



CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS

Recredenciado pela Portaria Ministerial nº 1.162, de 13/10/16, D.O.U. nº 198, de 14/10/2016
AELBRA EDUCAÇÃO SUPERIOR - GRADUAÇÃO E PÓS-GRADUAÇÃO S.A.

João Paulo Santos Maciel

REALIZAÇÃO DE TESTE DE STRESS NO SISTEMA SIG EXTENSÃO DO
CEULP/ULBRA

Palmas – TO

2020

João Paulo Santos Maciel
REALIZAÇÃO DE TESTE DE STRESS NO SISTEMA SIG EXTENSÃO DO
CEULP/ULBRA

Trabalho de Conclusão de Curso (TCC) II elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. Esp. Fábio Castro Araújo.

Palmas – TO

2020

João Paulo Santos Maciel
REALIZAÇÃO DE TESTE DE STRESS NO SISTEMA SIG EXTENSÃO DO
CEULP/ULBRA

Trabalho de Conclusão de Curso (TCC) II elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. Esp. Fábio Castro Araújo.

Aprovado em: ____/____/____

BANCA EXAMINADORA

Prof. Esp. Fábio Castro Araújo

Orientador

Centro Universitário Luterano de Palmas – CEULP

Prof. Dra. Parcilene Fernandes de Brito

Centro Universitário Luterano de Palmas – CEULP

Prof. Me. Jackson Gomes de Souza

Centro Universitário Luterano de Palmas – CEULP

Palmas – TO

2020

AGRADECIMENTOS

Em agradecimento ao meu professor e orientador Fábio Castro Araújo, pela paciência, apoio e confiança nesse trabalho. A todos os demais professores que estiveram presentes ao longo de todo esse curso e que me inspiraram e me encorajam a permanecer firme. Aos meus pais, por todo o amor e apoio, principalmente incentivando nas horas difíceis e de cansaço. E aos demais que diretamente e indiretamente fizeram parte de minha formação.

RESUMO

MACIEL, João Paulo Santos. **Realização de Teste de Stress no sistema SIG Extensão do CEULP/ULBRA**. 2020. 73 f. Trabalho de Conclusão de Curso (Graduação) – Curso de Sistemas de Informação, Centro Universitário Luterano de Palmas, Palmas/TO, 2020¹.

O objetivo deste trabalho foi a realização de testes de software, mais especificamente um teste não-funcional dedicado a performance, sendo ele o Teste de Stress. Esse teste foi aplicado sobre o sistema SIG Extensão do CEULP/ULBRA, uma ferramenta que divulga todos os eventos desenvolvido dentro da instituição. Com isso, buscou-se desenvolver um teste direcionado à capacidade de carga de usuários simultâneos suportada pelo SIG. Para tanto, foi desenvolvido um Plano de Teste, o Módulo de Teste e os Casos de Teste, além de documentos para registro do trabalho de teste e dos resultados, como o Relatório de Erros. Para fim de testar os cenários de estresse foram gerados os Casos de Teste contendo as cargas de 50, 100, 500 e 1000 usuários. Foram usadas as ferramentas JMeter e Htop para aplicação dos testes e acompanhamento do consumo de recursos por parte do sistema. Sendo observado resultados em que a aplicação testada teve baixo desempenho e alto consumo de recursos. E, com isso, essas informações foram analisadas e descritas no Relatório de Erros, que apresenta a descrição desses resultados e identifica os principais erros, densidade e tendências que ocorreram no sistema durante os testes.

¹ Elemento incluído com a finalidade de posterior publicação do resumo na internet. Sua formatação segue a norma ABNT NBR 6023, por isto o alinhamento e o espaçamento diferem do padrão do texto.

LISTA DE FIGURAS

Figura 1 - Etapas do desenvolvimento	16
Figura 2.1 - APDEX Caso de Teste 001 (50 usuários)	22
Figura 2.2 - Resumos de Solicitações - Caso de Teste 001 (50 usuários)	22
Figura 3.1 - APDEX Caso de Teste 002 (100 usuários)	23
Figura 3.2 - Resumos de Solicitações - Caso de Teste 002 (100 usuários)	24
Figura 4.1 - APDEX Caso de Teste 003 (500 usuários)	25
Figura 4.2 - Resumos de Solicitações - Caso de Teste 003 (500 usuários)	25
Figura 5.1 - Resumos de Solicitações - Caso de Teste 004 (1000 usuários)	26
Figura 5.2 - APDEX Caso de Teste 004 (1000 usuários)	27

LISTA DE TABELAS

Tabela 1 - Tabela de URL	20
Tabela 2 - Tabela de Erros	24
Tabela 3 - Tabela de Erros	26
Tabela 4 - Tabela de Erros	27

LISTA DE ABREVIATURAS E SIGLAS

Apdex - Índice de Desempenho da Aplicação

CEULP - Centro Universitário Luterano de Palmas

GB - Gigabyte

HTTP - Hypertext Transfer Protocol

KO - knockout

RAM - Random Access Memory

SIG - Sistema de Informações Gerenciais

SSD - solid-state drive

Stress - Estresse

URL - Uniform Resource Locator

ULBRA - Universidade Luterana do Brasil

WebAPP - Aplicação Web

SUMÁRIO

1 INTRODUÇÃO	9
2 REFERENCIAL TEÓRICO	11
2.1 Teste de Software	11
2.1.1 Plano de Teste	11
2.1.2 Módulos de Teste	11
2.1.3 Casos de Teste	11
2.1.4 Executar Teste	12
2.1.5 Coletar dados de Testes	12
2.1.6 Documentar os resultados dos testes	12
2.2 Testes não-funcionais	13
2.3 Teste de Stress	13
3 MATERIAIS E MÉTODOS	14
3.1 Materiais	14
3.1.1 JMeter	14
3.1.2 Htop	14
3.1.3 Apdex	15
3.1.4 Servidor	15
3.2 Métodos	16
3.2.1 Criar Plano de Testes	16
3.2.2 Gerar Módulos de Testes	16
3.2.3 Gerar Casos de Teste	17
3.2.4 Executar Trabalhos de Testes	17
3.2.5 Coleta de Dados de Teste.	18
3.2.6 Documentação de Resultados de Teste	18
4 RESULTADOS E DISCUSSÃO	19
4.1 Plano de Testes	19
4.2 Módulos de Teste	19
4.3 Casos de Teste	20
4.4 Relatório de Erros	21
4.4.1 Análise Individual dos Resultados dos Casos de Testes	22
4.4.1.1 Análise do Caso de Testes 001 (50 usuários em simultâneos)	22

4.4.1.2 Análise do Caso de Testes 002 (100 usuários em simultâneos)	23
4.4.1.3 Análise do Caso de Testes 003 (500 usuários em simultâneos)	24
4.4.1.4 Análise do Caso de Testes 004 (1000 usuários em simultâneos)	26
5 CONSIDERAÇÕES FINAIS	29
REFERÊNCIAS	31
APÊNDICES.....	32
APÊNDICE A – Plano de Teste	33
APÊNDICE B – Módulo de Teste	41
APÊNDICE C – Casos de Teste	45
APÊNDICE D – Relatório de Erros	51

1 INTRODUÇÃO

Com a popularidade e difusão do acesso à internet, vários serviços têm migrado do físico para as aplicações *web*. Desta forma, há a necessidade de deixar as aplicações com a maior disponibilidade e capazes de suportar grandes quantidades de usuários ativos ao mesmo tempo consumindo tais serviços. Uma das formas de simular essa grande quantidade de usuários acessando os serviços é através de testes de carga, volume ou estresse.

O teste de stress de software é um teste não-funcional caracterizado como de performance e tem o objetivo de buscar erros sob condições extremas de acessos simultâneos. Esse teste é ideal para verificar como o aplicativo se degrada e o seu consumo de recursos (MAXIM, 2016, p.573). Esse teste é basicamente a realização da qualificação de um produto por meio da realização de estresse.

O SIG (Sistema de Informações Gerenciais) é um software desenvolvido pela Fábrica de Software do CEULP/ULBRA, que visa integrar os diversos serviços tanto para administração quanto para a extensão. O software possui um módulo que atua no gerenciamento de eventos, que é o foco deste trabalho. O software permite o gerenciamento do evento, suas inscrições e atividades. Vários cursos utilizam o software para a realização de seus eventos de extensão.

Por se tratar de uma ferramenta que possui um alto número de acesso, e por não existir um padrão definido de teste de *stress* na fábrica de software, este sistema se tornou alvo para a aplicação do teste de *stress* que é objeto desse estudo deste trabalho. Como objetivo do trabalho foi definida a identificação de cenários de estresse suportado pela aplicação. Assim, buscou-se a aplicação de testes de carga para verificar a performance da aplicação diante de cenário de estresse. Buscou-se também acompanhar o uso e comportamentos de recursos físicos da máquina.

Nesta ferramenta são registrados os principais eventos que estão sendo realizados pela Instituição e com isso é possível encontrar um volume de acesso por parte do corpo docente, coordenadores de cursos e outros funcionários ligados a esse setor de ensino com o objetivo de cadastrar e acompanhar os eventos. Mais também é utilizado pelo corpo discente que busca por estes eventos e isso pode contribuir e gerar um grande acesso ao sistema.

Por ser uma aplicação importante para o CEULP, tornou-se necessária a realização de teste de carga e volume para identificar a capacidade do SIG com acessos simultâneos. E,

com esses números identificados, a realização do estresse da aplicação com variações em cima dos resultados dos testes citados anteriormente.

Com isso, foi possível gerar relatórios e analisar os resultados obtidos e o desempenho de carregamento das principais páginas com a aplicação do teste de *stress* no SIG Extensão do CEULP/ULBRA. Além disso, foi possível observar o comportamento do tempo de resposta do SIG Extensão com as realizações do teste e, com isso, criar relatórios com os resultados dos testes de stress para que possam ser apresentados à Fábrica de Software para a realização das devidas correções e melhorias na aplicação.

