso de um framework como esse, o gerenciamento do site torna-se mais fácil e seguro. Entretanto, acopla o desenvolvedor a uma arquitetura muitas vezes demasiada para a necessidade da página. Todas as características que acompanham um framework devem ser cuidadosamente avaliadas quanto a sua real aplicabilidade no desenvolvimento do site.

Com o objetivo de extrair a máxima eficiência da página da universidade, foram aplicadas as técnicas 2, 3, 4, 5, 6, 7 e 8. Apenas as ações 1 (Remover 404), 9 (keep-alive) e 10 (gzip) não foram aplicadas. Isto porque o site não apresentou nenhuma referência inválida e keep-alive e gzip já estavam habilitados.

Vale ressaltar que, dada a impossibilidade de acessar o código fonte da página da UFSC, os resultados referentes a tempo de carregamento aqui obtidos foram estimados, baseando-se na proporção de redução de tamanho e redução de conexões necessárias.

De uma maneira geral, a intervenção da página da UFSC resultou em ganho em todos os indicativos com destaque para a redução em 68% de requisições e 48% do tempo total de carregamento. O tempo de início de renderização foi reduzido em 36,9%

e o tamanho da página ficou 17% menor. A avaliação de YSlow subiu de 80 para 85, o que é considerável. A Figura 5.43 ilustra os resultados obtidos.

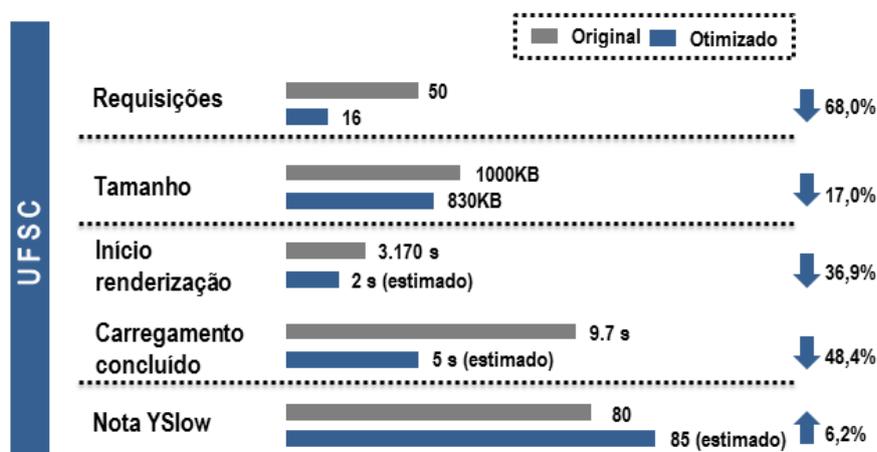


Figura 3: Resultados de Intervenções em UFSC.br

Embora o Website da UFSC utilize o framework WordPress como base e o mesmo ofereça certa inflexibilidade quanto a customizações, é possível alcançar certo ganho sem comprometer a estrutura padrão. Por ser um framework baseado em extensões, WordPress oferece outras facilidades, como o plugin W3 Total Cache, que promete otimizar facilmente o desempenho de aplicações WordPress.

## 5.2. Estudo de Caso 2: [www.korin.com.br](http://www.korin.com.br)

Nessa seção será abordada a análise realizada sobre o Website institucional da Korin Agricultura Natural. A Korin é uma empresa especializada em agricultura orgânica situada em São Paulo, e seu site utiliza a plataforma Microsoft .NET (MICROSOFT, ?). A página principal do site da Korin apresenta um design rico e um rotacionador automático de imagens que utiliza o framework jQuery (JQUERY, 2006) como base.

Das 10 técnicas apresentadas, oito foram aplicáveis no site da Korin, sendo elas 1, 4, 5, 6, 7, 8, 9, 10. Pela alta complexidade das imagens utilizadas no design do site, não foi possível criar CSS sprites ou DATA:URI's. É importante observar que os passos 9 e 10 não estavam efetivamente ativados nesse caso. Ao habilitar ambas opções a economia foi considerável.

