

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO
PAULO - IFSP
ÁREA DE INFORMÁTICA
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - ADS**

**ANDHERSON ANTUNES DOS SANTOS
MYKE AMBRIZZI RODRIGUES**

**MEEPO: UM GUIA DE RECOMENDAÇÕES PARA MELHORIA DE
DESEMPENHO *WEB* EM DISPOSITIVOS MÓVEIS**

TRABALHO DE CONCLUSÃO DE CURSO - TCC

CARAGUATATUBA

2013

**ANDHERSON ANTUNES DOS SANTOS
MYKE AMBRIZZI RODRIGUES**

**MEEPO: UM GUIA DE RECOMENDAÇÕES PARA MELHORIA DE
DESEMPENHO *WEB* EM DISPOSITIVOS MÓVEIS**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo, da Área de Informática, do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo.

Orientador: Prof. Renan Cavichi de Freitas

**CARAGUATATUBA
2013**

R123x

RODRIGUES, Myke Ambrizzi.

Meepo: Um Guia de Recomendações para Melhoria de Desempenho *Web* Móvel

/ Myke Ambrizzi Rodrigues – Caraguatatuba, 2013.

61f.

Orientador: Renan Cavichi de Freitas

Trabalho de Conclusão de Curso (Curso de Tecnologia em Análise de Sistemas) – Instituto Federal de São Paulo – IFSP.

1. Implementação das Recomendações.

I. Santos. Andherson Antunes dos.

II. Título.

IFSP-CAR / BC

CDU - xxx:000



Ministério da Educação
Instituto Federal de São Paulo
Campus Caraguatatuba

Nome do Diretor
Nome do Coordenador
Nome do Curso



TERMO DE APROVAÇÃO

MEEPO: UM GUIA DE RECOMENDAÇÕES PARA MELHORIA DE DESEMPENHO
WEB EM DISPOSITIVOS MÓVEIS

por

ANDHERSON ANTUNES DOS SANTOS

MYKE AMBRIZZI RODRIGUES

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 17 de Dezembro de 2013 como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas (ADS). Os candidatos foram arguidos pela Banca Examinadora composta pelos professores abaixo assinados, a qual após deliberação, considerou o trabalho aprovado.

Prof. Renan Cavichi de Freitas
Prof. Orientador

Prof. Dr. Lineu F. S. Mialaret
Presidente

Prof. Wanderson Santiago dos Reis
Membro

Dedico esse trabalho ao meu pai,
que infelizmente não pode participar deste
momento, minha família, e a minha
esposa, que me apoiou por estes três
anos.

Myke Ambrizzi Rodrigues

Aos meus amigos, pelas alegrias,
Tristezas e dores compartilhada.
A minha namorada, amiga e
companheira.
Ao meu parceiro Myke

Andherson Antunes dos Santos

AGRADECIMENTOS

A Deus, por nos iluminar e abençoar durante nossa trajetória.

A todos nossos professores que contribuíram com o nosso crescimento e amadurecimento ao longo de todo o curso.

Ao nosso orientador, Renan Cavichi, por nos apoiar com seus conhecimentos e dedicar seu tempo à nossa causa, sem nunca duvidar de nossa capacidade.

Aos nossos colegas de sala, pelos momentos compartilhados que deixarão saudades.

Agradecemos a CIP pela pró-atividade e presteza em resolver nossas dificuldades quanto a infraestrutura de TI encontradas ao longo do curso.

Enfim, a todos que mesmo indiretamente contribuíram para a realização desta pesquisa.

“Seja um padrão de qualidade. Algumas pessoas não estão acostumadas a um ambiente onde se espera a excelência. “

- STEVE JOBS

RESUMO

Com o crescente aumento do acesso à internet através de dispositivos móveis, como celulares e *tablets*, surgem novos desafios e paradigmas para o desenvolvimento de *sites*, de modo que estes possam ser acessados com o mesmo desempenho de quando são visitados em computadores de mesa. Neste trabalho de pesquisa identificam-se as principais características e deficiências do ambiente *web* móvel, para propor um guia de recomendações para o melhoramento de desempenho, testando e analisando em um estudo de caso, soluções para aumento da velocidade carregamento de páginas, visando à satisfação dos usuários.

Palavras-chave: Desempenho *Web*.Internet Móvel.Alta Latência.

ABSTRACT

With the increasing of the online access through mobile devices, as cellphones and tablets, we face new challenges and paradigms in *websites* development, to make the access performance as good as when they are visited on desktop. In this research, we identify the main features and issues of the mobile *web* environment, to propose a guide of recommendations for better performance, testing and analyzing in a use case, some solutions for increased load speed, in order to improve user's experience.

Keywords: *Web* Performance. Mobile Network. High Latency.

LISTA DE FIGURAS

Figura 1 – Requisições HTTP durante o carregamento da página.....	15
Figura 2 – Largura de Banda.....	23
Figura 3 – Arquitetura conceitual do MEEPO.....	33
Figura 4 – Ciclo de uma requisição HTTP com redirecionamento	39
Figura 5 – Teclado virtual de diversos dispositivos móveis.....	41

LISTA DE GRÁFICOS

Gráfico 1 – Teste de desempenho de Javascript	20
Gráfico 2 – Gráfico da velocidade média da internet móvel no Brasil	25
Gráfico 3 – Gráfico da Latência média da internet móvel no Brasil.....	25
Gráfico 4 – Gráfico da velocidade média da internet móvel em alguns países	26
Gráfico 5 – Tempo da primeira visualização.	50
Gráfico 6 – Tamanho total da página.	51
Gráfico 7 – Tempo de carregamento da página.	51
Gráfico 8 – Número de requisições HTTP.....	52
Gráfico 9 – Curva de ganho de desempenho por métrica.	53
Gráfico 10 – Ganho de desempenho individual para cada recomendação	53
Gráfico 11 – Resultados finais.....	54

LISTA DE TABELAS

Tabela 1 - Diferença do cache nos navegadores	28
Tabela 2 - Quantidade de conexões paralelas por navegador	28
Tabela 3 - Etapas de Aplicação das recomendações do MEEPO.....	46
Tabela 4 - Resultados obtidos no teste da versão sem otimização.....	47
Tabela 5 - Resultados obtidos no teste executado após a aplicação da etapa 1.....	47
Tabela 6 - Resultados obtidos no teste executado após a aplicação da etapa 2.....	48
Tabela 7 - Resultados obtidos no teste executado após a aplicação da etapa3.....	48
Tabela 9 - Resultados obtidos no teste executado após a aplicação da etapa 4.....	49
Tabela 10 - Resultados obtidos no teste executado após a aplicação da etapa 5. ...	49

SUMÁRIO

1	INTRODUÇÃO.....	14
1.1	OBJETIVOS.....	14
1.2	ESCOPO	15
1.3	REQUISITOS DE PESQUISA	15
1.4	ORGANIZAÇÃO DO TRABALHO.....	16
2	LEVANTAMENTO BIBLIOGRÁFICO.....	17
2.1	LINGUAGENS	17
2.1.1	HTML.....	17
2.1.2	HTML5.....	17
2.1.3	CSS.....	18
2.1.4	JAVASCRIPT	19
2.2	SOLUÇÕES PARA INTERFACE DO USUÁRIO	20
2.2.1	<i>Design</i> Responsivo.....	20
2.2.2	<i>Design</i> específico ou dedicado.....	21
2.3	REDES	21
2.3.1	HTTP.....	22
2.3.2	DNS.....	22
2.3.3	TCP.....	22
2.3.4	Latência e Largura De Banda.....	23
2.3.5	Tecnologias De Internet Móvel.....	24
2.3.6	A Internet Móvel Brasileira.....	24
2.4	SISTEMAS OPERACIONAIS PARA DISPOSITIVOS MÓVEIS.....	26
2.5	NAVEGADORES	27
2.5.1	Cache	27
2.5.2	Conexões paralelas.....	28
3	TRABALHOS RELACIONADOS	30
4	MEEPO - RECOMENDAÇÕES PARA MELHORIA DE DESEMPENHO DE SITES PARA DISPOSITIVOS MÓVEIS.....	32
4.1	PROJETO.....	33
4.1.1	P1- Solução de Interface	33
4.1.2	P2 - Bibliotecas CSS/JS	34
4.2	MELHORIAS.....	34
4.2.1	Cliente	35

4.2.2	Servidor	42
5	ESTUDO DE CASO.....	45
5.1	FERRAMENTAS UTILIZADAS.....	45
5.2	METODOLOGIA E MÉTRICAS UTILIZADAS.....	46
5.3	EXECUÇÃO DOS TESTES	46
5.3.1	Aplicação das recomendações MEEPO	47
6	ANÁLISE E DISCUSSÃO DOS RESULTADOS	50
6.1	RESULTADOS DOS TESTES.....	50
6.2	CONCLUSÃO SOBRE OS RESULTADOS	52
7	CONCLUSÃO.....	56
8	REFERÊNCIAS	58

1 INTRODUÇÃO

O acesso à internet não se limita mais apenas a computadores de mesa. Em 2011, 69% (115,4 milhões) dos brasileiros possuíam telefone celular [12] e 77% das pessoas com acesso à internet fizeram seu uso a partir destes dispositivos [1].

A qualidade na experiência vivenciada pelo usuário pode ser crucial para sua fidelização. Em um estudo de caso realizado em 2011, um aumento proposital em 1 segundo no tempo de resposta de uma loja virtual, obteve uma queda de 9,4% na quantidade de páginas visitadas e um aumento de 8,3% na taxa de rejeição e, mesmo após o fim do experimento, os compradores que deixaram de acessar, não voltaram a visitar o *site* [6].

Usuários esperam uma experiência rápida independentemente da plataforma em que naveguem. Segundo estudo de Jacob Nielsen [20], o tempo de espera ideal para o carregamento de uma página deveria ser de no máximo 1 segundo, não havendo assim tempo hábil do usuário dispersar enquanto espera o carregamento, porém a realidade encontrada hoje em dispositivos móveis está longe do ideal, atingindo médias de 7 segundos [13].

“First and foremost, we believe that speed is more than a feature. Speed is the most important feature” [29].

1.1 OBJETIVOS

Este trabalho de pesquisa tem por objetivo investigar técnicas para melhoria de desempenho *web* para dispositivos móveis, buscando encontrar soluções, verificar e validar sua efetividade, considerando as características e limitações da internet móvel, com o propósito de melhorar a velocidade de carregamento da página e conseqüentemente a experiência do usuário.

1.2 ESCOPO

A *Web*, como um sinônimo de *Internet*, funciona a partir de um modelo chamado cliente/servidor, onde um programa cliente, que roda em um sistema final, pede e recebe informações de um servidor que roda em outro sistema final.

Cerca de 80-90% do tempo do carregamento de uma página é realizado no lado cliente [26], realidade ilustrada na Figura 1. Dada a relevância do lado cliente no desempenho, essa pesquisa se concentra principalmente na melhoria das tecnologias e abordagens inerentes a ele, podendo também utilizar algumas técnicas do lado servidor, focando soluções que melhorem a velocidade de carregamento das páginas em dispositivos móveis a partir de conexões de internet de alta latência.

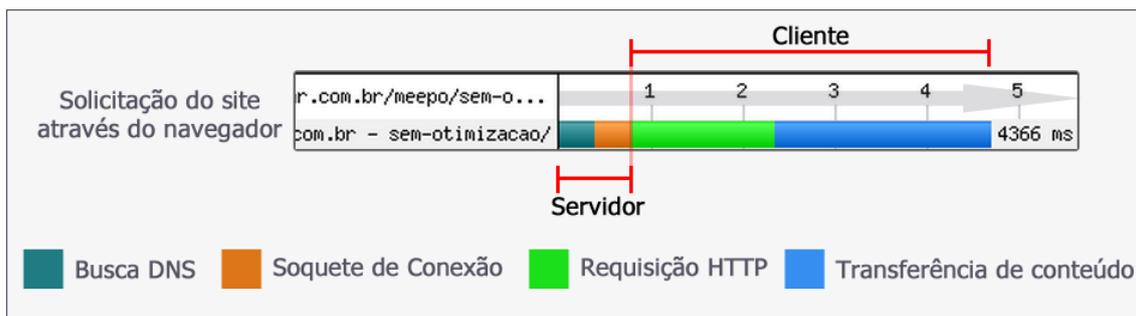


Figura 1 – Requisições HTTP durante o carregamento da página.

Esta pesquisa não abordará nenhuma linguagem específica do lado servidor.

1.3 REQUISITOS DE PESQUISA

Este trabalho expõe o resultado de uma pesquisa que resultou em uma visão geral das principais tecnologias relacionadas à melhoria de desempenho de páginas *web*, de forma a entendê-las e conhecer a relevância de cada uma delas para a melhoria de desempenho das mesmas.

Com base no estudo desenvolvido, propõem-se um guia de melhoria de desempenho, onde são listadas, classificadas e descritas as práticas mais

efetivas no ganho de desempenho quanto à velocidade de carregamento da página em dispositivos móveis, assim como um estudo de caso onde serão expostos os resultados da aplicação do guia proposto, a fim de verificar e validar sua eficiência.

1.4 ORGANIZAÇÃO DO TRABALHO

O capítulo 1 realiza a introdução do trabalho, abordando a importância da velocidade de carregamento de *sites*, além de definir o escopo e a organização do trabalho.

O capítulo 2 apresenta os principais termos, tecnologias, protocolos, linguagens e dados, nos quais essa pesquisa se baseia.

