Responsivo

Plano de Testes de Desempenho



Conteúdo

[1. Introdução 3](#_Toc444785151)

[**1.1.** Objetivos 3](#_Toc444785152)

[**1.2.** Sistema 3](#_Toc444785153)

**1.3.** Abordagem do Teste ...........................................................................................................4

[2. Escopo](#_Toc444785154) 4

[**2.1.** Detalhes do Escopo](#_Toc444785155) 4

[3. Características de Ambiente](#_Toc444785156) 5

[**3.1.** Aplicação em Teste](#_Toc444785157) 5

[**3.2.** Infraestrutura para Testes](#_Toc444785158) 5

**3.3.** Relatório .........................................................................................................................6

[4. Objetivos dos Testes](#_Toc444785159) 6

[**4.1.** Detalhes dos Testes](#_Toc444785160) 6

[5. Recursos](#_Toc444785161) 7

[**5.1.** Detalhes dos Recursos](#_Toc444785162) 7

[6. Projeto de Testes](#_Toc444785163) 7

[**6.1.** Detalhes](#_Toc444785164) 7

[7. Critérios de Aceite](#_Toc444785165) 8

[**7.1.** Critérios de Conclusão](#_Toc444785166) 8

[8. Premissas](#_Toc444785167) 9

[**8.1.** Detalhes das Premissas](#_Toc444785168) 9

[9. VALORES PARA SERVIDORES REMOTOS](#_Toc444785167) 10

[10. Glossário](#_Toc444785167) 11

# Introdução

|  |
| --- |
| Objetivos |
| Esse documento compõe-se dos seguintes objetivos:   * Identificar informações de projeto existentes e os cenários que devem ser testados. * Identificar os recursos necessários e prover uma estimativa dos esforços de teste. * Encontrar o ponto de saturação e dizer qual a vazão suportada pela aplicação. * Realizar um teste com uma carga previamente conhecida a fim de atestar se o novo sistema suportará a atual carga de produção com um tempo de resposta previamente acordado. * Saber qual será o comportamento da aplicação quando exposta por um longo período à uma carga consideravelmente alta e previamente conhecida. * Elencar quais são os gargalos que impedem a escalabilidade da aplicação. * Saber como a aplicação se recupera após alguma (s) falha (s) causada pelo teste de stress. |

|  |
| --- |
| Sistema |
| O teste de desempenho será realizado no sistema Responsivo(cvc). O Responsivo é o novo web site que substituíra o em produção. O mesmo fornece uma visualização abrangente das vendas durante todo o ciclo de vida da compra. Se tratando do sistema ele possui mas funcionalidades onde cliente consegue comprar diversos produtos na mesma tela, outro benefício é que ele é multiplataforma(tablet, smathphone). |

|  |
| --- |
| Abordagem do Teste |
| Tipos de testes   * Teste de Carga (Load Test): O objetivo é testar o software em seu ambiente normal de operação para observar seu comportamento com um aumento progressivo da carga de dados. * Teste de desempenho (Performance Test): Visa garantir que o sistema atende os níveis de desempenho e tempo de resposta combinados com os usuários e definidos nos requisitos. * Estabilidade: Identificar o Tempo de Resposta das transações especificadas para número determinado de usuários, por um período de tempo estendido. * Teste de Stress (Stress Test): Submeter o sistema a uma carga superior ao pico de carga conhecido pela aplicação para identificar o comportamento do software.   O objetivo deste tipo de teste é avaliar o comportamento do software sob condições críticas, tais como pouca memória, pouca área em disco, pouca CPU e outras restrições, o teste de stress é realizado colocando o software sobre condições mínimas de operação. |

# Escopo

|  |
| --- |
| Detalhes do Escopo |
| Será realizado teste na(s) seguinte(s) funcionalidade(s):  Tela homepage, busca de produtos e checkout onde será criado diversas threads(usuários) sendo enviados simultaneamente em cada tela, com o objetivo de visualizar como cada uma se comportará após um grande fluxo de requisições. Ferramenta utilizada para teste Jmeter com a tecnologia java. |