Por ser um site repleto de imagens, os passos referentes a compactação exerceram maior impacto, resultando em 67% de redução no tamanho total da página. Outro fator que contribuiu para esse resultado foi a remoção de bibliotecas jQuery duplicadas e não minificadas, visto que seu tamanho é considerável. Outro valor que se destaca é o tempo de início de renderização, que apresentou ganho de 27%, reduzindo de 3 para 2.2 s. A Figura 4 ilustra os resultados obtidos.

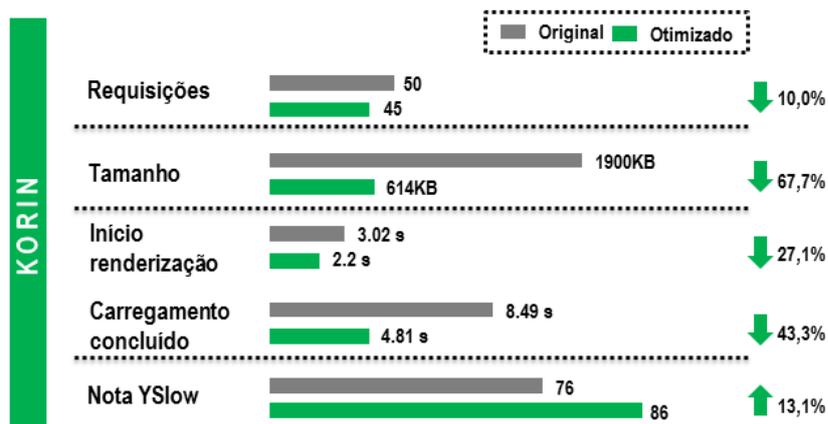


Figura 4: Resultados de Intervenções em [www.Korin.com.br](http://www.Korin.com.br)

O tempo de carregamento total da página caiu 43%, de 8.49s para 4.81, o que demonstra grande melhora no desempenho. Para finalizar, a avaliação YSlow subiu 10 pontos, alcançando a marca dos 86 pontos.

### 5.3. Estudo de Caso 3: [www.aproxima.com.br](http://www.aproxima.com.br)

Aproxima é uma agência de comunicação localizada na cidade de São Paulo. Seu site funciona com WordPress e a plataforma é Microsoft .NET. Em seguida será demonstrada a análise realizada no site <http://www.aproxima.com.br>. O Website da Aproxima tem aspecto de blog e apresenta um design limpo com poucas imagens, porém todas com tamanho significativo. Apenas o passo 8 (Subdomínios) não foi aplicado pois resultou em perda de desempenho de carregamento da página. A conexão persistente coberta pelo passo 9 já estava habilitada. A intervenção no site da Aproxima foi sem dúvida a mais significativa desse trabalho, com ganhos como 57% na redução do tamanho da página e 78% menos requisições realizadas. A Figura 5 apresenta os resultados compilados.

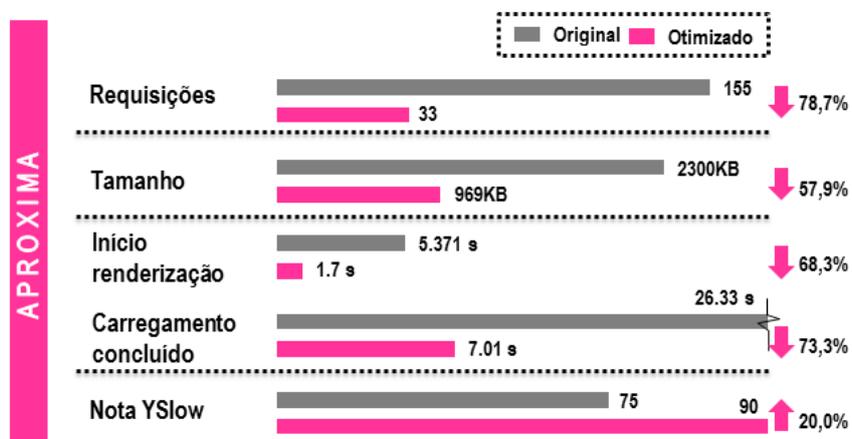


Figura 5: Resultados de Intervenções em [www.Aproxima.com.br](http://www.Aproxima.com.br)