Em seguida, no capítulo 3, são apresentados os principais trabalhos relacionados ao estudo realizado, demarcando os pontos nos quais cada um se relaciona à referida pesquisa.

Na sequência, no capítulo 4, encontra-se a principal contribuição desse trabalho, um guia de recomendações de melhoria de desempenho *web* para dispositivos móveis.

O Capítulo 5 descreve o estudo de caso realizado, no qual as técnicas apresentadas no capítulo anterior foram aplicadas e seu desempenho medido, a fim de verificar e validar sua efetividade no ganho de velocidade de carregamento.

O capítulo 6 trata-se da análise e discussão dos resultados obtidos no estudo de caso.

E, por fim, são apresentadas a conclusão, as recomendações e as sugestões para trabalhos futuros.

2 LEVANTAMENTO BIBLIOGRÁFICO

Este capítulo apresenta uma visão geral das linguagens, técnicas, conceitos, protocolos e estruturas relacionados a esse trabalho de pesquisa, sintetizando a base de conhecimento abordada nos capítulos subsequentes.

2.1 LINGUAGENS

Algumas técnicas de melhoria de desempenho para *web* móvel, relacionam-se com a implementação correta das linguagens da interface do usuário, como: HTML, CSS e Javascript.

2.1.1 HTML

O termo HTML provém da abreviação do inglês *Hyper Text Markup Language*, que significa Linguagem de Marcação de Hipertexto, que se trata da linguagem padrão utilizada para a produção de páginas *web*.

Criada por Tim Berners-Lee, derivou-se da linguagem pioneira SGML (Standard Generalized Markup Language), em meados dos anos 90, e teve sua primeira especificação formal na mesma década [22].

2.1.2 HTML5

Descontentes com o rumo que a *Web* tomava, um grupo de desenvolvedores de empresas como Mozilla, Apple e Opera, em meados de 2004, fundou uma organização chamada WHATWG (*Web Hypertext Application Technology Working Group*), em que o foco era trabalhar em uma

nova versão do HTML com o intuito de trazer mais flexibilidade para a produção de *websites* e sistemas baseados na *Web* [22].

A partir de 2006, o trabalho desenvolvido pelo grupo passou a ser reconhecido mundialmente, inclusive pelo W3C – *World Wide Web Consortium*, a principal organização internacional de padrões para a *web*. Em outubro do mesmo ano, Tim Berners-Lee anunciou que faria parte da equipe do WHATWG trabalhando com o desenvolvimento do novo HTML [22].

Entre suas principais características, o HTML5 facilita a manipulação dos elementos do DOM¹, pois possibilita ao desenvolvedor alterar características dos objetos de forma não invasiva, e oferece recursos que facilitam e melhoram o uso do CSS e Javascript. Ele também cria novas *tags* e altera a função de algumas antigas, padronizando a criação de sessões e nomenclatura de IDs, classes e *tags*.

O HTML5 torna o código mais semântico e organizado e também exige a necessidade de instalação de alguns complementos, sem perder interatividade e desempenho. Ele cria código pronto para futuros dispositivos, que facilita a reutilização da informação de diversas formas.

A nova implementação do HTML apresenta também tecnologias que permitem à aplicação salvar dados em dispositivos clientes. Estes podem ser sincronizados com o servidor ou permanecer sempre no cliente, de acordo com a necessidade do desenvolvedor.

2.1.3 CSS

O CSS, do inglês *Cascading Style Sheet*, que significa folhas de estilo em cascata, tem por finalidade fornecer ao navegador informações sobre como os elementos devem ser apresentados. De acordo com Maurício Samy Silva “Folha de estilo em cascata é um mecanismo simples para adicionar estilos (por exemplo: fontes, cores, espaçamentos) aos documentos *web*” [23].

¹ O DOM, Documento Object Model, é uma especificação criada pelo W3C para uma interface multiplataforma que representa como as marcações, independentemente da linguagem, são organizadas e lidas pelo navegador.

2.1.4 JAVASCRIPT

Como principal linguagem de programação do lado cliente, o Javascript é uma linguagem interpretada e orientada a objetos. Foi desenvolvida inicialmente para gerar *scripts* capazes de interagir com o usuário sem a necessidade de processamento no servidor.

Foi desenvolvido por Brendan Eich por volta de 1995, e apesar do nome, ela não é uma linguagem baseada em Java. Com o advento de técnicas que a utilizam para atualização de conteúdo nas páginas de forma assíncrona, ganhou força e tornou-se a mais popular linguagem da *web*, recebendo diversos *frameworks* e bibliotecas que hoje possibilitam o seu uso até mesmo do lado servidor [18].

2.1.4.1 Execução de Javascript

O tempo de processamento do Javascript tem impacto direto no desempenho de carregamento das páginas em dispositivos móveis.

Como se observa no Gráfico 1, o desempenho da linguagem em dispositivos móveis chega a ser até 17 vezes mais lento quando comparado ao de um computador de mesa, mostrando que o Javascript pode ter grande peso na melhoria de desempenho de um *site*.

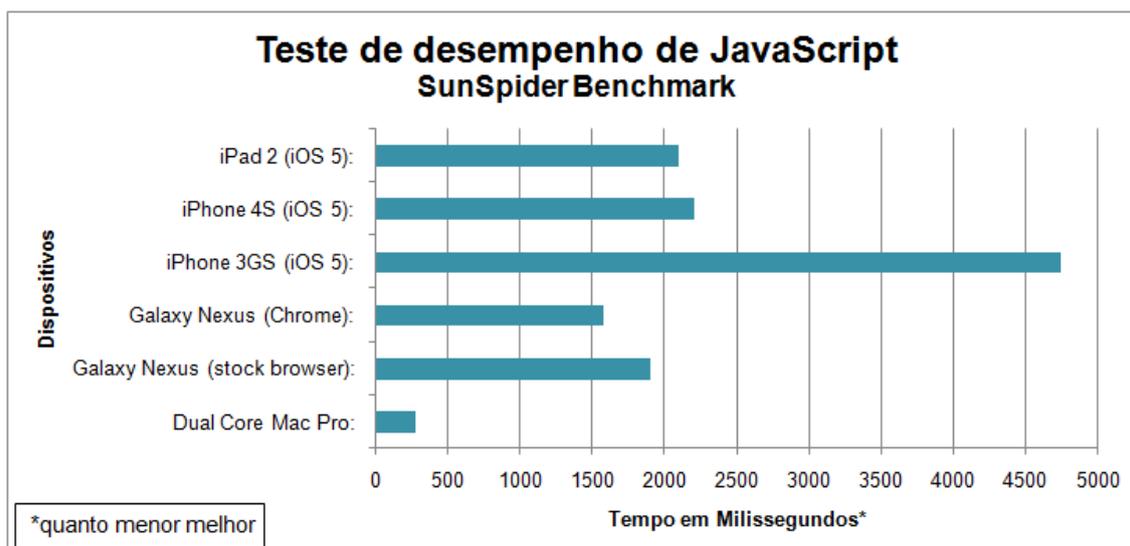


Gráfico 1 – Teste de desempenho de Javascript

2.2 SOLUÇÕES PARA INTERFACE DO USUÁRIO

O desenvolvimento tradicional de interfaces para *sites* com tamanhos predefinidos tornou-se ineficaz com o advento de diferentes dispositivos com acesso à internet. Surgiu então a necessidade de preparar os *sites* para serem acessados de maneira adequada em diferentes plataformas.

Descreve-se a seguir duas abordagens amplamente utilizadas na resolução de tal problema, as quais podem receber as técnicas de melhoria descritas nos próximos capítulos, de modo a obter ganhos de desempenho significativos no carregamento das páginas.

2.2.1 *Design* Responsivo

Um *site* responsivo adapta sua exibição de acordo com as características do dispositivo em que estiver sendo processado, evitando assim a necessidade do desenvolvimento de uma página específica para cada

dispositivo. Outro benefício importante é o uso de endereços únicos para cada página, facilitando a gestão de conteúdo.

2.2.2 *Design* específico ou dedicado

Trata-se da prática de desenvolver interfaces específicas para cada situação. Como por exemplo, uma página para a visualização em computadores de mesa e outra versão para ser carregada apenas a partir de dispositivos móveis.

Essa abordagem geralmente possui páginas com arquivos menores, pois somente os recursos que são necessários para aquele dispositivo são carregados. Isso permite explorar melhor as características de cada dispositivo, como uso de telas sensíveis ao toque ou através de cliques do *mouse*. O endereço do *site* na versão para dispositivos móveis normalmente é acompanhada da adição de um prefixo ao endereço da página, por exemplo, <http://m.site.com.br> ou mobile.site.com.br.

2.3 REDES

Uma rede de computadores é um conjunto de computadores interligados. A *Internet* é um conjunto de redes de computadores, de forma a tornar tudo uma grande rede.

Elas podem ser implementadas utilizando vários meios de comunicação, tais como cabos, linhas telefônicas, satélites entre outras.

Sua finalidade é permitir a troca de dados entre computadores, permitindo também o compartilhamento de recursos de hardware e software.

Todo o trabalho de pesquisa descrito neste documento depende da rede. A seguir, será feita uma descrição breve de diversos termos relevantes ao entendimento do mesmo.

2.3.1 HTTP

O Hypertext Transfer Protocol (HTTP), em português Protocolo de Transferência de Hipertexto, trata-se de um protocolo que pertence à camada de aplicação do modelo OSI, do inglês *Open Systems Interconnection*, que significa Interconexão de Sistemas Abertos. Este protocolo é a base para a comunicação em sistemas de informação *web*.

Ele funciona no modelo de cliente e servidor como um protocolo de requisição e resposta, por meio de um sistema de endereços chamado de URI (*Universal Resource Identifier* ou Identificador Universal de Recurso), e depende que os Protocolos TCP e IP estejam realizando a conexão entre cliente e servidor através de seus soquetes.

2.3.2 DNS

Existem duas formas de acessar uma página na *internet*: pelo nome de domínio ou pelo endereço IP. Para que não seja preciso digitar a sequência de números no navegador sempre que quiser visitar um *site*, o DNS, *Domain Name System*, ou Sistema de Nomes de Domínios, converte os nomes de domínios para números IP, permitindo a comunicação entre os dispositivos na rede mundial.

2.3.3 TCP

O TCP, *Transmission Control Protocol*, ou Protocolo de Controle de Transmissão, é um protocolo de comunicação da camada de transporte do Modelo OSI. É o principal protocolo de comunicação na *internet*. Sua versatilidade e robustez tornaram-no adequado às redes globais, já que este

verifica se os dados são enviados de forma correta, na sequência apropriada e sem erros, pela rede.

2.3.4 Latência e Largura De Banda

Imaginando a *Internet* como uma série de tubos interligados, a largura de banda seria a largura do tubo, já a latência seria o tempo necessário para um dado transitar do início ao fim deste tubo.

Tanto para a conexão com pouca largura de banda como para a transmissão com grande largura de banda, a latência é a mesma para cada bit de dados transmitidos. Porém no segundo caso a conexão transmite o arquivo mais rapidamente porque mais dados podem trafegar paralelamente.

A Figura 2 ilustra a Largura de Banda.

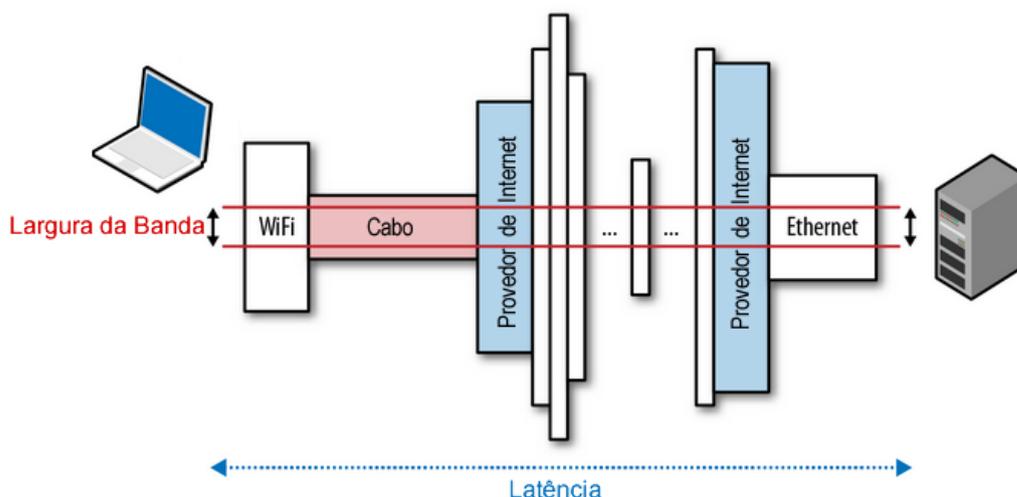


Figura 2 – Largura de Banda
Adaptado de: High Performance Browser Network, 2013 [10]

Até o início da década passada, o impacto da latência estava mascarado pelo fato das conexões de Internet terem pouca largura de banda.

Hoje em dia, não é incomum, em um navegador *web* baixando uma imagem pequena, esperar 100 a 150ms de resposta de conexão para depois demorar 5 ms para baixar toda a imagem. Ou seja, a latência nestes casos pode ser responsável por 90-95% do tempo total para baixar o recurso.

2.3.5 Tecnologias De Internet Móvel

As gerações de internet móvel são conhecidas pela numeração que identifica sua geração, seguido de um “G”, acrônimo de Geração.

A tecnologia 2G, implementada no início da década de 90, está atualmente em desuso, mas ainda é comum ser fornecida pelas operadoras de telecomunicações quando não há a disponibilidade do 3G.

