



**UFOP**

Universidade Federal  
de Ouro Preto

**Universidade Federal de Ouro Preto  
Instituto de Ciências Exatas e Aplicadas  
Departamento de Computação e Sistemas**

**Estudo de mecanismos e fatores que  
impactam no desempenho de  
aplicações *Web***

**Manoel Gonçalves Carneiro Júnior**

**TRABALHO DE  
CONCLUSÃO DE CURSO**

ORIENTAÇÃO:  
Prof. Fernando Bernardes de Oliveira

**Setembro, 2017  
João Monlevade–MG**

**Manoel Gonçalves Carneiro Júnior**

**Estudo de mecanismos e fatores que impactam  
no desempenho de aplicações *Web***

Orientador: Prof. Fernando Bernardes de Oliveira

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

**Universidade Federal de Ouro Preto**

**João Monlevade**

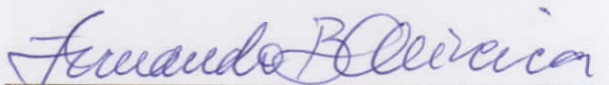
**Setembro de 2017**

# FOLHA DE APROVAÇÃO DA BANCA EXAMINADORA


## Estudo de mecanismos e fatores que impactam no desempenho de aplicações *Web*

Manoel Gonçalves Carneiro Júnior

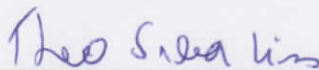
Monografia apresentada ao Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto como requisito parcial da disciplina CSI499 – Trabalho de Conclusão de Curso II do curso de Bacharelado em Sistemas de Informação e aprovada pela Banca Examinadora abaixo assinada:



Prof. Fernando Bernardes de Oliveira  
Doutor  
DECSI – UFOP



Prof<sup>ª</sup>. Daniela Rodrigues Dias  
Mestre  
Examinador  
DECSI – UFOP



Prof. Theo Silva Lins  
Mestre  
Examinador  
DECSI – UFOP

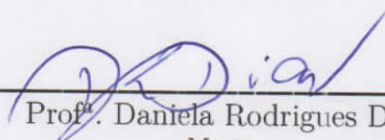
João Monlevade, 5 de setembro de 2017

## ATA DE DEFESA

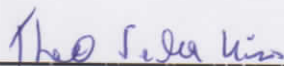
No dia 5 do mês de Setembro de 2017, às 15:00 horas, na sala C304 do Instituto de Ciências Exatas e Aplicadas, foi realizada a defesa de Monografia pelo(a) aluno(a) **Manoel Gonçalves Carneiro Júnior**, sendo a Comissão Examinadora constituída pelos professores: Prof. Fernando Bernardes de Oliveira, Prof<sup>o</sup>. Daniela Rodrigues Dias, Prof. Theo Silva Lins. O(a) candidato(a) apresentou a monografia intitulada: "**Estudo de mecanismos e fatores que impactam no desempenho de aplicações Web**". A comissão examinadora deliberou, por unanimidade, pela aprovação do candidato, com nota 9,5 (atue pontos e meio), concedendo-lhe o prazo de 15 dias para incorporação das alterações sugeridas ao texto final. Na forma regulamentar, foi lavrada a presente ata que é assinada pelos membros da Comissão Examinadora e pelo(a) graduando(a).



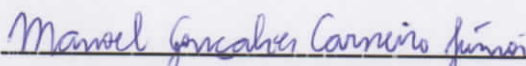
Prof. Fernando Bernardes de Oliveira  
Doutor  
DECSI - UFOP



Prof. Daniela Rodrigues Dias  
Mestre  
Examinador(a)  
DECSI - UFOP



Prof. Theo Silva Lins  
Mestre  
Examinador(a)  
DECSI - UFOP



Manoel Gonçalves Carneiro Júnior

## TERMO DE RESPONSABILIDADE

Eu, **Manoel Gonçalves Carneiro Júnior** declaro que o texto do trabalho de conclusão de curso intitulado “*Estudo de mecanismos e fatores que impactam no desempenho de aplicações Web*” é de minha inteira responsabilidade e que não há utilização de texto, material fotográfico, código fonte de programa ou qualquer outro material pertencente a terceiros sem as devidas referências ou consentimento dos respectivos autores.

João Monlevade, 5 de setembro de 2017

Manoel Gonçalves Carneiro Júnior

Manoel Gonçalves Carneiro Júnior

*Este trabalho é dedicado à minha mãe Dinéia de Lima Ribeiro Carneiro, meu maior exemplo de vida, perseverança e fé. Sou extremamente orgulhoso em ser filho de uma verdadeira guerreira! Dedico também à minha tia Geralda Diva Ribeiro, que está de camarote ao lado de Deus orando e torcendo por mim e por toda a família, saudades eternas.*

# Agradecimentos

Agradeço primeiramente a Deus, pois sem Ele eu nada seria. Aos meus pais Dinéia e Manoel, que me proporcionaram o dom da vida, são minha fortaleza e me apoiaram em todos os momentos dessa jornada, amo muito vocês. De maneira especial agradeço à minha irmã Danielle, que participou comigo dessa jornada, por estar sempre ao meu lado, me incentivando e não me deixando desistir, não sei o que seria de mim sem você, te amo sem limites. À minha irmã Nathalia, pelo apoio e por entender meus momentos de ausência e estudos, amo você. Com todo meu amor agradeço à minha namorada Bárbara, pelo incansável apoio, amor incondicional e por acreditar mais em mim do que eu mesmo, sua presença me acalmou por diversas vezes. Eu te amo muito e sou grato a Deus por ter me dado esse presente único. Às minhas avós, tios e tias, primos e primas (em especial à Bianca e Beatriz), e a toda minha família no geral, pelo incentivo e força nas horas difíceis e por acreditarem que esse dia chegaria. O meu agradecimento também à minha madrinha Flávia, por me incentivar a buscar meus sonhos e objetivos de vida. Ao meu orientador e professor Fernando, por toda a ajuda e suporte, me encorajando nas horas de desânimo e por confiar em mim acima de tudo, sem você esse trabalho não seria possível. Aos meus amigos de caminhada na UFOP: Eduardo, Fernanda, Lanna e Michel, pelas horas de descontração (nossos lanches eram os melhores) e pelos choros compartilhados nos intervalos. Nossos caminhos não foram fáceis, mas sei que no final tudo terá valido a pena. Por fim, agradeço a todos que direta ou indiretamente contribuíram para que eu chegasse até aqui. A todos o meu MUITO OBRIGADO!!!

*“Science is more than a body of knowledge; it is a way of thinking.”*

— Carl Sagan (1934 – 1996),  
*in: The Demon-Haunted World: Science as a Candle in the Dark.*



# Resumo

A performance das aplicações *Web* é um fator essencial para o bom funcionamento das mesmas, devido ao crescimento acentuado das funcionalidades, aliado à crescente demanda dos usuários. Por esses e outros fatores, a garantia de máxima eficiência desses serviços é primordial e desejável. Deste modo, o presente trabalho aborda um estudo acerca dos possíveis aspectos que podem influenciar na performance, no consumo de dados e na velocidade das aplicações. Para isso, foram realizados dois experimentos práticos a fim de verificar se alguns parâmetros previamente estabelecidos impactam na eficiência dos *softwares* em questão. O primeiro foi feito por meio da ferramenta *Apache JMeter* para avaliar o desempenho das páginas *Web* do [ICEA](#) e [ICSA](#), com variações simuladas da quantidade de usuários e iterações. Já para o segundo experimento foi criada uma página *Web* com o intuito de verificar a eficiência dela com algumas métricas consideradas relevantes. Para tal, foram adotadas as ferramentas *GTmetrix* e *WebPagetest*, de acordo com combinações de parâmetros criadas. O experimento com o *JMeter* não apontou diferenças estatísticas significativas entre as páginas testadas. Porém, foi possível identificar alguns aspectos relevantes presentes na análise, como a taxa de erros e vazão das páginas, induzindo a conclusões sobre a eficiência das mesmas em situações extremas de uso. Já os resultados do segundo experimento sugerem que alguns fatores selecionados causaram impactos significantes na página que foi desenvolvida para testes, como a localização do código *JavaScript*, além de algumas tendências de outras métricas, como a minificação do código e localização do [CSS](#). Ao final do estudo, um relatório de boas práticas foi elaborado com sugestões para o desenvolvimento das aplicações.

**Palavras-chaves:** *Web*. aplicações. performance. técnicas. experimentos.

# Abstract

*The performance of Web applications is an essential factor for the proper functioning of the same, due to the accentuated growth of the functionalities, together with the increasing demand of the users. For these and other factors, ensuring maximum efficiency of these services is paramount and desirable. In this way, the present work approaches a study about the possible aspects that can influence in the performance, the data consumption and the speed of the applications. For this, two practical experiments were carried out to verify if some parameters previously established impact on the efficiency of the software in question. The first was done using the Apache JMeter tool to evaluate the performance of the [ICEA](#) and [ICSA](#) Web pages, with simulated variations of the number of users and iterations. For the second experiment, a Web page was created with the purpose of verifying its efficiency with some metrics considered relevant. To this end, the GTmetrix and WebPagetest tools were adopted according to combinations of parameters created. The experiment with JMeter did not show statistically significant differences between the tested pages. However, it was possible to identify some relevant aspects present in the analysis, such as the error rate and page flow, leading to conclusions about their efficiency in extreme situations of use. The results of the second experiment, however, suggest that some selected factors have caused significant impacts on the page that was developed for testing, such as the location of the JavaScript code, as well as some trends in other metrics such as code minification and [CSS](#) localization. At the end of the study, a good practice report was prepared with suggestions for application development.*

**Key-words:** *Web. applications. performance. techniques. experiments.*

# Lista de ilustrações

Figura 1 – Tela de criação do Plano de Teste . . . . .	27
Figura 2 – Caminho para a criação do Grupo de Usuários . . . . .	28
Figura 3 – Opções de edição do Grupo de Usuários . . . . .	29
Figura 4 – Caminho para inclusão da Requisição HTTP . . . . .	30
Figura 5 – Campos possíveis de edição da Requisição HTTP . . . . .	31
Figura 6 – Inclusão do Relatório de Sumário . . . . .	32
Figura 7 – Campos que podem ser editados do Relatório de Sumário . . . . .	33
Figura 8 – Alerta do <i>JMeter</i> . . . . .	33
Figura 9 – <i>Boxplot</i> com o tempo mínimo para responder à requisição HTTP . . . . .	34
Figura 10 – <i>Boxplot</i> com o tempo máximo para responder à requisição HTTP . . . . .	35
Figura 11 – <i>Boxplot</i> com o tempo médio para responder à requisição HTTP . . . . .	36
Figura 12 – <i>Boxplot</i> com os erros encontrados durante os testes . . . . .	37
Figura 13 – <i>Boxplot</i> com a vazão das páginas utilizadas nos testes . . . . .	38
Figura 14 – Vista parcial da página criada . . . . .	40
Figura 15 – Vista parcial da página criada . . . . .	41
Figura 16 – Apresentação dos resultados do <i>GTmetrix</i> . . . . .	42
Figura 17 – Apresentação dos resultados do <i>WebPagetest</i> . . . . .	43
Figura 18 – <i>Boxplots</i> com o <i>First View</i> e os parâmetros criados . . . . .	44
Figura 19 – <i>Boxplots</i> com o <i>Full Load</i> e os parâmetros criados . . . . .	45
Figura 20 – Resultado do teste de <i>Tukey</i> para o parâmetro <i>First View</i> . . . . .	47
Figura 21 – Resultado do teste de <i>Tukey</i> para o parâmetro <i>Full Load</i> . . . . .	48
Figura 22 – Exemplo de <i>Boxplot</i> . . . . .	59

# Lista de tabelas

Tabela 1 – Valor- $P$ dos parâmetros de entrada . . . . .	38
Tabela 2 – Combinações de parâmetros criada . . . . .	41
Tabela 3 – Sumário dos resultados dos experimentos . . . . .	46
Tabela 4 – Resultados dos testes com o <i>Apache JMeter</i> . . . . .	55
Tabela 5 – Resultados dos testes com a página <i>Web</i> criada - Parte 1 . . . . .	56
Tabela 6 – Resultados dos testes com a página <i>Web</i> criada - Parte 2 . . . . .	57

# Lista de abreviaturas e siglas

**ANOVA** Análise de Variância

**CCTCI** Comissão de Ciência e Tecnologia, Comunicação e Informática

**CSS** *Cascading Style Sheets*

**FOUC** *Flash of Unstyled Content*

**HTML** *Hypertext Markup Language*

**HTTP** *HyperText Transfer Protocol*

**IBGE** Instituto Brasileiro de Geografia e Estatística

**ICEA** Instituto de Ciências Exatas e Aplicadas

**ICSA** Instituto de Ciências Sociais Aplicadas

**JS** *JavaScript*

**MANOVA** Análise de Variância Multivariada

**ms** milissegundos

**PHP** *PHP: Hypertext Preprocessor*

**PL** Projeto de Lei

**Telebrasil** Associação Brasileira de Telecomunicações

**UFOP** Universidade Federal de Ouro Preto

**VVT** Validação, Verificação e Teste

**WWW** *World Wide Web*

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	O problema de pesquisa	16
1.2	Objetivos	16
1.3	Metodologia	17
1.4	Organização do trabalho	17
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>19</b>
2.1	Teste de <i>software</i>	21
2.1.1	Testes de performance, carga e <i>stress</i>	21
2.1.2	Ferramentas de teste	22
2.2	Projeto de Lei (PL) 174/2016   PL 7182/2017	23
2.3	Considerações finais	24
<b>3</b>	<b>TESTES COM O APACHE JMETER</b>	<b>26</b>
3.1	Configurando o <i>Apache JMeter</i>	26
3.1.1	Criando um Plano de Teste ( <i>Test Plan</i> )	26
3.1.2	Criando um Grupo de Usuários ( <i>Thread Group</i> )	27
3.1.3	Configurando a quantidade de Usuários Virtuais e Contador de Iterações ( <i>Number of Threads e Loop Count</i> )	27
3.1.4	Adicionando o tipo de Requisição ao Grupo de Usuários	27
3.1.5	Adicionando Relatório de Sumário ( <i>Summary Report</i> )	28
3.2	Executando os testes	29
3.3	Testes realizados	30
3.4	Resultados dos testes com o <i>Apache JMeter</i>	33
3.5	Considerações finais	39
<b>4</b>	<b>TESTES COM UMA PÁGINA WEB</b>	<b>40</b>
4.1	Página <i>Web</i> criada	40
4.2	Configuração e execução do teste	41
4.3	Resultados do experimento	43
4.4	Considerações finais	46
<b>5</b>	<b>CONCLUSÃO</b>	<b>49</b>
5.1	Relatório de boas práticas	50
5.2	Trabalhos futuros	50

	<b>REFERÊNCIAS</b> . . . . .	<b>52</b>
	<b>APÊNDICES</b>	<b>54</b>
	<b>APÊNDICE A – TABELAS ELABORADAS PELO AUTOR</b> . . . . .	<b>55</b>
	<b>APÊNDICE B – FERRAMENTA GRÁFICA UTILIZADA NA ANÁLISE DOS RESULTADOS</b> . . . . .	<b>58</b>
<b>B.1</b>	<i>Boxplot</i> . . . . .	<b>58</b>

# 1 Introdução

A Internet, também conhecida como *Web*, têm se tornado uma notável ferramenta de revolução na forma de comunicação e relacionamento interpessoal entre a sociedade nos dias atuais. Ela modificou significativamente a forma de interação entre as pessoas, tanto no contexto familiar quanto em outros grupos sociais existentes. Também mudou, de certa maneira, a forma de viver, ensinar, aprender, trabalhar, comprar e divertir do mundo atual. A *Web* apresenta ainda benefícios no uso de tecnologias ligadas ao conhecimento, na inclusão social, na relação entre pessoas e organizações e na criação de valores.

Segundo informação divulgada pelo Instituto Brasileiro de Geografia e Estatística (IBGE) <sup>1</sup>, o Brasil possui uma população total estimada em torno de 208 milhões de habitantes. De acordo com um balanço da Associação Brasileira de Telecomunicações (Telebrasil) <sup>2</sup> do mês de junho, a quantidade de dispositivos com acesso às redes 3G e 4G chegou a 202 milhões no Brasil. Segundo esse balanço, 5.030 municípios já contam com infraestrutura móvel de Internet, cidades nas quais moram 98,4% da população.

Paralelamente aos fatores citados acima, têm-se o crescimento e utilização constantes das aplicações *Web*. Elas necessitam, cada vez mais, da obtenção do máximo desempenho possível, uma vez que o número de usuários cresce paralelamente a elas. Os usuários, por sua vez, requerem a melhor performance de uso, na qual quanto mais rapidamente ele obtiver as solicitações feitas, melhor ele avaliará a aplicação utilizada.

Tendo-se em vista que o desempenho é um fator primordial para avaliar a qualidade de uma aplicação *Web*, logo se associa a isso as características da estrutura de *hardware* existente no ambiente dos usuários. Porém, não adianta apenas se ter uma boa estrutura física para se garantir o máximo de desempenho possível, pois vários outros fatores podem impactar diretamente nesse quesito.

O tema do projeto é um estudo acerca de métodos e fatores que influenciam na velocidade e na performance de aplicações *Web*. O estudo será realizado por meio de um levantamento bibliográfico, visando detectar os fatores citados, além da realização de alguns experimentos práticos, de forma a fundamentar a importância do levantamento dos pontos identificados. A quantidade de aplicações *Web* existentes e o crescimento acentuado das mesmas justificam a importância desse projeto.

---

<sup>1</sup> Página disponível para visualização no endereço <<http://www.ibge.gov.br/apps/populacao/projecao/>>. Acessado em: 21 Ago 2017

<sup>2</sup> Informações sobre o balanço disponíveis em <<http://www.telebrasil.org.br/sala-de-imprensa/releases/8284-brasil-chega-a-200-milhoes-de-acessos-moveis-a-internet>>. Acessado em: 21 Ago 2017



## 1.1 O problema de pesquisa

Um exemplo do problema relacionado à performance é fruto de uma pesquisa realizada pela *Amazon*, uma empresa de comércio eletrônico dos Estados Unidos. De acordo com [Viegas \(2016\)](#), a pesquisa apresenta que o faturamento da *Amazon* aumenta 1% a cada 100 milissegundos (ms) poupados do carregamento. Ou seja, quanto menor a latência (tempo gasto para um pacote de dados ir de um ponto a outro numa rede), maior o faturamento da empresa. No caso citado, observa-se uma relação direta entre o desempenho de uma aplicação *Web* e o lucro de uma grande organização.