2 REFERENCIAL TEÓRICO

2.1 Testes de Software

“Os testes de software são uma das funções de controle de qualidade com um objetivo principal: encontrar erros” (MAXIM, 2016, p.450). Sendo assim, é um processo formal de avaliar todo um software ou parte dele por meios manuais ou automatizado com o uso de aplicações para verificar se ele satisfaz os requisitos especificados no desenvolvimento (IEEE, 1983, 729). Com isso, “as atividades de testes são executadas paralelamente ao desenvolvimento de *software* e sistema, não apenas na conclusão do esforço de desenvolvimento” (IEEE 2008, 610).

2.1.1 Plano de Testes

Um software deve ser testado várias vezes durante o seu desenvolvimento, com isso equipes desenvolvem planos de teste a partir de uma estratégia que envolve etapas e lógica (WU, 2009). Para isso, é necessário que seja criado um roteiro de teste do software, como também uma forma de registrar os testes anteriores para serem reutilizados ou para criar testes para ser executado (JASKIEL, 2001).

Um bom plano deve identificar quais tipos de teste melhor se aplica ao software, de forma que simule os diversos cenários que podem ser impostos em ambiente real, e isso vai além de prazo e orçamento.

2.1.2 Módulos de Testes

“O módulo de teste é um módulo que facilita a execução de tarefas de teste” (WU, 2009) e é usado para testar o desempenho de aplicações. Ainda segundo o artigo George Wu (2009), “o módulo de teste pode ser uma instanciação de um objeto gerado pela estrutura de teste a partir de informações de configuração armazenadas”.

Associado ao plano de teste, o módulo pode facilitar no momento da execução dos testes, e a partir disso servir como mecanismo de ponteiro para identificar em qual etapa ou fase do plano está sendo executado pela a equipe de teste.

2.1.3 Casos de Teste

“Um Caso de Teste mostra os caminhos percorridos por um módulo” (ENDLER, 2018). Os casos servem como base para a execução de testes manuais ou como requisito para

a automatização dos testes. Com isso é esperado que contenha os resultados esperados do comportamento do software.

Segundo Endler (2018), “os Casos de Teste devem cobrir o máximo de situações possíveis”. Ou seja, os casos devem representar o máximo de situações possíveis da vida real que o software poderá passar sendo submetido por ações dos usuários. Para isso, os testadores devem ter consciência do modelo de negócio, que tipo de público é atendido e as condições em que pode ser usado.

2.1.4 Executar Testes

Na etapa de execução, trabalhos de testes tratam de colocar em prática tudo o que foi planejado. “O trabalho de teste pode enviar comandos ou dados para um componente de software cliente já em execução, instruindo-o a fazer determinadas solicitações de um componente de software para servidor já em execução” (WU, 2009).

Nessa fase são preparados e aplicados os testes e registrados os resultados de cada teste para que possam ser comparados aos resultados esperados definidos nos casos de teste. Com isso, a execução trabalhos de testes podem ser dividida em sub etapas.

2.1.5 Coletar dados dos testes.

A coleta de dados vem logo após a finalização dos testes, quando a equipe passa a reunir todos os dados relatados ou dos relatórios e dados estatísticos gerados pelas ferramentas de automação e monitoramento usado para o teste do software. “Cada relatório de dados pode incluir dados de séries temporais, entradas de *log* baseadas em texto ou dados tabulares, juntamente com metadados para identificar, entre outras coisas, as métricas de desempenho relevantes” (WU, 2009). Com isso, com todos esses dados, passa-se para a fase de documentação onde os resultados serão analisados.

2.1.6 Documentar os resultados dos testes

Na fase de documentação, são apresentados os dados coletados na etapa de coleta de dados de teste, que podem ser desde a forma de relatórios, até estatísticas ou dados tabulares. Serão analisados e transformado em informações e colocados em um documento, podendo ser ele uma parte do documento do desenvolvimento do *software* ou uma parte específica para os testes. “Eles conferem confiabilidade ao *software*, reorientam o desenvolvimento do

design e do código, e poupam gastos desnecessários, quando detectam erros nas fases iniciais do desenvolvimento de um software” (VIANA, 2012).

“Os documentos deste padrão descrevem os planejamentos, a especificação e a geração de relatórios de testes” (BIANCHINI, 2004). Com isso, independente do formato desse documento, ele deve ser repassado para a equipe de desenvolvimento. E esse processo pode ser circular entre as equipes de Teste e de Desenvolvimento podendo se repetir até que o produto atinja a qualidade desejada.

2.2 Testes não-funcionais

Testes funcionais apenas testam as funcionalidades, embora essencial, não é suficiente (SOASTA, 2011). Para isso é necessário submeter o WebAPP a testes não-funcionais para se assegurar da segurança e performance. Esses testes são projetados para simular situações de carga do mundo real (MAXIM, 2016, p. 561). Para verificar o comportamento do sistema, como também a usabilidade, certificar a capacidade de carga e recuperação e nível de segurança e uso de recursos por parte do sistema. Com isso, é possível ter uma noção de como ele reagirá em ambiente real.

2.3 Teste de Stress

“O teste de esforço (stress) de aplicativos móveis tenta encontrar erros que vão ocorrer sob condições operacionais extremas” (MAXIM, 2016, p. 573). Apesar do conceito se referir a aplicativos móveis, ele também é aplicado a outros tipos de *software*, principalmente para verificar como essas aplicações degradam e se essa degradação não compromete a segurança.

Segundo Soasta (2011), para a realização desse teste, deve-se ter uma diversidade de ambiente para refletir com precisão os possíveis cenários de uso por parte dos usuários. Com isso, aproximando do modo de uso dos usuários finais, certificando assim que o desempenho da aplicação em condições extremas não afeta o usuário final.

3 MATERIAIS E MÉTODOS

3.1 MATERIAIS

Nesta seção são apresentadas as ferramentas utilizadas para a realização do teste de estresse neste trabalho. Foram utilizados o *Apache JMeter* para a realização de diversos tipos de requisições ao sistema e as ferramentas *Htop* para monitoração do consumo de recurso da máquina que hospeda a aplicação.

3.1.1 JMeter

O aplicativo *Apache JMeter* foi projetada pela *Apache Software Foundation* com código aberto para realização de testes e medir desempenho de aplicações Web. Com isso, a ferramenta disponibiliza vários tipos de requisições e valores modais além de diversos controladores lógicos e de *threads* para serem usados para testar o desempenho em recursos estáticos e dinâmicos.

Para isso, “ele pode ser usado para simular uma carga pesada em um servidor, grupo de servidores, rede ou objeto para testar sua força ou analisar o desempenho geral em diferentes tipos de carga” (APACHE JMETER, 2019) testando assim desde aplicativos e servidores até protocolos. A exemplo, pode ser usado para testar HTTP, HTTPS, serviços SOAP / REST, FTP, bancos de dados via JDBC, LDAP, TCP e entre outros.

3.1.2 Htop

“O Htop é um visualizador de processos gratuito baseado em *ncurses* (GPL) para Linux” (DIE.NET, 2017). Ou seja, uma ferramenta de monitoramento de processos visual e interativa que “na versão 2.0 o Htop se tornou multiplataforma, sendo suportado por Linux, FreeBSD, OpenBSD e MacOS” (Tedesco, 2017). Assim como no Gerenciador de Tarefas do Windows, a ferramenta é capaz de listar programas em execução no computador e exibir dados de desempenho e de consumo de recursos.

A ferramenta se divide em duas formas de exibição, em que o topo mostra informações do processador e uso dos recursos da máquina. E na parte de baixo exibe uma lista com várias informações, como *id* de processos, nível de prioridade, porcentagem e quantidade de recursos consumidos pelos processos dos programas abertos.

3.1.3 Apdex (Índice de Desempenho da Aplicação)

O Índice de Desempenho da Aplicação (Apdex) é uma ferramenta de informações gerenciais que serve para medir o índice de desempenho da aplicação em relação a satisfação do usuário. Esse método foi uma especificação proposta pela Apdex Alliance “o objetivo fundamental do Apdex é simplificar o relatório das medições do tempo de resposta do aplicativo, possibilitando a representação dessas medições usando uma métrica comum” (Apdex, 2010).

A realização da pontuação conforme o index é necessário medir o tempo gasto entre a requisição e a resposta, ou seja, o “tempo de resposta”. A pontuação do índice é o resultado da fórmula que envolve o tempo de resposta considerados satisfatórios, toleráveis e frustrantes podendo ser usado como tempo padrão para definir como 500 milissegundos, porem esse tempo padrão pode ser definido de acordo com a instituição ou equipe de desenvolvimento.

Para calcular o Apdex usa-se a seguinte formula (BRADY, 2009):

$$Apdex = \frac{\text{Número de resultados Satisfatorios} + \frac{\text{Número de resultados Toleraveis}}{2}}{\text{Total de Requisições}}$$

O resultado desse cálculo encontra-se dentro da escala de 0 á 1 que vai ser referente a pontuação Apdex. Essa pontuação indica uma margem de desempenho que pode representar o nível de satisfação para o usuário que pode ser de satisfatório, tolerável e frustrante. As margens desses níveis são definidas pela equipe de qualidade ou o cliente e a pontuação Apdex deve-se encontrar dentro de um desses.