O 3G trata-se da internet móvel que utiliza principalmente as tecnologias WCDMA ou CDMA. O WCDMA inclui as tecnologias HSPA e a sua evolução, o HSPA+, conhecido como 3G+ com potencial de alcançar 21 Mbps ²(entretanto, os planos mais comuns no Brasil são limitados a 1mbps).

A tecnologia 4G, ainda sendo implantada no Brasil, prevê tráfego de dados em até 100 Mbps [10] e utiliza a tecnologia LTE (acrônimo de *Long Term Evolution*, em português Evolução de Longo Prazo).

2.3.6 A Internet Móvel Brasileira

De acordo com dados apresentados no Gráfico 2, pode-se observar que as taxas de transferência nas redes móveis do Brasil, são relativamente baixas, se comparadas às de outros países, conforme Gráfico 4, e a latência, vista no Gráfico 3, é extremamente alta, o que resulta em uma baixa qualidade de conexão.

Tais fatores comprometem ainda mais o desempenho de carregamento das páginas *web*, tornando ainda mais importante a adoção de boas práticas para o desenvolvimento *web* móvel.

² Taxa de transferência de dados em megabits por segundo.

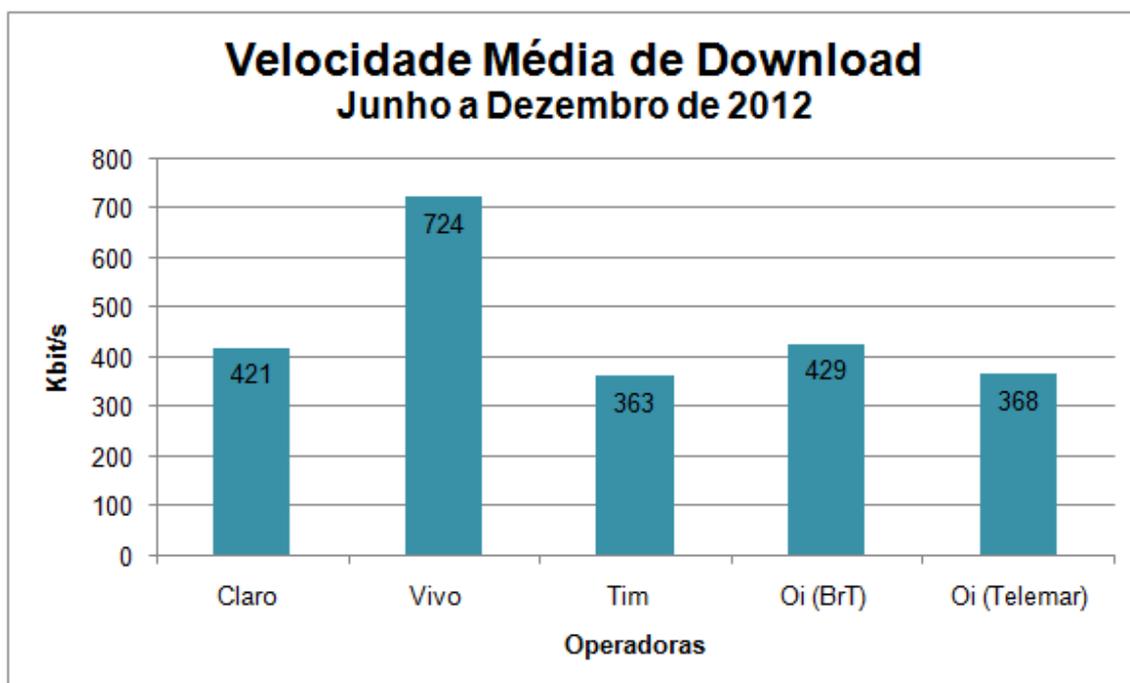


Gráfico 2 – Gráfico da velocidade média da internet móvel no Brasil
Fonte: Nic.br - Campus Party 6 Medindo a Qualidade da Internet no Brasil

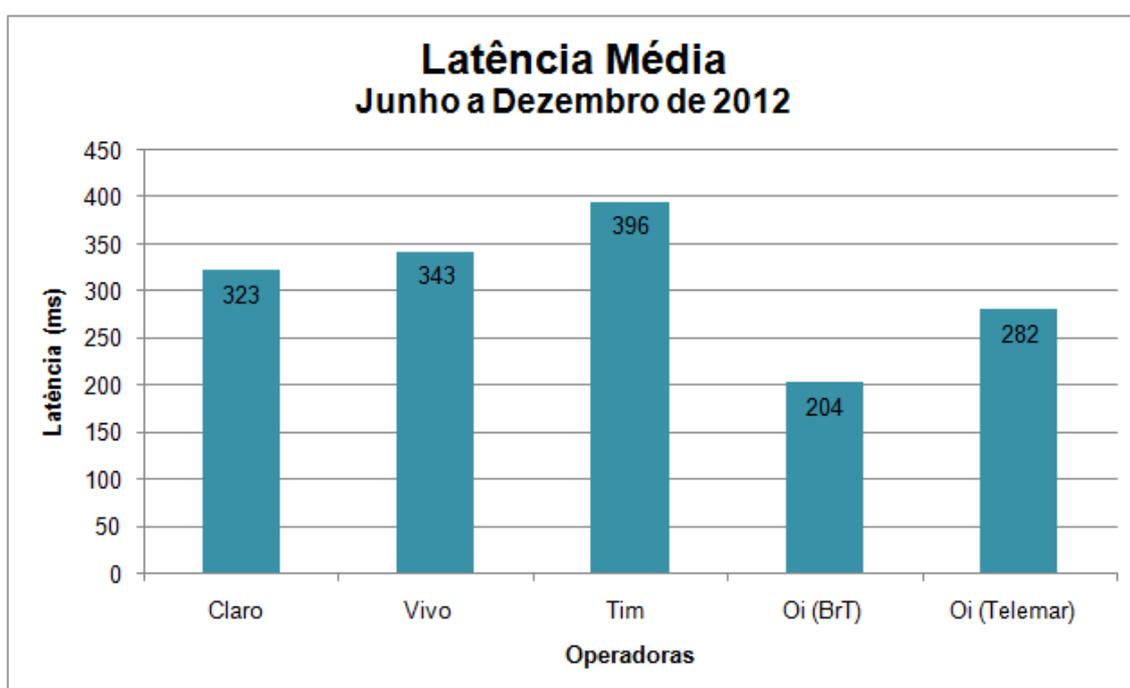


Gráfico 3 – Gráfico da Latência média da internet móvel no Brasil
Fonte: Nic.br - Campus Party 6 Medindo a Qualidade da Internet no Brasil

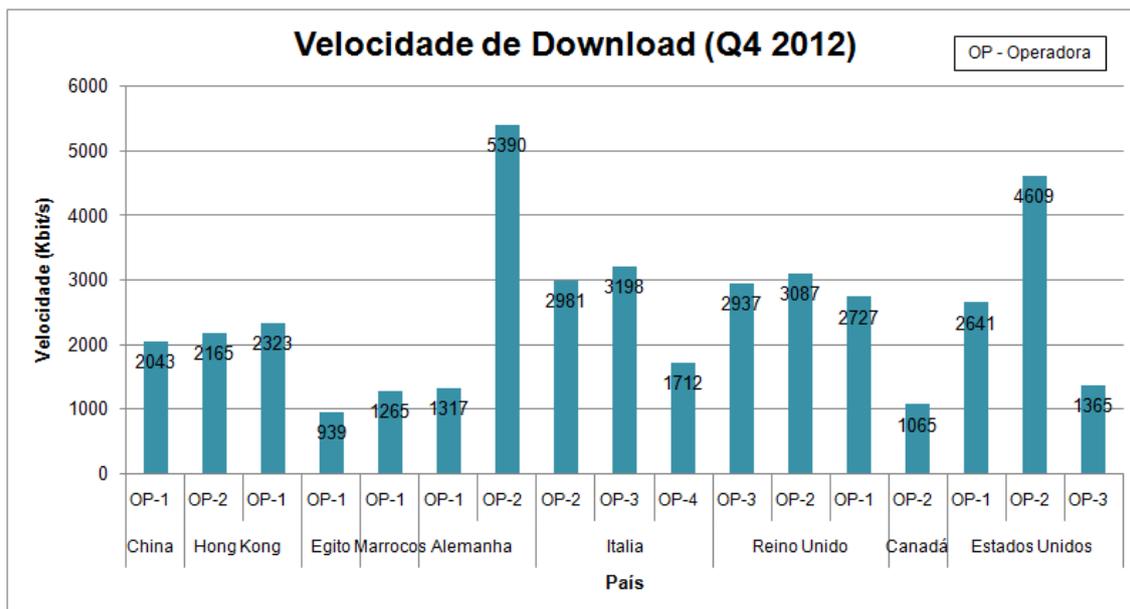


Gráfico 4 – Gráfico da velocidade média da internet móvel em alguns países
Fonte: Akamai - The State of the Internet 4th quarter 2012 Report

2.4 SISTEMAS OPERACIONAIS PARA DISPOSITIVOS MÓVEIS

O sistema operacional é um programa de controle de um computador, responsável por alocar recursos de hardware e escalonar tarefas, além de prover uma interface para o usuário, fornecendo a ele uma maneira de acesso aos recursos do computador.

Alguns sistemas operacionais utilizados nos dispositivos móveis:

- Android:

Atualmente na versão 4.4, conhecida como KitKat, o Android é um sistema operacional baseado no núcleo do Linux para dispositivos móveis, desenvolvido pela *Open Handset Alliance*, liderada pelo *Google*. Esta com código aberto desde 21 Outubro de 2008 sob a licença Apache.

- IOS:

Sistema operacional de código fechado da Apple Inc., lançado em junho de 2007, desenvolvido inicialmente para os celulares iPhone da empresa, hoje também é usado em outros produtos da empresa. Atualmente na versão 7, conhecida como iOS7.

- Windows Phone 8:

Sistema operacional de código fechado para dispositivos móveis criado pela Microsoft, lançado no final de 2010. É o sistema sucessor da plataforma Windows Mobile e seus aplicativos não são compatíveis entre si.

2.5 NAVEGADORES

Um navegador é um programa que permite que seus usuários interajam com páginas na *internet*. Diferem entre si na forma e capacidade de processar os recursos utilizados na criação de interface do usuário. Alguns exemplos de navegadores em dispositivos móveis:

- Navegador padrão do Android – pré-instalado no Android.
- Chrome - Disponível para iOS e nos Android 4.x
- Safari - Navegador padrão do iOS, disponível também para Android e Windows Phone na versão 7.x
- Internet Explorer - Navegador padrão do Windows Phone, só é compatível oficialmente com o mesmo.
- Firefox - Disponível para Android, Windows Phone e parcialmente no iOS.
- Opera Mini - Disponível para Android, iOS e Windows Phone.

2.5.1 Cache

Um dos recursos de um navegador é o cache, que é a capacidade de armazenar informações para uma futura reexibição do mesmo conteúdo ao usuário. A correta manipulação deste recurso prevê grandes benefícios em relação ao desempenho do *site*. A Tabela 1 mostra algumas características destes recursos em diferentes navegadores móveis.

Tabela 1 - Diferença do cache nos navegadores

Versão	Navegador	Tamanho máximo do cache
Móvel	Android 2.3	8 MB
Móvel	iOS 5	100 MB
Móvel	Android Chrome	250 MB
Computador	Chrome, Firefox.	512 MB, ou mais

Fonte: *Web caching on smartphones: ideal vs reality.*

Conforme Tabela 1, observa-se que o cache é bem mais limitado nos navegadores para dispositivos móveis. Outro agravante é em relação a como esse cache é manipulado [8], já que diversas técnicas comuns utilizadas hoje não se aplicam, ou aplicam-se de maneira limitada em navegadores móveis.

2.5.2 Conexões paralelas

Cada referência externa contida em uma página, como uma imagem ou um arquivo CSS, é transferida através de uma conexão paralela a qual a página está sendo baixada. A quantidade de conexões paralelas que um navegador consegue realizar também possui grande variação em suas versões móveis, conforme Tabela 2.

Tabela 2 - Quantidade de conexões paralelas por navegador

Conexões Totais	Conexões por domínio	Navegador
4	4	Android anterior à versão 3.0
256	6	Android posterior à versão 3.0
30	4	IOS 4
52	6	IOS 5
256	6	Chrome para Android

Fonte: *Velocity 2012 - Taming the Mobile Beast*

Ao alcançar o limite de conexões, o próximo arquivo entra em um fila e só é transferido quando alguma conexão for liberada, podendo trazer grandes impactos negativos em relação à performance do *site*.

3 TRABALHOS RELACIONADOS

Neste capítulo serão apresentados alguns trabalhos que se relacionam ao estudo realizado nesta pesquisa.

Steve Souders [25] [26], coletou as melhores práticas enquanto otimizava algumas das mais visitadas páginas da internet, que, junto a outros especialistas no assunto, geraram duas obras que foram utilizadas como referência neste trabalho de pesquisa.

As práticas mais relevantes destas obras para a proposta deste trabalho explicam como melhorar o desempenho do HTML, CSS, Javascript e imagens, além de técnicas de como escrever um código eficiente, criar aplicações responsivas e carregar *scripts* sem bloquear outros componentes da página. Também demonstram como compartilhar recursos através de múltiplos domínios, reduzir tamanho de imagens sem perder qualidade e usar codificação fragmentada para processar páginas mais rapidamente.