Por apresentar um design limpo, o Website da Aproxima tende a ter desempenho elevado. Entretanto, apresentava problemas como imagens não otimizadas e conexões massivas com elementos de redes sociais. Tais otimizações renderam 20 pontos adicionais na escala YSlow o que comprova o grande resultado obtido.

## 5.4. Resultados Experimentais

De uma maneira geral, os três Websites aqui analisados foram passíveis de melhoria e apresentaram consideráveis ganhos de desempenho. A Figura 5.11 apresenta a média dos ganhos em cada fator indicativo.

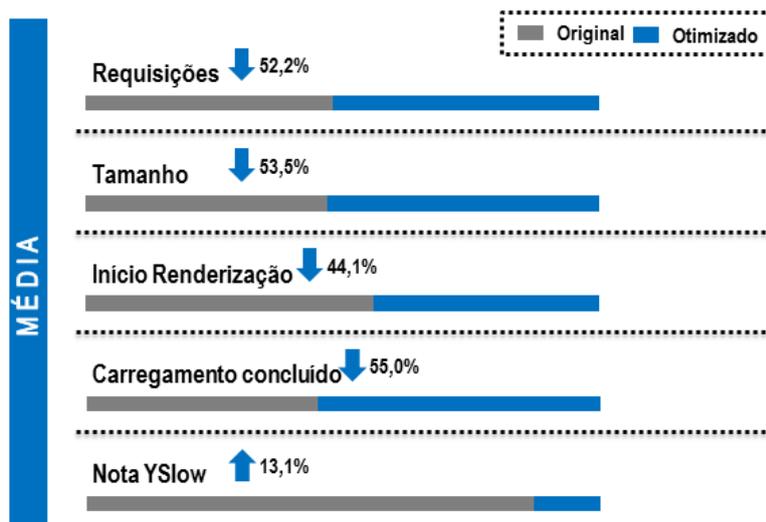


Figura 6: Média de melhorias dos casos

A quantidade de requisições reduziu 52%, as páginas ficaram 53% menores e houve 13% de crescimento em média na qualificação YSlow. Quanto aos tempos de carregamento, os navegadores iniciaram a renderização da tela 44% mais rápido e levou 55% menos tempo para carregar páginas por completo.

## 6. Conclusão

Esse trabalho consistiu em uma *survey* para melhoria de desempenho de aplicações Web, incluindo 10 técnicas úteis no desenvolvimento de qualquer aplicação Web. Diferente de outros fluxos de pesquisas voltados a otimização algorítmica e banco de dados, esse documento foca no tempo consumido no navegador do usuário, o que corresponde a até 80% do tempo de carregamento de páginas.

Com o intuito de demonstrar a eficiência das técnicas aqui abordadas, foram realizadas três intervenções em Websites reais, sites esses diferentes por natureza quanto a arquitetura e plataforma. Os resultados apresentaram, em média, redução de 55% no tempo de carregamento, chegando em 73% em um dos testes. O tráfego necessário reduziu em média 53%, o que é um bom indicativo de eficiência. Embora existam inúmeras combinações de plataformas e frameworks para desenvolvimento Web, provou-se que a adoção de tais técnicas é factível a qualquer aplicação Web.

## Referências

ABERDEEN, B. S. The Performance of Web Applications. Customers Are Won or Lost in One Second. Aberdeen. [S.l.]. 2008.

AKAMAI, C. R. Consumer Response to Travel Site Performance. Akamai. [S.l.]. 2010.

AMAZON. Make Data Useful. Stanford Data Mining. [S.l.]: [s.n.]. 2006.