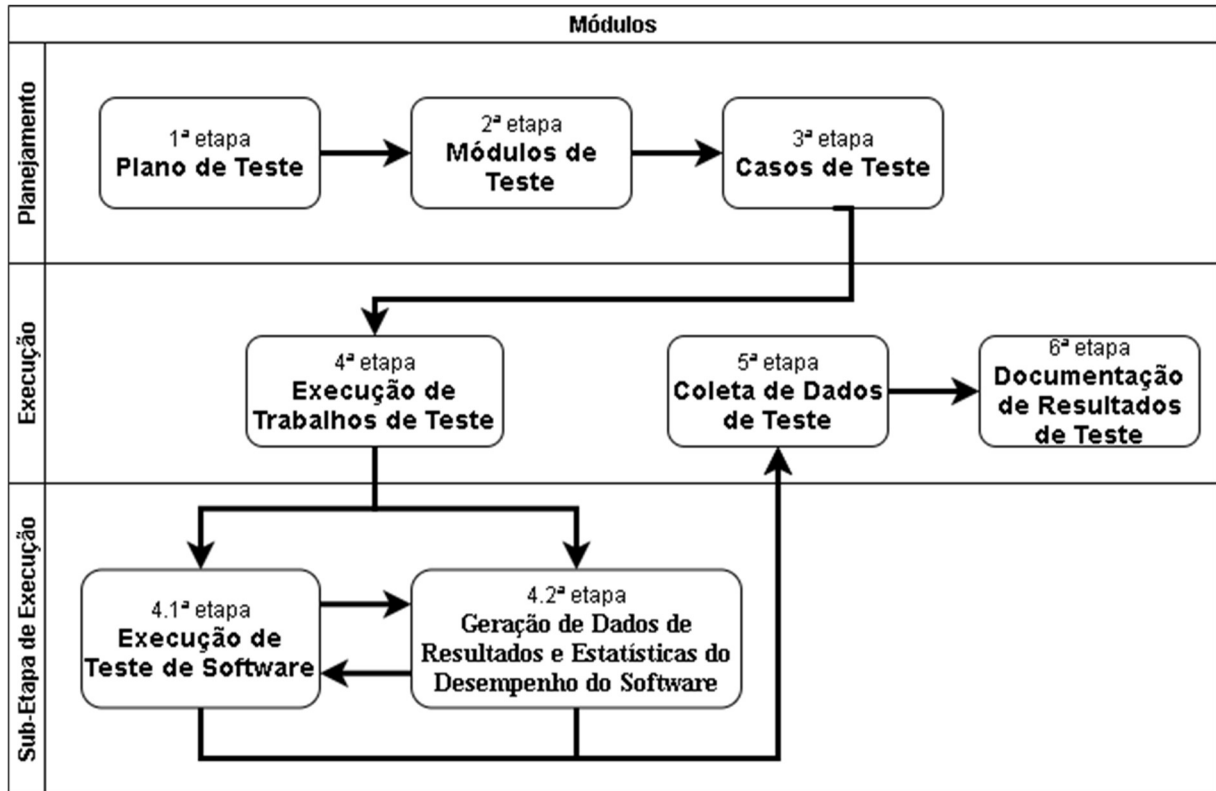
3.1.4 Servidor

Para a realização desse trabalho foi utilizado uma máquina virtual com sistema operacional Linux Ubuntu de 64 bits, com processador de 4 núcleos e com memorias de 4 GB de RAM e SSD de 80 GB.

3.2 MÉTODOS

Esta seção apresenta as etapas que foram seguidas para a realização do trabalho. A Figura 1 apresenta as fases do desenvolvimento do trabalho.

Figura 1 - Etapas do desenvolvimento.



3.2.1 Criar Plano de Testes

Para a aplicação do teste de *stress*, foi desenvolvido um plano de teste contendo um roteiro de execução, modelos de documentos e condições aceitáveis que os testes devem ocorrer. Desta forma, padronizando e organizando procedimentos e documentos de forma que alinhem e garantam que o máximo de erros possam ser encontrados.

3.2.2 Gerar Módulos de Testes

Os módulos são uma parte importante na hora do planejamento do teste para definir o modelo que foi aplicado e qual roteiro seguir. Então foram desenvolvidos módulos que são identificados como etapas selecionadas e trabalhadas de acordo com o tipo de teste aplicado. Com isso, é possível criar as etapas e realizar o refinamento delas para cada cenário essencial para a aplicação do plano de teste de stress.

E assim, para a realização deste trabalho, foram definidas nesta etapa os módulos: Plano de Teste, Módulos de Teste, Casos de Teste, Execução de Trabalho de Teste, Coleta de Dados de Teste e Documentação de Resultados de Teste. Durante a etapa de planejamento, foram definidos os modelos de documentos, o que seria feito e o detalhamento de cada etapa.

Com isso, para facilitar durante a execução do teste, o módulo de Execução de Trabalho de Teste foi dividido em duas sub-etapas: Execução de Teste no Software e Geração de Dados de Resultados do Desempenho do Software. Isso foi realizado para que fosse possível executar o teste e certificar de que ele está sendo aplicado corretamente e para um melhor acompanhamento da geração de dados do teste.

3.2.3 Gerar Casos de Teste

Os casos de testes foram definidos para a realização de *stress* no sistema. Porém, como não tem especificado a carga de acessos simultâneos suportado pelo SIG Extensão, os casos de teste seguem os volumes de cargas definidos durante o planejamento que se encontra especificado no Plano de Teste, assim como o modelo de documentação de caso de teste. Assim, buscando identificar o limite e cobrindo o máximo de cenários de estresse possível.

3.2.4 Executar Trabalhos de Testes

Com os casos de teste prontos e refinados, então inicia-se os trabalhos de testes. Sendo assim, foi preparado o JMeter com os dados e, após esse procedimento, passou-se a aplicar os testes no sistema SIG que se encontra em um ambiente controlado de produção.

A execução dos trabalhos de testes trata-se de uma tarefa complexa. Não basta executar os testes, deve-se levar em consideração que a(s) ferramenta(s) definidas no plano de teste e que serão usadas para testar possuem algumas necessidades, que pode ser de dados ou de pré-requisitos para funcionar corretamente. Por mais que algumas ferramentas executem os testes de modo automático, elas precisam ser alimentadas com dados e informações relevantes para realizar qualquer ação sobre o sistema a ser testado.

Para facilitar a execução dos trabalhos, esse processo foi dividido em duas sub-etapas, sendo elas: Execução de Teste no Software e Registro e Geração de Dados de Resultados e Estatísticas do Desempenho do Software. O objetivo desta divisão teve como propósito melhor gerenciar o que está sendo executado e os resultados que estão sendo gerados.

O objetivo da sub-etapa de execução de teste é a realização do teste sobre o sistema propriamente dito. A execução segue o roteiro determinado pelos casos de testes, alimentando a ferramenta de teste com as devidas informações necessárias para executar o estresse sobre o sistema SIG. Para isso, foi realizada a execução dos testes de carga e volume com o objetivo de identificar o ponto de estresse da aplicação e, a partir deste momento, realizar variações de carga para que o leve a trabalhar de forma extrema sendo assim aplicado o Teste de Stress.

Nesta etapa, a de Geração de Dados de Resultados e Estatísticas do Desempenho do Software, praticamente é a geração de relatórios, seja de modo manual e/ou automática pela ferramenta de teste, como também a realização de otimização com prévia análise por meio do uso de técnicas estatísticas para melhor representar alguns resultados dos testes.

3.2.5 Coleta de Dados de Teste.

A etapa de Coleta de Dados de Teste visa reunir e organizar os resultados gerados pelos testes, sendo eles de modo automático, pelas ferramentas usados durante todo o procedimento, ou manual, por meio de realização de anotações e geração de dados estatísticos. E para organizar este volume de dados foram utilizados os padrões determinados durante o Plano de Teste para classificar e filtrar os dados que fossem relevantes para o teste. E, com isso, centralizando os resultados que serão repassados para a próxima etapa.

3.2.6 Documentação de Resultados de Teste

Com os resultados reunidos e organizados na etapa de coleta, passa-se a realizar uma análise mais profunda nos resultados dos testes. A partir disso, insere informações em um modelo de documento já pré-definido durante o Plano de Teste para que este possa ser repassado para a equipe de desenvolvimento do sistema. Com isso, os documentos podem ser simples e objetivos ou serem mais bem detalhados, isso, dependendo da complexidade e nível de abrangência dos testes aplicados.

4 RESULTADOS E DISCUSSÃO

Este capítulo tem como objetivo apresentar como se deu o desenvolvimento do trabalho. Aqui é descrito os resultados de cada etapa da aplicação dos testes propostos utilizando os materiais e a metodologia descrita em capítulos anteriores.

Para fim de realização deste trabalho, foi usado a média de consumo de recursos por parte das máquinas e demais aplicações a meia de 0,7% da capacidade de processamento. As margens de pontos Apdex definidas para os níveis de frustrante, tolerável e satisfatório foram de 0 a 0,3 como frustrante, de 0,301 a 0,699 como tolerável e de 0.7 a 1 como satisfatório. E definido a limiar de tempo tolerável (T) como de 500 milissegundos e de frustrante (F) de 1500 milissegundos (ou 1 segundo e 500 milissegundos).

4.1 Plano de Testes

O objetivo do plano de teste desenvolvido foi garantir a excelência durante a execução dos testes. Para isso, foi necessário desenvolver uma estratégia que cobrisse o máximo de cenários possíveis. Nele foi definido tarefas e ferramentas e também os modelos de documentação para a transcrição dos resultados e informações dos testes. Este plano aborda apenas um tipo de teste sendo ele não-funcionais, mais especificamente os testes de *stress*.

Com a estratégia descrita na documentação, passa para a etapa de descrição de Recursos onde no plano de testes descrevemos os recursos humanos, ou seja, dados da equipe e a especialidade e responsabilidade de cada um no projeto. E os recursos de ferramentas, onde foram listadas e descritas as características, funcionalidades, e versões destas no plano de testes.

Por fim, os Produtos de Trabalho onde vão ser documentado todo o processo de relatórios registros de dados dos resultados e que contêm a análise dos resultados de testes.

4.2 Módulos de Teste

Na modulação é onde ocorreu a divisão da realização dos testes em etapas. Porém, para facilitar quando se alcançar a etapa executar trabalho de testes, ela foi subdividida em duas sub etapas sendo elas: Executar Testes e Registro e Estatísticas dos Resultados. Estas sub etapas ocorrem de modo cíclico, facilitando para o testador identificar em qual etapa ele se encontra, a exemplo: saber se é para executar algum teste ou se já foram realizados para que possa ser coletado os resultados.

4.3 Casos de Teste

Nesta etapa, buscou-se definir quais as situações que devem ser testadas no software. Quais os cenários que o teste de estresse deveria simular. Com isso, os casos de testes foram gerados para saber os caminhos a serem percorridos durante a execução dos testes e com isso cobrindo situações com 50, 100, 500 e 1000 usuários em acessos simultâneos nas páginas descritas pela tabela de URLs.