Os pontos mais efetivos expostos nos livros, em relação a este trabalho, estão listados a seguir:

- Evitar requisições HTTPs;
- Utilizar uma CDN ou Rede de Fornecimento de Conteúdo;
- Adicionar um Cabeçalho de Expiração (*Expires Header*);
- Ativar componentes de Gzip;
- Colocar Folhas de Estilo no topo do código;
- Colocar *scripts* no final da página;
- Evitar expressões CSS;
- Fazer Javascript e CSS externo;
- Reduzir pesquisas de DNS;
- Minificar o Javascript;
- Evitar redirecionamentos;
- Remover *scripts* duplicados;

O autor Ilya Grigorik, no livro sobre alto desempenho de navegação na rede [10], demonstra as melhores práticas de melhoria de desempenho para os protocolos TLS (Transport Layer Security), TCP (Transmission Control

Protocol), UDP (User Datagram Protocol), e explica os requisitos de melhorias de redes móveis sem fio.

No livro é demonstrado como acelerar o desempenho de rede 3G/4G sobre redes móveis, desenvolver aplicações móveis rápidas e energeticamente eficientes, encontrar gargalos de endereço no HTTP 1.X e outros protocolos de navegação, planejar e oferecer o melhor desempenho HTTP 2.0, ativar streaming em tempo real no navegador de forma eficiente, criar videoconferências ponto a ponto e aplicações de baixa latência em tempo real, todos esses pontos têm papel importante no trabalho aqui apresentado

Outro trabalho importante e que merece destaque é o “Yahoo! *Exceptional Performance*” [30]. Trata-se de uma equipe de desenvolvedores do Yahoo! que promove práticas para melhorar o desempenho de *websites*.

Entre as ferramentas desenvolvidas pelo time do Yahoo! encontra-se o YSlow, uma ferramenta que analisa o desempenho de páginas e sugere maneiras de melhorá-lo.

O Projeto “*Pagespeed*” [9] do Google é uma família de ferramentas desenvolvida para auxiliar o desenvolvedor na melhoria de desempenho das páginas *web*, ajudando a identificar boas práticas que podem ser aplicadas à página, podendo até automatizar o processo de otimização das mesmas. Dentre essas ferramentas destacam-se:

- *PageSpeed Insights* - Ferramenta *online* que analisa a página e, em seu resultado, sugere formas de otimizar seu *site*;
- *PageSpeed (Plugins* para o Chrome& Firefox) - Permite executar análises diretamente do navegador utilizando as sugestões do *PageSpeed* para otimizar a página;
- Bibliotecas populares - O Google disponibiliza as bibliotecas mais populares de código aberto de Javascript a partir de sua infraestrutura de alta performance para serem incorporadas diretamente dentro das páginas de qualquer *site*;
- *WebP* - Formato de imagem de código aberto que fornece uma compressão de imagem até 30% mais leve que os formatos tradicionais [9].

4 MEEPO - RECOMENDAÇÕES PARA MELHORIA DE DESEMPENHO DE *SITES* PARA DISPOSITIVOS MÓVEIS

Este capítulo apresenta o MEEPO, um guia com recomendações de melhoria de desempenho *web* para dispositivos móveis, como principal contribuição deste trabalho de pesquisa.

O tema otimização de *sites*, de maneira genérica, possui um vasto conteúdo de informações, seja em livros, convenções ou guias de boas práticas, porém os *sites* quando carregados em dispositivos móveis, muitas vezes não se beneficiam destes aprimoramentos.

O desenvolvimento para ambiente móvel possui alguns desafios extras que devem ser levados em consideração antes da aplicação de alguma técnica de otimização, como a alta latência e baixa largura de banda da internet e do cachê, conexões paralelas e desempenho de execução de javascript limitados dos navegadores em suas versões móveis.

O MEEPO, acrônimo da expressão em inglês “*MobilE web Performance Optimization*”, em tradução livre, “otimização de desempenho *web* móvel”, agrupa dentre as muitas recomendações pesquisadas, as que possuem maior eficiência na melhoria de desempenho relativas a dispositivos móveis.

A Figura 3 apresenta a arquitetura conceitual proposta para o MEEPO. Essa arquitetura apresenta uma análise de alto nível para os principais elementos que compõem as recomendações para melhoria de desempenho sem fornecer detalhes técnicos de sua implementação.

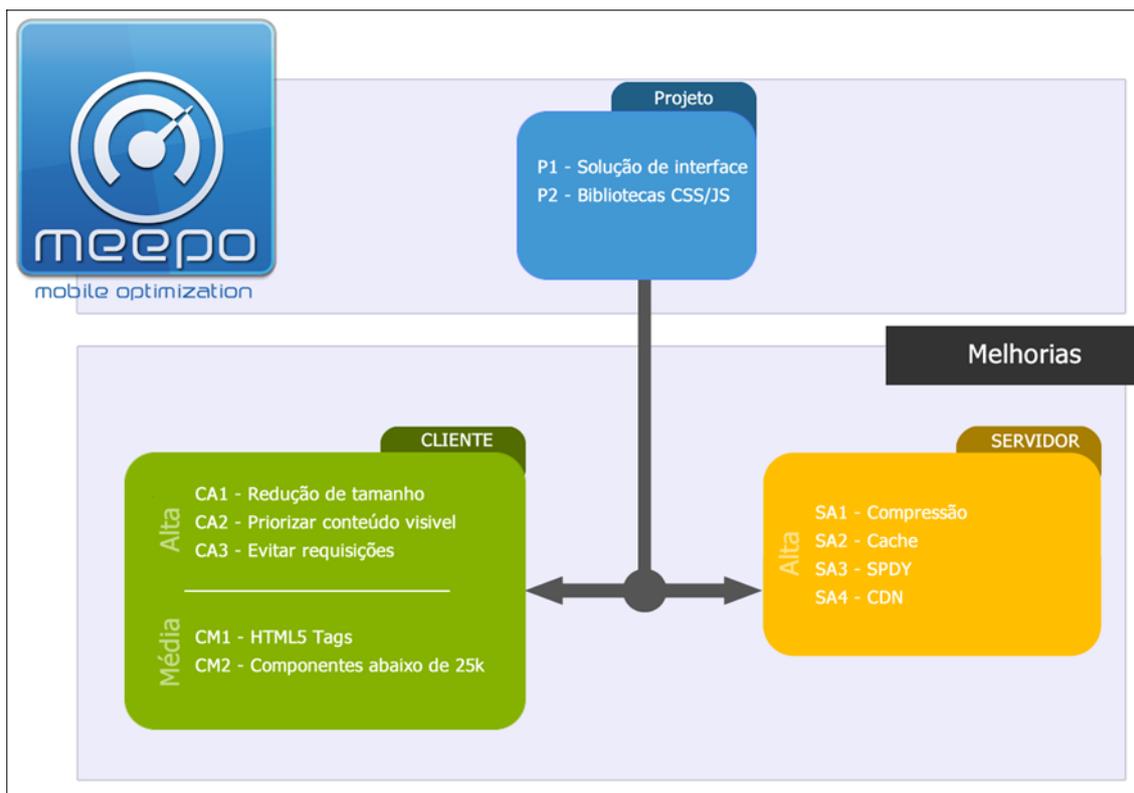


Figura 3– Arquitetura conceitual do MEEPO

4.1 PROJETO

Nas abordagens relacionadas ao projeto, analisam-se as características e objetivos do *site*. As decisões tomadas neste momento irão determinar o ambiente da aplicação que irá influenciar como diversas técnicas de melhoria serão aplicadas.

4.1.1 P1- Solução de Interface

O objetivo do *site* irá determinar a escolha do tipo de solução de interface a ser utilizada, sendo da mais genérica para a mais específica: a responsiva, a dedicada responsiva e a dedicada.

No que tange o desempenho, quanto mais complexo for o *site*, mais rápida tende a ser a solução específica em relação à genérica. Porém a

escolha da solução não é algo trivial e pode variar em cada caso, mas todas podem se beneficiar das técnicas de melhoria.

4.1.2 P2 - Bibliotecas CSS/JS

O uso de bibliotecas auxilia no desenvolvimento aumentando a produtividade, mas tende a diminuir o desempenho e deve ser evitado sempre que possível.

A escolha de bibliotecas deve levar em conta os objetivos do *site*, bem como a escolha da solução de interface a ser utilizada. Do ponto de vista do desempenho, deve-se sempre buscar utilizar bibliotecas específicas para as soluções propostas ao invés de soluções genéricas.

Citando como exemplo o JQuery, popular biblioteca de Javascript, que, devido ao fato de buscar um funcionamento estável em todos os navegadores e em qualquer plataforma, possui tamanho demasiadamente grande para utilização em dispositivos móveis e deve ser evitado[16]. O Bootstrap, biblioteca popular de CSS, permite selecionar os componentes de maneira individual, devendo sempre escolher somente os que serão utilizados no projeto.

4.2 MELHORIAS

Neste item, serão abordadas as técnicas de melhoria de *sítes* em dispositivos móveis, as quais estão divididas em duas frentes, cliente e servidor, e classificadas quanto à eficiência, de acordo com o disposto na Figura 3.

4.2.1 Cliente

As principais técnicas utilizadas para melhoria de desempenho a serem implementadas no lado cliente estão divididas em dois grupos, um de alta eficiência e outro de média eficiência, cada um contendo conceitos de otimização, os quais serão expostos do mais efetivo para o menos efetivo.

4.2.1.1 CA1 - Redução de tamanho

4.2.1.1.1 Minificação de arquivos HTML, CSS e Javascript

A minificação, como tradução informal do termo em inglês *minify*, é uma técnica que remove todos os bytes extras da página ao retirar espaços vazios, comentários, caracteres desnecessários, apresentando algumas vantagens, que motivam sua utilização, como:

Redução de tamanho - com a redução dos bytes desnecessários, o código passa a ser menor, pois todo o conteúdo “inútil” foi removido;

Melhora na compressão - com a remoção de conteúdos desnecessários a compressão pode ser aumentada ainda mais;

Carregamento mais rápido - conteúdos menores reduzem o tempo de transferência e por isso são executados mais rapidamente.

Existem diversas ferramentas disponíveis para a automatização da minificação, aplicando esta técnica em códigos Javascript, CSS, e em alguns casos HTML [26].

4.2.1.1.2 Compressão de arquivos

A compressão de arquivos é um recurso de servidor tratado com mais detalhes no item 4.2.2.1. Algumas práticas do cliente podem tornar o uso da compactação ainda mais efetiva:

- A compactação não deve ser utilizada em arquivos binários como imagens, vídeos, pdfs entre outros. Estes já possuem suas próprias formas de compactação e a utilização de uma nova compressão poderá torná-los maiores.[26]
- Minifique seus arquivos HTML, CSS e Javascript, pois o uso da mesma pode tornar a compactação mais eficiente. [26]
- Use letras minúsculas sempre que possível.
- Especifique atributos das tags HTML sempre na mesma ordem, utilize ordem alfabética.

4.2.1.1.3 Melhoria de Imagens

Na melhoria de imagens existem algumas regras simples, porém altamente relevantes para o bom desempenho de carregamento do *site* [25].

Definir sempre a largura e altura de uma imagem é importante, pois isso irá reduzir o tempo de processamento. Assim como tentar sempre utilizar imagens com tamanhos otimizados para cada situação.

Arquivos de imagens possuem informações desnecessárias quando são utilizados em *sites*, como informações internas com dados sobre o modelo da câmera utilizada, data, local, etc. Existem ferramentas de otimização de imagens que removem todas essas informações desnecessárias reduzindo o tamanho do arquivo e mantendo-o visualmente idêntico.

Em dispositivos móveis, devido ao reduzido tamanho da tela, a percepção da qualidade de uma imagem é bem menor. Deve-se comprimir a imagem até uma qualidade aceitável.

Uma forma de melhorar o desempenho percebido pelo usuário é usar arquivos de imagens progressivos, que são exibidas conforme são carregados.

4.2.1.2 CA2 – Priorização de conteúdo visível

Essa abordagem está relacionada à estruturação do código, com o objetivo de desenvolver tentando exibir informações relevantes ao usuário o mais rápido possível, gerando assim maior percepção de velocidade no carregamento.

4.2.1.2.1 Utilização CSS diretamente no código da página

A utilização de folhas de estilo diretamente no código da página permite que o DOM seja construído à medida que é carregado, processando conteúdo imediatamente após ser baixado. Deve-se usar esta técnica nos elementos mais críticos no processamento da interface, ou seja, aqueles que possuem maior relevância para o usuário ou são necessários para construção da estrutura da página.

O restante das folhas de estilo externas devem ser adicionadas ao topo do *site*, para que sejam carregadas o quão rápido seja possível, evitando assim o bloqueio dos elementos definidos por eles.

4.2.1.2.2 Carregamento de scripts de maneira assíncrona

A utilização do carregamento assíncrono, evita o bloqueio da construção do DOM já que, dessa forma, ele é construído de maneira independente do carregamento completo do script [25].

Além disso, é uma boa prática adicionar *scripts* no final da página, de forma que eles possam ser carregados após o término do carregamento do DOM.

4.2.1.3 CA3 - Evitar requisições

A sobrecarga de requisições é um fator de extrema relevância na performance quando se trata da internet móvel. Como explicado nos capítulos anteriores, a alta latência das redes móveis pode causar grandes gargalos no desempenho.

4.2.1.3.1 Evitar Redirecionamentos

Redirecionamentos de páginas geram requisições extras, o que em ambientes móveis pode causar um aumento no tempo de carregamento das páginas [26].