APROXIMA. Aproxima Comunicação em Contato, ? Disponível em: <<http://aproxima.com.br/>>. Acesso em: 13 maio 2012.

AZION. Azion Content Delivery Network (CDN) – Mapa de Presença. Azion Technologies, 2012. Disponível em: <<http://www.azion.com.br/network-br/>>. Acesso em: 10 maio 2012.

BEHESHTI, H. Waterfalls 101: How to Read a Waterfall Chart. Strangeloop Networks Blog, 2011. Disponível em: <<http://www.strangeloopnetworks.com/resources/webinars/waterfalls-101-how-to-read-a-waterfall-chart/>>. Acesso em: 10 maio 2012.

BIXBY, J. Our need for web speed: It's about neuroscience, not entitlement. Web Performance Today, 2012. Disponível em: <<http://www.webperformancetoday.com/2012/03/21/neuroscience-page-speed-web-performance/>>. Acesso em: 21 abr. 2012.

BOUTELL, T. PNG (Portable Network Graphics) Specification. World Wide Web Consortium, 1996. Disponível em: <<http://www.w3.org/TR/REC-png-multi.html>>. Acesso em: 10 maio 2012.

CGI.BR. TIC Domicílios e Empresas. CGI.br. [S.l.]. 2010.

COYIER, C. CSS Sprites: What They Are, Why They're Cool, and How To Use Them. CSS-Tricks, 2009. Disponível em: <<http://css-tricks.com/css-sprites/>>. Acesso em: 10 maio 2012.

CROCKFORD, D. JSMIn, The JavaScript Minifier. Wrrrld Wide Web, 2003. Disponível em: <<http://www.crockford.com/javascript/jsmin.html>>. Acesso em: 10 maio 2012.

CSLONGWANG. CS Long Wang CDN. CS Long Wang Hosting, 2012. Disponível em: <<http://www.cslongwang.com/other/HostCdn.asp>>. Acesso em: 10 maio 2012.

DANTAS, M. Redes de Comunicação e Computadores: Abordagem Quantitativa. 1a. ed. Florianópolis: Visual Books, 2010.

DIXON, P. Shopzilla's Site Redo - You Get What you Measure. Velocity. [S.l.]: [s.n.], 2009.

DOJO. Dojo Toolkit. Dojo Toolkit, 2012. Disponível em: <<http://dojotoolkit.org>>. Acesso em: 10 maio 2012.

FACEBOOK. Facebook. Facebook, 2012. Disponível em: <<http://www.facebook.com>>. Acesso em: 10 maio 2012.

FIREBUG. What is Firebug? Firebug: Development Evolved., ? Disponível em: <<http://getfirebug.com/whatisfirebug>>. Acesso em: 13 maio 2012.

GARRETT, J. J. Ajax: A New Approach to Web Applications. Adaptive Path, 2005. Disponível em: <<http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>>. Acesso em: 10 maio 2012.

GENTILCORE, T. Beyond Gzipping. Velocity. San Jose: [s.n.]. 2009.

GOMEZ. When Seconds Count Survey. Compuware. [S.l.]. 2010.

GOMEZ. Why Web Performance Matters: Is Your Site Driving Customers Away? GOMEZ. [S.l.]. 2010.

HARRISON, C. et al. Rethinking the progress bar. 20th Annual ACM Symposium on User interface Software and Technology. Newport, Rhode Island, USA, October 07 - 10, 2007: UIST '07. ACM. 2007. p. 115-118.

HTTPARCHIVE. Trends and Stats. HTTP Archive, 2012. Disponível em: <<http://httparchive.org>>. Acesso em: 10 abr. 2012.

JQUERY. jQuery. Write Less, Do More, 2006. Disponível em: <<http://jquery.com/>>. Acesso em: 10 maio 2012.