Tabela de URLs	
URL	Detalhes
/sig/extensao/login	Na página de Login, a autenticação é necessária para a que possa ter acesso aos demais funções e URLs do SIG Extensão.
/sig/extensao/home	Na página Inicial, pode ser verificado a lista de eventos que o usuário está inscrito e outra lista de demais ações de extensão.
/sig/extensao/inscricoes	Na página Minhas Inscrições, é possível acompanhar e ver com mais detalhes os Eventos de Ação de Extensão que o usuário está inscrito.
/sig/extensao/certificados	Na página Meus Certificados, encontra-se listado e disponível para baixar todos os certificados dos eventos que o usuário participou.
/sig/extensao/acoes	Na página de Ações de Extensão, encontra-se todas os eventos de Ações de Extensão realizado pela instituição podendo analisar a data de cada um.
/sig/extensao/acoes/471/detalhes/	Ao acessar qualquer evento, passa para a página de detalhes, que apresenta todas as informações detalhadas e dá acesso à URL do site onde o usuário poderá realizar a inscrição do evento.

Nessa tabela, estão listados todas as URLs de cada página que foi testado, os acessos simultâneos pelo trabalho, e seus detalhes, descrevendo o que cada uma possui e o que pode ser usufruído pelos usuários. Com isso, representando a navegação dentro do sistema, SIG Extensão, seguindo a ordem a qual foi desenvolvido e executado todos os casos de teste.

4.4 Relatório de Erros

Todos os casos de testes definidos durante a fase de planejamento e contidos nos documentos de Casos de Testes e foram realizados conforme o previsto no Plano de Teste e com isso os seguindo a ordem conforme no Módulos de Teste. E assim, registrando e classificando os defeitos e para que se possa ser repassado para a equipe de desenvolvimento.

Quase todos os testes foram concluídos, porém o sistema SIG Extensão não suportou a carga do Caso de Testes 004, de mil usuários simultâneos, ocasionando na queda da aplicação. E por meio da análise dos defeitos, sendo identificado os mais recorrentes em quase todos os casos testados os erros *500/Internal Server Error* e *400/Bad Request*, além de um alto consumo de recursos de processamento de dados. Podendo ser classificado como mediano para crítico uma vez que a finalidade do SIG é atender uma instituição de ensino que é o CEULP/ULBRA.

Para a análise dos resultados dos testes com os casos com carga de 100, 500, 1000 usuários foram considerados os relatórios de resultados gerados pela ferramenta JMeter e anotações relacionados ao acompanhamento do ambiente de teste por meio do Htop. Para qualificação foi considerada a pontuação Apdex (Índice de Desempenho da Aplicação), presentes nos relatórios do JMeter, que é um método que mede a satisfação que um usuário referente ao cenário de teste. Esse método considera que a pontuação de Apdex devem estar entre 1, considerado satisfatório, 0,5, é tolerável, e 0, frustrante.

Com isso, foi notado que em apenas um dos casos de testes aplicado o resultado Apdex ficou ligeiramente acima da pontuação tolerável com os demais ficando abaixo. Sendo que na carga de 1000 usuários teve a pior índice, de modo que mediu quase zero. Além disso, foi a carga que proporcionou a falha da aplicação no ambiente proposto, possuindo a maior variação de erros dentre os testes.

Em relação a erros, o mais presente em quase todos os casos de testes foi o erro *500/Internal Server Error* que indica a ocorrência de mau funcionamento do SIG Extensão, podendo se tratar de um componente ou função que possa está gerando os erro com relação ao volume de carga de requisição usado nos testes de estresse. É um problema comum em todos os cenários de testes foi o alto consumo de recurso, principalmente de processamento de dados, sendo que o consumo de memória RAM se manteve estabilizado com pequenas variações, porém o SIG chega a consumir quase 1GB deste recurso.

4.4.1 Análise Individual dos Resultados dos Casos de Testes

4.4.1.1 Análise do Caso de Testes 001 (50 usuários em simultâneos)

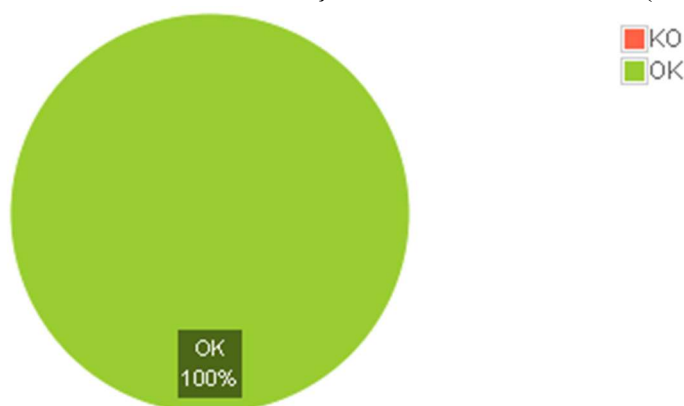
O SIG com uma carga de estresse de 50 usuários em solicitação simultâneos manteve estável e sem recorrência de erro. Com o primeiro caso de teste sobre o SIG tendo um tempo médio de resposta de um minuto e os índice de desempenho da aplicação (Apdex) com média de 0.594, sendo considerado como tolerável. Conforme representado junto com as requisições na Figura 2.1

Figura 2.1 - APDEX Caso de Teste 001 (50 usuários)

Apdex ^	T (Toleration threshold) ◆	F (Frustration threshold) ◆	Label ◆
0.594	500 ms	1 sec 500 ms	Total
0.267	500 ms	1 sec 500 ms	Requisição HTTP-0
0.508	500 ms	1 sec 500 ms	Requisição HTTP
0.858	500 ms	1 sec 500 ms	HTTP Request
0.872	500 ms	1 sec 500 ms	Requisição HTTP-1

Com isso, pode ser obtido o seguinte gráfico: Figura 2.2, sendo KO (knockout) Nocaute nas solicitações (Erros) e OK para as Requisições bem sucedidas.

Figura 2.2 - Resumos de Solicitações - Caso de Teste 001 (50 usuários)



Com isso pode ser observado que a carga de estresse de 50 usuários simultâneos, não compromete o SIG Extensão. Mais essa carga já apresenta uma significativa quantidade de

consumo significativo de recursos da máquina. Sendo que o sistema chegou a consumir quase 1GB de RAM e usando em média 87% da capacidade de processamento do processador durante os momentos de pico do teste referente às solicitações.

4.4.1.2 Análise do Caso de Testes 002 (100 usuários em simultâneos)

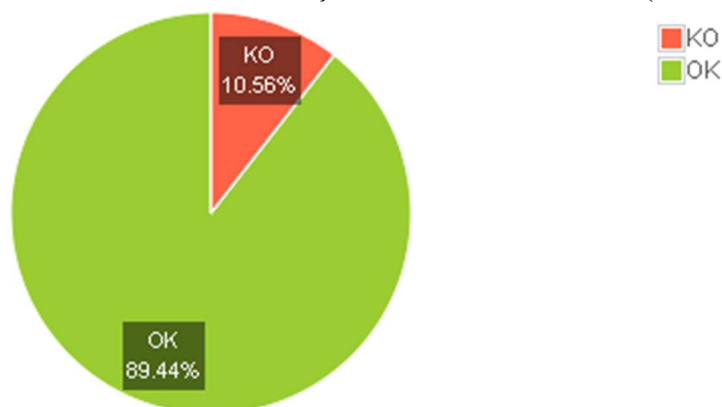
Ao aplicar uma carga de solicitações de 100 usuários em simultâneos o sistema SIG já apresentou um pequeno aumento do consumo de recursos em comparação ao teste com carga de 50 usuários, e com mais de 10% de erros nas requisições. O tempo de duração do teste foi de aproximadamente 2 minutos e com índice de desempenho da aplicação (apdex) com média de 0.334, ou seja, menor que com a carga do Caso de Teste 001. Conforme representado junto com as requisições na Figura 3.1.

Figura 3.1 - APDEX Caso de Teste 002 (100 usuários)

Apdex ▲	T (Toleration threshold) ◆	F (Frustration threshold) ◆	Label ◆
0.334	500 ms	1 sec 500 ms	Total
0.130	500 ms	1 sec 500 ms	Requisição HTTP-0
0.203	500 ms	1 sec 500 ms	Requisição HTTP
0.480	500 ms	1 sec 500 ms	HTTP Request
0.645	500 ms	1 sec 500 ms	Requisição HTTP-1
0.750	500 ms	1 sec 500 ms	HTTP Request-1
1.000	500 ms	1 sec 500 ms	HTTP Request-0

Ao final do teste, foi identificado a ocorrência de 10,56 % de erros com uma taxa de sucesso de 89,44% de das requisições do teste. Representado pelo gráfico da figura 3.2, sendo KO (knockout) Nocaute nas solicitações (Erros) e OK para as Requisições bem sucedidas.

Figura 3.2 - Resumos de Solicitações - Caso de Teste 002 (100 usuários)



Sendo assim identificado o tipo, a quantidade e a porcentagem de representativa dos erros conforme a Tabela de Erros:

TABELA DE ERROS			
Tipo de Erro	Quantidade	% dos erros	% em toda a amostra
Error 500: Internal Server Error	236	86,13%	9,10%
Error 400: Bad Request	38	13,87%	1,46%

Como pode ser notado, apesar de apresentar erros o consumo de recursos chegou a ser superior a 1GB de RAM e usando em média 90% da capacidade de processamento do processador durante os momentos de pico com isso não deixando margem podendo ocasionar em lentidão.

4.4.1.3 Análise do Caso de Testes 003 (500 usuários em simultâneos)

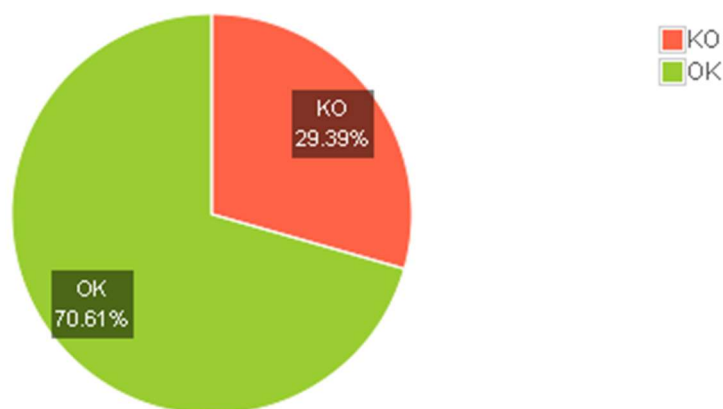
A carga de teste de 500 usuários realizando requisições em simultâneos no SIG Extensão, levou cerca de 13 minutos e com isso o número de erro e de consumo de recursos aumentou consideravelmente em comparação a carga de 100 usuários. E com quase 30% de erros em relação às requisições e com índice de desempenho da aplicação (apdex) com média de 0.228, onde pode ser considerado que é quase totalmente frustrante. Conforme representado junto com as requisições na Figura 4.1.