Quando ocorre um redirecionamento, é necessário acrescentar mais três vezes o tempo de latência para cada redirecionamento executado, pois cada redirecionamento gera uma nova requisição, que necessita de uma nova busca no servidor DNS, uma nova conexão TCP e um novo envio.

O redirecionamento mais comum a ser evitado é quando existe um *site* dedicado para dispositivos móveis e redireciona-se o usuário para ele quando necessário, conforme ilustrado na Figura 4. É possível eliminar este redirecionamento utilizando técnicas no servidor, que quando identificado o dispositivo do usuário, ao invés de redirecionar para outro endereço, forneça o conteúdo específico diretamente.

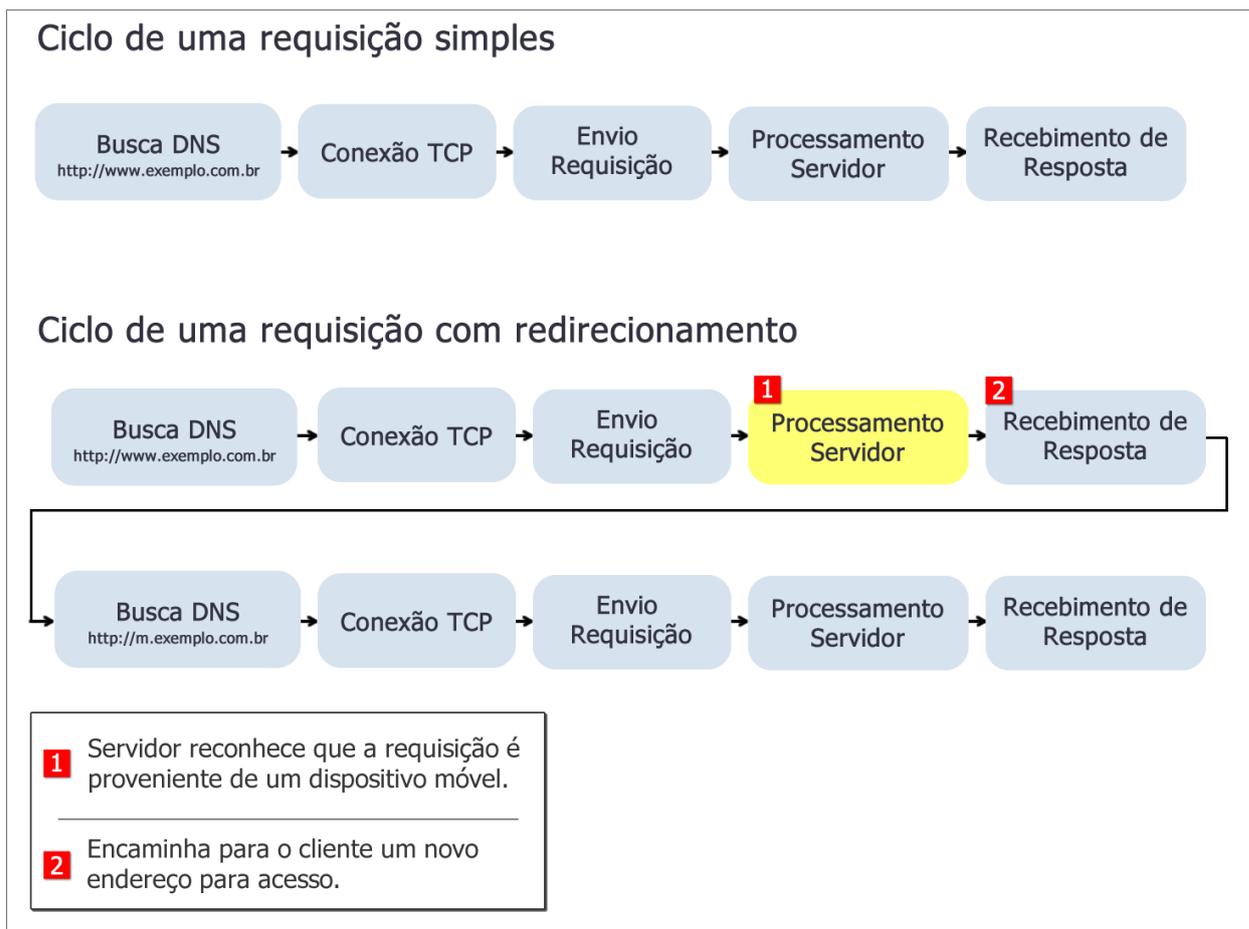


Figura 4 – Ciclo de uma requisição HTTP com redirecionamento

4.2.1.3.2 Utilização de sprites CSS

Sprites CSS são diversas imagens combinadas em uma única imagem que é manipulada por CSS, de forma a exibir somente o quadro referente à imagem desejada [26].

Esta técnica, bastante eficaz, pode reduzir drasticamente o número de requisições de uma página, além de em alguns casos também ser responsável por reduzir o peso da página por meio do uso das melhorias corretas [25].

4.2.1.3.3 Remoção de Scripts, bibliotecas e folhas de estilo duplicados

Evite utilizar bibliotecas, *scripts* e folhas de estilo contendo o mesmo conteúdo proveniente de endereços diferentes. Essa prática gera requisições extras aumentando o tempo de carregamento da página. Além disso, é uma boa prática evitar *scripts* que executam uma mesma função dentro do mesmo arquivo [26].

4.2.1.3.4 Concatenação de arquivos

Também é uma prática altamente desejável, a concatenação dos arquivos que devem ser carregados na página. O uso de tal prática diminui o número de requisições, porém a mesma deve ser analisada, pois existe a possibilidade de haver perda de desempenho, já que conteúdos mais importantes podem ser bloqueados pelo restante do arquivo [26].

4.2.1.4 CM1 - Otimização em Tags HTML

O HTML5 possui novas tags com funções específicas para e-mail, URL, número, alcance, data e tempo, que são eficientemente processadas pelo navegador e já possuem funções de validação, eximindo assim a necessidade do uso de Javascript quando for necessário utilizar alguma destas funções, reduzindo o peso da página e melhorando o tempo de carregamento [16]. A Figura 5 mostra o teclado virtual de diversos dispositivos utilizando-se de algumas tags citadas.

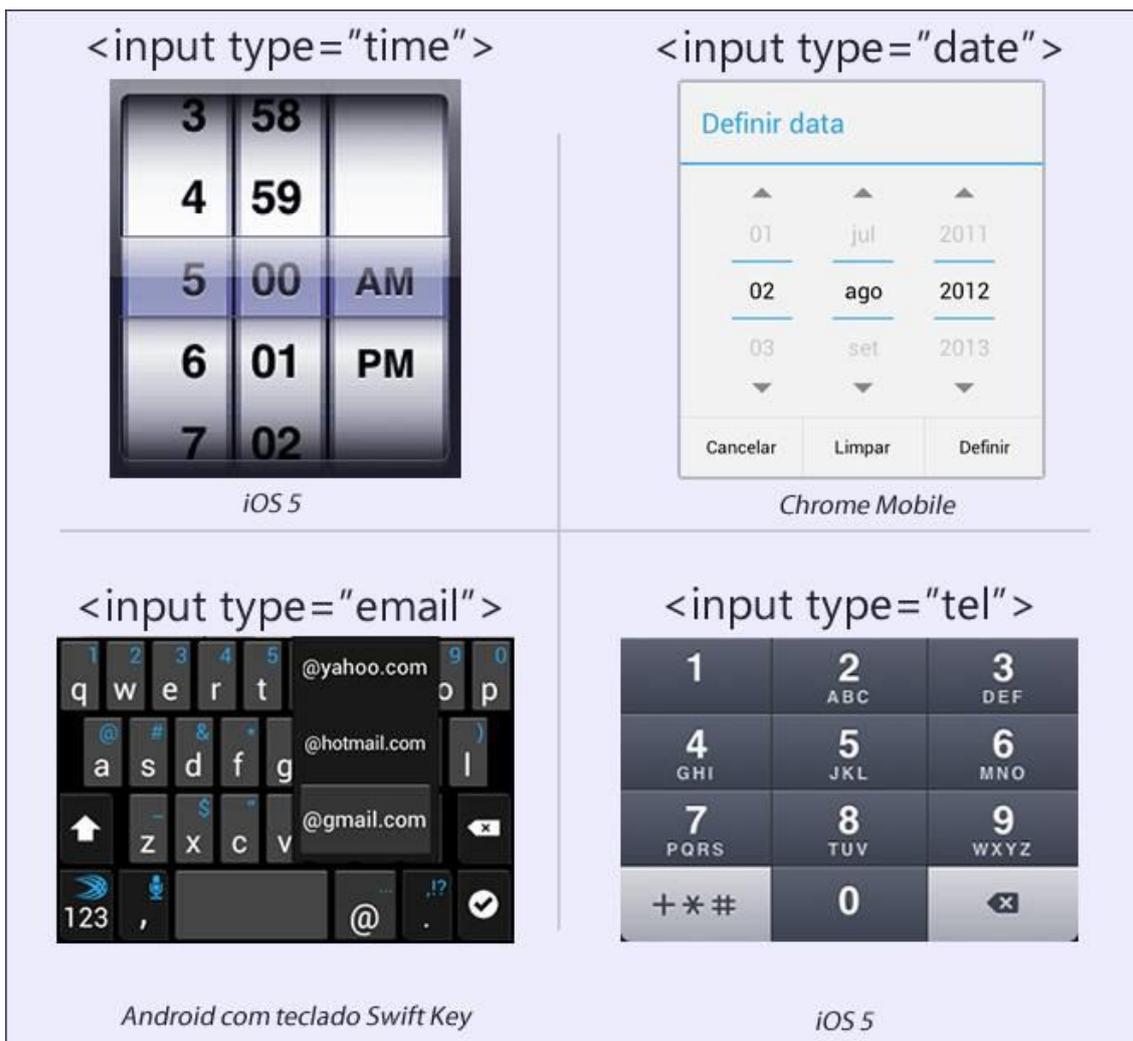


Figura 5 - Teclado virtual de diversos dispositivos móveis.

Além disso, algumas práticas como definir os atributos em ordem alfabética, utilizar fonte minúscula e padronizar o uso das aspas, usando sempre duplas ou simples. Tais práticas podem aperfeiçoar o desempenho da compressão e descompressão dos arquivos.

4.2.1.5 CM2 - Manter componentes abaixo de 25k

Esta prática é efetiva somente para iPhones desenvolvidos antes da versão 4S, devido ao fato destes dispositivos não realizarem o cache de arquivos com mais de 25kb.

Tal prática classifica-se como tendo média efetividade, pelo fato de a mesma não ser funcional na grande maioria dos dispositivos.

4.2.2 Servidor

Nesta seção, serão citadas configurações que devem ser implementadas nos servidores, capazes de promover ganho de desempenho no carregamento das páginas, no lado cliente, e outras para que algumas das técnicas utilizadas no capítulo anterior tenham efeito.

4.2.2.1 SA1 - Compressão

Recurso do servidor que comprime em tempo de execução os dados HTML, CSS ou Javascript, tornando-o menor, sendo então transmitido através da rede de forma mais rápida e eficiente. Esse arquivo é reconhecido e descomprimido pelo navegador, e então processado [26].

É preciso cuidado no momento de aplicar as técnicas de compressão na página, pois, em alguns casos dependendo do tamanho do arquivo, pode causar um efeito indesejado na otimização, não é recomendado comprimir arquivos muito pequenos (ex: menos que 150 bytes), pois o tempo de compressão e descompressão pode também sobrepor o ganho de tempo com a transferência.

As principais tecnologias de compressão disponíveis no servidor são Gzip e Deflate. É possível também, configurar o servidor para que as páginas sejam pré-comprimidas e colocadas em cache para utilizações futuras.

4.2.2.2 SA2 - Cache

A grande maioria das páginas *web* possuem diversos conteúdos os quais não sofrem alterações em grande parte do tempo, tais como arquivos de imagem, CSS e Javascript.

Estes recursos podem ser armazenados no cliente utilizando-se o cache HTTP. O uso do cache pode trazer duas vantagens: a redução de tempo de carregamento e redução de tamanho da página.

Para configuração do cache é necessário configurar os seguintes atributos: expiração no atributo “*Expires*” do cabeçalho HTTP, ou uma idade máxima, “*Cache-Control: max-age*”, no mesmo cabeçalho. Estes atributos definem o tempo no qual o recurso ficará armazenado em cache e deve seguir algumas regras [26] [25].

Para recursos os quais não se sabe quando serão alterados, recomenda-se a utilização de um prazo de permanência em cache entre um mês e um ano, não devendo ultrapassar este prazo. Já nos casos em que se sabe exatamente quando um recurso será alterado, é correto utilizar datas de expiração mais curtas, porém se não há certeza de que esse recurso será alterado em breve, mesmo que exista uma probabilidade alta de isso acontecer, a data de expiração deve ser marcada como a mais longa possível.

Caso esses atributos não sejam configurados no cabeçalho do HTTP no servidor, os navegadores utilizam suas próprias políticas de gerenciamento de cache, o que não é recomendado.

Para que se aproveite ao máximo este recurso, é recomendada a sua aplicação ao máximo de conteúdo estático na página.

Outro atributo importante do cabeçalho HTTP é o *Last-Modified*. Este atributo marca a data da última alteração do recurso e ajuda o navegador a definir quando o arquivo deve ou não ser atualizado no cache local [26].

4.2.2.3 SA3 - SPDY

Segundo Bryan McQuade e Matthew Steele, o SPDY (pronunciado "Speedy") é um protocolo que permite que os *sítes* sejam transmitidos de forma mais eficiente, resultando em melhorias de tempo de carregamento de páginas de até 55% [10].