De acordo com [Rosa e Litwak \(2016\)](#), alguns países já possuem operadoras que adotam a prática de limitação do consumo de banda larga fixa, como os Estados Unidos e Canadá. As empresas desses países oferecem planos com e sem limite de acesso à Internet. Essa prática gera uma grande disparidade de ideias e opiniões entre os usuários.

Como apresentado em [Digital \(2016\)](#), os Estados Unidos também enfrentam polêmica com a limitação da Internet. Grandes empresas já se manifestaram a respeito, como a *Netflix*, que possui mais de 80 milhões de assinantes. Ela seria uma das empresas mais afetadas com essa limitação, por conta do grande consumo de Internet necessário para a exibição de filmes e séries.

Os fatores citados, dentre vários outros, sugerem que um estudo acerca de questões relacionadas à performance e consumo de dados e velocidade das aplicações *Web* é importante. Desse modo, este trabalho propõe o estudo de técnicas para se reduzir o consumo excessivo de dados na *Web*, bem como mecanismos que podem influenciar no seu desempenho.

## 1.2 Objetivos

O objetivo geral deste trabalho é identificar os fatores que podem influenciar no desempenho de aplicações *Web*, sejam sites em geral quanto sistemas. A proposta é estudar se fatores como a hospedagem, linguagens e arquitetura, métodos e técnicas podem afetar a velocidade e a performance das aplicações, bem como observar o impacto de cada um deles.

Este trabalho possui os seguintes objetivos específicos:

- Revisar a bibliografia acerca de métodos e técnicas relacionadas ao desempenho das aplicações, bem como avaliar estudos de casos e as respectivas influências no contexto em questão;
- Identificar e avaliar testes de performance que podem ser realizados na identificação de problemas;

- Planejar e realizar experimentos para avaliar a performance de aplicações *Web*;
- Analisar e discutir os resultados obtidos, além de identificar possíveis melhorias e considerações gerais sobre o processo;
- Elaborar um breve relatório contendo as boas práticas que devem ser observadas no desenvolvimento de aplicações *Web* e que podem influenciar diretamente no desempenho das mesmas.

### 1.3 Metodologia

O objeto de pesquisa deste trabalho é a obtenção dos fatores principais que podem influenciar o desempenho das aplicações *Web* nos dias atuais, considerando-se o crescente número delas dia após dia. Espera-se, por meio de testes de performance, averiguar se alguns critérios previamente estabelecidos podem impactar direta ou indiretamente na eficiência das aplicações.

A performance das aplicações será avaliada por meio de dois experimentos. O primeiro foi realizado com 2 páginas *Web* por meio de execuções da ferramenta *Apache JMeter*. As quantidades de usuários e iterações foram variadas a cada observação, adotando-se o critério aleatório entre as execuções. Já o segundo experimento consiste em testes feitos numa página *Web* criada pelo autor, utilizando-se para tal as ferramentas *GTmetrix* e *WebPagetest*, que serão descritas no Capítulo 2. As execuções têm como entrada variações de alguns parâmetros pré-definidos, explicados no Capítulo 4.

Os passos para execução deste trabalho são assim definidos:

- Revisão da literatura, contendo definições úteis para o entendimento do trabalho, além de casos envolvendo questões sobre performance;
- Testes com o *Apache JMeter* para identificar as possíveis diferenças existentes entre as páginas utilizadas, consistindo na primeira bateria de testes realizada;
- Testes com uma página *Web* criada pelo autor para avaliar se os fatores estabelecidos causam impactos diretos na performance da aplicação;
- Análise e discussão dos resultados apresentando os principais pontos observados na realização do trabalho.

### 1.4 Organização do trabalho

O restante deste trabalho é organizado como se segue. O Capítulo 2 apresenta uma revisão bibliográfica abordando vários pontos para o embasamento teórico do trabalho.

---

No Capítulo 3 têm-se a descrição dos testes realizados utilizando-se a ferramenta *Apache JMeter*, incluindo os resultados observados. Já o Capítulo 4 aborda a execução de testes de performance realizados na página *Web* desenvolvida pelo autor. O Capítulo 5 apresenta as conclusões e as considerações finais obtidas, além de sugestões para trabalhos futuros.

## 2 Revisão bibliográfica

Este capítulo engloba uma breve revisão acerca do tema do trabalho, servindo como base para o desenvolvimento do projeto. Nele são definidos alguns conceitos relevantes, e citadas algumas notícias e estudos na área de pesquisa em questão.

Antes de tudo, a definição de alguns termos e siglas faz-se necessária para o entendimento completo do trabalho. Com base no conteúdo descrito em [Caelum \(2017\)](#), têm-se:

- *Cascading Style Sheets (CSS)*: é uma linguagem de programação para definição de estilos que define o *layout* de páginas *Web*, como cores, fontes, larguras e alturas de formas, dentre outros;
- *JavaScript (JS)*: é uma linguagem de programação muito utilizada em páginas *Web*. Com ela, pode-se exibir mensagens e outras informações relevantes, fazer verificações ou alterar dinamicamente o visual das páginas, por exemplo;
- *Hypertext Markup Language (HTML)*: é uma linguagem de marcação - define formatos, maneiras de visualização e padrões dentro de um documento - utilizada para desenvolver aplicações, sendo a única que o navegador consegue interpretar para a exibição de conteúdo;
- *HyperText Transfer Protocol (HTTP)*: definido como um protocolo de comunicação usado basicamente para transferência de páginas HTML de um computador para a rede *World Wide Web (WWW)*.
- *PHP: Hypertext Preprocessor (PHP)*: uma linguagem interpretada livre, usada no desenvolvimento de aplicações no lado do servidor, sendo capaz de produzir conteúdo dinâmico na Internet.

Um caso semelhante ao da *Amazon*, como apresentado na Seção 1.1, foi notado com a *Walmart*, uma das maiores redes internacionais de lojas de departamento. De acordo com [Everts e Kadlec \(2016\)](#), uma pesquisa realizada chegou à conclusão que há um aumento de até 2% nas conversões para cada 1 segundo de melhoria no tempo de carga. Além disso, foi observado que a cada 100ms poupados no carregamento resulta em um aumento de até 1% na receita da rede. Já o *Financial Times*, jornal diário internacional da Inglaterra, realizou um teste acrescentando um atraso de um segundo a cada visualização de página, e percebeu uma considerável queda de 4,9% na quantidade de artigos lidos pelos usuários dentro de 7 dias. Outro atraso, porém de dois segundos, ocasionou uma queda de 4,4%, e

um atraso de três segundos obteve baixa de 7,2%. A *Netflix*, uma das maiores provedoras de filmes e séries de televisão *online*, notou uma queda de 43% nos valores gastos com largura de banda depois de implementar o *GZip*, que, segundo [Gailly e Adler \(2003\)](#), é uma técnica de compressão de dados a partir de algoritmos específicos.

É importante notar que alguns países têm discutido acerca de fatores de limitação de Internet. Essa prática reflete diretamente nas aplicações *Web*, uma vez que se torna necessário o consumo mínimo possível de dados. Existe no Brasil uma brecha na legislação (marco civil da Internet), que possibilita se estabelecer um limite de tráfego da banda larga fixa. Essa brecha ocasionou a criação de um Projeto de Lei ([PL](#)) visando a proibição de se impor limite à Internet, conforme será descrito na Seção 2.2.

Em casos nos quais se há o limite de tráfego, a observação de algumas questões da infraestrutura de *hardware* e *software* utilizados é imprescindível, como: o protocolo utilizado (HTTP 1.1 x 2.0, por exemplo), largura de banda e latência da rede, número de solicitações HTTP, número de redirecionamentos, versões desatualizadas de ferramentas e técnicas de desenvolvimento, banco de dados sobrecarregado, *hardware* e *software* obsoletos, dentre outras.

Buscando a resolução dessas e outras questões, observou-se na literatura algumas técnicas e ferramentas para se otimizar o máximo possível as aplicações *Web*. Segundo [Lopes \(2011\)](#), algumas das principais técnicas são:

- Habilitar GZIP: técnica em que todo conteúdo textual ([HTML](#), [CSS](#), [JS](#), etc) é comprimido antes de ser enviado para o cliente, chegando a cortar mais de 50% do tráfego total;
- Minificar *Javascript* e [CSS](#): remover caracteres desnecessários (para a aplicação), tais como espaços, tabulações, comentários, abreviar variáveis;
- Comprimir o [HTML](#): remover espaços, comentários e até certas aspas e fechamento de *tags* desnecessários;
- Não redimensionar imagens no [HTML](#) ou no [CSS](#): não deixar para diminuir imagens setando o *width* e o *height* no [HTML](#) ou [CSS](#). Sempre redimensione o arquivo original e referencie o tamanho direto;
- Avaliar formato das imagens (JPEG, PNG): essencial, pois as imagens costumam ser a parte mais pesada das páginas Web. Alguns formatos apresentam qualidade superior e consumo de dados menor que outros, otimizando o tempo de carregamento total;
- Inserir o código [JS](#) no final do arquivo: o [JS](#) bloqueia a renderização - que segundo [Moreira \(2013\)](#) é basicamente um processo de obtenção do produto final de um

processamento digital - de tudo que vem abaixo dele no código. Para evitar esse bloqueio, indica-se a inserção do **JS** ao final do corpo do código. O **JS** mal posicionado deixa a exibição da tela em branco durante muito tempo, atrasando a renderização;

- Inserir o código **CSS** no topo: o **CSS** também bloqueia a renderização mas, colocá-lo para baixo causaria o efeito denominado *Flash of Unstyled Content (FOUC)*, no qual a página é renderizada sem estilo e, quando o **CSS** acaba de baixar, ela é redesenhada com o estilo. Embora isso torne a primeira visualização da página mais rápida, mostrar uma página sem estilos não tem muito valor para o usuário. O *layout* fica sem estrutura, sem organização, e o efeito é aparentar que a página está demorando mais para carregar;
- Escrever código eficiente: **Tavares (2016)** define como eficiente um código que é enxuto, fácil de entender, direto, sem duplicidade, dentre outros aspectos. O desempenho pode ser bastante afetado por más práticas de codificação de **HTML**, **CSS** ou **JS**, por exemplo.

A base prática deste trabalho se fundamenta na realização de testes de *software*, buscando-se verificar a aplicação dos conceitos e técnicas citadas acima. A próxima Seção aborda os conceitos e os tipos de testes, um dos temas da área da Engenharia de *Software*.

## 2.1 Teste de *software*

Segundo **Delamaro et al. (2007)**, a fase de Teste de *Software* pode ser caracterizada como uma série de atividades, coletivamente chamadas de Validação, Verificação e Teste (**VVT**). Juntas, essas atividades buscam a garantia que tanto a forma pelo qual o *software* está sendo desenvolvido quanto o produto final em si estejam de acordo com a especificação inicial. As etapas de **VVT** não se aplicam apenas ao produto final (*software* pronto), mas podem e devem acontecer durante toda a vida útil do processo de desenvolvimento do mesmo.

Já **Pressman (2011)** define Teste de *Software* como uma sequência de atividades que são planejadas antecipadamente e executadas sistematicamente. Segundo aquele autor, uma estratégia de teste de *software* deve abordar testes de baixo nível, utilizados para avaliar se um pequeno trecho de código foi desenvolvido corretamente, bem como testes de alto nível, para avaliar as funções principais do sistema de acordo com os requisitos impostos pelo cliente.

### 2.1.1 Testes de performance, carga e *stress*

Na atualidade, a Internet está presente em vários lugares e acessível a uma grande parcela da população mundial. A quantidade de aplicações existentes se torna fundamental

aos indivíduos e empresas, visto que a dependência por essas funcionalidades é cada vez maior.

O volume de dados produzidos e trafegados pela *Web* aumenta proporcionalmente a quantidade de usuários e aplicações existentes. Por esse e outros motivos, garantir a máxima eficiência dessas aplicações tornou-se uma necessidade nos dias atuais. Para aferir essa eficiência, são necessários alguns testes. Seguem abaixo as definições dos tipos de testes, de acordo com [Integração \(2013\)](#):

- **Teste de performance:** têm como função observar o desempenho da aplicação em condições de uso tidas como normais, obtendo-se importantes informações sobre o uso das principais funcionalidades;
- **Teste de carga:** diferentemente dos testes de performance, os testes de carga visam verificar o comportamento do sistema com uma quantidade de usuários (reais ou virtuais) previamente estabelecida. Busca-se estabelecer um número estimado de usuários o mais próximo das condições reais possível;
- **Teste de *stress*:** o teste de *stress* tem como objetivo explorar os limites da aplicação, aumentando-se a carga indefinidamente.

### 2.1.2 Ferramentas de teste

Buscando mensurar a capacidade, a disponibilidade em situações extremas ou somente verificar o tempo de resposta de determina funcionalidade da aplicação, se faz necessário a utilização de ferramentas que permitam a simulação do uso e a devida documentação dos resultados obtidos.

Existem várias ferramentas *online* que podem ser utilizadas para realizar testes de aplicações *Web*. Dentre as possibilidades, foram selecionadas as seguintes ferramentas:

- *PageSpeed Tools*: pontua as falhas identificadas pelo teste e sugere as possíveis soluções para contorná-las. (<<https://developers.google.com/speed/pagespeed/insights/>>. Acessado em: 15 Jul 2017)
- *WebPagetest*: possibilita a escolha do país/cidade que serão feitos os testes, além do tipo de navegador, mostrando no final alguns dados relevantes, como o *First View* e *Full load*. Esses itens serão detalhados no Capítulo 4. (<<https://www.webpagetest.org/>>. Acessado em: 10 Ago 2017)
- *GTmetrix*: também possibilita a escolha do país/cidade que serão feitos os testes, além do tipo de navegador, e permite ainda a comparações entre dois sites. Algumas recomendações de alterações são apresentadas ao final do teste. (<<https://gtmetrix.com/>>. Acessado em: 10 Ago 2017)

- *Apache JMeter*: o *JMeter*<sup>1</sup> é um *software* de código aberto, desenvolvido na linguagem Java, utilizado para a realização de testes comportamentais, sendo possível mensurar o desempenho do conteúdo testado. O *JMeter* possibilita a realização de testes de desempenho relativo aos recursos estáticos, dinâmicos e aplicativos dinâmicos da *Web*. Ele possui a capacidade de simular alta carga em um servidor (grande quantidade de usuários e iterações), grupo de servidores, redes ou objeto para aferir sua robustez ou para avaliar o desempenho sob diferentes tipos de carga. (<<http://jmeter.apache.org/>>. Acessado em: 25 Jul 2017)

Pôde-se perceber a importância da realização dos testes de *software* para avaliação da qualidade e eficiência de aplicações. Uma vez se tendo os parâmetros de testes muito bem definidos, a confiabilidade dos mesmos tende a aumentar, elevando-se a classificação positiva das aplicações pelos usuários. Como visto, os testes algumas vezes expandem o uso das funcionalidades das aplicações ao extremo, como é o caso dos testes de carga, mencionados anteriormente. Esses experimentos podem vir a causar um notável consumo de Internet, no caso da aplicação testada ser *online*, por exemplo.

Sabe-se que no Brasil o tema limitação da Internet têm chamado atenção há algum tempo. Essa prática pode refletir negativamente na realização de testes e uso das aplicações *Web*. Em relação ao contexto mencionado, no cenário político se discute um Projeto de Lei (PL) que trata justamente essa temática, que será abordado na Seção seguinte.

## 2.2 Projeto de Lei (PL) 174/2016 | PL 7182/2017

Como apresentado em Brasil (2016), em abril de 2016 o senador Ricardo Ferraço propôs um Projeto de Lei, intitulado PL 174/2016. Trata-se de uma proposta de emenda ao Marco Civil da Internet, lei promulgada em 2014, que é considerada uma espécie de Constituição da Internet Brasileira. Na proposta, o senador solicita a inclusão, no artigo 7º, que trata do direito dos internautas, do seguinte trecho: “vedar a implementação de franquia limitada de consumo nos planos de Internet banda larga fixa”.

O Projeto de Lei seguiu todo o trâmite previsto no Senado Federal, incluindo: encaminhamentos às comissões especiais, designação do relator do projeto, consulta popular sobre o projeto, criação e aprovação do relatório, dentre outros passos. O Senado aprovou o projeto, após várias movimentações, no dia 23 de março de 2017, conforme mostrado em Brasil (2016). Ele foi em seguida encaminhando para a Câmara dos Deputados.

Já na Câmara, o PL 174/2016 sofreu modificação de nomenclatura, passando para PL 7182/2017. Conforme mostrado em Brasil (2017), o PL passou por várias tramitações,

---

<sup>1</sup> O *Apache JMeter* está disponível para *download* no seguinte endereço: (<[http://jmeter.apache.org/download\\_jmeter.cgi](http://jmeter.apache.org/download_jmeter.cgi)>). Acessado em: 25 Jul 2017



sendo que seu *status* mais atual é: “Aguardando Parecer do Relator na Comissão de Ciência e Tecnologia, Comunicação e Informática (CCTCI)”. Foi observado que o projeto está caminhando pela Câmara, de acordo com o andamento da casa.

Caso seja aprovado pelo plenário da câmara, o PL beneficiará diretamente toda a população, incluindo: usuários finais, empresas que utilizam aplicações *Web*, instituições de ensino, empresas que trafegam grande volume de dados pela *Web*, enfim, todos que direta ou indiretamente consomem e pagam pela utilização da Internet banda larga fixa no Brasil.

Em contrapartida, se têm o lado das empresas fornecedoras da banda larga, que não concordam com o PL, uma vez que a limitação de consumo as beneficiaria. Muitas dessas empresas adotam a prática do consumo limitado, já que a legislação não as proíbe. Há ainda algumas empresas que fornecem os 2 tipos de serviço, com e sem limitação, sendo que a segunda opção é, na maioria das vezes, mais cara, porém mais "vantajosa" para o consumidor.

Portanto, há uma disparidade de ideias acerca do Projeto de Lei 7182/2017. Mas pode-se dizer que a disputa está prestes a acabar, conforme demonstrado nas movimentações do projeto disponível em [Brasil \(2017\)](#). Vê-se que alguns políticos se interessam pela causa do senador Ricardo Ferraço e estão dispostos a fazer o projeto se encaminhar para o fim, a favor da população brasileira.