# Características de Ambiente

|  |
| --- |
| Aplicação em Teste |
| A aplicação que será testada está implantada e configurada de seguinte forma:]  (Falta o gráfico) |

|  |
| --- |
| Infraestrutura para Testes |
| Será necessária 1 (uma) máquina linux disponível a cada 500 usuários concorrentes ou 1 (uma) máquina Windows disponível a cada 200 usuários concorrentes, com as seguintes configurações mínimas:   * Memória RAM: 8GB; * Processamento: Dual core 1.6GHz * Rede: 1 Gbit/s * Espaço em Disco: 5GB; * Disco rígido de 5.400 RPM * Placa de vídeo compatível com DirectX 9 para resolução de tela de 1024 x 768 ou superior |

|  |
| --- |
| Relatório |
| Quantidade de processos, horários, datas e maquinas:   * Quantidade Total de Usuários Virtuais: * Quantidade de Scripts Executados: * Data e Horas de Início: * Data e Hora de Término: * Duração do Teste: * Quantidade de Máquinas Injetoras: |

# Objetivos dos Testes

|  |
| --- |
| Detalhes dos Testes |
| Os testes terão como objetivo, elucidar as seguintes questões:   * Qual a concorrência de usuários suportada sem que o sistema apresente erros? * Nesta carga, qual a mediana e o tempo máximo de resposta por tipo de requisição? * Nesta carga, quais pontos de gargalo são apresentados? * Qual a concorrência de usuários suportada sem que o sistema apresente mediana superior a 05 segundos? * O sistema suportará a carga de pico prevista (6.500 usuários concorrentes) |

# Recursos

|  |
| --- |
| Detalhes dos Recursos |
| Para realização deste plano, serão necessários os seguintes recursos humanos:   * **Especialista em Testes de Desempenho:** Profissional responsável pela criação e gestão da execução dos testes de desempenho, assim como pela consolidação dos indicadores obtidos e elaboração do relatório de testes. * **Analista de Infraestrutura:** Profissional responsável pela disponibilização do ambiente e monitoramento dos indicadores demandados pelo solicitante. * **Database Administrator:** Profissional responsável pela disponibilização, manutenção e monitoramento da(s) base(s) de dados necessárias ao correto funcionamento da aplicação. |

# Projeto de Testes – Funcionalidades a serem testadas

|  |
| --- |
| Detalhes |
| 1. Na primeira bateria de testes haverá adição gradual de 1 usuário por segundo até o limite de 30 usuários durante 5 minutos. A cada erro, o respectivo usuário virtual será finalizado. O objetivo dessa execução é obter a quantidade de usuários concorrentes que o sistema suporta sem apresentar falhas. 2. Na segunda bateria de testes será utilizada a carga obtida na execução anterior durante 10 minutos, com o objetivo de identificar a mediana e tempo máximo de resposta por tipo de requisição, assim como identificar os gargalos visíveis. 3. Na terceira bateria de testes será aplicada carga de 20 usuários concorrentes durante 5 minutos com o objetivo de coletar os tempos de resposta, os percentuais e tipos de erro apresentados. 4. Serão executadas baterias de teste reduzindo gradativamente em 10% a quantidade de usuários concorrentes até ser obtida concorrência onde a mediana de cada uma das requisições é menor ou igual a 10 segundos e o percentual de erros é inferior a 1%. |

# Critérios de Aceite

|  |
| --- |
| Critérios de Conclusão |
|  |

# Premissas

|  |
| --- |
| Detalhes das Premissas |
| * O(s) ambiente(s) foco dos testes deverá(ão) estar disponível(eis); * Caso seja disponibilizada nova versão da aplicação em teste, os scripts deverão ser criados novamente; * O cronograma pode ser alterado devido a alterações de escopo, disponibilização de novas versões, solicitação de novas baterias de teste, entre outras possibilidades. * O sistema que será testado deverá estar implantado e completamente operacional; * Deverão ser disponibilizadas janelas para a execução das baterias de teste, onde a infraestrutura esteja sendo utilizada de forma reduzida por usuários reais. * Disponibilização de recursos de nuvem (máquinas, navegadores, acesso à internet) ou local com diversas maquinas. * Cada área responsável deverá prover de monitoramento de seu ambiente de infraestrutura (servidores, banco de dados, web services) de acordo com as necessidades, utilizando-se ou não de ferramentas específicas. O principal ponto de monitoração de um teste de performance é o tempo de resposta em função da quantidade de usuários simultâneos. O relatório de testes deve ser preenchido não somente pelo analista responsável pelos testes, mas também pelos envolvidos na execução. Os responsáveis pela infraestrutura, por exemplo, devem relatar o comportamento dos servidores de aplicação e de banco de dados durante e após a execução das baterias de testes. Posteriormente essas métricas armazenadas e cruzadas entre si permitem uma análise em detalhe de como o sistema se comporta, de modo que é possível detectar gargalos por meio dessas métricas. A ferramenta utilizada pela área de teste será o JMeter, sendo o resultado do teste passado pela mesma para as diversas áreas. * A infraestrutura deverá apresentar informações referentes ao ambiente (servidores de aplicação, banco de dados, recursos de rede, balanceador de carga, arquitetura e etc.), expectativas sobre os critérios para aceitação do produto. * Se caso o teste for executado em nuvem, a aplicação alvo de teste deve estar disponível para acesso, pois as máquinas não estarão em uma rede local (LAN) e sim na rede internet (WAN). * Dispositivo ou software de negação de serviço desabilitado (DOS Denial of Service) e DDOS (Distributed Denial of Service) para que seja possível simular os acessos ou requisições sem bloqueios das maquinas de carga. |