Trata-se de um protocolo experimental desenvolvido pela Google que tem como objetivo fazer um uso mais eficiente do protocolo TCP, introduzindo uma nova camada binária para permitir requisições e respostas multiplexadas, priorização, minimizar e até eliminar a latência de rede desnecessária.

Apesar de seu grande potencial, está classificado neste trabalho como de média eficiência devido ao fato de ainda ser experimental e não haver suporte a esta tecnologia nos principais servidores de hospedagem brasileiros.

4.2.2.4 SA4 - CDN

CDN, Content Delivery Network ou rede de distribuição de conteúdo é uma coleção de servidores *web* distribuídos em vários locais, tem por objetivo entregar conteúdo aos usuários de maneira mais eficiente [8]. Quando há uma solicitação do usuário, é selecionado o servidor com capacidade de prover as informações de maneira mais rápida, podendo ser o servidor mais próximo fisicamente ou o com o menor número de saltos de rede ou o servidor com o tempo de resposta mais rápido.

Apesar de seu grande potencial, principalmente em relação à melhora dos problemas relativos à latência, está classificado neste trabalho como de média eficiência devido ao fato de não haver suporte a esta tecnologia nos principais servidores de hospedagem brasileiros, porém, há uma grande oferta deste serviço principalmente nos Estados Unidos, sendo viável a sua utilização caso haja necessidade de prover acesso de alta performance para os usuários desta região.

5 ESTUDO DE CASO

Este capítulo descreve o estudo de caso realizado com a aplicação do MEEPO.

Como objeto de estudo foi utilizado uma página desenvolvida na disciplina Análise de Sistemas II, cursada no segundo semestre de 2012, que faz parte da graduação tecnológica em Análise e Desenvolvimento de Sistemas, do mesmo curso em que este trabalho de pesquisa se enquadra.

A página trata-se de uma vitrine virtual de imóveis, a qual é utilizada para vendas de imóveis na região de Campinas. Na execução dos testes foi adotada a versão estática da página inicial, sem processamentos lógicos ou conexões com banco de dados, excluindo assim qualquer possível influência do tempo de processamento do servidor no resultado obtido, não sendo contemplada nenhuma subpágina do *site*. A página criada foi hospedada na mesma empresa em que o *site* original está hospedado atualmente.

Os testes foram executados a partir de um ambiente controlado, através da ferramenta GTmetrix, melhor explicada no item 5.1, que simula uma conexão 3G com velocidade e latência constantes de 1.6 mbps e 200 ms, através de um dispositivo móvel marca Samsung, modelo Galaxy Nexus, com Android 4.2, utilizando navegador Chrome v1.4.3. Tal ambiente foi selecionado ao invés de um teste real devido à instabilidade das redes móveis nacionais que durante os testes iniciais possuíam diversos picos com alto e baixo desempenho, não permitindo aferir com precisão a eficiência das técnicas aplicadas, forçando assim o uso de um ambiente controlado.

5.1 FERRAMENTAS UTILIZADAS

Foi utilizada a ferramenta online GTmetrix [11] que possui uma área específica para testes de desempenho de páginas em dispositivos móveis, na qual a partir da escolha de um dispositivo da lista, é executado um teste em um

dispositivo real e em seguida é exibido um relatório com informações detalhadas do carregamento incluindo gráficos e gravações de vídeo.

O *WebPageTest* é uma ferramenta similar ao *GTmetrix*, possuindo alguns recursos a mais, porém não é específica para o uso em dispositivos móveis.

5.2 METODOLOGIA E MÉTRICAS UTILIZADAS

As recomendações do MEEPO foram aplicadas gradativamente analisando os resultados obtidos a cada etapa do processo conforme ilustrado na Tabela 3.

Tabela 3 - Etapas de Aplicação das recomendações do MEEPO

Etapa	Recomendações Aplicadas
1	CA1
2	CA1 + CA2
3	CA1 + CA2 +CA3
4	CA1 + CA2 +CA3 +SA1
5	CA1 + CA2 +CA3 +SA1 + SA2

A métrica utilizada no estudo de caso foi baseada na medição do tempo de carregamento total, tempo para exibir a primeira informação para o usuário, quantidade de requisições e tamanho total do *site* antes e depois de cada teste.

5.3 EXECUÇÃO DOS TESTES

Os testes foram executados na ferramenta *GTmetrix*, configurada para utilizar o navegador *Chrome* em um dispositivo *Android*, *GalaxyNexus*. Conectado de um servidor de *Vancouver*, *Canadá*.

O primeiro teste foi realizado a partir da versão estática do *site* real, na qual foram removidas todas as intervenções de banco de dados e

processamento de lógica de negócio. Esta versão será usada como base para os testes deste estudo de caso.

Os principais resultados obtidos nesta versão podem ser visualizados na Tabela 4.

Tabela 4 - Resultados obtidos no teste da versão sem otimização.

Primeira aparição	Tempo total de carregamento	Tamanho da Página	Total Requisições
4.01 seg	7.15 seg	1.17 mb	43

5.3.1 Aplicação das recomendações MEEPO

A partir do *site* estático padrão foram aplicadas as recomendações do MEEPO. Para cada iteração foi gerado um relatório de resultados de performance obtidos.

5.3.1.1 Aplicando a Etapa 1

Nesta etapa foi possível aplicar todas as recomendações da etapa 1 do MEEPO. Os arquivos Javascript e CSS foram minificados, as imagens foram otimizadas e as dicas para melhora de compressão foram aplicadas.

A Tabela 5 exibe os principais resultados obtidos nesta etapa.

Tabela 5 - Resultados obtidos no teste executado após a aplicação da etapa 1.

Primeira aparição	Tempo total de carregamento	Tamanho da Página	Total Requisições
2.98 seg	5.53 seg	612 kb	43

5.3.1.2 Aplicando a Etapa 2

Nesta etapa também foi possível aplicar todas as recomendações CA2 do MEEPO. As folhas de estilo foram aplicadas diretamente na página e todos os *Javascripts* foram alterados para serem carregados de maneira assíncrona.

Os principais resultados obtidos após esta etapa de aprimoramento, estão na Tabela 6.

Tabela 6 - Resultados obtidos no teste executado após a aplicação da etapa 2.

Primeira aparição	Tempo total de carregamento	Tamanho da Página	Total Requisições
1.75 seg	4.74 seg	599 kb	37

5.3.1.3 Aplicando a Etapa 3

As recomendações CA3 foram parcialmente utilizadas nesta etapa. Devido ao fato do *site* não utilizar redirecionamento, não foi preciso aplicar a recomendação de removê-la. O *site* também não possui imagens que pudessem se beneficiar do uso de *sprites* CSS e não havia recursos duplicados que precisassem ser removidos.

A recomendação utilizada nesta etapa foi a de concatenar arquivos Javascript e CSS.

Os principais resultados obtidos após esta etapa de aprimoramento, compõem a Tabela 7.

Tabela 7 - Resultados obtidos no teste executado após a aplicação da etapa 3.

Primeira aparição	Tempo total de carregamento	Tamanho da Página	Total Requisições
1.43 seg	4.36 seg	598 kb	35

5.3.1.4 Aplicando o Etapa 4

Nesta etapa, o recurso de compressão do lado servidor foi ativado.

Os principais resultados obtidos após esta etapa de aprimoramento, são listados na Tabela 9

Tabela 8 - Resultados obtidos no teste executado após a aplicação da etapa 4.

Primeira aparição	Tempo total de carregamento	Tamanho da Página	Total Requisições
1.33 seg	3.89 seg	321 kb	33

5.3.1.5 Aplicando a Etapa 5

Nesta etapa, os recursos de cache do lado servidor foram configurados.

Os principais resultados obtidos após esta etapa de aprimoramento estão dispostos na Tabela 10.

Tabela 9 - Resultados obtidos no teste executado após a aplicação da etapa 5.

Primeira aparição	Tempo total de carregamento	Tamanho da Página	Total Requisições
1.33 seg	3.89 seg	321 kb	33

5.3.1.6 Recomendações não utilizadas

As recomendações de média eficiência do lado cliente do MEEPO, CM1 e CM2 não foram utilizadas.

O *site* não possui recursos explorados por Javascript que podem ser trocados por HTML5 e os componentes que podem ser reduzidos para menos 25 kb já foram reduzidos em etapas anteriores.

As recomendação SM1 e SM2 não foram implementadas, pois o servidor utilizado para os testes não possuía suporte às tecnologias SPDY e CDN.

Devido a esses fatos, estas recomendações não agregaram valor ao processo de melhoria deste estudo de caso.

6 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Neste capítulo, os resultados dos testes de aplicação do MEEPO obtidos no capítulo anterior serão analisados e discutidos a fim de aferir se os seus objetivos foram cumpridos.

6.1 RESULTADOS DOS TESTES

A finalização do carregamento do DOM representa o momento em que é apresentado algo ao usuário pela primeira vez. Analisando o Gráfico 5, observa-se que a aplicação do MEEPO foi altamente eficaz, reduzindo ao final do experimento em 1/3 o tempo de carregamento do DOM, destacando o CA1 e CA2 que foram responsáveis por 84% deste resultado.

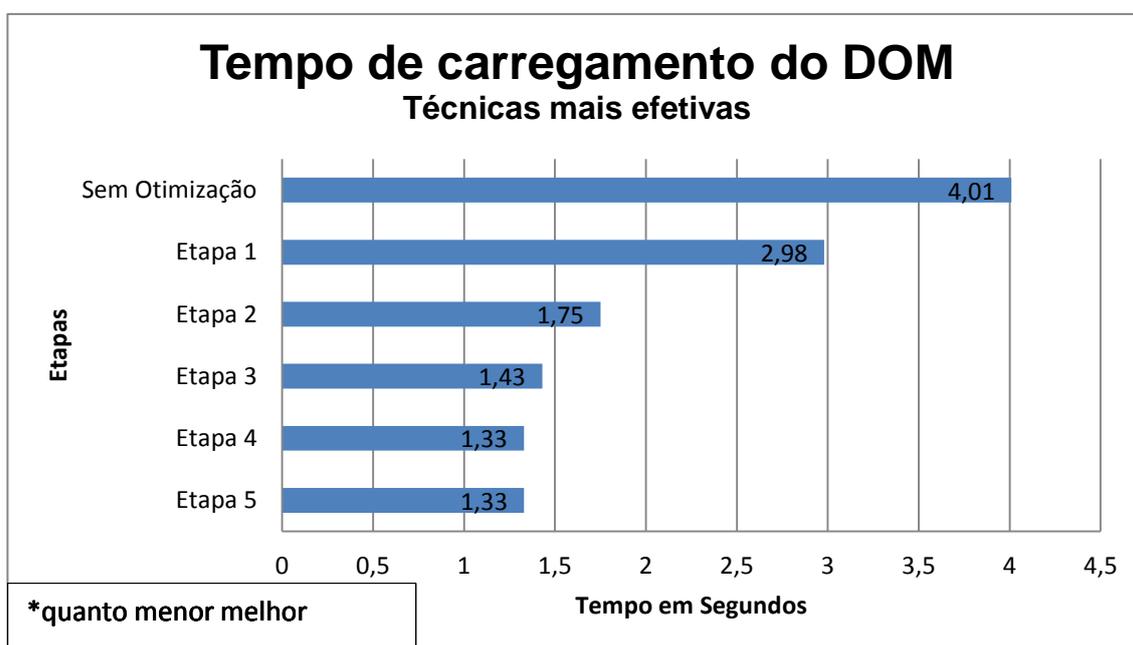


Gráfico 5 – Tempo da primeira visualização.

Devido à baixa largura de banda das redes móveis, o tamanho do *site* possui grande influência no tempo de carregamento. Analisando o Gráfico 6, observa-se que a aplicação das recomendações CA1 e SA1, reduziram em 3.6 vezes o tamanho do *site* ao final do experimento. Outras recomendações não afetaram de maneira efetiva o tamanho total da página.

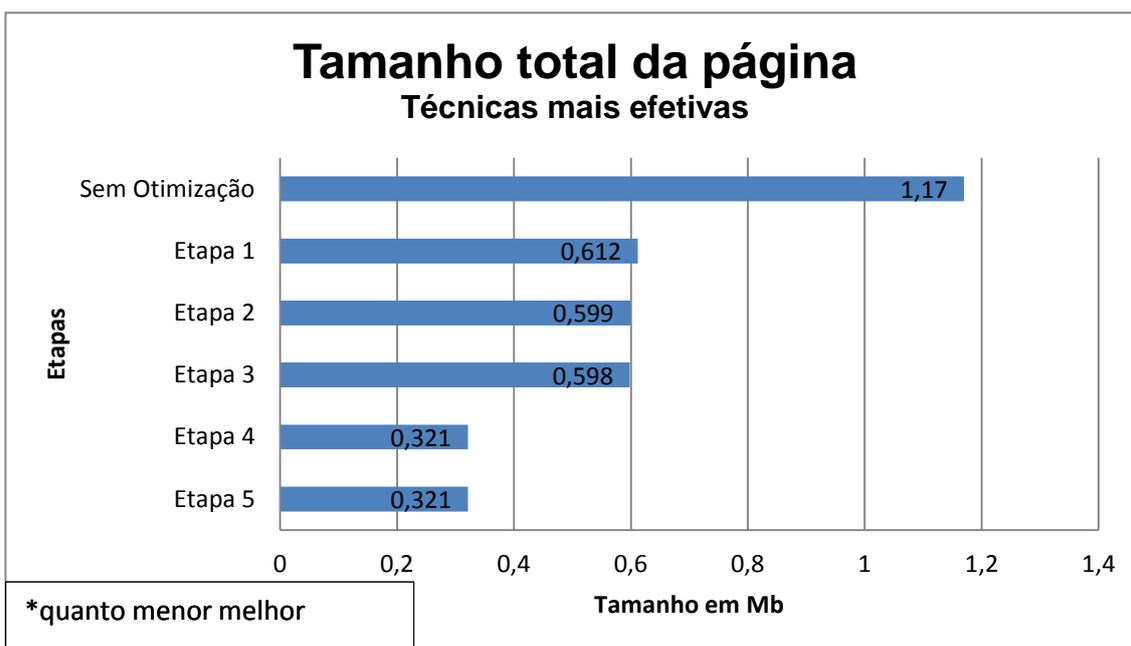


Gráfico 6 – Tamanho total da página.