Caso o PL seja aprovado e sancionado pelo presidente da república sem vetos, a população seria altamente beneficiada, possibilitando um tráfego de dados maior, ampliando ainda mais o universo das aplicações *Web* pelo país afora. Em caso contrário, as empresas estariam livres para a limitação e posterior cobrança pelo excedente navegado, ocasionando aumento dos valores praticados pelo mercado, indo contra a maioria dos usuários da Internet brasileira.

Em suma, a limitação da Internet banda larga no Brasil implicaria diretamente nas aplicações *Web*, uma vez que estabelecendo um limite de navegação, se notaria um cenário oposto ao constante crescimento de aplicações, que necessitam cada vez mais de uma banda larga de qualidade e sem limites de tráfego. Como visto, a performance *Web* e a limitação da Internet banda larga fixa são assuntos amplamente discutidos na atualidade e merecem uma atenção especial sobre suas vertentes.

## 2.3 Considerações finais

Este capítulo englobou o embasamento teórico do trabalho, tratando de questões sobre testes de *software* e ferramentas úteis para a análise e verificação da performance *Web*. Foram incluídas algumas definições de termos úteis para o entendimento do trabalho,

além de alguns casos específicos sobre performance de aplicações de grandes empresas mundiais. O Projeto de Lei que tramita na Câmara dos Deputados sobre a proibição do limite da Internet no Brasil foi apresentado. Foram analisados os cenários positivo e negativo sobre a aprovação do [PL](#). O próximo capítulo descreverá a primeira bateria de testes realizada, utilizando-se o *Apache JMeter* para verificar duas páginas *Web*.

## 3 Testes com o *Apache JMeter*

Este capítulo apresenta o primeiro experimento, realizado com aplicações *Web* por meio da ferramenta *Apache JMeter*. O objetivo do experimento é verificar se existe alguma discrepância significativa entre as páginas do [ICEA](#) e [ICSA](#), e as possíveis causas e efeitos dessa diferença, caso exista.

Inicialmente, a configuração do *JMeter* é apresentada. Após isso, têm-se a descrição do modo de se executar os testes, seguido das configurações do *hardware* utilizado nos testes, o detalhamento dos testes realizados propriamente dito e a análise dos resultados observados. Por fim, são abordadas as considerações finais sobre o experimento realizado.

### 3.1 Configurando o *Apache JMeter*

As configurações utilizadas no experimento foram baseadas em [Integração \(2013\)](#), adaptando-se ao contexto de uso do trabalho. As etapas necessárias e os itens utilizados na definição do plano na ferramenta conforme o teste definido são descritas a seguir.

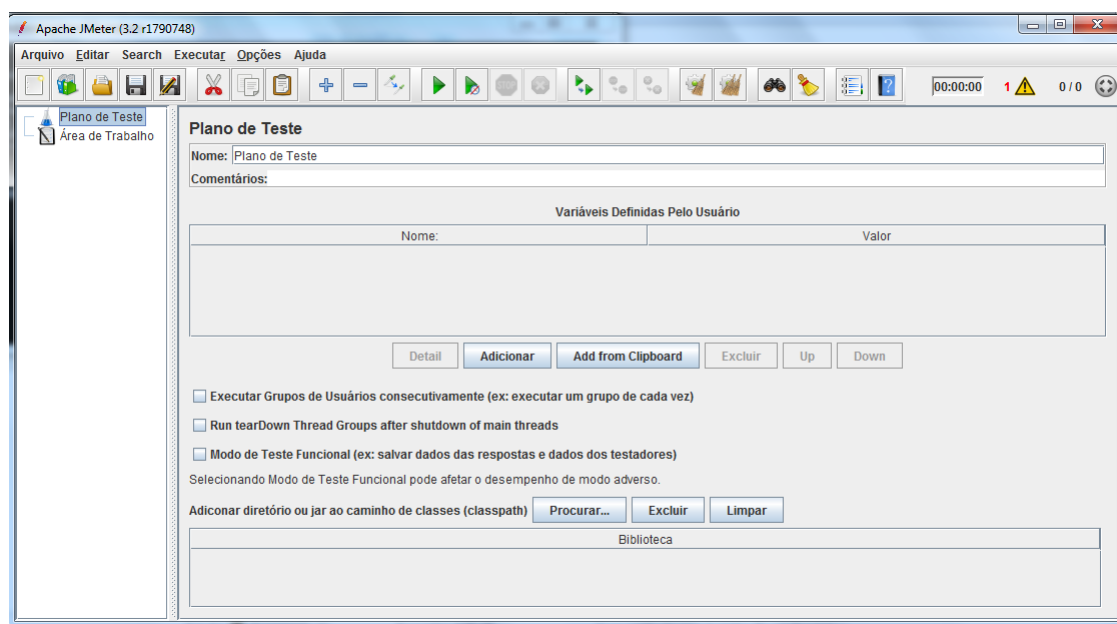
#### 3.1.1 Criando um Plano de Teste (*Test Plan*)

O Plano de Teste é o elemento básico para a criação do *script* (arquivo com extensão `.jmx`), e descreve os passos da ferramenta na execução dos testes. Ele é um projeto da ferramenta, no qual são adicionados todos os elementos úteis aos testes que serão executados. Os principais componentes do *Test Plan* que foram utilizados no trabalho são:

- Grupo de Usuários (*Thread Groups*): representa um grupo de usuário que executa as solicitações que serão configuradas;
- Ouvinte (*Listener*): responsáveis por exibir os resultados gerados pelo plano de teste e apresentá-los em determinado formato;
- Testador (*Sampler*): tipo da solicitação, podendo ser HTTP, FTP, SOAP, Java, dentre outros.

Para criar o Plano de Teste, basta seguir o caminho “*File->New*” ou “Arquivo->Novo” e configurar com o nome desejado. A Figura 1 mostra a tela de criação do Plano de Teste.

Figura 1 – Tela de criação do Plano de Teste



Fonte: *Apache JMeter*

### 3.1.2 Criando um Grupo de Usuários (*Thread Group*)

No Grupo de Usuários são agregados todos os componentes do teste. Para criá-lo basta seguir o caminho “*Edit->Add->Threads(Users)->Thread Groups*” ou “*Editar->Adicionar->Threads(Users)->Grupo de Usuários*”. A Figura 2 representa o caminho para a criação do Grupo de Usuários.

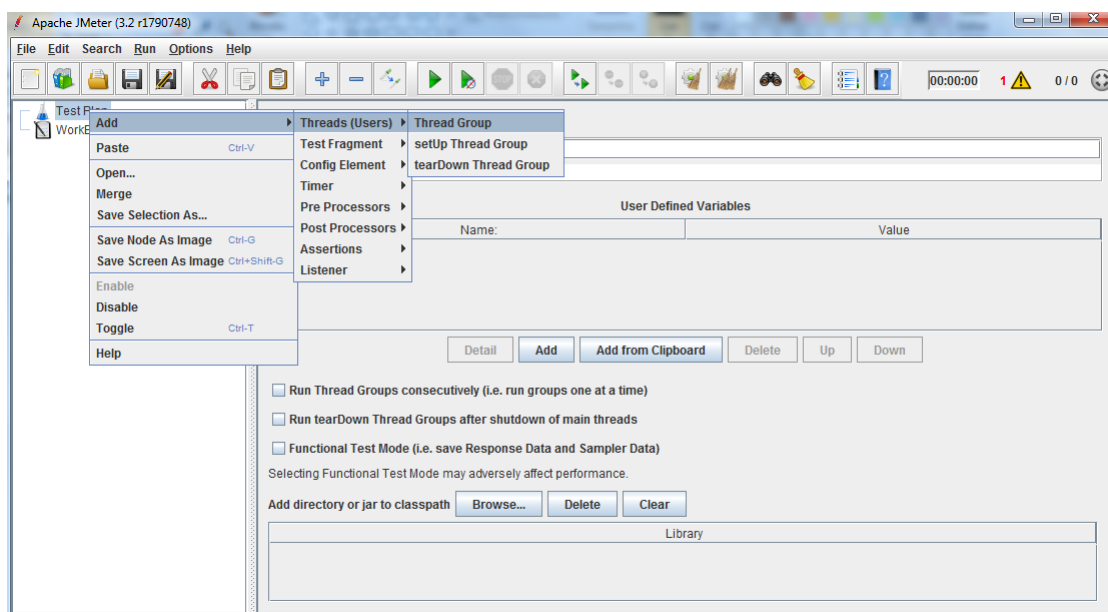
### 3.1.3 Configurando a quantidade de Usuários Virtuais e Contador de Iterações (*Number of Threads e Loop Count*)

Após a criação do Grupo de Usuários, é possível a configuração de alguns parâmetros no campo “*Thread Properties*” ou “*Propriedades do Usuário Virtual*”, como a quantidade de usuários virtuais (*Number of Threads*) e iterações (*Loop Count*), sendo definidos de acordo com a necessidade dos testes. Os valores iniciais são configurados como 1. A Figura 3 mostra as opções de edição do Grupo, como: nome, número de usuários virtuais e contador de iterações.

### 3.1.4 Adicionando o tipo de Requisição ao Grupo de Usuários

O *JMeter* possibilita a inserção de vários tipos de Testador (*Sampler*), como: *HTTP Request*, *FTP Request*, *SOAP/XML-RPC Request* e *Java Request*. Como este trabalho engloba aplicações *Web*, adotou-se o tipo *HTTP Request*, que é responsável por gerenciar

Figura 2 – Caminho para a criação do Grupo de Usuários



Fonte: *Apache JMeter*

as requisições HTTP trafegadas para uma página *Web*. O caminho para a inclusão é “*Edit->Add->Sampler->HTTP Request*” ou “*Editar->Adicionar->Testador->Requisição HTTP*”.

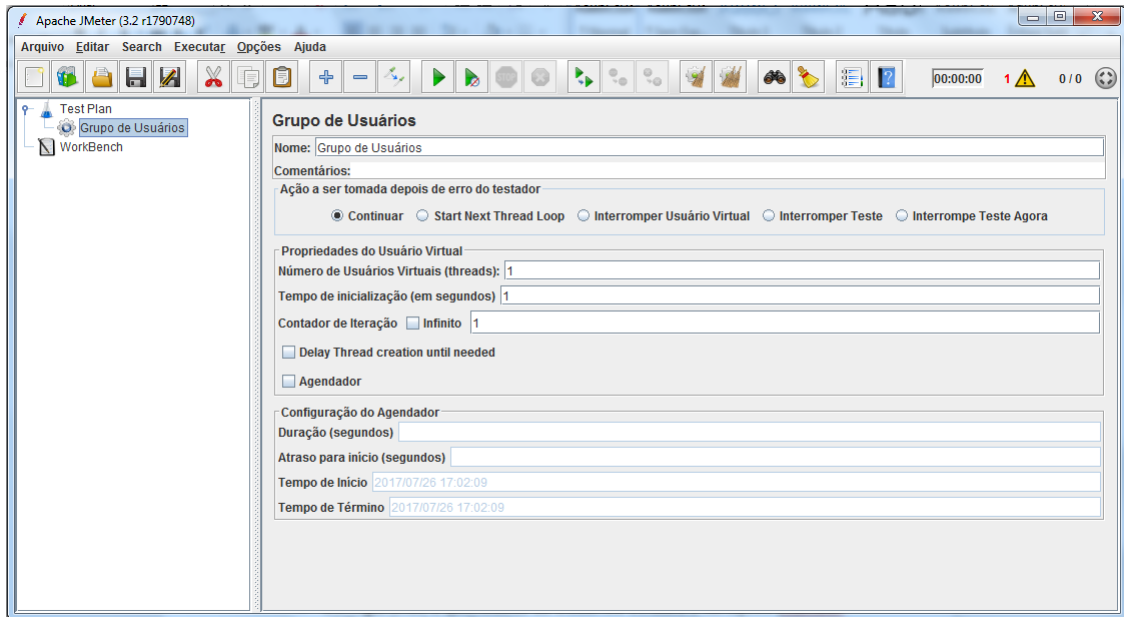
Após a inclusão, é possível a configuração de algumas variáveis, como: protocolo utilizado, nome do servidor, número da porta, método, dentre outras. A variável que foi modificada para cada teste é o Nome do Servidor ou IP (*Server Name or IP*), que é o endereço da aplicação a ser testada. A Figura 4 representa o caminho para inclusão da Requisição HTTP. Já a Figura 5 ilustra os campos possíveis de edição, como: protocolo, nome do servidor ou IP e número da porta.

### 3.1.5 Adicionando Relatório de Sumário (*Summary Report*)

O Relatório de Resumo contém as principais informações úteis para a análise dos testes. O caminho para adicionar esse recurso é “*Edit->Add->Listerner->Summary Report*” ou “*Editar->Adicionar->Ouvinte->Relatório de Sumário*”. As Figuras 6 e 7 mostram, respectivamente, a inclusão do Relatório de Sumário e os campos que podem ser editados, como: nome do relatório e comentários.

Ao final de todo esse processo, pode-se salvar o Plano de Teste, para usos futuros. Para isso, basta seguir o caminho “*File->Save Test Plan as*” ou “*Arquivo->Salvar Plano de Teste como*” e escolher o nome e local no qual se deseja salvar o projeto.

Figura 3 – Opções de edição do Grupo de Usuários



Fonte: *Apache JMeter*

## 3.2 Executando os testes

É possível a execução do teste diretamente pela interface gráfica do *JMeter*. Entretanto, o aplicativo quando é iniciado alerta que para se use o modo com interface apenas para criação e depuração de teste. É aconselhado, conforme mostra a Figura 8 que, para testes de carga, se use o modo terminal, adaptando a ferramenta aos requisitos de teste necessários.

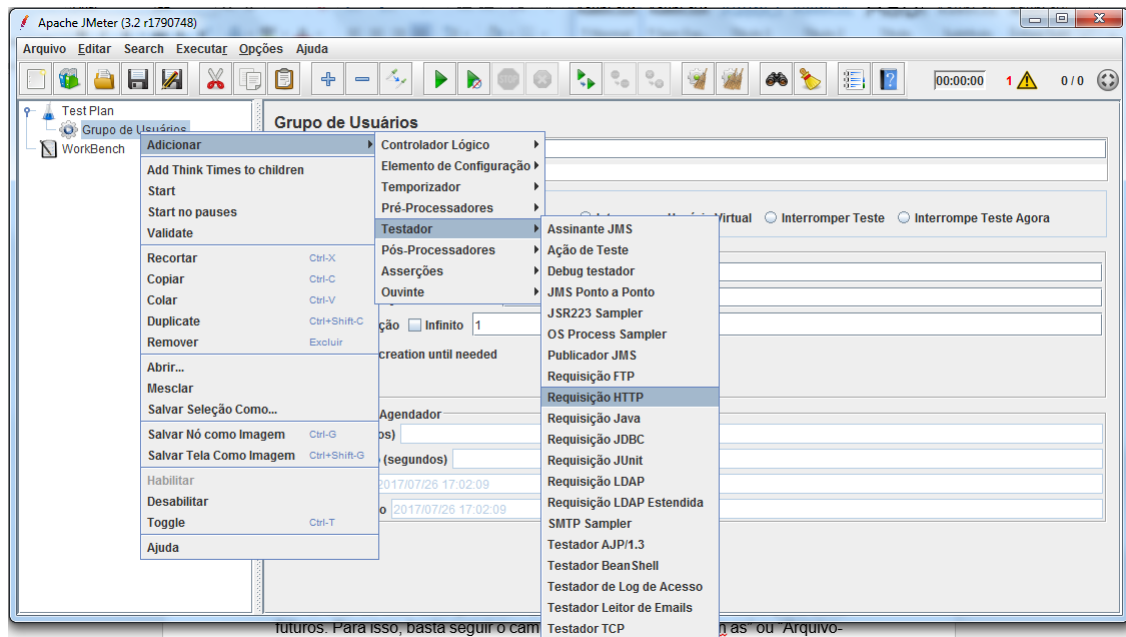
Segue abaixo um exemplo prático de execução do *script* de teste:

```
.\jmeter -n -t PlanoDeTeste.jmx -l ArquivoDeResultados -e -o LocalOndeSalvar
```

Legenda do *script*:

- `.\jmeter`: comando para executar a ferramenta *JMeter*.
- `-n`: especifica que o *JMeter* deve ser executado no modo de terminal (sem interface gráfica).
- `-t PlanoDeTeste.jmx`: comando para indicar o nome do arquivo com extensão JMX que contém o Plano de teste.
- `-l ArquivoDeResultados`: nome do arquivo para registrar os resultados do teste.

Figura 4 – Caminho para inclusão da Requisição HTTP



Fonte: *Apache JMeter*

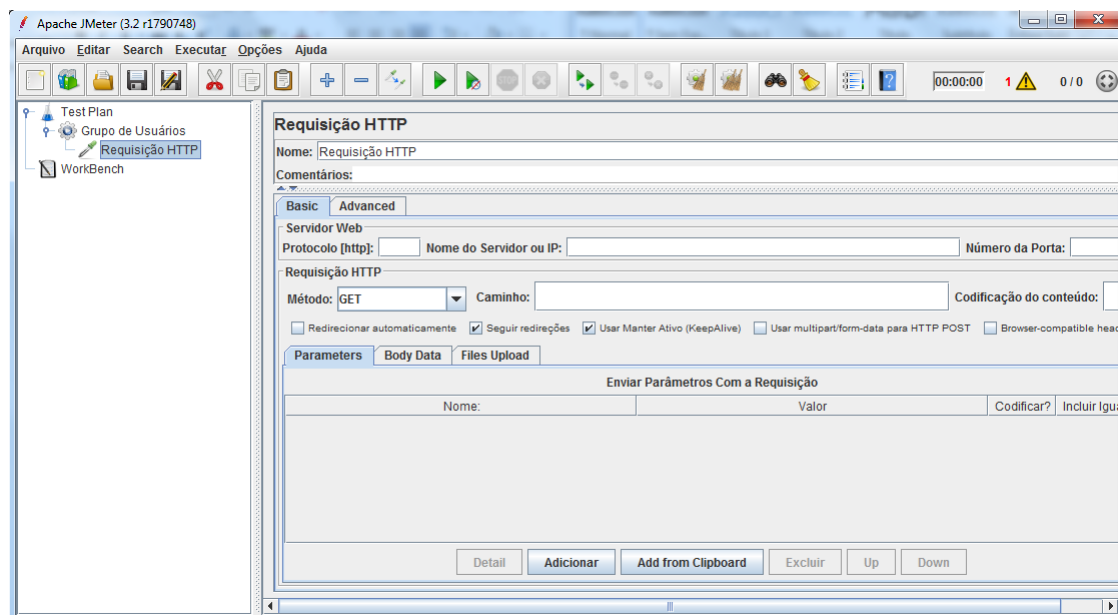
- -e: comando para gerar o relatório somente após o teste de carga.
- -o LocalOndeSalvar: comando que define a pasta de saída após o teste de carga, que não deve existir ou estar vazia.