Figura 4.1 - APDEX Caso de Teste 003 (500 usuários)

Apdex [▲]	T (Toleration threshold) [◆]	F (Frustration threshold) [◆]	Label [◆]
0.228	500 ms	1 sec 500 ms	Total
0.056	500 ms	1 sec 500 ms	Requisição HTTP
0.263	500 ms	1 sec 500 ms	Requisição HTTP-0
0.358	500 ms	1 sec 500 ms	HTTP Request-1
0.384	500 ms	1 sec 500 ms	HTTP Request
0.605	500 ms	1 sec 500 ms	Requisição HTTP-1
0.736	500 ms	1 sec 500 ms	HTTP Request-0

Ao final do teste, foi identificado a ocorrência de 29,39% de erros com uma taxa de sucesso de apenas 70,61% das requisições do teste. Representado pelo gráfico da figura 4.2, sendo KO (knockout) Nocaute nas solicitações (Erros) e OK para as Requisições bem sucedidas.

Figura 4.2 - Resumos de Solicitações - Caso de Teste 003 (500 usuários)



Como pode ser visto, a carga de teste aplicada sobre o SIG Extensão apresentou novos erros mais com o erro *500/Internal Server Error* sendo o que mais ocorre. Outro fator identificado foi o consumo de recursos, que pode ser considerado um problema recorrente uma vez que consumo por parte da aplicação foi aproximadamente a 1GB de RAM mas ainda se manteve estável em comparação aos casos de testes anteriores e usando em média 93,8% da capacidade do processador durante os momentos de pico. Com isso gerando uma alta concorrência pelo processamento refletindo em lentidão na aplicação.

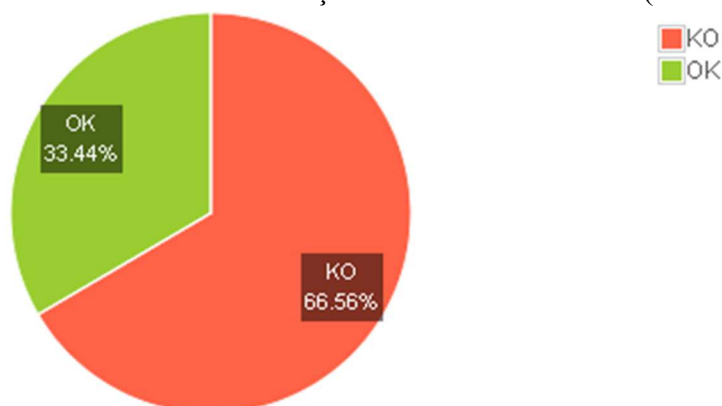
Sendo assim identificado o tipo, a quantidade e a porcentagem de representativa dos erros conforme a Tabela de Erros:

TABELA DE ERROS			
Tipo de Erro	Quantidade	% dos erros	% em toda a amostra
Error 500: Internal Server Error	2026	61,51%	18,08%
Error 403: Forbidden	573	17,40%	5,11%
Non HTTP response code: java.net.SocketException/Non HTTP response message: Connection reset	336	10,20%	3,00%
Non HTTP response code: org.apache.http.conn.HttpHostConnectException/N on HTTP response message: Connect to 127.0.0.1:8000 [V127.0.0.1] failed: Tempo esgotado para conexão (Connection timed out)	239	7,26%	2,13%
Error 400: Bad Request	120	3,64%	1,07%

4.4.1.4 Análise do Caso de Testes 004 (1000 usuários em simultâneos)

A carga de teste de 1000 usuários realizando requisições em simultâneos no SIG Extensão ocasionou no aumento do volume e aparição de novos erros. Além da carga provocar uma sobrecarga tanto do sistema como dos recursos o que ocasionou em um colapso derrubando a aplicação. Com isso, na execução do caso de teste, proporcionou 66,56% de *KO* (*knockout*) de requisições, sendo a maior taxa entre os casos de testes aplicados neste trabalho. Conforme apresentado pela figura 5.1.

Figura 5.1 - Resumos de Solicitações - Caso de Teste 004 (1000 usuários)



E ainda, foi verificado que o sistema teve sua pior pontuação Apdex sendo de 0,061, praticamente próximo de 0 (zero), onde, no método Apdex é considerado frustrante para os usuários. Representado pela figura 5.2.

Figura 5.2 - APDEX Caso de Teste 004 (1000 usuários)

Apdex	T (Toleration threshold)	F (Frustration threshold)	Label
0.061	500 ms	1 sec 500 ms	Total
0.012	500 ms	1 sec 500 ms	Requisição HTTP
0.034	500 ms	1 sec 500 ms	HTTP Request
0.161	500 ms	1 sec 500 ms	Requisição HTTP-0
0.326	500 ms	1 sec 500 ms	HTTP Request-1
0.356	500 ms	1 sec 500 ms	Requisição HTTP-1
0.457	500 ms	1 sec 500 ms	HTTP Request-0

Com isso, foi possível identificar uma alteração considerável, onde neste caso de teste, a qual o erro mais recorrente sendo de conexão recusada (*Connection refused*) seguido de erros relacionados a falhas de resposta, conexão e entre outros. Como pode ser identificado o tipo, a quantidade e a porcentagem de representativa dos erros conforme a Tabela de Erros:

TABELA DE ERROS			
Tipo de Erro	Quantidade	% dos erros	% em toda a amostra
Non HTTP response code: org.apache.http.conn.HttpHostConnectException/ Non HTTP response message: Connect to 127.0.0.1:8000 [V127.0.0.1] failed: Conexão recusada (Connection refused)	10167	77,01%	51,26%
Non HTTP response code: org.apache.http.NoHttpResponseException/Non HTTP response message: 127.0.0.1:8000 failed to respond	1013	7,67%	5,11%

Error 403: Forbidden	687	5,20%	3,46%
Non HTTP response code: java.net.SocketException/Non HTTP response message: Connection reset	604	4,58%	3,04%
500/Internal Server Error	539	4,08%	2,72%
Non HTTP response code: org.apache.http.conn.HttpHostConnectException/ Non HTTP response message: Connect to 127.0.0.1:8000 [V127.0.0.1] failed: Tempo esgotado para conexão (Connection timed out)	136	1,03%	0,69%
400/Bad Request	55	0,42%	0,28%
Non HTTP response code: java.net.SocketException/Non HTTP response message: Conexão fechada pela outra ponta (connect failed)	1	0,01%	0,01%

Outro problema recorrente, apesar de manter padrão o consumo aproximado de 1GB de RAM, é o alto consumo de recursos de processamento de dados durante os momentos de pico, onde, a execução do caso de teste chegou a usar 99,8% da capacidade de processamento do reservado para o ambiente de teste.

A densidade dos defeitos identificados durante a execução dos Testes de Stress sobre o sistema SIG Extensão por meio de níveis de gravidade, de crítica, alta, média e baixa e método Apdex, de satisfatório a frustrante conforme já citado na seção 4. O **desempenho do sistema**, seguindo o método **Apdex**, chegou a ficar um pouco acima do Tolerável, porém os demais casos tendiam a Frustrante.

O **Nível de Gravidade** dos erros, que passaram a aparecer a partir do Caso de Teste 002 com carga de 100 usuários em simultâneos, sendo considerados entre **Médio** a **Grave** uma vez que corresponde de 10,56% a 66,56% dos erros gerados pelas cargas de estresse aplicadas ao sistema chegando a ocorrer a queda da aplicação.

5 CONSIDERAÇÕES FINAIS

O desenvolvimento de testes de stress torna-se importante para certificar que a ferramenta garante o máximo de disponibilidade possível, com o desempenho e a segurança que o modelo de negócio exige. Esse tipo de teste pode simular condições reais de operação e desta forma apresentar erros antes da disponibilização ou para melhorias de atualização.

No caso do SIG Extensão, o teste de estresse teve o objetivo de identificar os cenários de estresse suportado com a hipótese de aplicação de carga e volume para verificar a performance e o consumo de recurso por parte da aplicação. Com isso, foi desenvolvido um Plano de Teste a qual fosse objetivo e de simples leitura, além do Módulo de Teste que divide o trabalho de teste em etapas e auxilia na orientação da equipe de teste. Para a execução de teste e identificação dos cenários, foram gerado os Casos de Teste contendo as cargas de 50, 100, 500 e 1000 usuários em acessos simultâneos.

Para a realização dos testes com os cenários destacados pelos casos de testes, foram usadas as ferramentas JMeter e Htop, a primeira para aplicar os testes e a segunda para monitorar os recursos de *hardware*. Para fim de medir o desempenho do sistema, foi usado no Relatório de Teste o método de Apdex, ou Índice de Desempenho da Aplicação, também foi realizado a identificação dos erros e a avaliação do consumo de recursos.

Com isso, todos os resultados foram analisados, por exemplo, em relação a erros, em quase todos os casos foi verificado o erro *500/Internal Server Error*, que indica a ocorrência de mau funcionamento do SIG Extensão, tendo também baixos resultados de desempenho com Apdex entre Tolerável e Frustrante. Outro problema recorrente em todos os cenários de testes foi o alto consumo de recurso, destacando-se o de processamento, podendo chegar a mais de 99% dependendo do caso de teste, consumindo 1 GB de memória RAM.

O teste de stress pode ser simples ou complexo, mas isso pode depender do modelo de negócio e recursos disponíveis para a aplicação. No caso do SIG Extensão, foi utilizado um teste simples em que o objetivo era analisar o seu comportamento, uma vez que se trata de uma ferramenta de eventos de uma instituição de ensino e, portanto, deve suportar uma grande quantidade de usuários simultâneos.