O tempo para finalização do carregamento da página representa a quantidade de tempo necessário para o *site* carregar completamente. Analisando o Gráfico 7, observa-se que a aplicação do MEEPO proporcionou uma redução progressiva a cada etapa de otimização, chegando a quase a metade do tempo de carregamento no fim do experimento.

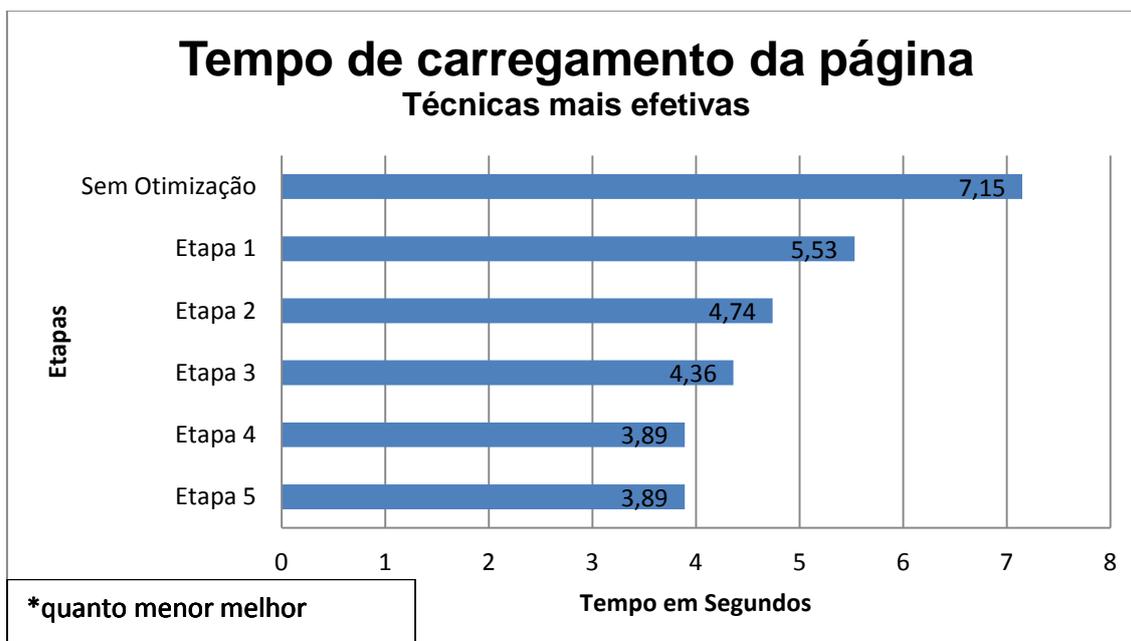


Gráfico 7 – Tempo de carregamento da página.

A quantidade de requisições de uma página possui interferência no tempo de carregamento de uma página em dispositivos móveis, pois cada uma sofre influência do tempo de latência da conexão e são limitadas a quantidade de conexões paralelas que o navegador suporta. Analisando o Gráfico 8, observa-se que ao final do experimento a aplicação do MEEPO proporcionou um redução de 25% nas requisições.

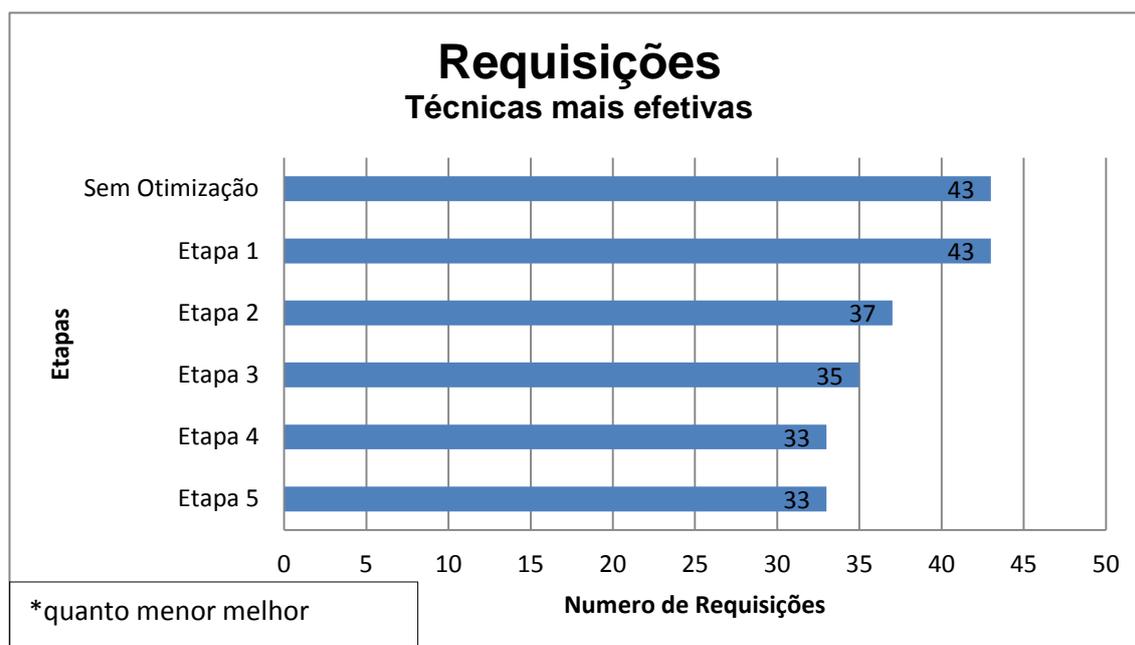


Gráfico 8 – Número de requisições HTTP.

6.2 CONCLUSÃO SOBRE OS RESULTADOS

Analisando os Gráficos 9 e 10 pode-se verificar um ganho significativo de desempenho no *site* otimizado comparando o *site* padrão. Há um ganho de 23% na velocidade do *site* só na aplicação do CA1, enquanto o uso do CA2, apesar representar apenas 14% de melhora no carregamento do *site*, obteve um ganho de 41% no tempo para exibir algum conteúdo ao usuário, variante com muito mais relevância. Além disso, após a aplicação do CA1 e do SA1 houve uma diminuição de 84% no tamanho da página. Os ganhos de desempenho das outras recomendações tiveram menos impacto.

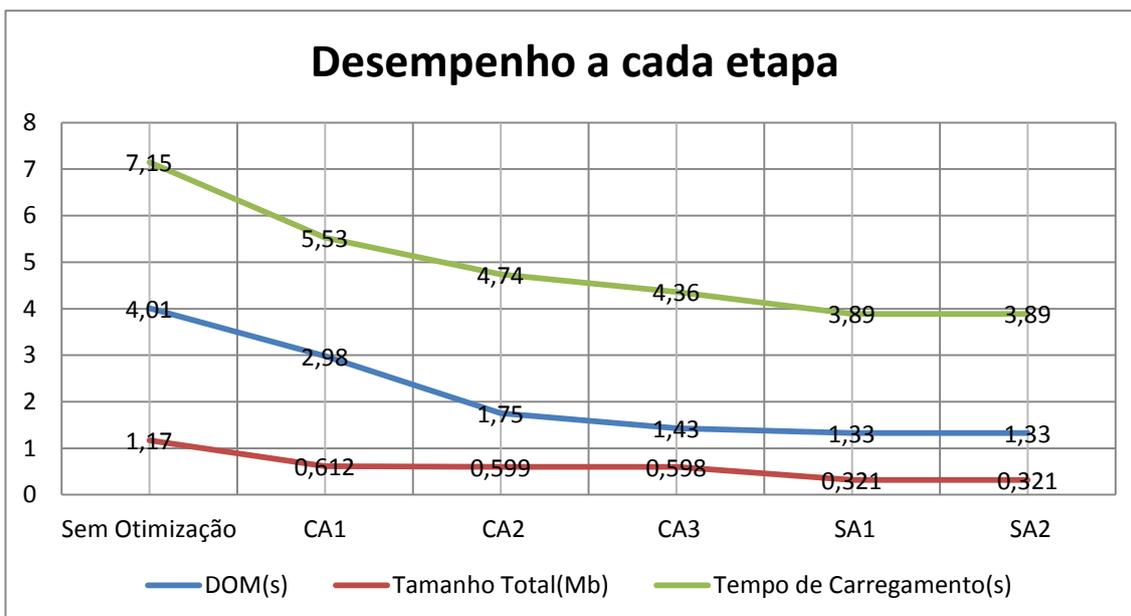


Gráfico 9 – Curva de ganho de desempenho por métrica.

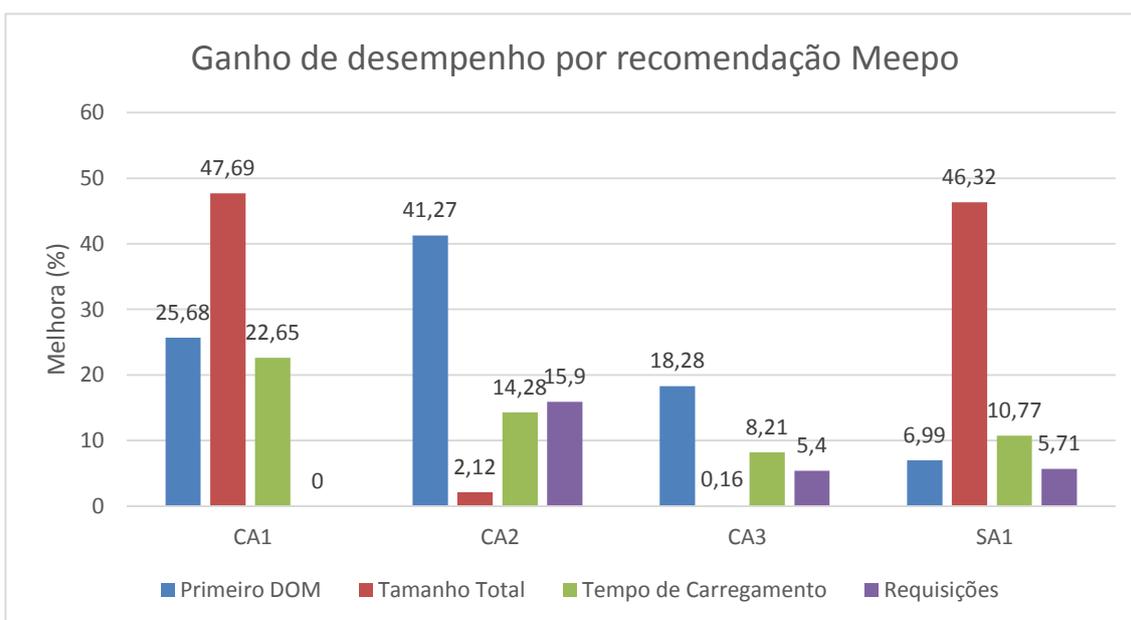


Gráfico 10 – Porcentagem de ganho de desempenho individual para cada recomendação

O resultado do experimento, representado no Gráfico 11, demonstra que a utilização do MEEPO mostrou-se eficiente, com ganhos de 66,83% no tempo de carregamento para primeira exibição ao usuário e 45,59% no tempo de carregamento final, obteve uma redução de 72,56% no tamanho total da página e uma queda de 23,25% nas requisições. Sobretudo, levando-se em consideração a experiência do usuário, a redução no tempo de carregamento

do DOM permitiu que o usuário visualizasse informações na tela em apenas 1,33 segundos, contra os mais de 4 segundos originais.