KOLICH, M. S. HOWTO: Use Apache mod\_deflate To Compress Web Content (Accept-Encoding: gzip), 2009. Disponível em: <<http://mark.koli.ch/2009/04/howto-use-apache-mod-deflate-to-compress-web-content-obsessed-with-speed-of-kolichcommobi.html>>. Acesso em: 10 maio 2012.

KORIN. Korin Agricultura Natural. Korin Agricultura Natural, ? Disponível em: <<http://www.korin.com.br>>. Acesso em: 13 maio 2012.

KUROSE, J. F.; ROSS, W. K. Redes de Computadores e a Internet uma Abordagem Top-Down. 5a. ed. São Paulo: Pearson Education, 2010.

LIVSHITS , B.; KICIMAN, E. Doloto. Microsoft Research, 2008. Disponível em: <<http://research.microsoft.com/en-us/projects/doloto/>>. Acesso em: 10 maio 2012.

LOHR, S. For Impatient Web Users, an Eye Blink Is Just Too Long to Wait. The New York Times, New York, p. 29, fev. 2012.

MEENAN, P. About WebPagetest.org. WebPagetest.org, 2008. Disponível em: <<http://www.webpagetest.org/about>>. Acesso em: 13 maio 2012.

MICROSOFT..NET Framework Developer Center, ? Disponível em: <<http://msdn.microsoft.com/net/>>. Acesso em: 13 maio 2012.

MINIFYCSS. CSS Compressor and Minifier. CSS Compressor and Minifier, ? Disponível em: <<http://www.minifycss.com/css-compressor/>>. Acesso em: 13 maio 2012.

MINIFYJAVASCRIPT. JavaScript Compressor. JavaScript Compressor, ? Disponivel em: <<http://www.minifyjavascript.com/>>. Acesso em: 13 maio 2012.

MOOTOOLS. MooTools, a Compact JavaScript Framework, 2006. Disponivel em: <<http://mootools.net/>>. Acesso em: 10 maio 2012.

MOZZILA, B. C. Firefox & Page Load Speed - Part II. Blog of Metrics, 2010. Disponivel em: <<http://blog.mozilla.org/metrics/2010/04/05/firefox-page-load-speed-%E2%80%93-part-ii/>>. Acesso em: 14 abr. 2012.

NIELSEN, J. The Need for Speed. useit.com. [S.l.]. 1997.

NIELSEN, J. Website Response Times. Userit.com, 2010. Disponivel em: <<http://www.useit.com/alertbox/response-times.html>>. Acesso em: 21 abr. 2012.

PAUKA, D. Opções de CDN no Brasil. Developer\*.com, 2009. Disponivel em: <<http://www.dirceupauka.com/opcoes-de-cdn-no-brasil>>. Acesso em: 10 maio 2012.

PINGDOM. Full Page Test. Pingdom, ? Disponivel em: <<http://tools.pingdom.com/fpt/>>. Acesso em: 13 maio 2013.

RFC1951. DEFLATE Compressed Data Format Specification version 1.3. IETF. [S.l.]. 1996.

RFC1952. GZIP file format specification version 4.3. IETF. [S.l.]. 1996.

RFC2318. The text/css Media Type. Internet Engineering Task Force. [S.l.]. 1998.

RFC2397. The "data" URL scheme. IETF. [S.l.]. 1998.

RFC2557. MIME Encapsulation of Aggregate Documents, such as HTML (MHTML). IETF. [S.l.]. 1999.

RFC2616. Hypertext Transfer Protocol -- HTTP/1.1. IETF. [S.l.]. 1999.

RFC6265. HTTP State Management Mechanism. Internet Engineering Task Force. [S.l.]. 2011.

RFC793. Transmission Control Protocol. Internet Engineering Task Force. [S.l.]. 1981.

SCHURMAN, E.; BRUTLAG, J. The User and Business Impact of Server Delays, Additional Bytes, and HTTP Chunking in Web Search. O'Reilly's Velocity. San Jose: [s.n.]. 2009.

SEOW, S. C. Designing and Engineering Time: The Psychology of Time Perception in Software. [S.l.]: Addison-Wesley Professional, 2008.