A definição desse tipo de teste é bem genérica, uma vez que simplesmente diz que se trata da provocação de estresse do sistema. Isso contribui para uma grande variação de formas de aplicação do teste de stress de software, sendo que esse estresse pode ser de modo a testar a segurança contra ataques, desempenho com aplicação de carga e volume, consumo de recursos que pode ser de *hardware* ou a junção de todos os citados. Assim, é possível

desenvolver testes de estresse de diversas complexibilidades, o que pode gerar uma dificuldade na hora de decidir aplicar esse teste.

Por ser um teste que possui uma grande gama de variação, é bem explorado comercialmente, porém é pouco explorado na área de trabalhos científicos diretamente relacionado ao Teste de Stress para software. Apesar disso, é possível encontrar uma boa quantidade de discussões relacionadas a esse tipo de teste. Uma delas está ligado à nomenclatura de Teste de Esforço onde há os que consideram como outro tipo teste ou como uma parte de teste de estresse, enquanto outros que afirmam que se trata do Teste de Stress com outro nome.

REFERÊNCIAS

- “Padrões de desenvolvimento: processos e tecnologias”, Fábrica de Software CEULP/ULBRA. Palmas, 2019.
- PRESSMAN, Roger S. e MAXIM, Bruce R. **Engenharia de Software: uma abordagem profissional**. 8. ed. São Paulo AMGH, 2016.
- IEEE. **829- IEEE Standard for Software and System Test Documentation**. Nova York, 2008.
- IEEE. **729 - IEEE Standard Glossary of Software Engineering Terminology**. Nova York, 1983.
- IEEE. **Std 610.12-1990 - IEEE Standard Glossary of Software Engineering Terminology**. Nova York, 1990.
- IEEE. **IEEE Standard for Software and System Test Documentation**. Nova York, 2008.
- SOASTA. **Five Strategies for Performance Testing Mobile Applications**. Mountain, 2011.
- APACHE, Apache JMeter, JMeter, the Apache feather, and the Apache JMeter logo are trademarks of the Apache Software Foundation. **Apache JMeter**. Delaware, 1999 - 2019. Disponível em: < <https://jmeter.apache.org/>> Acesso em: 18 mar. 2020.
- HP, HP Development Company, **Suporte ao Cliente HP - Base de Conhecimento**. Brasil, 2020. Disponível em: <<https://support.hp.com/br-pt/product/hp-450-notebook-pc/5224999/document/c03692720>> Acesso em: 09 abr. 2020.
- GEORGE WU, Sunnyvale, **Execuring Software Performace Test Jobs In a Cloustered System**. Estados Unidos, 2009.
- VIANA, Phil. **Criação e Geração de Planos de Teste de Software**. China, 2012.
- BIANCHINI, Juliano. **Testes de Funcional de Software a Partir de Diagramas de Casos de Uso**. Blumenau, 2004.
- APIDEX, Apidex Alliance. **Core Apdex Qualities**. Maio de 2010. Disponível em <<https://www.apdex.org/index.php/2010/05/core-apdex-qualities/>> Acesso em: 11 Jul. 2020.
- BRADY, James F. **The Apdex Index Vs Traditional Management Information Decision Tools**. Nevada, 2009.

APÊNDICES

APÊNDICE A – Plano de Teste

(Sistemas de Informações Gerenciais)

SIG Extensão do CEULP/ULBRA

PLANO DE TESTES (Teste de Stress)

Histórico de Revisão

Data	Versão	Descrição	Autor
03/06/2020	1.0	Plano de testes de stress sobre o SIG Extensão	João Paulo S. Maciel

Sumário:

1. OBJETIVOS	03
2. ESCOPO	03
3. REFERENCIA	03
4. REQUISITOS DE TESTE	04
4.1 Teste de Stress	04
5. ESTRATÉGIA DE TESTE	05
6. RECURSOS	06
6.1 Recursos Humanos	06
6.2 Recursos de Ferramentas	07
7. PRODUTOS DE TESTE	08

1. OBJETIVOS

1.1 Objetivos Geral:

- Realizar Testes de Stress no SIG Extensão do CEULP/ULBRA para analisar os resultados e identificar aspectos e indicar alguns pontos do sistema que precise de melhorias.

1.2 Objetivos Específicos

- Observar comportamento do SIG Extensão com as realizações do teste de stress;
- Relatar do teste de stress;
- Analisar os resultados dos testes.
- Documentar o processo de Testes de Stress

2. ESCOPO

Este Plano de Teste é aplicado ao teste de stress sobre o sistema SIG Extensão realizando um estresse por meio de uma carga extrema de acesso simultâneos. O teste de stress de software, a qual se trata de um dos testes não-funcionais caracterizado como de performance, onde tem o objetivo de buscar erros sob condições extremas de acessos simultâneos. Esse teste é ideal para verificar como o aplicativo se degrada e o seu consumo de recursos, Maxim (2016, p.573). Esse teste é basicamente a realização da qualificação de um produto por meio da realização de estresse.

Como a carga suportada pelo SIG é desconhecida para a aplicação do estresse sobre a aplicação. Contará também com os testes de Carga e Volume para a identificação dos limites da aplicação e por meio desta informação a aplicação do Teste de Stress.

3. REFERÊNCIAS

1. “Padrões de desenvolvimento: processos e tecnologias”, Fábrica de Software CEULP/ULBRA. Palmas, 2019.
2. PRESSMAN, Roger S. e MAXIM, Bruce R. Engenharia de Software: uma abordagem profissional. 8. ed. São Paulo AMGH, 2016.
3. IEEE. 729 - IEEE Standard Glossary of Software Engineering Terminology. Nova York, 1983.

4. IEEE. Std 610.12-1990 - IEEE Standard Glossary of Software Engineering Terminology. Nova York, 1990.
5. IEEE. IEEE Standard for Software and System Test Documentation. Nova York, 2008.
6. SOASTA. Five Strategies for Performance Testing Mobile Applications. Mountain, 2011.
7. APACHE, Apache JMeter, JMeter, the Apache feather, and the Apache JMeter logo are trademarks of the Apache Software Foundation. Apache JMeter. Delaware, 1999 - 2019. Disponível em: < <https://jmeter.apache.org/>> Acesso em: 18 mar. 2020.
8. HP, HP Development Company, Suporte ao Cliente HP - Base de Conhecimento. Brasil, 2020. Disponível em: < <https://support.hp.com/br-pt/product/hp-450-notebook-pc/5224999/document/c03692720>> Acesso em: 09 abr. 2020.
9. GEORGE WU, Sunnyvale, Execuring Software Performace Test Jobs In a Cloustered System. Estados Unidos, 2009.

4. REQUISITOS DE TESTE

Buscou-se listar todos os itens como casos de uso e requisitos não funcionais considerados necessário para a aplicação do Teste de Stress.

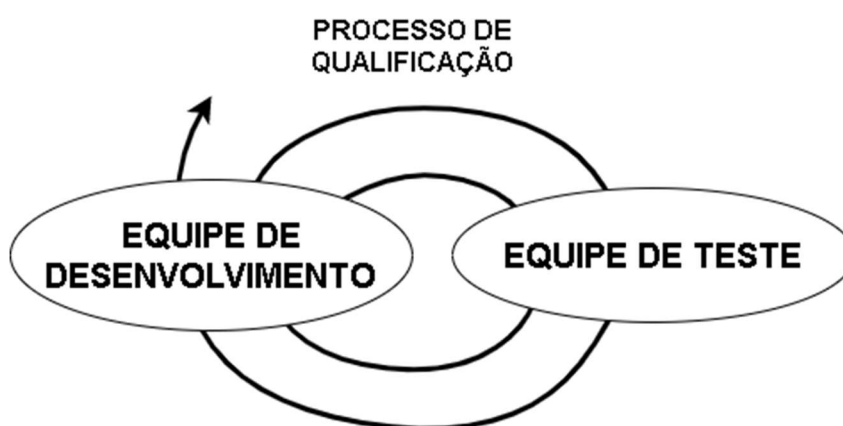
4.1 Teste de Stress

- Verificar a resposta do sistema SIG Extensão durante a aplicação da carga de estresse.
- Verificar a resposta do sistema SIG Extensão durante a aplicação da carga de estresse por tempo prolongado.
- Verificar a resposta do sistema SIG Extensão durante a aplicação de uma variação de entre a carga mínima e de estresse.
- Verificar o consumo de recursos por parte do sistema SIG Extensão.

5. ESTRATÉGIA DE TESTE

Para a aplicação do plano de testes foi pensado em conceito de uma espiral. Onde, para se atingir a qualidade desejada o Software passará pelas equipes de desenvolvimento e de teste de acordo com o que vai avançando com o processo de qualificação, figura 1. Para que a aplicação desenvolvida passe de uma equipe para a outra em um processo que segue uma sequência de testes e correções com o intuito de se atingir a melhor qualidade possível de acordo com a finalidade da aplicação.

Figura 1 - Espiral de processo de qualificação de Software



Com isso, outra vantagem desse conceito é que ele independe de estar ligado diretamente a desenvolvimento da aplicação, podendo ser empregado em produtos prontos a qual se deseja refinar, podendo selecionar os tipos de testes a serem aplicados. Outro fator é que permite que as equipes não estejam devidamente no mesmo ambiente ou em maior contato, permitindo assim a terceirização dos serviços. E isso possibilitou a realização do processo de teste sem que fosse preciso encontrar-se no mesmo ambiente de produção da Fábrica de Software do CEULP/ULBRA.

Para fim de realização do teste no SIG Extensão, foi realizado a apresentação da proposta de submeter o sistema ao Teste de Stress para a avaliação de condições extremas de cenários de pico a qual pode ocorrer em uma instituição de ensino como o CEULP. Para este fim, foi apresentado os requisitos contidos na seção 4 deste documento.

Para isso, durante o planejamento, foram identificado e gerados os módulos de teste para definir o modo que o Teste de Stress seria aplicado e qual roteiro seguir para que todos os cenários possam ser cobertos e garanta um resultado final que confirme ou aponte para melhorias que garantam a qualidade do sistema.