### 3.3 Testes realizados

As coletas de dados foram realizadas no campus do [ICEA](#), utilizando um computador pessoal (*notebook*) com as seguintes especificações:

- Marca: *Acer*;
- Modelo: *Aspire 4736Z*;
- Processador: *Pentium(R) Dual-Core CPU T4400 @ 2.20GHz*;
- Memória RAM: *3 GB*;
- Disco Rígido: *250 GB*;
- Sistema operacional: *Windows 7 Ultimate Service Pack1 64 bits*;
- Versão do *Apache JMeter*: *3.2*.

Figura 5 – Campos possíveis de edição da Requisição HTTP



Fonte: *Apache JMeter*

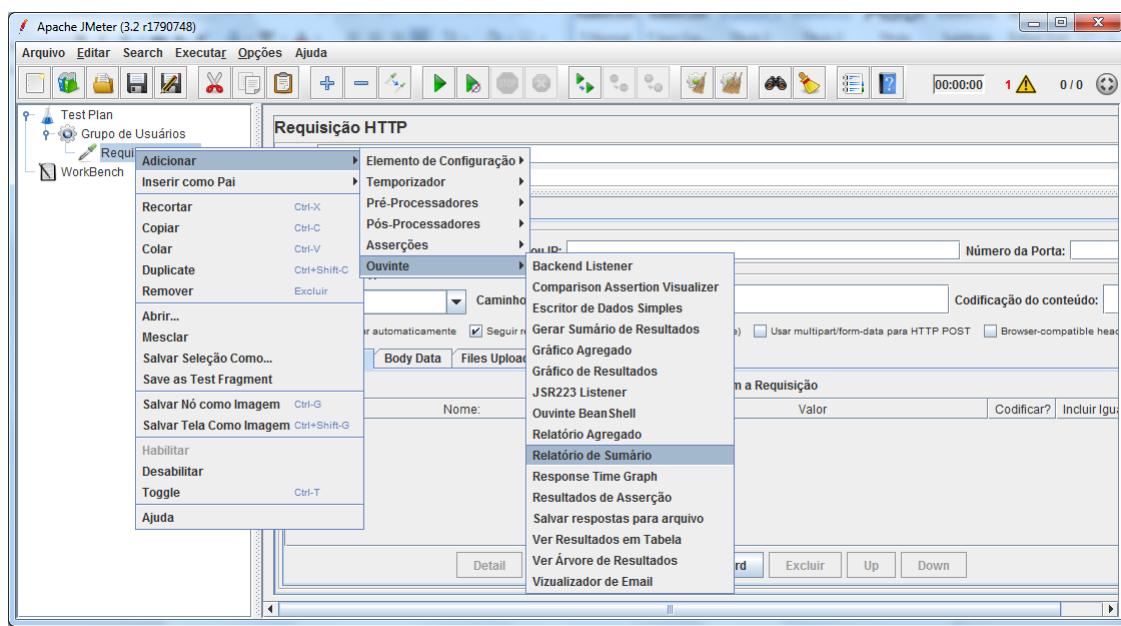
Os testes realizados abordam direta ou indiretamente os conceitos de performance, carga e *stress* citados anteriormente. Com o uso do *JMeter*, é possível englobar ambos os testes, conforme será observado posteriormente. Considerando o contexto do presente trabalho, foram realizados testes em aplicações utilizando o *JMeter*. Os testes foram estruturados da seguinte maneira:

- Duas aplicações, sendo elas: as páginas do Instituto de Ciências Exatas e Aplicadas (ICEA), disponível em (<http://www.icea.ufop.br>). Acessado em: 20 Jul 2017) e do Instituto de Ciências Sociais Aplicadas (ICSA), disponível em (<http://www.icsa.ufop.br>). Acessado em: 20 Jul 2017). Ambas foram desenvolvidas utilizando o *Joomla*<sup>1</sup>, que é uma plataforma de aplicações *Web* que possibilita a criação de *sites* na Internet e o gerenciamento dos conteúdos. As páginas estão hospedadas no mesmo servidor (servidor da UFOP, localizado em Ouro Preto);
- Números de usuários virtuais pré-definidos: 100 e 1000, visando oscilar a carga para verificar o comportamento das aplicações em questão;
- Quantidade de iterações pré-definida: 30 e 300, com o propósito de observar o comportamento dos *sites* sob altas e insistentes demandas.

<sup>1</sup> Leia mais na documentação do *Joomla*, disponível em [https://docs.joomla.org/Portal:Learn\\_More/pt-br](https://docs.joomla.org/Portal:Learn_More/pt-br). Acessado em: 20 Jul 2017



Figura 6 – Inclusão do Relatório de Sumário



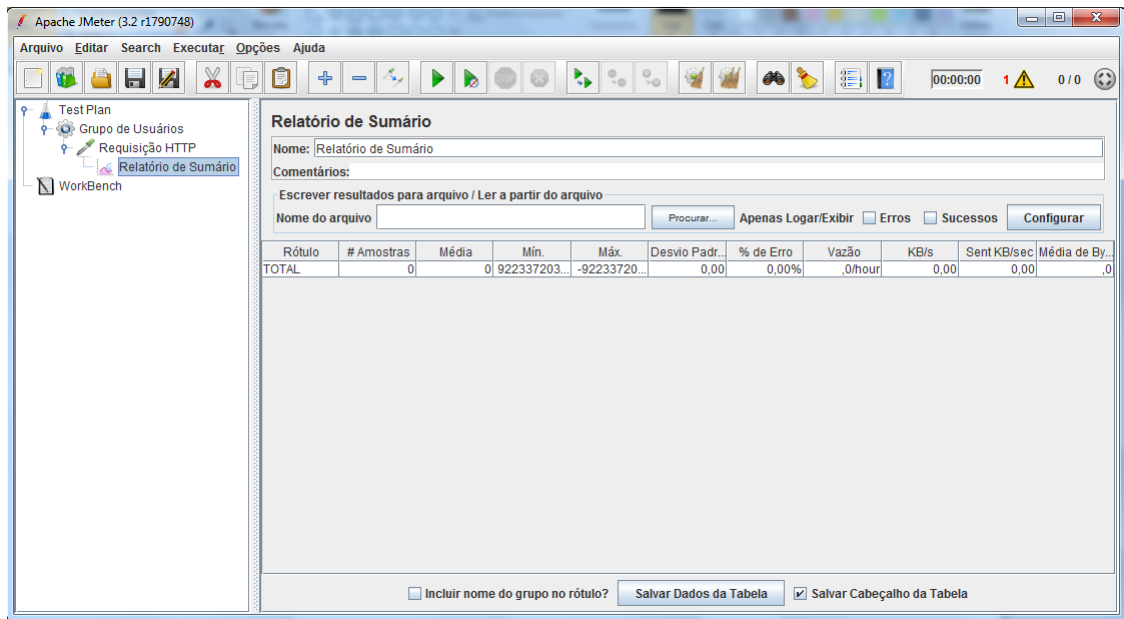
Fonte: *Apache JMeter*

Considerando o planejamento fatorial do experimento, são 8 observações (duas aplicações, duas quantidades de usuários e duas quantidades de iterações,  $2^3$ ), e para cada observação página-usuários-iterações foram realizadas 5 replicações. Assim, o experimento possui 40 observações totais. A ordem das observações foi aleatorizada. Foram realizadas 8 observações por dia, obedecendo à aleatoriedade, ambas realizadas no mesmo local, mesmo computador e horários de coleta próximos. Para mais informações sobre este tipo de experimento, sugere-se a leitura de Planejamento Fatorial  $2^n$  em [Montgomery e Runger \(2011\)](#).

Os resultados dos testes realizados no *JMeter* trazem vários dados sobre a aplicação testada. Foram extraídas, dentre as várias possibilidades, as seguintes variáveis de resposta, que foram consideradas relevantes para o estudo:

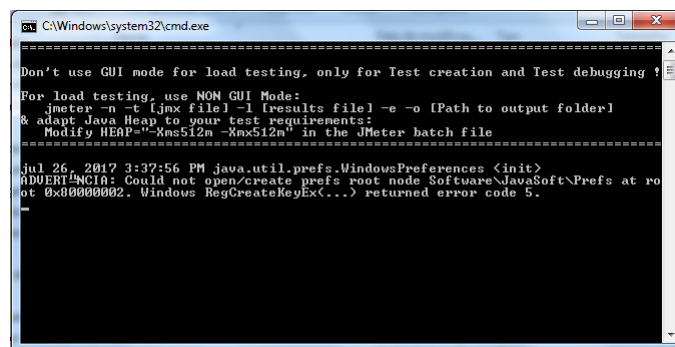
- tempo mínimo das respostas às requisições HTTP, em milissegundos;
- tempo máximo das respostas às requisições HTTP, em milissegundos;
- tempo médio das respostas às requisições HTTP, em milissegundos;
- tempo total de cada observação, em minutos;
- quantidade total de requisições de cada observação, que basicamente é a relação Usuários x Iterações;
- a vazão relativa a cada observação, que refere-se a quantidade de requisições por segundo;

Figura 7 – Campos que podem ser editados do Relatório de Sumário



Fonte: *Apache JMeter*

Figura 8 – Alerta do JMeter



Fonte: *Apache JMeter*

- g) quantidade de erros encontrados durante a observação;
- h) percentual de erros encontrados durante a observação.

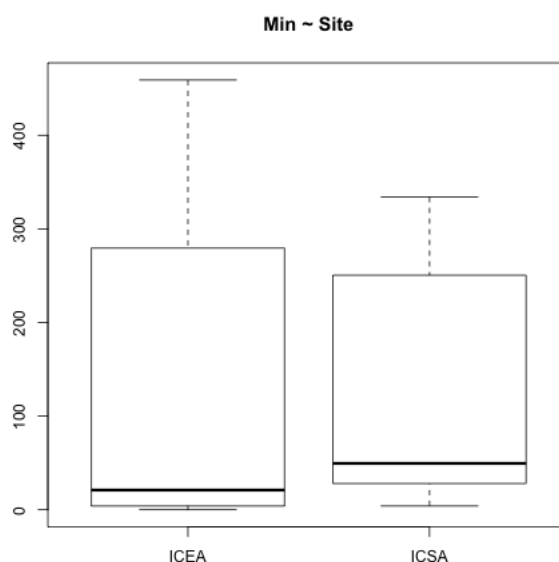
### 3.4 Resultados dos testes com o *Apache JMeter*

Conforme citado anteriormente, os dados coletados dos testes com o *JMeter* são apresentados na Tabela 4, disponível no Apêndice A. A partir deles foram gerados *Boxplots*

<sup>2</sup> de modo a possibilitar uma análise preliminar das variáveis de resposta mais relevantes, sendo elas: tempo mínimo, tempo máximo, tempo médio, quantidade de erros e vazão relativa.

A Figura 9 apresenta o *Boxplot* do tempo mínimo de respostas à requisição HTTP. Conforme pode-se notar, ambas as páginas (ICEA e ICSA) possuem medianas entre 20 e 50 milissegundos, sendo um bom tempo de resposta mínimo. Embora a página do ICEA apresente uma mediana menor do que o ICSA, é possível observar que o maior valor (amplitude do *Boxplot*) é justamente o do ICEA (acima de 400 ms). Isso pode sugerir um tempo mínimo de resposta do ICEA maior que do ICSA.

Figura 9 – *Boxplot* com o tempo mínimo para responder à requisição HTTP

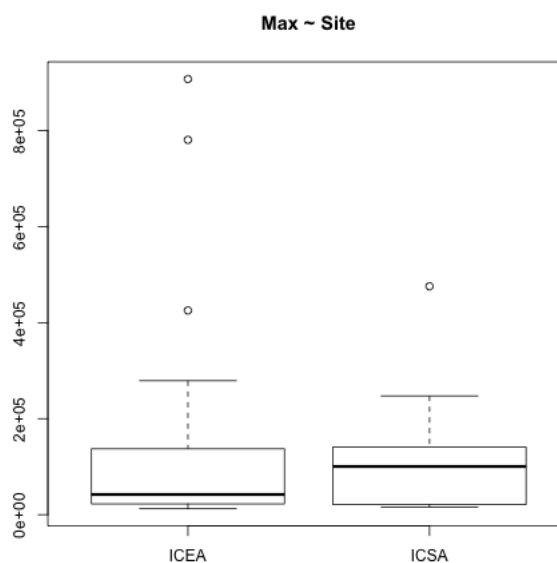


Fonte: elaborado pelo autor

Já a Figura 10 mostra o *Boxplot* do tempo máximo de respostas à requisição HTTP. Conforme nota-se, ambas as páginas (ICEA e ICSA) possuem medianas próximas, embora novamente a página do ICEA apresente uma mediana menor do que o ICSA. Observa-se que, por mais que a mediana da página do ICSA seja maior, o que poderia sugerir um resultado inferior ao do ICEA, mais uma vez o tempo máximo do ICEA é superior. Isso pode sugerir um tempo máximo de resposta do ICEA maior que do ICSA.

Além disso, alguns *outliers* são reportados na figura. Essas marcações indicam valores atípicos (fora do padrão) notados na amostra. A página do ICEA trouxe mais valores atípicos do que a do ICSA, o que sugere uma maior instabilidade da página do ICEA em relação ao ICSA, embora sejam poucos valores discrepantes.

<sup>2</sup> Para mais informações sobre *Boxplot*, sugere-se a leitura do Apêndice B

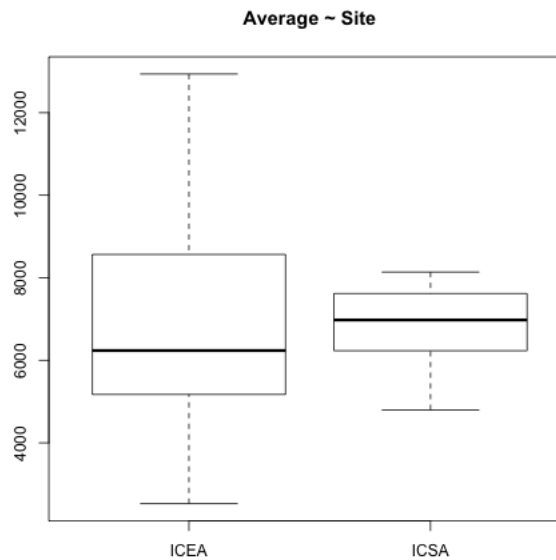
Figura 10 – *Boxplot* com o tempo máximo para responder à requisição HTTP

Fonte: elaborado pelo autor

A próxima variável é o tempo médio de resposta às requisições HTTP, o qual é apresentado pela Figura 11. Nota-se a mesma relação citada anteriormente, em que a página do ICEA possui uma melhor mediana, porém a amplitude do ICSA é menor. Começa-se a observar um certo padrão de comportamento entre os resultados apresentados até o momento, sendo que uma certa vantagem de uma página pode ser compensada por alguma outra característica em relação à outra página.

Em relação à variável de erros encontrados durante a execução dos testes, a página do ICEA demonstrou uma ligeira vantagem se comparado à do ICSA. A mediana de erros do ICEA está um pouco abaixo da mediana do ICSA, conforme pode ser observado na Figura 12. Pode-se perceber ainda que o valor máximo da página do ICSA é bastante superior ao valor do ICEA, apesar de serem percebidos alguns valores atípicos no *Boxplot* do ICEA.

Levando-se em consideração apenas a informação dos erros encontrados, pode-se sugerir que a página do ICSA é um pouco mais instável em relação ao ICEA. No âmbito mais amplo, foram detectadas quantidades significativas de erros, conforme mostrado na Tabela 4. Percebe-se que algumas combinações, como por exemplo o site do ICSA com 1000 usuários e 300 iterações (observação 27 da Tabela 4), obtiveram altas taxas de erros (93,19%). Isso pode sugerir alguma falha em questões de alta escalabilidade de uso da página, como no exemplo citado (300.000 requisições).

Figura 11 – *Boxplot* com o tempo médio para responder à requisição HTTP

Fonte: elaborado pelo autor

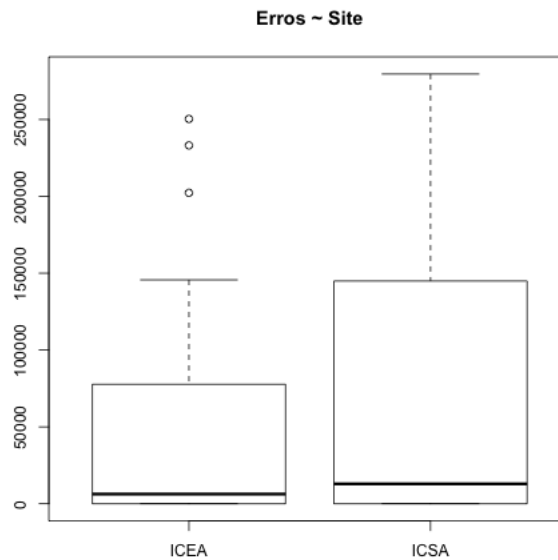
A taxa de erros observada poderia ser considerada catastrófica, dependendo principalmente da categoria da aplicação. Por exemplo, se ela fosse crítica, sendo que os erros são evitados ao máximo e praticamente inadmissíveis, como um *software* financeiro *online* (*Internet Banking*). No caso citado, erros podem vir a gerar prejuízos, causando um declínio acentuado na confiabilidade e credibilidade da empresa detentora da aplicação.