SHEA, D. CSS Sprites: Image Slicing's Kiss of Death. A List Apart, 2004. Disponivel em: <<http://www.alistapart.com/articles/sprites>>. Acesso em: 22 abr. 2012.

SMUSH.IT. Smush.it. Smush.it, ? Disponivel em: <<http://www.smush.it/>>. Acesso em: 13 maio 2012.

SOUDERS, S. High Performance Web Sites. 1a. ed. [S.l.]: O'Reilly, 2008.

SOUDERS, S. Even Faster Web Sites. 1a. ed. [S.l.]: O'Reilly, 2009.

SOUDERS, S. Control JS, 2010. Disponível em: <<http://stevesouders.com/controljs/>>. Acesso em: 10 maio 2012.

SOUDERS, S. WPO - Web Performance Optimization. High Performance Web Sites, 2010. Disponível em: <<http://www.stevesouders.com/blog/2010/05/07/wpo-web-performance-optimization/>>. Acesso em: 22 abr. 2012.

SOUDERS, S. the Performance Golden Rule. High Performance Websites, 2012. Disponível em: <<http://www.stevesouders.com/blog/2012/02/10/the-performance-golden-rule/>>. Acesso em: 22 abr. 2012.

SPRITEME. SpriteMe, ? Disponível em: <<http://spriteme.org/>>. Acesso em: 13 maio 2012.

SPRITESTITCH. Pacman Cross Stic and Needlepoint Patterns. Sprite Stitch, 2008. Disponível em: <<http://www.spritestitch.com/?p=89>>. Acesso em: 10 maio 2012.

SRINIVASAN, R. Speed Web delivery with HTTP compression - A look at the page-delivery effects of data compression in HTTP 1.1. TATA Consultancy. [S.l.]. 2003.

STEFANOV, S. Progressive Rendering via Multiple Flushes. phpied.com, 2010. Disponível em: <<http://www.phpied.com/progressive-rendering-via-multiple-flushes/>>. Acesso em: 10 maio 2012.

STEFANOV, S. Psychology of Performance. Velocity. [S.l.]: [s.n.]. 2010.

THEURER, T. Performance Research, Part 1: What the 80/20 Rule Tells Us about Reducing HTTP Requests. Yahoo! User Interface Blog, 2006. Disponível em: <<http://www.yuiblog.com/blog/2006/11/28/performance-research-part-1/>>. Acesso em: 22 abr. 2012.

THEURER, T. Performance Research, Part 3: When the Cookie Crumbles. Yahoo! User Interface Blog, 2007. Disponível em: <<http://yuiblog.com/blog/2007/03/01/performance-research-part-3/>>. Acesso em: 10 maio 2012.

THEURER, T. Performance Research, Part 4: Maximizing Parallel Downloads in the Carpool Lane. Yahoo! User Interface Blog, 2007. Disponível em: <<http://yuiblog.com/blog/2007/04/11/performance-research-part-4/>>. Acesso em: 10 maio 2012.

UFSC.BR. Universidade Federal de Santa Catarina, ? Disponível em: <<http://www.ufsc.br>>. Acesso em: 13 maio 2012.

WILSON, F. The 10 Golden Principles of Successful Web Apps. Future of Web Apps. [S.l.]: [s.n.]. 2010.

WORDPRESS. WordPress, 2003. Disponível em: <<http://wordpress.org/>>. Acesso em: 13 maio 2012.

YAHOO! YUI Compressor. Yahoo! Developer Network, 2009. Disponível em: <<http://developer.yahoo.com/yui/compressor/>>. Acesso em: 10 maio 2012.

YAHOO! Yahoo! Performance Rules. Yahoo! Developers, 2012. Disponível em: <<http://developer.yahoo.com/performance/>>. Acesso em: 10 maio 2012.

YSLOW. YSlow. YSlow, ? Disponível em: <<http://yslow.org/>>. Acesso em: 13 maio 2012.