# Valores para Servidores remotos

Microsoft Azure: 24 maquinas equivalentes: 12.000 usuários.

* **1 dia:** R$341,28/MÊS
* **2 dias:** R$682,56/MÊS
* **3 dias:** R$1.023,84/MÊS
* **4 dias:** R$1.365,12/MÊS
* **5 dias:** R$1.706,40/MÊS
* **6 dias:** R$2.047,68/MÊS
* **7 dias:** R$2.388,96/MÊS

https://azure.microsoft.com/pt-br/pricing/calculator/

Blazemeter: 24 maquinas equivalentes: 12.000 usuários.

Dólar: 1.855.445.2285 = Real 5.9163

<https://www.blazemeter.com/pricing>

# Glossário

**Ramp-Up(Tempo de inicialização):** Tempo em segundos para inicializar cada usuário exemplo: 5 usuários em 10 segundos(Ramp-Up), 10/5 =2, um usuário a cada 2 segundos.

**Think Time:** Tempo de espera entre ações. O “think time” é muito útil para simular o tempo que um usuário real leva entre uma ação e outra, incluindo o tempo de preenchimento de um formulário, por exemplo.

**Identificador:** Responsáveis por suas devidas áreas.

**Median(mediana):** Quando 50% das requisições atingiram aquele número, ignora-se outros 50% com diversos valores e aceita apenas o resultado dos 50% iguais.

**Gargalos(Bottenecks):** Quando o sistema sofre um pico significativo naquele momento em que a requisição está operando, exemplo: No momento de rodar o teste percebe-se que ao inserir os dados do cliente na tela de checkout, a aplicação no gráfico do jmeter apresenta um pico significativo, sendo como tarefa do analista de teste, informar o pessoal do database para consultar qual a falha que poderia ser, como possíveis razões ele poderia ser desde tabela não seguidas com padrões, querys mal escritas e etc.

**Avg. RESPONSE TIME:** é a quantidade de tempo que o sistema leva para processar uma solicitação depois de ter recebido uma. Este parâmetro não inclui o tempo que ele precisava para transmitir HTTP / s pedidos sobre a rede para web-servidor em teste. Ele é medido por ms(milissegundo).

**VU(Thread):** Usuário online.

**Avg. Bandwidth:** velocidade da banda em Kilobits.

**Media(average):** O tempo médio de resposta é a soma de todos os tempos de resposta divididos pelo número de transações avaliadas. O tempo médio é um indicador importante, pois ele agrega os tempos de resposta e facilita a identificação de páginas ou APIs mais demoradas, mas ele não é e nunca deve ser visto como o critério definitivo de aceite, pois ele é um número muito maleável e pode facilmente enganar.

**Asserções:** verifica no teste se aquele elemento(tag) especificado está na homepage.

**Vazão(Througput):** é a quantidade de requests que o sistema consegue atender em um segundo (ou minutos em alguns casos). Esse número está intimamente atrelado ao tempo médio de resposta. Quanto mais rápido o meu sistema responder, maior será o meu throughput.

**Initial Delay, Sec:** Tempo que deve aguardar o início da primeira thread.

**Startup time:** Mesmo que ramp-up.

**Hold Load for, sec:** Tempo para desligar as threads.

**Shutdown Time:** Tempo para desligar as Threads exemplo: 25 threads + 25 shutdowns time = 1 thread sendo desligada por Segundo.