Outra informação considerada relevante para o presente estudo é a vazão de ambas as páginas utilizadas para os testes. *Throughput* ou vazão, conforme Ribeiro (2013), é a quantidade de vezes que se consegue receber uma resposta completa de uma página por segundo. Ela pode influenciar diretamente no desempenho da página testada. Caso a vazão seja baixa, quer dizer que a aplicação demonstra certa instabilidade quando submetida à cargas maiores de operações, e vice-versa. A vazão é calculada conforme a Equação 3.1.

$$Vazão = \frac{Total\ de\ requisições}{Tempo\ em\ segundos} \quad (3.1)$$

Conforme pode-se observar na Figura 13, as vazões de ambas as páginas possuem valores de medianas bem próximos (algo em torno de 50). Porém, a página do ICEA apresenta uma pequena vantagem (o valor máximo é superior ao do ICSA).

Semelhante ao critério de erros, considerando-se apenas os valores para vazão, pode-se dizer que a página do ICEA demonstra certa dianteira em relação ao ICSA, apesar de pequena. O ICEA apresenta valores máximos e mínimos para a vazão superiores aos

Figura 12 – *Boxplot* com os erros encontrados durante os testes

Fonte: elaborado pelo autor

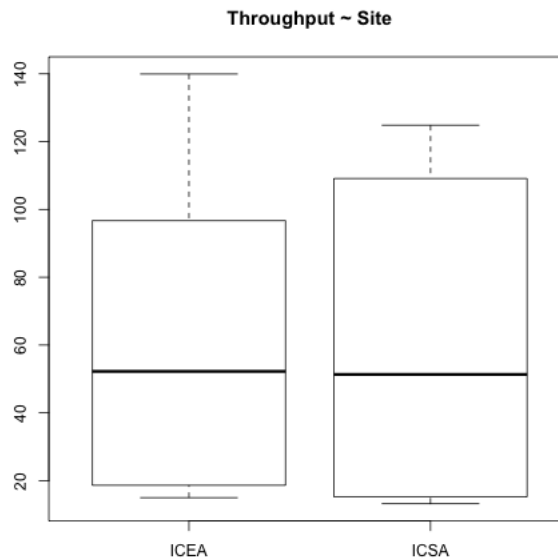
valores do [ICSA](#), comprovando a vantagem citada, visto que quanto maior a vazão, melhor.

Os resultados preliminares mencionados anteriormente, embora sejam discrepantes em alguns aspectos, sugerem a equivalência da performance das aplicações numa média geral. Para avaliar os resultados, um teste estatístico foi aplicado. As considerações acerca desse teste são descritas a seguir.

Considerando que o experimento possui mais de uma variável de resposta, utilizou-se a Análise de Variância Multivariada ([MANOVA](#)), a qual é uma maneira análoga multivariante do método Análise de Variância ([ANOVA](#)). A ANOVA é uma técnica estatística que possibilita a avaliação de afirmações sobre as médias de populações amostrais, verificando se existe uma diferença significativa entre os valores ([MONTGOMERY; RUNGER, 2012](#)).

Ambas as técnicas utilizam como parâmetro de comparação uma variável  $\alpha$ , que tem valor definido como 0,05 e é o nível mínimo de significância necessário. Como resultados finais, a [MANOVA](#) gera um valor- $P$  para cada conjunto de entradas da análise. Caso o valor- $P$  seja menor que  $\alpha$ , é dito que os resultados do parâmetro em questão sugerem diferença estatística significativa e caso seja maior, é dito o contrário (que não sugerem diferença significativa). A Tabela 1 exibe os resultados da análise [MANOVA](#) para as cinco variáveis de resposta: tempo mínimo, máximo, médio, vazão e erros, através do valor  $P$  obtido para cada parâmetro de entrada.

Como pode-se notar, para todas as variáveis avaliadas o valor  $P$  é superior a 0,05.

Figura 13 – *Boxplot* com a vazão das páginas utilizadas nos testes

Fonte: elaborado pelo autor

Tabela 1 – Valor-*P* dos parâmetros de entrada

Métrica	Valor- <i>P</i>
Tempo Mínimo	0,6654
Tempo Máximo	0,4296
Tempo Médio	0,9268
Vazão	0,8402
Erros	0,4916

Fonte: elaborado pelo autor

Considerando o ambiente experimental estabelecido, os resultados sugerem que não existe diferença estatística significativa entre as páginas averiguadas. Isso pode ser justificável por alguns motivos, dentre eles:

- As duas páginas são hospedadas no mesmo servidor (UFOP);
- As duas aplicações foram desenvolvidas usando a mesma plataforma, o *Joomla*;
- A estrutura e os recursos das duas páginas são basicamente os mesmos (menus, submenus, notícias, formulário de contato, logomarcas, barra de pesquisa, dentre outros).

## 3.5 Considerações finais

Esse capítulo descreveu os testes iniciais realizados com o *Apache JMeter* nas páginas do [ICEA](#) e [ICSA](#). Os resultados encontrados sugeriram que não existem diferenças estatísticas significativas entre as páginas. Porém, foi possível a identificação de alguns aspectos interessantes presentes na análise, como a taxa de erros e vazão das páginas, induzindo a conclusões sobre a eficiência das mesmas em situações extremas de uso. Além disso, obteve-se também as medianas das amostras para os tempos mínimo, médio e máximo de resposta, que não apresentaram oscilações representativas, embora tenha-se visto alguns valores atípicos. Já o próximo capítulo trata sobre testes realizados em uma página desenvolvida pelo autor, a fim de verificar alguns fatores específicos que podem influenciar no desempenho de uma aplicação.



## 4 Testes com uma página *Web*

Este capítulo descreve os testes realizados com uma página *Web* desenvolvida pelo autor. O objetivo do experimento é verificar se alguns fatores pré-estabelecidos causam ou não impactos diretos na performance da aplicação. Foram utilizadas as ferramentas *GTmetrix* e *WebPagetest*, citadas anteriormente, para coletar os parâmetros utilizados na elaboração dos resultados finais.

Inicialmente será apresentada a estrutura da página *Web* criada. Após isso, a definição do experimento é discutida, seguida da análise dos resultados. Por fim, as considerações finais sobre o experimento são apresentadas.

### 4.1 Página *Web* criada

A fim de averiguar quais fatores influenciam de fato na performance das aplicações *Web*, foi desenvolvida uma página principal utilizando **PHP** combinado com **HTML**, **CSS** e **JS**. Os conteúdos dessa página foram utilizados única e exclusivamente para os testes, como: imagens, textos, logomarca, menu, dentre outros. Esses itens são meramente ilustrativos e representativos, sendo retirados da página oficial do **ICEA**<sup>1</sup> e da **UFOP**<sup>2</sup>. As Figuras 14 e 15 ilustram a página criada.

Figura 14 – Vista parcial da página criada

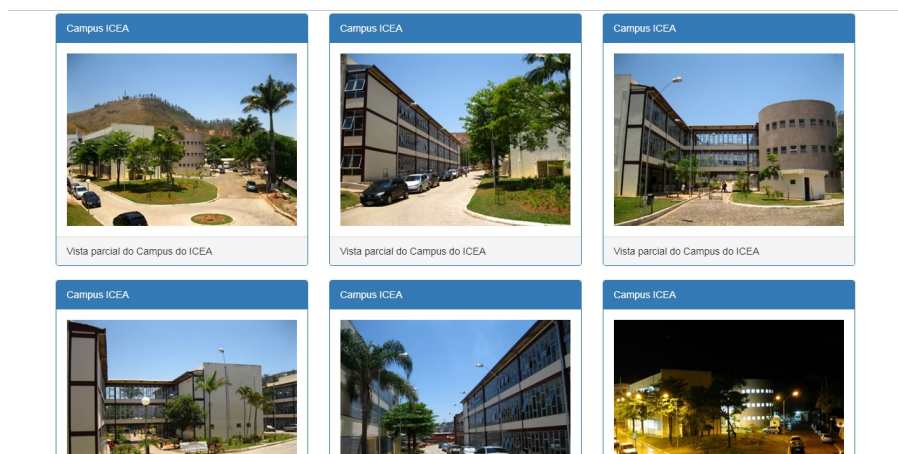


Fonte: elaborado pelo autor

<sup>1</sup> Página disponível em <<http://www.icea.ufop.br/site/>>. Acessado em: 05 Ago 2017

<sup>2</sup> Página disponível em <<http://www.ufop.br>>. Acessado em: 05 Ago 2017

Figura 15 – Vista parcial da página criada



Fonte: elaborado pelo autor

## 4.2 Configuração e execução do teste

A linguagem **PHP** permite a criação de parâmetros no próprio código fonte da página e posteriormente a invocação desses parâmetros via URL, por exemplo. Na página criada foram consideradas as configurações: conteúdo **CSS** no topo ou no final da página; conteúdo **CSS** minificado ou não minificado; conteúdo **JS** no topo ou no final da página; conteúdo **JS** minificado ou não minificado. A Tabela 2 apresenta as combinações criadas.

Para se realizar os testes, é necessário informar uma URL à ferramenta, com o endereço que a aplicação está hospedada. Como a página utilizada é armazenada localmente, a URL passada segue o padrão: "http://200.239.154.203/tcc/index.php?css=0&js=4", considerando o endereço IP do computador na rede, o caminho da página e os parâmetros **CSS** e **JS** definidos. O servidor no computador é o Apache versão 2.2.27. O ambiente computacional é o mesmo descrito na Seção 3.3.

Tabela 2 – Combinações de parâmetros criada

Item	Localização	Minificado	Parâmetro
<b>CSS</b>	Topo	Sim	0
<b>CSS</b>	Final	Sim	1
<b>CSS</b>	Topo	Não	2
<b>CSS</b>	Final	Não	3
<b>JS</b>	Topo	Sim	4
<b>JS</b>	Final	Sim	5
<b>JS</b>	Topo	Não	6
<b>JS</b>	Final	Não	7

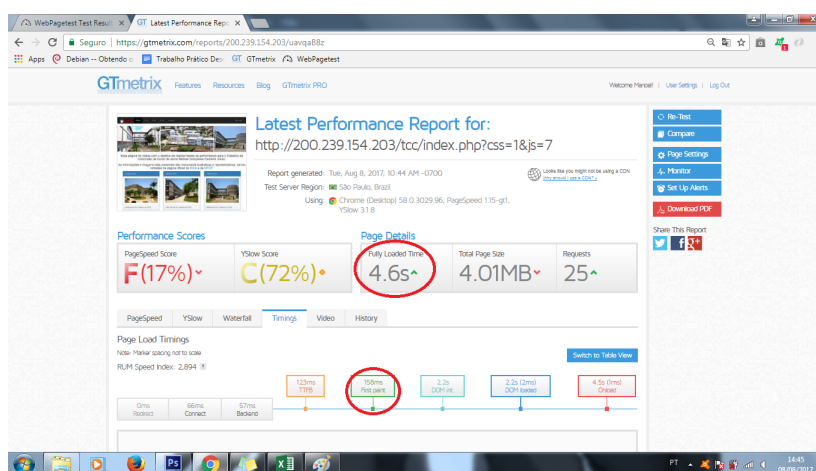
Fonte: elaborado pelo autor

Considerando o planejamento fatorial do experimento, são 32 observações (duas ferramentas, duas localizações do **CSS** - topo ou final, duas formas do **CSS** - minificado ou não, duas localizações do **JS** - topo ou final, duas formas do **JS** - minificado ou não,  $2^5$ ), e para cada observação da página foram realizadas 5 replicações. Assim, o experimento possui 160 observações totais. A ordem das observações foi aleatorizada, sendo elas realizadas no mesmo local, mesmo computador e horários próximos. Para mais informações sobre este tipo de experimento, sugere-se a leitura de Planejamento Fatorial  $2^n$  em [Montgomery e Runger \(2011\)](#).

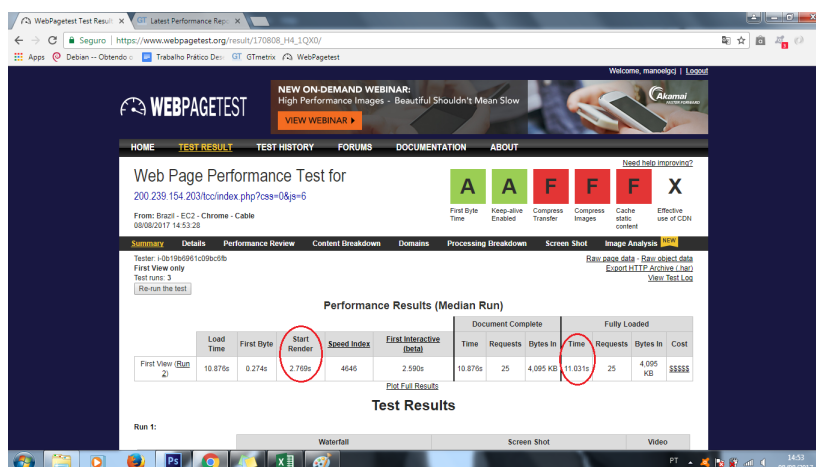
Tendo os parâmetros definidos, iniciou-se a execução dos testes. Para tal, como apresentado anteriormente, foram utilizadas as ferramentas *GTmetrix* e *WebPagetest*. Os dados coletados são apresentados nas Tabelas 5 e 6, as quais estão disponíveis no Apêndice A.

Foram utilizadas duas variáveis de resposta das ferramentas. O *First view*, o qual corresponde ao tempo gasto para se obter a primeira visualização parcial da página, e o *Full load*, que representa o tempo total gasto para o carregamento de toda a página. No *GTmetrix* esses dados são, respectivamente, o *First paint* e o *Fully Loaded Time*. Esses campos estão destacadas em vermelho na Figura 16. Já no *WebPagetest*, o *First view* se refere ao *Start Render* e o *Full load* é representado pelo *Time* decrescido de 2 segundos, uma vez que a documentação do *WebPagetest* cita que o tempo considerado por essa variável é o total gasto para executar o testes, acrescido de 2 segundos de inatividade. A documentação do *WebPagetest* está disponível em <https://sites.google.com/a/webpagetest.org/docs/>. Acessado em: 20 Ago 2017. A Figura 17 ilustra os dados que foram coletados durante os testes (parte demarcada em vermelho).

Figura 16 – Apresentação dos resultados do *GTmetrix*



Fonte: retirado do *GTmetrix* e adaptado pelo autor

Figura 17 – Apresentação dos resultados do *WebPagetest*

Fonte: retirado do *WebPagetest* e adaptado pelo autor

### 4.3 Resultados do experimento

A partir dos dados coletados pelas ferramentas de teste, foram elaborados *Boxplots* para a análise preliminar dos resultados. A Figura 18 ilustra os *Boxplots* dos valores médios do *First View*. No eixo horizontal estão os parâmetros dos testes, e no vertical constam os tempos médios, em segundos, de execução dos testes com cada parâmetro. Para facilitar a visualização foram inseridas letras (de A a P) em cada caixa mostrada pelo *Boxplot*.

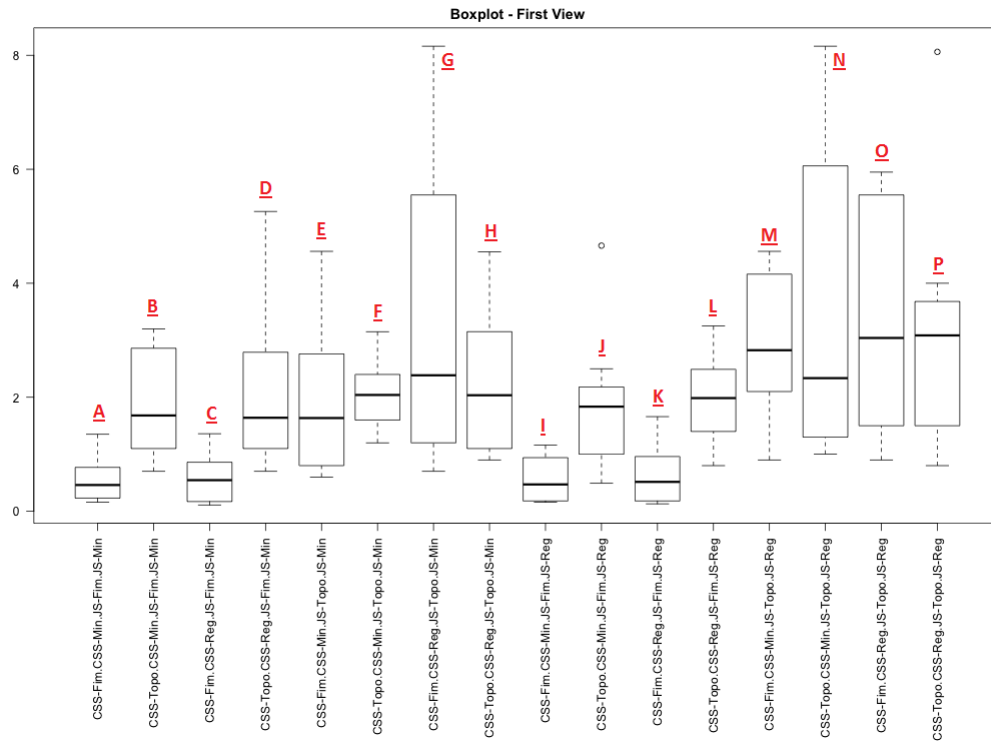
Nota-se que as caixas A, C, I e K apresentam as menores médias de toda a figura, ambas com tempos abaixo de 1 segundo. Se pode observar que, justamente nesses resultados, os parâmetros envolvidos são **CSS** e **JS** no final da página, variando entre minificados ou não. Assim, numa primeira análise, percebe-se que a localização do **CSS** e **JS** no final da página tendem a apresentar melhores resultados do que o contrário.

As demais caixas apresentam médias próximas umas das outras, com valores próximos de 2 até pouco mais de 3 segundos, o que sugere pouca variação dos resultados encontrados. Nota-se ainda que quando o **JS** é inserido no topo da página, como nas caixas G, M, O e P, a média percebida são as mais altas da amostra, o que sugere o fato citado anteriormente, que o **JS** inserido no final da página apresenta resultados inferiores do que no topo.

As caixas J e P mostram 2 valores atípicos, apresentados como formas circulares acima do valor máximo da representação gráfica. Não foi notada nenhuma relação desses valores com os parâmetros de testes. Essa discrepância pode ter sido proveniente de alguma interferência externa, como algum atraso para o servidor da ferramenta de teste responder, ou ainda interno, como a arquitetura do computador usado para os testes e a conexão de

Internet utilizada.