Com isso, sendo idealizado para este plano de teste os seguintes processo com os respectivos módulos: Plano de Teste, Módulos de Teste, Casos de Teste, Execução de Trabalho de Teste e suas sub-etapas Execução de Teste no Software e Geração de Dados de Resultados e Estatísticas do Desempenho do Software, Coleta de Dados de Teste e Documentação de Resultados de Teste. Na respectiva ordem onde acabaria sendo contida em um modelo de diagrama documentado em Módulos de Testes que será abordado mais à frente.

Sendo assim, os Módulos definidos, para documentar todo o conteúdo gerado foi definido os seguintes modelos de documentação. Sendo ele para o Plano de Testes, dividido em: Objetivo, Escopo, Referências, Requisitos de Teste, Estratégia, Recursos e Produtos. Já o Módulo de testes o documento contém o quadro com o diagrama de sequências de módulos e em seguida os detalhamentos de cada um.

A documentação de Módulo de Testes ficou dividida em diagrama e a explicação sintetizada de cada módulo. E para os Casos de Teste foi dividido de acordo com os cenários de teste sendo eles: 50, 100, 500 e 1000 usuários em acesso simultâneos.

Com tudo, para fim de conter os resultados e análise dos testes foi pensado de forma que o documento fosse objetivo e simples facilitando a leitura dos resultados pela equipe de teste. Sendo assim o, documento da seção de Documentação de Resultados de Testes, Relatório de Teste sendo dividido em: Objetivos, Escopo, Referências, Introdução aos Teste, Cobertura do Teste, Análise de Defeitos e Ações Sugeridas.

6.RECURSOS

Nesta seção buscou-se listar os recursos necessários para a realização dos trabalhos de teste no sistema SIG Extensão, identificado assim as suas responsabilidade ou configurações e recursos.

6.1 Recursos Humanos

Esta seção apresenta os recursos humanos usados para a realização dos trabalhos de Testes de Stress.

Equipe e responsabilidades:		
Trabalhador	Recurso mínimo	Responsabilidade/Observações
Gerente de Teste	1- Fabio Castro	Gerenciamento do Desenvolvimento de Testes e administração da execução dos testes.
Designer/Engenheiro	1- João Paulo Maciel	Desenvolvimento do Plano de Testes. Arquitetação de módulos de testes. Identificação de Ferramentas. Realização de Trabalhos de Testes. Análise de Resultados de testes e documentação.
Implementador		
Administrador		
Testador		

6.2 Recursos de Ferramentas

Nesta seção buscou-se listar os recursos de hardware e software necessários para a realização dos trabalhos de teste no sistema SIG Extensão, identificado assim as configurações e recursos necessário para a devida avaliação.

Recursos de Ferramentas	
Recurso	Tipo\Versão\Configuração
Servidor	Máquina com Linux Ubuntu 64 bit com processador 4 núcleos, 4 GB de RAM e SSD de 80 GB.
Apache JMeter	Aplicativo de código aberto para realização de testes e medir desempenho de aplicações Web. É usado para testar HTTP, HTTPS, serviços SOAP / REST, FTP, bancos de dados via JDBC, LDAP, TCP e entre outros. Desenvolvido pela: <i>Apache Software Foundation</i> Versão: 5.3

Htop	<p>O Htop é uma ferramenta de monitoramento de processos visual e interativa que “na versão 2.0 o Htop se tornou multiplataforma, sendo suportado por Linux, FreeBSD, OpenBSD e MacOS” (Tedesco, 2017). É capaz de listar programas em execução no computador e exibir dados de desempenho e de consumo de recursos.</p> <p>Desenvolvido pela: --</p> <p>Versão:2.0</p>
------	---

7.PRODUTOS DE TRABALHO

- Plano de Pestes
- Módulos de Teste
- Casos de Testes
- Resultados de Testes
 - Relatórios da Ferramenta
- Relatório de Resultados de Testes

APÊNDICE B – Módulo de Teste

(Sistemas de Informações Gerenciais)

SIG Extensão do CEULP/ULBRA

MÓDULO DE TESTES

(Teste de Stress)

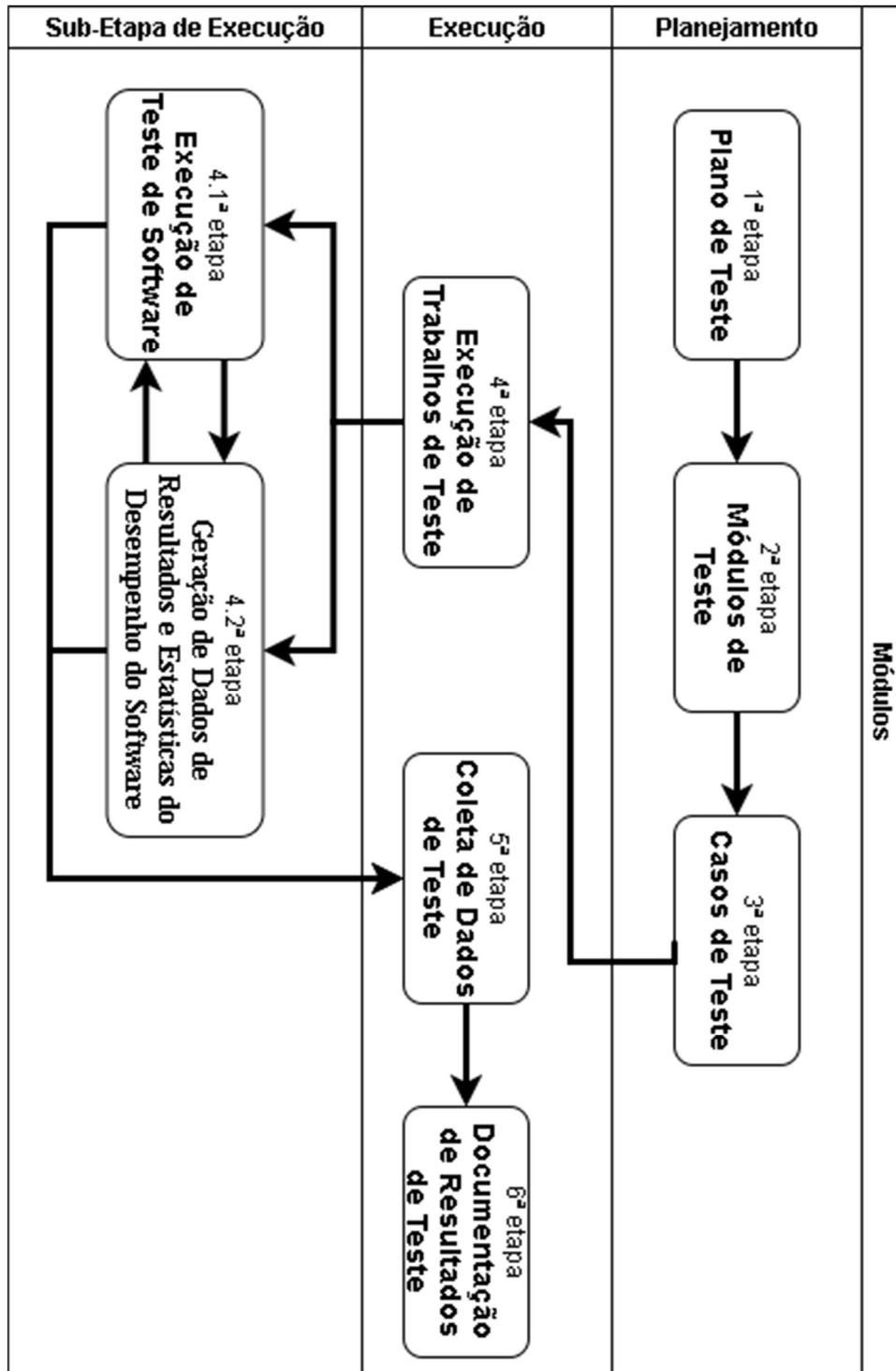
Histórico de Revisão

Data	Versão	Descrição	Autor
03/06/2020	1.2	Módulos de Teste de Stress que será aplicado sobre os Trabalhos de Testes do SIG Extensão.	João Paulo S. Maciel

Palmas - TO

2020

Para a realização deste trabalho, foram definidas nesta etapa os módulos: Plano de Teste, Módulos de Teste, Casos de Teste, Execução de Trabalho de Teste, Coleta de Dados de Teste e Documentação de Resultados de Teste. Seguindo a rota definida neste documento representado pelo diagrama a seguir:



1 MÓDULO: PLANO DE TESTE

O plano de teste que traça o roteiro de execução, define modelos de documentos e condições aceitáveis que os testes devem ocorrer. Padronizando e organizando procedimento e documentando todo o processo de Trabalho de Teste.

2 MÓDULO: MÓDULOS DE TESTE

Os módulos são as etapas definidos de acordo com os testes que foi aplicado e que constitui o roteiro a ser seguido conforme determina a seção de Estratégia do Plano de Teste e assim definindo o modelo a ser aplicado a do mesmo modo que define este modelo de documento.

3 MÓDULO: CASOS DE TESTE

Os casos de testes foram definidos para a realização de stresse no sistema. Porém, como não tem especificado a carga de acessos simultâneos suportado pelo SIG Extensão os casos de teste seguem os volumes de cargas definido durante o planejamento que se encontra especificado no Plano de Teste assim como o modelo de documentação de caso de teste. Assim, buscando identificando o limite e cobrindo o máximo de cenários de estresse possíveis.

4 MÓDULO: EXECUTAR TRABALHOS DE TESTE

Os Trabalhos de Teste é dividido em duas sub-etapas, sendo elas: Execução de Teste no Software e Registro e Geração de Dados de Resultados e Estatísticas do Desempenho do Software.

4.1 Execução de Teste no Software

O objetivo é a execução dos testes sobre o sistema. Segundo o que é definido pelo documento que possui os Casos de Testes.

4.2 Geração de Dados de Resultados e Estatísticas do Desempenho do Software

Aqui ocasiona a geração de relatórios, seja de modo manual, por meio de realização de anotações e geração de dados estatísticos, e/ou automática, gerados pelas ferramentas de teste como também a realização de otimização com prévia análise por meio do uso de técnicas estatísticas para melhor representar alguns resultados dos testes.