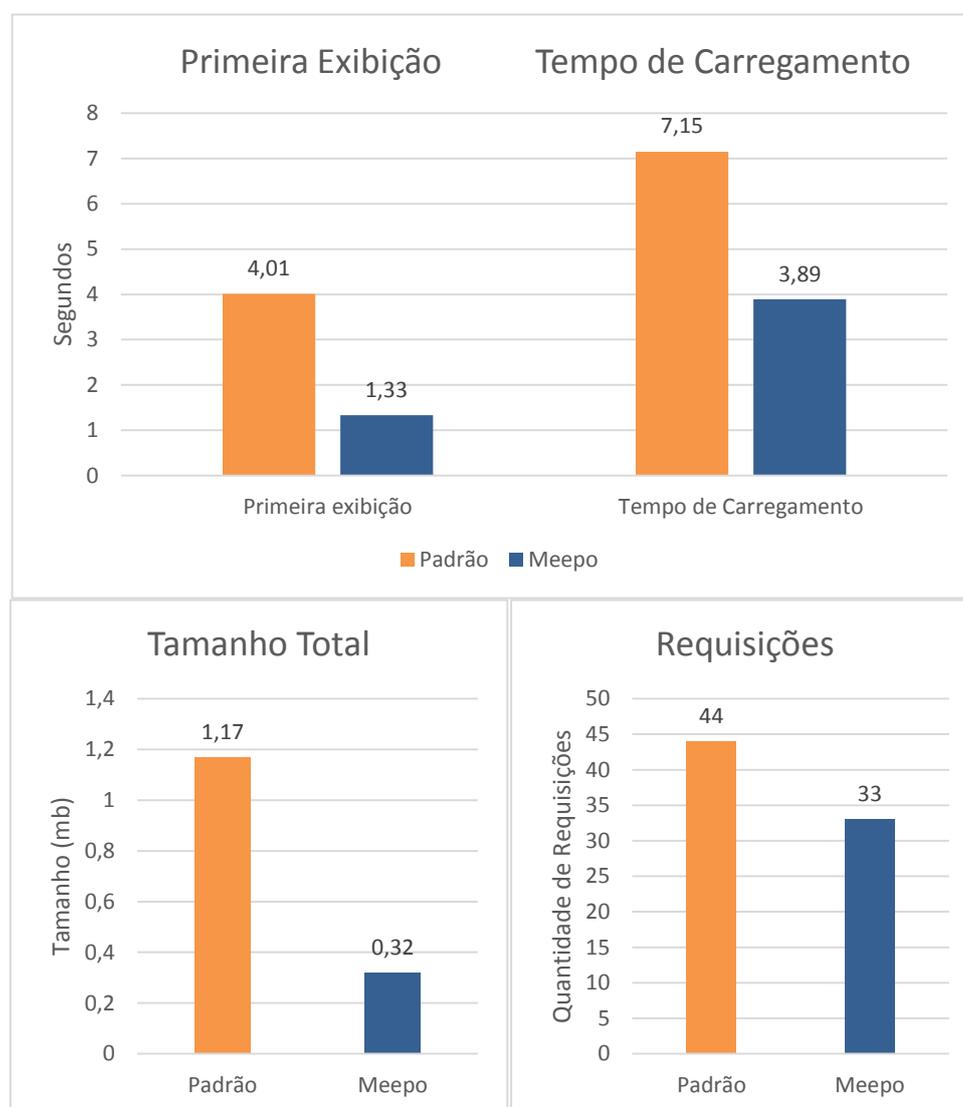


Gráfico 11 – Resultados finais

Em suma, todas as recomendações tiveram resultados positivos, mas deve-se destacar eficiência e facilidade de aplicação das recomendações SA1, CA1 e CA2, nesta ordem. Em cenários onde há a necessidade de melhoria em um curto espaço de tempo, estas são, sem dúvidas, as primeiras a serem utilizadas.

Conforme esperado, o uso do SA2 não obteve ganho de desempenho, pois a sua aplicação só prevê melhoria a partir do segundo acesso, teste que devido à limitação da ferramenta, não foi possível realizar.

Infelizmente não foi possível testar as outras recomendações relativas ao servidor, SA3 e SA4, pois tais recomendações não se encaixam no exemplo escolhido para uso no estudo de caso.

7 CONCLUSÃO

Este trabalho de pesquisa apresentou a importância do desempenho dos *sites* e os principais desafios para melhorá-los em dispositivos móveis. A partir do estudo de diversas técnicas de melhoria de desempenho de *site*, foram selecionadas as que possuíam maior eficiência em dispositivos móveis. Com base nestes estudos foi proposto um guia de recomendações de melhorias de desempenho *web* para dispositivos móveis, nomeado como MEEPO, como uma das principais contribuições desta pesquisa.

Considerando os resultados obtidos nos testes, apesar do estudo de caso não contemplar um ambiente que utilize todas as recomendações do MEEPO, o objetivo proposto para o guia foi alcançado.

No estudo de caso, a aplicação do MEEPO proporcionou ganhos de 66,83% no tempo de carregamento para primeira exibição ao usuário e 45,59% no tempo de carregamento final, além de obter uma redução de 72,56% no tamanho total da página e uma queda de 23,25% nas requisições. Após a aplicação do MEEPO, o usuário passou a receber informações na tela em apenas 1,33 segundos, contra os mais de 4 segundos originais.

Como apresentado neste trabalho, os principais desafios encontrados no ambiente de internet móvel como, alta latência e baixa largura de banda, baixo poder de processamento dos dispositivos e as limitações de cache e conexões paralelas, são minimizados com a aplicação das recomendações do MEEPO.

A aplicação de suas recomendações também prevê ganho de desempenho quando o acesso é provido em outras plataformas, com destaque para aquelas que utilizam-se de redes de alta latência em suas conexões.

Como recomendações sugerem-se que sejam aplicadas as técnicas não utilizadas no estudo de caso desta pesquisa, a fim de aferir sua eficácia, além de pesquisar outras técnicas de melhoria que possam vir a agregar ao MEEPO, como melhorias específicas da linguagem Javascript. Recomenda-se também que outro estudo de caso seja conduzido, aplicando o MEEPO desde o início do desenvolvimento da aplicação.

Por fim, como trabalhos futuros, mantendo o mesmo contexto de melhoria de desempenho de *sites* em dispositivos móveis, vislumbra-se a

pesquisa de melhorias relacionadas ao lado servidor que venham a agregar ganho de desempenho no carregamento dos *sites*.

8 REFERÊNCIAS

- [1] ACCENTURE. **Móvel Web Watch**, 2011.[1]
- [2] AKAMAI. **The State of Internet 4th Quarter, 2012 report**, 2012.
- [3] AKAMAI. **The State of Internet 3th Quarter, 2012 report**, 2012.
- [4] BIXBY, J. **Case studies from the móvel frontier: The relationship between faster mobile sites e business KPIs**. In: VELOCITY BERLIN, 2011, Berlin, DE-BE.
- [5] CERF, G.; KAHN, Robert E. **A Protocol for Packet Network Intercommunication**, *IEEE Transactions on Communications*, Vol. 22, No. 5, maio 1974 pp. 637-648.
- [6] ELGELMANN, K. **Entendendo a diferença entre Largura de Banda e Latência**. 19 ,mar. 2013. Disponível em: <<https://learningnetwork.cisco.com/docs/DOC-20318>>. Acesso em: 08 dez. 2013.
- [7] ENDES, A. **Arquitetura de Software: desenvolvimento orientado para arquitetura**. Rio de Janeiro: Campus, 2002.
- [8] ERMAN, J.; GERBER, A.; HUANG, J.; MAO, Z.; QUIAN, F.; QUAH, K.; SEN, S.; SPATSCHECK, O. **Web caching on smartphones: ideal vs. reality**. *ACM New York, NY, USA*, v. 1, n. 1, p. 124-140, 2012.
- [9] GOOGLE, **Make the Web Faster**. 09. jul. 2013. Disponível em: <<https://developers.google.com/speed/>>. Acesso em: 08 dez. 2013.
- [10] GRIGORIK, I. **High Performance Browser Networking**. Sebastopol, CA-US: O'Reilly Media, 2013.
- [11] GTmetrix, Disponível em: <<http://gtmetrix.com/dashboard.htm>>. Acesso em: 08 dez. 2013.
- [12] IBGE - Instituto Brasileiro de Geografia e Estatística - **Acesso à Internet e posse de telefone móvel celular para uso pessoal 2011**.
- [13] JAIN, A. **Is the web getting faster?**. In: VELOCITY CONFERENCE, 2013, Santa Clara, US-CA.
- [14] JAIN, A. **Global Site Speed Overview: How Fast Are Websites Around The World?**. 19 ab. 2012. Disponível em: <<http://analytics.blogspot.com.br/2012/04/global-site-speed-overview-how-fast-are.html>>. Acesso em: 08 dez. 2013.
- [15] LE HÉGARET, Philippe. Document Object Model (DOM). 19 jan. 2005.

Disponível em: <<http://www.w3.org/DOM/>>. Acesso em: 08 dez. 2013.

[16] LOPES, S. **A Web Mobile: Programe para um mundo de muitos dispositivos**. Casa do Código, 2013.

[17] MENDES, A. **Arquitetura de Software: desenvolvimento orientado para arquitetura**. Editora Campus. Rio de Janeiro - RJ, 2002

[18] MORRISON, M. **Use a Cabeça Javascript**. Alta Books, 2008.

[19] NIC.BR, **Campus Party 6 Medindo a Qualidade da Internet no Brasil**. 2012.

[20] NIELSEN, J. **Usability Engineering**. San Francisco: Morgan Kaufmann, 1993.

[21] SCHURMAN, E.; Brutlag, J. **The User and Business Impact of Server Delays, Additional Bytes, and HTTP Chunking in Web Search**. In: VELOCITY CONFERENCE, 2009, San Jose, US-CA.

[22] SILVA, A. **HTML5 A linguagem de marcação que revolucionou a web**. São Paulo: Novatec, jul. 2011.

[23] SILVA, M. **CSS3 - Desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das css3**. São Paulo: Novatec, dez. 2011.

[24] SINGHAL, A. **Using site speed in web search ranking**.(s.l) 31 out. 2013. Disponível em: <<http://googlewebmastercentral.blogspot.com.br/2010/04/using-site-speed-in-web-search-ranking.html>>. Acesso em: 08 dez. 2013.

[25] SOUDERS, S. **Even Faster Webpages**. Sebastopol, CA-US:O'Reilly Media, 2009.

[26] SOUDERS, S. **High Performance Web Sites**. Sebastopol, CA-US:O'Reilly Media, 2007.

[27] *WebPageTest*, Test a *website's* performance. Disponível em:<www.webpagetest.org>. Acesso em: 08 dez. 2013.

[28] WELSH, M. **Taming the mobile beast**. In: VELOCITY CONFERENCE, 2012, Santa Clara, US-CA.

[29] WILSON, F. **10 Golden Principles of Successful Web Apps**. In: FUTURE OF WEB APPS, 2010, Miami, US-FL.

[30] YAHOO, **Exceptional Performance**. Disponível em: <<http://developer.yahoo.com/performance/>>. Acesso em: 08 dez. 2013.

9 GLOSSÁRIO

Bit	Menor unidade de informação que pode ser armazenada ou transmitida
Byte	Unidade de informação digital equivalente a oito bits
Cache	Memória temporária, usada para guardar os dados a serem armazenados em um dispositivo de armazenamento, até que o referido dispositivo esteja pronto para receber esses dados;
Design	Concepção de um projeto ou modelo; planejamento.
DNS	Sistema de gerenciamento de nomes hierárquico e distribuído visando resolver nomes de domínios em endereços de rede
DOM	Especificação criada pelo W3C para uma interface multiplataforma que representa como as marcações, independentemente da linguagem, são organizadas e lidas pelo navegador.
Framework	Abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica.
Hardware	Conjunto de unidades físicas, componentes, circuitos integrados, discos e mecanismos que compõem um computador ou seus periféricos.
Href	Símbolo (tag) HTML que em um link, indica o endereço para o qual ser conduzido.
Id	Id é a sigla para Identity, que significa identidade, em português.
Internet	Internet é um conjunto de redes mundial, e o nome tem origem inglesa, onde inter vem de internacional e net significa rede, ou seja, rede de computadores mundial.
IP	Identificação de um dispositivo (computador, impressora, etc) em uma rede local ou pública.
JPEGs	Acrônimo de <i>Joint Photographics Experts Group</i> , é um método de compressão de imagens fotográficas e também é considerado como umformato de arquivo.
Mbps	Significa megabit por segundo, e é uma unidade de transmissão de dados equivalente a 1.000 kilobits por segundo ou 1.000.000 bits por segundo.
Minify	Técnica que remove todos os bytes extras da página ao retirar espaços vazios, comentários, caracteres desnecessários
Mouse	Dispositivo de entrada dotado de um a três botões, que repousa em uma superfície plana sobre a qual pode ser deslocado, e que, ao ser movimentado, provoca deslocamento análogo de um cursor na tela

ms	Unidade de medida de tempo. Corresponde a 10^{-3} segundos, ou seja, um milésimo de segundo. Atualmente é usado para medir o tempo que demoramos a enviar software pela internet, por exemplo: $1\text{ms} = 0,001\text{s}$ $1\text{s} = 1000\text{ms}$
Online	<i>Online</i> ou <i>on-line</i> é um anglicismo advindo do uso da Internet, sendo em linha uma tradução literal de <i>on-line</i> , pouco usada no português.
PDF	Formato de arquivo, desenvolvido pela Adobe Systems em 1993, para representar documentos de maneira independente do aplicativo, do <i>hardware</i> e do sistema operacional usados para criá-los.
Script	Conjunto de instruções que executam uma função, usado normalmente com uma linguagem macro ou linguagem em lote.
Sites	Conjunto de páginas <i>web</i> , isto é, de hipertextos acessíveis geralmente pelo protocolo HTTP na internet.
Software	Qualquer programa ou grupo de programas que instrui o <i>hardware</i> sobre a maneira como ele deve executar uma tarefa, inclusive sistemas operacionais, processadores de texto e programas de aplicação.
Sprites	São diversas imagens combinadas em uma única imagem que é manipulada por CSS, de forma a exibir somente o quadro referente à imagem desejada.
Streaming	Fluxo de mídia, forma de distribuir informação multimídia numa rede através de pacotes.
Tags	Estruturas de linguagem de marcação contendo instruções, tendo uma marca de início e outra de fim para que o navegador possa renderizar uma página.
W3C	World Wide Web Consortium é a principal organização de padronização da World Wide Web.
Web	Designa a rede que conecta computadores por todo mundo, a World Wide Web