Figura 18 – *Boxplots* com o *First View* e os parâmetros criados

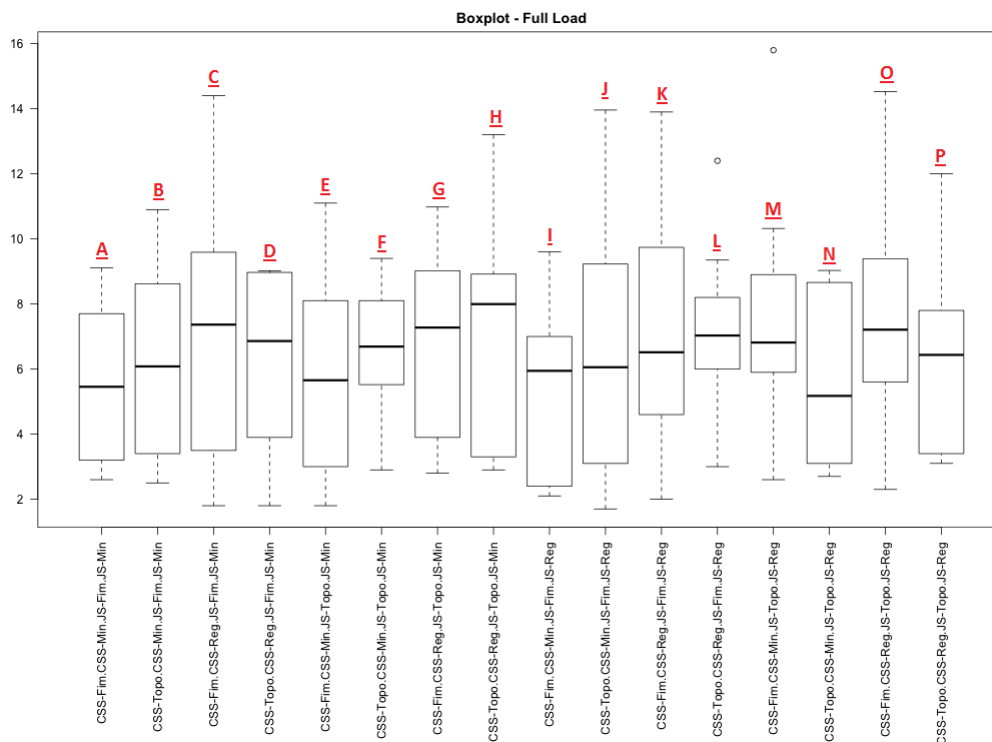


Fonte: elaborado pelo autor

Já a Figura 19 apresenta os *Boxplots* com os valores médios do *Full Load* coletados no experimento. Analisando-se inicialmente as caixas de A a P se pode mencionar que a variável de resposta apresenta leve variação nos tempos médios de execução dos testes. Os valores oscilam nas proximidades de 5 e 8 segundos, o que não é tão discrepante na exibição e carregamento da página Web, considerando que ela possui várias imagens e textos. Em algumas médias são notados valores considerados altos, em torno de 14 segundos ou um pouco mais. Porém, na média os testes mostraram tempos próximos, como apresentado anteriormente.

Novamente percebem-se alguns pontos atípicos, dessa vez nas caixas L e M. Tais pontos não interferem diretamente no resultado final da pesquisa, pelo fato de serem poucos pontos isolados num total mais amplo. As oscilações presentes nos resultados podem ser causadas por fatores diferentes dos critérios de análise estabelecidos na pesquisa.

A Tabela 3 apresenta o sumário dos resultados avaliados. Foram considerados os valores *First View* e *Full load* para as métricas: localização do código CSS (topo ou final da página), tipo do CSS (minificado ou não), localização do código JS (topo ou final da página), tipo do JS (minificado ou não). Os valores indicados referem-se ao valor-*P*

Figura 19 – *Boxplots* com o *Full Load* e os parâmetros criados

Fonte: elaborado pelo autor

mencionado anteriormente no Capítulo 3. Conforme visto, considera-se que para obter relevância significativa, os valores encontrados precisam ser inferiores a 0,05 (variável  $\alpha$ ). Como notado, o fator *JsLocation* para o *First View* apresentou um valor bem abaixo de  $\alpha$ , o que sugere que a localização do **JS** é um critério que influencia fortemente o tempo de carregamento total da página, considerando o ambiental experimental utilizado.

Outra métrica que estabelece certa relação de influência no desempenho da página avaliada é o *CssLocation*, também para o *First View*, que obteve um valor-*P* menor que a variável  $\alpha$ . Pode-se inferir que a localização do **CSS** tende a influenciar no carregamento da página, embora o valor-*P* dela não seja tão mais baixo que  $\alpha$ . Os demais argumentos não apresentaram valores que sugerem interferência direta nos resultados.

Uma outra ferramenta gráfica utilizada na avaliação dos resultados foi obtida a partir do teste de *Tukey*. Esse teste tem como entrada os valores da análise **MANOVA**, e seus resultados apontam quais variáveis da análise feita mais influenciou de fato nas variações observadas<sup>3</sup>. A Figura 20 ilustra os resultados do teste de *Tukey* em relação ao parâmetro *First View* das páginas. O teste sugere diferença significativa nos seguintes fatores: localização do código **CSS** (topo ou final), tipo do **CSS** (minificado ou não),

<sup>3</sup> Para mais informações sobre o teste de *Tukey*, recomenda-se a leitura de <<http://www.portaaction.com.br/anova/31-teste-de-tukey>>. Acessado em: 20 Ago 2017

Tabela 3 – Sumário dos resultados dos experimentos

<i>First View</i>			
<i>CssLocation</i>	<i>CssType</i>	<i>JsLocation</i>	<i>JsType</i>
0.01646	0.21438	-4.74	0.12407
<i>Full Load</i>			
<i>CssLocation</i>	<i>CssType</i>	<i>JsLocation</i>	<i>JsType</i>
0.7705	0.1522	0.8348	0.7582

Fonte: dados da pesquisa

localização do código **JS** (topo ou final) e tipo do **JS** (minificado ou não).

As retas horizontais localizadas à frente dos parâmetros indicam o nível de predominância de certa característica em relação à outra. Por exemplo: a primeira reta refere-se à localização do código **CSS** (topo ou final), no qual o ponto central dela indica neutralidade do parâmetro, ou seja, a localização não interferiu nos resultados do experimento realizado. Quanto mais à esquerda da reta, mais a localização do **CSS** no topo interfere, e vice-versa. A reta vertical tracejada indica um ponto de referência para a localização das retas horizontais, sendo que: quanto mais à direita da reta pontilhada, maior foi a influência do parâmetro da direita, caso contrário, maior foi a influência do parâmetro da esquerda.

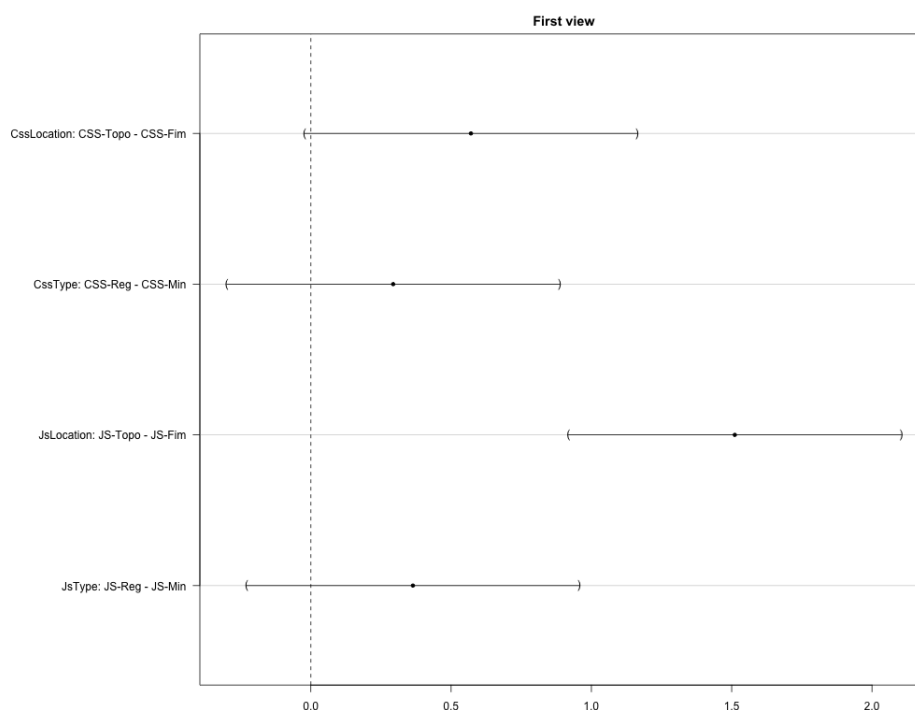
Observa-se na Figura 20 que o primeiro e terceiro parâmetros se sobressaem em relação à reta tracejada, sendo que os resultados sugerem que, no contexto experimental utilizado/estabelecido, a localização do **CSS** (*CSSLocation*) tende a apresentar melhores resultados no final da página, considerando-se a primeira reta horizontal. Já se tratando da localização do **JS** (*JSLocation*) é possível afirmar que ela apresenta resultados significativos quando inseridas ao fim da página, como se pode perceber na terceira reta horizontal.

Quando considera-se o fator *Full Load*, a análise se modifica. Conforme mostrado na Figura 21, nenhuma das métricas inseridas se desviou fortemente do ponto neutro. Vê-se que o tipo do código **CSS** (*CSSType*) é a métrica que mais apresenta uma certa tendência, que indica que o código minificado tende a predominar sobre o não minificado. Os demais parâmetros pouco se desviam da reta pontilhada.

## 4.4 Considerações finais

Esse capítulo apresentou a análise dos resultados encontrados com o experimento prático realizado com a página *Web* criada. Os resultados do experimento mostraram alguns fatores que influenciaram no desempenho da página testada. Notou-se que a localização do **JS** no final da página demonstrou diferenças estatísticas significativas no desempenho



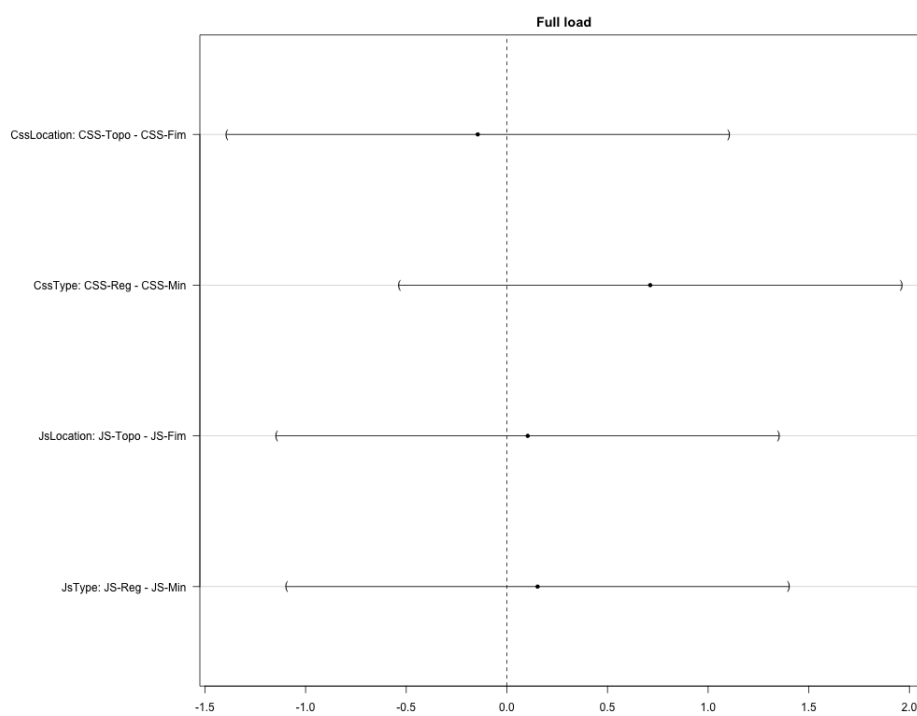
Figura 20 – Resultado do teste de *Tukey* para o parâmetro *First View*

Fonte: elaborado pelo autor

da mesma, analisando o *First View*. Já a localização do [CSS](#) apresentou certa tendência a ser mais significativa se inserida ao final do código, apesar dessa característica não ser estatisticamente tão relevante. Os demais parâmetros testados não apresentaram valor-*P* significativo, embora tenham sido notados algumas pequenas variações entre eles. O capítulo seguinte apresenta as considerações finais do projeto, indicando os desafios e limitações encontrados, além de sugestões para trabalhos futuros.



Figura 21 – Resultado do teste de *Tukey* para o parâmetro *Full Load*



Fonte: elaborado pelo autor

## 5 Considerações finais

Este trabalho apresentou questões sobre algumas técnicas e mecanismos para se reduzir o consumo excessivo de dados na *Web*, podendo influenciar no desempenho das aplicações, que crescem exponencialmente nos dias atuais. Além do consumo de dados, os fatores averiguados estão relacionados à satisfação e à percepção do usuário em relação às aplicações, pois caso elas apresentem atrasos ou deficiências na exibição e utilização das suas funcionalidades, os usuários provavelmente ficarão insatisfeitos, reduzindo o uso ou até mesmo deixando de utilizá-las.

Os primeiros testes foram realizados utilizando-se aplicações já existentes ([ICEA](#) e [ICSA](#)) com a ferramenta *Apache JMeter*. Esse conjunto de testes foi utilizado para se analisar aplicações que estão no contexto de uso cotidiano do ambiente acadêmico, a fim de se avaliar as questões envolvendo a performance de ambas as páginas. Os resultados sugeriram que as páginas testadas não apresentaram diferenças estatísticas significativas. É importante avaliar num trabalho futuro aplicações com escopos maiores e com mais funcionalidades, e identificar se os resultados são diferentes.

Já nos testes realizados com a página *Web* criada, alguns resultados apresentaram comportamentos interessantes. Considerando o ambiente experimental utilizado, os resultados sugerem que a localização do código [JS](#) pode estar diretamente relacionado com a performance da aplicação. Esse é um dos fatos citados no Capítulo 2 que foram verificados a partir da execução dos experimentos, sendo que inserir o código *Javascript* no início do arquivo irá bloquear a renderização de tudo que vem abaixo dele, atrasando na exibição do conteúdo.

Outro ponto interessante é a questão dos códigos estarem ou não minificados, que em alguns resultados apresentados pode-se notar certa tendência em influenciar a performance das aplicações, mesmo que de maneira pequena. Novamente, é importante avaliar em aplicações de maior porte se os resultados podem se modificar.

Quanto ao objetivo inicial do trabalho, que se resume em identificar os fatores que podem influenciar no desempenho de aplicações *Web*, ele foi abordado no decorrer da monografia. Durante o trabalho, algumas questões limitantes e/ou desafiadoras foram observadas. Por exemplo, verificar sobre como o tipo/local de hospedagem das aplicações podem influenciar no desempenho das mesmas. Essa questão não pôde ser verificada, visto que envolve questões mais complexas para a execução. Outra limitação encontrada se baseia na infraestrutura de rede utilizada para realizar os testes, que apresentou certa oscilação. Entretanto, pode não ter sido identificada nas análises realizadas.

Em suma, com a realização deste projeto foi possível se estabelecer alguns fatores

que afetam diretamente no desempenho *Web*, a partir do uso de ferramentas computacionais úteis e funcionais. Esses fatores deram embasamento para a geração do relatório de boas práticas que devem ser observadas no desenvolvimento de aplicações *Web*, que será descrito na próxima seção. Vale ressaltar que para os experimentos realizados, foram considerados os aspectos de *software* na elaboração e análise dos resultados. Já o contexto de *hardware* não foi o foco do trabalho.

## 5.1 Relatório de boas práticas

Com base em toda a pesquisa realizada no estudo, considera-se que alguns pontos são altamente relevantes para o escopo das aplicações *Web*. As boas práticas percebidas a partir do trabalho foram:

- Inserir o código *JavaScript* no final do arquivo: essa prática possibilitou uma redução no tempo de exibição inicial da página criada no trabalho, de acordo com os resultados alcançados com os experimentos realizados;
- Minificar JS: por mais que as variações percebidas no projeto foram poucas, para aplicações de grande porte espera-se que a prática da minificação do JS contribua para aumentar o desempenho das mesmas;
- Minificar CSS: ainda que as oscilações observadas no projeto sejam poucas, estima-se que para aplicações de grande porte essa prática contribua no desempenho das mesmas;
- Inserir o código CSS no final do arquivo: embora possa vir a afetar parcialmente a experiência de uso, de acordo com os experimentos realizados, foi notado que essa prática tende a contribuir para uma maior eficiência da página, ainda que inicialmente. Esse fato contradiz a informação abordada no Capítulo 2, que sugere a inserção no [CSS](#) no topo do código.

Essas são algumas das boas práticas notadas no estudo que podem contribuir na otimização das aplicações *Web*, em relação ao tempo de carregamento e exibição do conteúdo. Pelo fato do escopo de testes ter sido limitado e do surgimento de limitações, algumas questões não puderam ser verificadas.

## 5.2 Trabalhos futuros

As pesquisas em relação ao desempenho de aplicações *Web* podem ser realizadas em diversos outros contextos. Como propostas de continuidade deste trabalho, sugere-se a

---

realização de outros experimentos, como aplicações de portes maiores, utilização de outros *frameworks* e arquiteturas como *Laravel*, *CakePHP*, *Java*, *Ruby*, dentre outras. Outro ponto importante é a avaliação de outras variáveis de resposta nos experimentos, além das utilizadas no trabalho apresentado. Também seria interessante a avaliação dos experimentos realizados em outros ambientes de *hardware*, visando verificar se o comportamento das aplicações testadas se mantém.

## Referências

- BRASIL. *Senado Federal*: Projeto de lei nº 174/2016. 2016. <<https://www25.senado.leg.br/web/atividade/materias/-/materia/125599>>. [Online; Acessado em 7 Ago 2017]. Citado na página 23.
- BRASIL. *Câmara dos Deputados*: Projeto de lei nº 7182/2017. 2017. <<http://www.camara.gov.br/proposicoesWeb/fichadetramitacao?idProposicao=2126316>>. [Online; Acessado em 7 Ago 2017]. Citado 2 vezes nas páginas 23 e 24.
- CAELUM. *Desenvolvimento Web com HTML, CSS e JavaScript*. 2017. <<https://www.caelum.com.br/apostila-html-css-javascript/>>. [Online; Acessado em 15 Jul 2017]. Citado na página 19.
- DELAMARO, M. E.; MALDONADO, J. C.; JINO, M. *Introdução ao Teste de Software*. 4. ed. São Paulo: Elsevier Editora Ltda, 2007. Citado na página 21.
- DIGITAL, O. *EUA também enfrenta polêmica com limitação da internet e Netflix se posiciona*. 2016. <<http://olhardigital.uol.com.br/noticia/eua-tambem-enfrenta-polemica-com-limitacao-da/-internet-e-netflix-se-posiciona/62051>>. [Online; Acessado em 21 Nov 2016]. Citado na página 16.
- EVERTS, T.; KADLEC, T. *WPO Stats*. 2016. <<https://wpostats.com/>>. [Online; Acessado em 30 Out 2016]. Citado na página 19.
- GAILLY, J. loup; ADLER, M. *The gzip home page*. 2003. <<http://www.gzip.org/>>. [Online; Acessado em 15 Jul 2017]. Citado na página 20.
- INTEGRAÇÃO, P. *Manual de Utilização da Ferramenta JMeter*. Goiânia, 2013. Citado 2 vezes nas páginas 22 e 26.
- LOPES, S. *Por uma Web mais rápida: 26 técnicas de otimização de Sites*. 2011. <<http://blog.caelum.com.br/por-uma-web-mais-rapida-26-tecnicas-de-otimizacao-de-sites/>>. [Online; Acessado em 13 Jan 2017]. Citado na página 20.
- MAGALHÃES, M. N.; LIMA, A. C. P. de. *Noções de Probabilidade e Estatística*. 6. ed. São Paulo: Editora da Universidade de São Paulo, 2004. Citado na página 58.
- MONTGOMERY, D. C.; RUNGER, G. C. *Applied Statistics and Probability for Engineers*. 5. ed. [S.l.]: John Wiley & Sons, 2011. ISBN 9780470053041. Citado 2 vezes nas páginas 32 e 42.
- MONTGOMERY, D. C.; RUNGER, G. C. *Estatística Aplicada e Probabilidade para Engenheiros*. 5. ed. [S.l.]: Grupo Editorial Nacional, 2012. Citado na página 37.
- MOREIRA, V. *O que é renderizar?* 2013. <<https://oprofessorweb.wordpress.com/2013/12/10/o-que-e-renderizar/>>. [Online; Acessado em 21 Ago 2017]. Citado na página 20.
- PRESSMAN, R. S. *Engenharia de Software: Uma abordagem profissional*. 7. ed. São Paulo: AMGH Editora Ltda, 2011. Citado na página 21.

RIBEIRO, C. *Destilando JMeter I: Introdução e Conceitos*. 2013. <<http://www.bugbang.com.br/destilando-jmeter-i-introducao-e-conceitos/>>. [Online; Acessado em 21 Ago 2017]. Citado na página 36.

ROSA, B.; LITWAK, P. A. *Limite para banda larga fixa só é adotado nos EUA e no Canadá*. 2016. <<http://oglobo.globo.com/economia/negocios/limite-para-banda-larga-fixa-so-adotado-nos-eua-no-canada/-19143095>>. [Online; Acessado em 23 Nov 2016]. Citado na página 16.

TAVARES, L. *Boas práticas de programação*. 2016. <<http://www.devmedia.com.br/boas-praticas-de-programacao/31163>>. [Online; Acessado em 21 Ago 2017]. Citado na página 21.

VIEGAS, J. *A performance tem impacto na conversão de e-commerces?* 2016. <<http://blog.onedaytesting.com.br/performance-e-conversao-em-commerces/>>. [Online; Acessado em 23 Nov 2016]. Citado na página 16.

# Apêndices

# APÊNDICE A – Tabelas elaboradas pelo autor

Tabela 4 – Resultados dos testes com o *Apache JMeter*

Obs	Site	Usuários	Iterações	Replicação	Mín (ms)	Max (ms)	Médio (ms)	Vazão (Req/s)	Total Req.	Erros	Erros (%)	Tempo (minutos)
1	ICSA	100	300	1	93	112542	6813,22	14,52	30000	145	0,48333333	34,43526171
2	ICEA	100	30	1	361	13518	4808,75	19,94	3000	0	0	2,507522568
3	ICSA	1000	30	1	33	188896	7402,13	88,46	30000	26334	87,78	5,652272213
4	ICEA	100	300	1	266	187317	5630,10	17,14	30000	13	0,04333333	29,17152859
5	ICEA	1000	300	1	4	425402	12937,37	72,99	300000	136345	45,44833333	68,50253459
6	ICSA	100	30	1	239	21205	6069,57	14,74	3000	23	0,76666667	3,392130258
7	ICSA	1000	300	1	25	475767	7639,44	118,61	300000	275341	91,78033333	42,15496164
8	ICEA	1000	30	1	4	77000	8041,26	93,27	30000	12103	40,34333333	5,36078053
9	ICSA	100	30	2	252	21946	6692,16	13,2	3000	25	0,83333333	3,787878788
10	ICSA	100	300	2	26	56598	6929,28	14,23	30000	368	1,22666667	35,13703443
11	ICEA	100	30	2	260	23448	5775,36	14,99	3000	122	4,06666667	3,33557038
12	ICEA	1000	300	2	0	907505	7611,44	106,98	300000	145678	48,55933333	46,73770798
13	ICEA	100	300	2	38	279100	5236,58	18,63	30000	478	1,59333333	26,83843264
14	ICEA	1000	30	2	4	87537	11733,19	72,73	30000	12871	42,90333333	6,874742197
15	ICSA	1000	30	2	32	128628	8104,96	82,34	30000	26634	88,78	6,072382803
16	ICSA	1000	300	2	11	161877	8139,88	109,91	300000	277031	92,34366667	45,49176599
17	ICSA	1000	30	3	52	105655	7025,67	105,46	30000	26966	89,88666667	4,741134079
18	ICEA	1000	30	3	4	43898	9088,99	93,07	30000	17942	59,80666667	5,372300419
19	ICSA	1000	300	3	30	144450	7594,59	119,5	300000	274920	91,64	41,84100418
20	ICSA	100	30	3	256	15574	6002,44	16,09	3000	0	0	3,107520199
21	ICEA	100	30	3	273	23098	5277,61	18,1	3000	0	0	2,762430939
22	ICEA	100	300	3	260	14950	5324,31	18,54	30000	0	0	26,96871629
23	ICSA	100	300	3	299	19262	6570,56	15,05	30000	0	0	33,22259136
24	ICEA	1000	300	3	4	62515	9361,74	100,7	300000	202287	67,429	49,65243297
25	ICEA	100	30	4	459	12418	5110,97	18,68	3000	0	0	2,676659529
26	ICSA	100	300	4	249	27791	6393,75	15,46	30000	0	0	32,34152652
27	ICSA	1000	300	4	4	247473	7795,46	124,86	300000	279581	93,19366667	40,04485023
28	ICEA	1000	300	4	0	780738	7001,30	100,16	300000	250367	83,45566667	49,9201278
29	ICEA	1000	30	4	4	42407	12574,52	66,92	30000	14642	48,80666667	7,47160789
30	ICSA	1000	30	4	47	100753	8091,76	101,35	30000	25495	84,98333333	4,933399112
31	ICEA	100	300	4	377	21846	2603,69	37,56	30000	59	0,19666667	13,31203408
32	ICSA	100	30	4	334	21193	4915,55	19,34	3000	17	0,56666667	2,585315408
33	ICEA	1000	300	5	4	41776	6697,86	139,94	300000	233152	77,71733333	35,7295984
34	ICEA	100	30	5	286	21312	2523,89	33,19	3000	18	0,6	1,506477855
35	ICSA	100	30	5	266	21107	4797,06	18,99	3000	5	0,16666667	2,632964718
36	ICSA	1000	30	5	45	100095	7382,50	108,32	30000	25580	85,26666667	4,615952733
37	ICEA	1000	30	5	4	39394	8037,55	106,74	30000	19016	63,38666667	4,684279558
38	ICEA	100	300	5	324	25824	2703,51	36,24	30000	53	0,17666667	13,79690949
39	ICSA	100	300	5	239	21219	4852,68	20,34	30000	257	0,85666667	24,58210423
40	ICSA	1000	300	5	5	137440	7521,96	122,76	300000	262892	87,63066667	40,72987944

Fonte: dados da pesquisa



Tabela 5 – Resultados dos testes com a página Web criada - Parte 1

Obs	Replication	Tool	CssLocation	CssType	JsLocation	JsType	CSS	JS	FirstView	FullLoad	CommandLine
1	1	WebPagetest	Final	Min	Topo	Normal	1	6	2,97	6,75	http://200.239.154.203:8080/tcc/index.php?css=1&js=6
2	1	WebPagetest	Topo	Min	Final	Normal	0	5	1,78	9,55	http://200.239.154.203:8080/tcc/index.php?css=0&js=5
3	1	WebPagetest	Final	Normal	Topo	Normal	3	6	5,66	8,21	http://200.239.154.203:8080/tcc/index.php?css=3&js=6
4	1	GTmetrix	Topo	Normal	Final	Min	2	5	1,1	6,8	http://200.239.154.203:8080/tcc/index.php?css=2&js=5
5	1	GTmetrix	Final	Normal	Topo	Normal	3	6	1,3	5,9	http://200.239.154.203:8080/tcc/index.php?css=3&js=6
6	1	WebPagetest	Final	Min	Topo	Min	1	4	2,76	5,86	http://200.239.154.203:8080/tcc/index.php?css=1&js=4
7	1	WebPagetest	Topo	Normal	Topo	Min	2	4	1,97	7,84	http://200.239.154.203:8080/tcc/index.php?css=2&js=4
8	1	WebPagetest	Final	Normal	Topo	Min	3	4	8,16	10,98	http://200.239.154.203:8080/tcc/index.php?css=3&js=4
9	1	WebPagetest	Topo	Normal	Topo	Normal	2	6	3,09	6,85	http://200.239.154.203:8080/tcc/index.php?css=2&js=6
10	1	GTmetrix	Topo	Min	Topo	Min	0	4	1,6	6,9	http://200.239.154.203:8080/tcc/index.php?css=0&js=4
11	1	WebPagetest	Final	Min	Final	Min	1	5	0,69	7,7	http://200.239.154.203:8080/tcc/index.php?css=1&js=5
12	1	GTmetrix	Final	Normal	Final	Normal	3	7	0,13	13,9	http://200.239.154.203:8080/tcc/index.php?css=3&js=7
13	1	GTmetrix	Final	Min	Final	Min	1	5	0,23	3,6	http://200.239.154.203:8080/tcc/index.php?css=1&js=5
14	1	GTmetrix	Final	Normal	Final	Min	3	5	0,11	4,7	http://200.239.154.203:8080/tcc/index.php?css=3&js=5
15	1	GTmetrix	Topo	Min	Topo	Normal	0	6	1	3,1	http://200.239.154.203:8080/tcc/index.php?css=0&js=6
16	1	GTmetrix	Final	Normal	Topo	Min	3	4	0,7	3,2	http://200.239.154.203:8080/tcc/index.php?css=3&js=4
17	1	WebPagetest	Topo	Min	Final	Normal	0	7	1,69	6,3	http://200.239.154.203:8080/tcc/index.php?css=0&js=7
18	1	GTmetrix	Topo	Normal	Final	Normal	2	7	1,9	6,7	http://200.239.154.203:8080/tcc/index.php?css=2&js=7
19	1	GTmetrix	Final	Min	Topo	Normal	1	6	2,1	5,9	http://200.239.154.203:8080/tcc/index.php?css=1&js=6
20	1	GTmetrix	Topo	Normal	Topo	Normal	2	6	0,8	3,2	http://200.239.154.203:8080/tcc/index.php?css=2&js=6
21	1	WebPagetest	Topo	Min	Topo	Min	0	4	1,88	5,52	http://200.239.154.203:8080/tcc/index.php?css=0&js=4
22	1	GTmetrix	Final	Min	Topo	Min	1	4	0,6	2,9	http://200.239.154.203:8080/tcc/index.php?css=1&js=4
23	1	GTmetrix	Topo	Min	Final	Normal	0	7	0,49	1,7	http://200.239.154.203:8080/tcc/index.php?css=0&js=7
24	1	GTmetrix	Topo	Normal	Topo	Min	2	4	0,9	3,2	http://200.239.154.203:8080/tcc/index.php?css=2&js=4
25	1	WebPagetest	Final	Min	Final	Normal	1	7	0,88	6,01	http://200.239.154.203:8080/tcc/index.php?css=1&js=7
26	1	WebPagetest	Topo	Normal	Final	Normal	2	7	2,98	7,36	http://200.239.154.203:8080/tcc/index.php?css=2&js=7
27	1	WebPagetest	Final	Normal	Final	Normal	3	7	0,96	9,67	http://200.239.154.203:8080/tcc/index.php?css=3&js=7
28	1	GTmetrix	Topo	Min	Final	Min	0	5	0,7	2,5	http://200.239.154.203:8080/tcc/index.php?css=0&js=5
29	1	WebPagetest	Final	Normal	Final	Min	3	5	0,79	5,72	http://200.239.154.203:8080/tcc/index.php?css=3&js=5
30	1	WebPagetest	Topo	Normal	Final	Min	2	5	5,26	8,97	http://200.239.154.203:8080/tcc/index.php?css=2&js=5
31	1	GTmetrix	Final	Min	Final	Normal	1	7	0,16	4,6	http://200.239.154.203:8080/tcc/index.php?css=1&js=7
32	1	WebPagetest	Topo	Min	Topo	Normal	0	6	2,77	9,03	http://200.239.154.203:8080/tcc/index.php?css=0&js=6
33	2	GTmetrix	Topo	Min	Final	Min	0	5	3,2	10,9	http://200.239.154.203:8080/tcc/index.php?css=0&js=5
34	2	WebPagetest	Topo	Min	Final	Min	0	5	1,58	5,26	http://200.239.154.203:8080/tcc/index.php?css=0&js=5
35	2	WebPagetest	Topo	Min	Topo	Normal	0	6	6,25	8,64	http://200.239.154.203:8080/tcc/index.php?css=0&js=6
36	2	GTmetrix	Final	Normal	Final	Normal	3	7	0,18	6,3	http://200.239.154.203:8080/tcc/index.php?css=3&js=7
37	2	WebPagetest	Final	Min	Topo	Min	1	4	2,18	7,7	http://200.239.154.203:8080/tcc/index.php?css=1&js=4
38	2	WebPagetest	Final	Normal	Final	Min	3	5	0,79	12,84	http://200.239.154.203:8080/tcc/index.php?css=3&js=5
39	2	GTmetrix	Topo	Normal	Final	Normal	2	7	1,4	12,4	http://200.239.154.203:8080/tcc/index.php?css=2&js=7
40	2	WebPagetest	Topo	Normal	Topo	Min	2	4	3,15	8,15	http://200.239.154.203:8080/tcc/index.php?css=2&js=4
41	2	GTmetrix	Final	Min	Topo	Min	1	4	1,1	5,2	http://200.239.154.203:8080/tcc/index.php?css=1&js=4
42	2	GTmetrix	Topo	Min	Topo	Min	0	4	1,2	8,1	http://200.239.154.203:8080/tcc/index.php?css=0&js=4
43	2	GTmetrix	Topo	Normal	Topo	Normal	2	6	1,5	3,1	http://200.239.154.203:8080/tcc/index.php?css=2&js=6
44	2	WebPagetest	Topo	Min	Final	Normal	0	7	2,18	13,96	http://200.239.154.203:8080/tcc/index.php?css=0&js=7
45	2	GTmetrix	Final	Min	Topo	Normal	1	6	4,5	15,8	http://200.239.154.203:8080/tcc/index.php?css=1&js=6
46	2	WebPagetest	Topo	Min	Topo	Min	0	4	2,68	7,27	http://200.239.154.203:8080/tcc/index.php?css=0&js=4
47	2	WebPagetest	Topo	Normal	Topo	Normal	2	6	8,06	12	http://200.239.154.203:8080/tcc/index.php?css=2&js=6
48	2	GTmetrix	Final	Normal	Topo	Normal	3	6	3,1	5,6	http://200.239.154.203:8080/tcc/index.php?css=3&js=6
49	2	GTmetrix	Final	Min	Final	Min	1	5	0,34	2,9	http://200.239.154.203:8080/tcc/index.php?css=1&js=5
50	2	WebPagetest	Final	Min	Final	Min	1	5	0,77	6,61	http://200.239.154.203:8080/tcc/index.php?css=1&js=5
51	2	WebPagetest	Final	Min	Final	Normal	1	7	1,16	9,42	http://200.239.154.203:8080/tcc/index.php?css=1&js=7
52	2	WebPagetest	Final	Min	Topo	Normal	1	6	4,56	10,32	http://200.239.154.203:8080/tcc/index.php?css=1&js=6
53	2	GTmetrix	Final	Min	Final	Normal	1	7	0,25	7	http://200.239.154.203:8080/tcc/index.php?css=1&js=7
54	2	WebPagetest	Final	Normal	Topo	Normal	3	6	5,95	14,53	http://200.239.154.203:8080/tcc/index.php?css=3&js=6
55	2	GTmetrix	Topo	Normal	Topo	Min	2	4	2,1	9,2	http://200.239.154.203:8080/tcc/index.php?css=2&js=4
56	2	WebPagetest	Topo	Normal	Final	Min	2	5	2,76	9,02	http://200.239.154.203:8080/tcc/index.php?css=2&js=5
57	2	GTmetrix	Topo	Normal	Final	Min	2	5	1,1	3,5	http://200.239.154.203:8080/tcc/index.php?css=2&js=5
58	2	GTmetrix	Final	Normal	Topo	Min	3	4	2,3	10,1	http://200.239.154.203:8080/tcc/index.php?css=3&js=4
59	2	WebPagetest	Final	Normal	Topo	Min	3	4	2,47	5,98	http://200.239.154.203:8080/tcc/index.php?css=3&js=4
60	2	GTmetrix	Topo	Min	Topo	Normal	0	6	1,4	4	http://200.239.154.203:8080/tcc/index.php?css=0&js=6
61	2	GTmetrix	Topo	Min	Final	Normal	0	7	1,8	6,8	http://200.239.154.203:8080/tcc/index.php?css=0&js=7
62	2	GTmetrix	Final	Normal	Final	Min	3	5	0,3	14,4	http://200.239.154.203:8080/tcc/index.php?css=3&js=5
63	2	WebPagetest	Topo	Normal	Final	Normal	2	7	3,25	9,35	http://200.239.154.203:8080/tcc/index.php?css=2&js=7
64	2	WebPagetest	Final	Normal	Final	Normal	3	7	1,66	11,99	http://200.239.154.203:8080/tcc/index.php?css=3&js=7
65	3	GTmetrix	Topo	Normal	Final	Min	2	5	1,1	8,2	http://200.239.154.203:8080/tcc/index.php?css=2&js=5
66	3	GTmetrix	Topo	Normal	Topo	Min	2	4	2,5	13,2	http://200.239.154.203:8080/tcc/index.php?css=2&js=4
67	3	GTmetrix	Final	Min	Topo	Min	1	4	1,2	11,1	http://200.239.154.203:8080/tcc/index.php?css=1&js=4
68	3	GTmetrix	Topo	Min	Final	Normal	0	7	2,5	11,7	http://200.239.154.203:8080/tcc/index.php?css=0&js=7
69	3	WebPagetest	Final	Min	Topo	Min	1	4	2,95	8,1	http://200.239.154.203:8080/tcc/index.php?css=1&js=4
70	3	GTmetrix	Final	Normal	Final	Normal	3	7	0,32	4,6	http://200.239.154.203:8080/tcc/index.php?css=3&js=7
71	3	GTmetrix	Topo	Normal	Final	Normal	2	7	1,7	8,2	http://200.239.154.203:8080/tcc/index.php?css=2&js=7
72	3	WebPagetest	Final	Normal	Topo	Normal	3	6	3,18	9,88	http://200.239.154.203:8080/tcc/index.php?css=3&js=6
73	3	GTmetrix	Final	Min	Final	Min	1	5	0,27	4,3	http://200.239.154.203:8080/tcc/index.php?css=1&js=5
74	3	WebPagetest	Topo	Normal	Topo	Min	2	4	4,55	8,68	http://200.239.154.203:8080/tcc/index.php?css=2&js=4
75	3	GTmetrix	Final	Normal	Final	Min	3	5	0,26	3,5	http://200.239.154.203:8080/tcc/index.php?css=3&js=5
76	3	GTmetrix	Topo	Min	Final	Min	0	5	1,1	5,5	http://200.239.154.203:8080/tcc/index.php?css=0&js=5
77	3	WebPagetest	Final	Normal	Topo	Min	3	4	8,06	9,02	http://200.239.154.203:8080/tcc/index.php?css=3&js=4
78	3	GTmetrix	Final	Normal	Topo	Normal	3	6	2,4	8,4	http://200.239.154.203:8080/tcc/index.php?css=3&js=6
79	3	GTmetrix	Topo	Min	Topo	Min	0	4	2	9,4	http://200.239.154.203:8080/tcc/index.php?css=0&js=4
80	3	WebPagetest	Topo	Min	Topo	Min	0	4	3,15	9,17	http://200.239.154.203:8080/tcc/index.php?css=0&js=4

Fonte: dados da pesquisa

Tabela 6 – Resultados dos testes com a página *Web* criada - Parte 2

Obs	Replication	Tool	CssLocation	CssType	JsLocation	JsType	CSS	JS	FirstView	FullLoad	CommandLine
81	3	GTmetrix	Final	Normal	Topo	Min	3	4	1,2	3,9	http://200.239.154.203:8080/tcc/index.php?css=3&js=4
82	3	WebPagetest	Final	Min	Final	Min	1	5	1,35	9,11	http://200.239.154.203:8080/tcc/index.php?css=1&js=5
83	3	WebPagetest	Final	Min	Topo	Normal	1	6	4,16	8,9	http://200.239.154.203:8080/tcc/index.php?css=1&js=6
84	3	WebPagetest	Topo	Min	Final	Normal	0	7	4,66	9,23	http://200.239.154.203:8080/tcc/index.php?css=0&js=7
85	3	WebPagetest	Final	Normal	Final	Min	3	5	1,36	9,01	http://200.239.154.203:8080/tcc/index.php?css=3&js=5
86	3	GTmetrix	Topo	Normal	Topo	Normal	2	6	4	7,8	http://200.239.154.203:8080/tcc/index.php?css=2&js=6
87	3	GTmetrix	Final	Min	Topo	Normal	1	6	3	8,1	http://200.239.154.203:8080/tcc/index.php?css=1&js=6
88	3	WebPagetest	Topo	Normal	Final	Normal	2	7	2,07	8,11	http://200.239.154.203:8080/tcc/index.php?css=2&js=7
89	3	WebPagetest	Final	Normal	Final	Normal	3	7	0,78	6,73	http://200.239.154.203:8080/tcc/index.php?css=3&js=7
90	3	GTmetrix	Final	Min	Final	Normal	1	7	0,27	2,4	http://200.239.154.203:8080/tcc/index.php?css=1&js=7
91	3	WebPagetest	Topo	Normal	Final	Min	2	5	2,86	9,01	http://200.239.154.203:8080/tcc/index.php?css=2&js=5
92	3	WebPagetest	Topo	Normal	Topo	Normal	2	6	3,68	6,35	http://200.239.154.203:8080/tcc/index.php?css=2&js=6
93	3	GTmetrix	Topo	Min	Topo	Normal	0	6	1,9	3,3	http://200.239.154.203:8080/tcc/index.php?css=0&js=6
94	3	WebPagetest	Topo	Min	Final	Min	0	5	2,08	6,66	http://200.239.154.203:8080/tcc/index.php?css=0&js=5
95	3	WebPagetest	Topo	Min	Topo	Normal	0	6	3,5	6,35	http://200.239.154.203:8080/tcc/index.php?css=0&js=6
96	3	WebPagetest	Final	Min	Final	Normal	1	7	0,94	9,6	http://200.239.154.203:8080/tcc/index.php?css=1&js=7
97	4	WebPagetest	Topo	Min	Topo	Min	0	4	2,08	6,48	http://200.239.154.203:8080/tcc/index.php?css=0&js=4
98	4	WebPagetest	Topo	Normal	Final	Normal	2	7	2,29	6,16	http://200.239.154.203:8080/tcc/index.php?css=2&js=7
99	4	WebPagetest	Final	Min	Final	Normal	1	7	0,97	6,3	http://200.239.154.203:8080/tcc/index.php?css=1&js=7
100	4	GTmetrix	Final	Normal	Topo	Normal	3	6	1,5	2,5	http://200.239.154.203:8080/tcc/index.php?css=3&js=6
101	4	WebPagetest	Final	Normal	Topo	Min	3	4	4,15	8,57	http://200.239.154.203:8080/tcc/index.php?css=3&js=4
102	4	WebPagetest	Final	Normal	Final	Normal	3	7	1,16	9,74	http://200.239.154.203:8080/tcc/index.php?css=3&js=7
103	4	WebPagetest	Final	Normal	Final	Min	3	5	1,15	9,32	http://200.239.154.203:8080/tcc/index.php?css=3&js=5
104	4	GTmetrix	Final	Min	Topo	Normal	1	6	1,4	2,6	http://200.239.154.203:8080/tcc/index.php?css=1&js=6
105	4	WebPagetest	Topo	Min	Topo	Normal	0	6	6,06	8,66	http://200.239.154.203:8080/tcc/index.php?css=0&js=6
106	4	GTmetrix	Final	Min	Topo	Min	1	4	0,8	3	http://200.239.154.203:8080/tcc/index.php?css=1&js=4
107	4	GTmetrix	Topo	Min	Final	Normal	0	7	1	3,1	http://200.239.154.203:8080/tcc/index.php?css=0&js=7
108	4	WebPagetest	Final	Min	Topo	Min	1	4	4,56	8,75	http://200.239.154.203:8080/tcc/index.php?css=1&js=4
109	4	WebPagetest	Topo	Normal	Final	Min	2	5	2,18	6,92	http://200.239.154.203:8080/tcc/index.php?css=2&js=5
110	4	GTmetrix	Topo	Normal	Final	Normal	2	7	1,1	3	http://200.239.154.203:8080/tcc/index.php?css=2&js=7
111	4	GTmetrix	Final	Normal	Topo	Min	3	4	1,5	2,8	http://200.239.154.203:8080/tcc/index.php?css=3&js=4
112	4	GTmetrix	Topo	Normal	Topo	Min	2	4	1	2,9	http://200.239.154.203:8080/tcc/index.php?css=2&js=4
113	4	WebPagetest	Topo	Min	Final	Min	0	5	2,86	8,62	http://200.239.154.203:8080/tcc/index.php?css=0&js=5
114	4	WebPagetest	Topo	Normal	Topo	Normal	2	6	3,57	8,93	http://200.239.154.203:8080/tcc/index.php?css=2&js=6
115	4	GTmetrix	Topo	Min	Topo	Min	0	4	1,4	2,9	http://200.239.154.203:8080/tcc/index.php?css=0&js=4
116	4	GTmetrix	Topo	Min	Final	Min	0	5	1,1	3,3	http://200.239.154.203:8080/tcc/index.php?css=0&js=5
117	4	GTmetrix	Topo	Normal	Topo	Normal	2	6	1,5	3,4	http://200.239.154.203:8080/tcc/index.php?css=2&js=6
118	4	GTmetrix	Final	Normal	Final	Normal	3	7	0,35	2,7	http://200.239.154.203:8080/tcc/index.php?css=3&js=7
119	4	WebPagetest	Final	Normal	Topo	Normal	3	6	2,98	6,21	http://200.239.154.203:8080/tcc/index.php?css=3&js=6
120	4	GTmetrix	Topo	Min	Topo	Normal	0	6	1,3	2,7	http://200.239.154.203:8080/tcc/index.php?css=0&js=6
121	4	GTmetrix	Final	Normal	Final	Min	3	5	0,17	1,8	http://200.239.154.203:8080/tcc/index.php?css=3&js=5
122	4	WebPagetest	Final	Min	Final	Min	1	5	0,95	8,87	http://200.239.154.203:8080/tcc/index.php?css=1&js=5
123	4	GTmetrix	Final	Min	Final	Normal	1	7	0,18	2,3	http://200.239.154.203:8080/tcc/index.php?css=1&js=7
124	4	WebPagetest	Final	Min	Topo	Normal	1	6	2,68	6,88	http://200.239.154.203:8080/tcc/index.php?css=1&js=6
125	4	GTmetrix	Topo	Normal	Final	Min	2	5	0,7	1,8	http://200.239.154.203:8080/tcc/index.php?css=2&js=5
126	4	WebPagetest	Topo	Min	Final	Normal	0	7	1,87	5,8	http://200.239.154.203:8080/tcc/index.php?css=0&js=7
127	4	WebPagetest	Topo	Normal	Topo	Min	2	4	1,97	5,65	http://200.239.154.203:8080/tcc/index.php?css=2&js=4
128	4	GTmetrix	Final	Min	Final	Min	1	5	0,16	2,6	http://200.239.154.203:8080/tcc/index.php?css=1&js=5
129	5	GTmetrix	Final	Min	Final	Normal	1	7	0,16	2,1	http://200.239.154.203:8080/tcc/index.php?css=1&js=7
130	5	WebPagetest	Topo	Min	Topo	Normal	0	6	8,16	8,74	http://200.239.154.203:8080/tcc/index.php?css=0&js=6
131	5	GTmetrix	Topo	Normal	Topo	Normal	2	6	1,2	4,2	http://200.239.154.203:8080/tcc/index.php?css=2&js=6
132	5	WebPagetest	Final	Min	Topo	Normal	1	6	2,68	5,94	http://200.239.154.203:8080/tcc/index.php?css=1&js=6
133	5	WebPagetest	Topo	Normal	Final	Min	2	5	2,79	5,5	http://200.239.154.203:8080/tcc/index.php?css=2&js=5
134	5	GTmetrix	Final	Min	Topo	Normal	1	6	0,9	2,9	http://200.239.154.203:8080/tcc/index.php?css=1&js=6
135	5	WebPagetest	Final	Min	Topo	Min	1	4	2,07	5,45	http://200.239.154.203:8080/tcc/index.php?css=1&js=4
136	5	WebPagetest	Final	Normal	Topo	Min	3	4	5,55	8,91	http://200.239.154.203:8080/tcc/index.php?css=3&js=4
137	5	GTmetrix	Topo	Normal	Topo	Min	2	4	1,1	3,3	http://200.239.154.203:8080/tcc/index.php?css=2&js=4
138	5	WebPagetest	Topo	Min	Final	Normal	0	7	1,98	5,81	http://200.239.154.203:8080/tcc/index.php?css=0&js=7
139	5	WebPagetest	Topo	Normal	Topo	Normal	2	6	3,08	6,52	http://200.239.154.203:8080/tcc/index.php?css=2&js=6
140	5	GTmetrix	Topo	Normal	Final	Min	2	5	0,8	3,9	http://200.239.154.203:8080/tcc/index.php?css=2&js=5
141	5	GTmetrix	Final	Min	Topo	Min	1	4	0,7	1,8	http://200.239.154.203:8080/tcc/index.php?css=1&js=4
142	5	GTmetrix	Final	Normal	Final	Normal	3	7	0,17	2	http://200.239.154.203:8080/tcc/index.php?css=3&js=7
143	5	WebPagetest	Final	Normal	Topo	Normal	3	6	5,55	9,39	http://200.239.154.203:8080/tcc/index.php?css=3&js=6
144	5	GTmetrix	Topo	Min	Final	Normal	0	7	0,7	2,2	http://200.239.154.203:8080/tcc/index.php?css=0&js=7
145	5	GTmetrix	Final	Normal	Final	Min	3	5	0,16	2,2	http://200.239.154.203:8080/tcc/index.php?css=3&js=5
146	5	GTmetrix	Topo	Normal	Final	Normal	2	7	0,8	3,5	http://200.239.154.203:8080/tcc/index.php?css=2&js=7
147	5	WebPagetest	Topo	Normal	Final	Normal	2	7	2,49	6	http://200.239.154.203:8080/tcc/index.php?css=2&js=7
148	5	WebPagetest	Topo	Min	Final	Min	0	5	3,06	8,42	http://200.239.154.203:8080/tcc/index.php?css=0&js=5
149	5	GTmetrix	Final	Normal	Topo	Min	3	4	0,8	4,7	http://200.239.154.203:8080/tcc/index.php?css=3&js=4
150	5	WebPagetest	Final	Normal	Final	Normal	3	7	0,68	5,82	http://200.239.154.203:8080/tcc/index.php?css=3&js=7
151	5	GTmetrix	Final	Min	Final	Min	1	5	0,19	3,2	http://200.239.154.203:8080/tcc/index.php?css=1&js=5
152	5	GTmetrix	Topo	Min	Topo	Normal	0	6	1	2,7	http://200.239.154.203:8080/tcc/index.php?css=0&js=6
153	5	WebPagetest	Topo	Normal	Topo	Min	2	4	3,25	8,92	http://200.239.154.203:8080/tcc/index.php?css=2&js=4
154	5	WebPagetest	Final	Min	Final	Min	1	5	0,58	6,64	http://200.239.154.203:8080/tcc/index.php?css=1&js=5
155	5	GTmetrix	Final	Normal	Topo	Normal	3	6	0,9	2,3	http://200.239.154.203:8080/tcc/index.php?css=3&js=6
156	5	GTmetrix	Topo	Min	Topo	Min	0	4	2,4	3,9	http://200.239.154.203:8080/tcc/index.php?css=0&js=4
157	5	WebPagetest	Topo	Min	Topo	Min	0	4	2,08	5,77	http://200.239.154.203:8080/tcc/index.php?css=0&js=4
158	5	WebPagetest	Final	Min	Final	Normal	1	7	0,67	5,88	http://200.239.154.203:8080/tcc/index.php?css=1&js=7
159	5	GTmetrix	Topo	Min	Final	Min	0	5	0,7	3,4	http://200.239.154.203:8080/tcc/index.php?css=0&js=5
160	5	WebPagetest	Final	Normal	Final	Min	3	5	0,86	9,59	http://200.239.154.203:8080/tcc/index.php?css=3&js=5

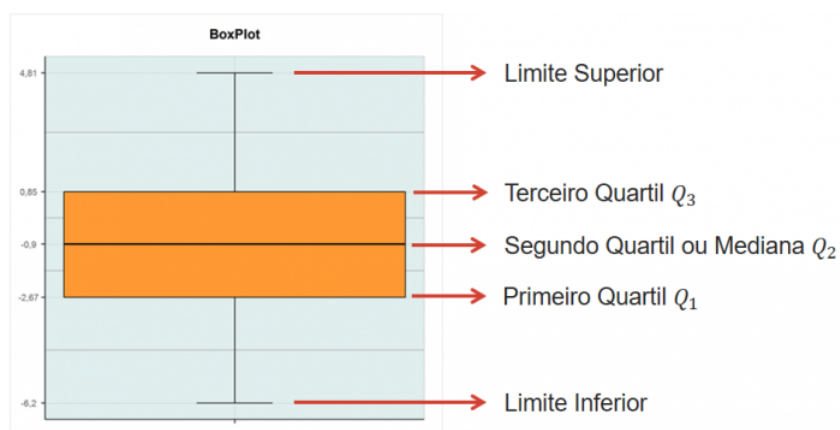
Fonte: dados da pesquisa

# APÊNDICE B – Ferramenta gráfica utilizada na análise dos resultados

## B.1 *Boxplot*

Resumidamente, pode-se dizer que o *Boxplot* (conhecido também como diagrama de caixa ou diagrama de extremos e quartis) é uma ferramenta gráfica que representa a variação dos dados observados referentes a uma variável numérica. Conforme mencionado em [Magalhães e Lima \(2004\)](#), o *Boxplot* trabalha com a ideia de quartil, definido como um dos três valores que divide o conjunto de dados em quatro partes iguais, nas quais cada parte representa 1/4 da amostra total. De acordo com a [Figura 22](#), que ilustra um exemplo dessa ferramenta, têm-se que o *Boxplot* fornece, dentre outras informações, os seguintes dados:

- Limite superior: representa o maior valor encontrado na amostra em questão;
- Q1: corresponde a 25% das medidas da amostra;
- Q2 ou Mediana: corresponde a 50% das medidas da amostra;
- Q3: corresponde a 75% das medidas da amostra;
- Limite inferior: menor valor observado na amostra.

Figura 22 – Exemplo de *Boxplot*

Fonte: <<http://www.portaaction.com.br/estatistica-basica/31-boxplot>>. Acessado em: 05 Ago 2017