5 COLETA DE DADOS DE TESTE

Esta etapa visa reunir e organizar os resultados gerados pelos testes, sendo eles de modo automático ou manual. E usando os requisitos para organizar e classificar de modo a filtrar os dados que sejam relevantes para o teste. E com isso, centralizando os resultados que serão repassa para a etapa de Documentação de Resultados de Teste.

6 Documentação de Resultados de Teste

Nesta etapa realiza a análise dos resultados dos testes e a documentação já pré-definido durante o Plano de Teste para que assim possa ser repassado para a equipe de desenvolvimento do sistema.

APÊNDICE C – Casos de Teste

(Sistemas de Informações Gerenciais)

SIG Extensão do CEULP/ULBRA

CASOS DE TESTES

(Teste de Stress)

Histórico de Revisão

Data	Versão	Descrição	Autor
03/06/2020	1.0	Casos de testes identificando cenários de stress sobre o SIG Extensão	João Paulo S. Maciel

Palmas - TO

2020

Sumário:

1. CASOS DE TESTES	3
Caso de Teste - 001 (50 usuários)	3
Caso de Teste - 002 (100 usuários)	4
Caso de Teste - 003 (500 usuários)	5
Caso de Teste - 004 (1000 usuários)	6

1. CASOS DE TESTES

Nesta etapa, buscou-se definir quais as situações que devem ser testadas no software. Quais os cenários que o teste de estresse deveria simular. Os casos de testes foram gerados para saber os caminhos a serem percorridos durante a execução dos testes e com isso cobrindo situações com 50, 100, 500 e 1000 usuário em acessos simultâneos. Para a executar dos testes foram montados os seguintes casos de testes:

Caso de Teste - 001 (50 usuários)			
id	Carga de Trabalho	Recurso	Ações
01	50 usuários em acesso simultâneo	Autenticação (/sig/extensao/login)	1. A ferramenta abre a url; 2. Ferramenta adiciona usuario e senha; 3. Ferramenta escolhe a opção “autenticar”; 4. E verificado o tempo e recursos consumidos para acessar a página.
02	50 usuários em acesso simultâneo	Página inicial (/sig/extensao/home)	1. A ferramenta abre a url; 2. E verificado o tempo e recursos consumidos para acessar a página.
03	50 usuários em acesso simultâneo	Página de inscrição (/sig/extensao/inscricoes)	1. A ferramenta abre a url; 2. E verificado o tempo e recursos consumidos para acessar a página.
04	50 usuários em acesso simultâneo	Página de Certificados (/sig/extensao/certificados)	1. A ferramenta abre a url; 2. E verificado o tempo e recursos consumidos para acessar a página.
05	50 usuários em acesso simultâneo	Páginas de Ações de extensão (/sig/extensao/acoes)	1. A ferramenta abre a url; 2. E verificado o tempo e recursos consumidos para acessar a página. 3. Ferramenta seleciona uma ação e opção “autenticar”;
06	50 usuários em acesso simultâneo	Detalhes de ação (/sig/extensao/acoes/471/detalhes/)	1. A ferramenta abre a url; 2. E verificado o tempo e recursos consumidos para acessar a página.

Caso de Teste - 002			
id	Carga de Trabalho	Recurso	Ações
01	100 usuários em acesso simultâneo	Autenticação (/sig/extensao/login)	1. A ferramenta abre a url; 2. Ferramenta adiciona usuario e senha; 3. Ferramenta escolhe a opção “autenticar”; 4. E verificado o tempo e recursos consumidos para acessar a página.
02	100 usuários em acesso simultâneo	Página inicial (/sig/extensao/home)	1. A ferramenta abre a url; 2. E verificado o tempo e recursos consumidos para acessar a página.
03	100 usuários em acesso simultâneo	Página de inscrição (/sig/extensao/inscricoes)	1. A ferramenta abre a url; 2. E verificado o tempo e recursos consumidos para acessar a página.
04	100 usuários em acesso simultâneo	Página de Certificados (/sig/extensao/certificados)	1. A ferramenta abre a url; 2. E verificado o tempo e recursos consumidos para acessar a página.
05	100 usuários em acesso simultâneo	Páginas de Ações de extensão (/sig/extensao/acoes)	1. A ferramenta abre a url; 2. E verificado o tempo e recursos consumidos para acessar a página. 3. Ferramenta seleciona uma ação e opção “autenticar”;
06	100 usuários em acesso simultâneo	Detalhes de ação (/sig/extensao/acoes/471/detalhes/)	1. A ferramenta abre a url; 2. E verificado o tempo e recursos consumidos para acessar a página.

Caso de Teste - 003			
id	Carga de Trabalho	Recurso	Ações
01	500 usuários em acesso simultâneo	Autenticação (/sig/extensao/login)	1. A ferramenta abre a url; 2. Ferramenta adiciona usuario e senha; 3. Ferramenta escolhe a opção “autenticar”; 4. E verificado o tempo e recursos consumidos para acessar a página.
02	500 usuários em acesso simultâneo	Página inicial (/sig/extensao/home)	1. A ferramenta abre a url; 2. E verificado o tempo e recursos consumidos para acessar a página.
03	500 usuários em acesso simultâneo	Página de inscrição (/sig/extensao/inscricoes)	1. A ferramenta abre a url; 2. E verificado o tempo e recursos consumidos para acessar a página.
04	500 usuários em acesso simultâneo	Página de Certificados (/sig/extensao/certificados)	1. A ferramenta abre a url; 2. E verificado o tempo e recursos consumidos para acessar a página.
05	500 usuários em acesso simultâneo	Páginas de Ações de extensão (/sig/extensao/acoes)	1. A ferramenta abre a url; 2. E verificado o tempo e recursos consumidos para acessar a página. 3. Ferramenta seleciona uma ação e opção “autenticar”;
06	500 usuários em acesso simultâneo	Detalhes de ação (/sig/extensao/acoes/471/detalhes/)	1. A ferramenta abre a url; 2. E verificado o tempo e recursos consumidos para acessar a página.

Caso de Teste - 004			
id	Carga de Trabalho	Recurso	Ações
01	1000 usuários em acesso simultâneo	Autenticação (/sig/extensao/login)	1. A ferramenta abre a url; 2. Ferramenta adiciona usuario e senha; 3. Ferramenta escolhe a opção “autenticar”; 4. E verificado o tempo e recursos consumidos para acessar a página.
02	1000 usuários em acesso simultâneo	Página inicial (/sig/extensao/home)	1. A ferramenta abre a url; 2. E verificado o tempo e recursos consumidos para acessar a página.
03	1000 usuários em acesso simultâneo	Página de inscrição (/sig/extensao/inscricoes)	1. A ferramenta abre a url; 2. E verificado o tempo e recursos consumidos para acessar a página.
04	1000 usuários em acesso simultâneo	Página de Certificados (/sig/extensao/certificados)	1. A ferramenta abre a url; 2. E verificado o tempo e recursos consumidos para acessar a página.
05	1000 usuários em acesso simultâneo	Páginas de Ações de extensão (/sig/extensao/acoes)	1. A ferramenta abre a url; 2. E verificado o tempo e recursos consumidos para acessar a página. 3. Ferramenta seleciona uma ação e opção “autenticar”;
06	1000 usuários em acesso simultâneo	Detalhes de ação (/sig/extensao/acoes/471/detalhes/)	1. A ferramenta abre a url; 2. E verificado o tempo e recursos consumidos para acessar a página.

APÊNDICE D – Relatório de Erros

(Sistemas de Informações Gerenciais)

SIG Extensão do CEULP/ULBRA

RELATÓRIO DE ERROS DO SIG EXTENSÃO
(Teste de Stress)

Histórico de Revisão

Data	Versão	Descrição	Autor
03/06/2020	1.0	Avaliação de Testes de Stress sobre o SIG Extensão	João Paulo S. Maciel

Sumário:

1 OBJETIVO	03
2 ESCOPO	03
3 REFERÊNCIAS	03
4 INTRODUÇÃO	03
5 COBERTURA DOS TESTE	03
6 ANÁLISE DE DEFEITOS	04
6.1 Análise Individual dos Resultados dos Casos de Testes	04
6.1.1 Análise do Caso de Testes 001 (50 usuários em simultâneos)	04
6.1.2 Análise do Caso de Testes 002 (100 usuários em simultâneos)	05
6.1.3 Análise do Caso de Testes 003 (500 usuários em simultâneos)	05
6.1.4 Análise do Caso de Testes 004 (1000 usuários em simultâneos)	07
6.2 Densidade dos Defeitos	11
6.3 Tendências dos Defeitos	11
6.4 Idade dos Defeitos	12

1. OBJETIVOS

Este Relatório de Avaliação de Teste tem como objetivo descrever os resultados de testes sobre o SIG Extensão do CEULP/ULBRA tendo como base os requisitos do Plano de Teste.

2. ESCOPO

Este relatório busca apresentar e analisar os resultados dos testes carga, volume e stress sobre a aplicação SIG Extensão. Com isso buscando:

- Avaliar o comportamento do sistema SIG sobre condições extremas do Teste de Stress.
- Identificar aprimoramentos que possam ampliar a capacidade do Teste de Stress para aumentar a qualidade dos testes.

3. REFERÊNCIAS

1. “Exemplo da Web do Projeto de Registro em Curso”, Versão 2001.03. IBM Corp., 2004.

4. INTRODUÇÃO

Todos os casos de testes definido durante a fase de planejamento e contidos nos documentos de Casos de Testes e foram realizados conforme o previsto no Plano de Teste e com isso os seguindo a ordem conforme no Módulos de Teste. E assim, registrando e classificando os defeitos e para que se possa ser repassado para a equipe de desenvolvimento.

Quase todos os testes foram concluídos, porém o sistema SIG Extensão não suportou a carga do Caso de Testes 004, de mil usuários simultâneos, ocasionando na queda da aplicação. E por meio da análise dos defeitos, sendo identificado os mais recorrentes em quase todos os casos testados os erros *500/Internal Server Error* e *400/Bad Request*, além de um alto consumo de recursos de processamento de dados. Podendo ser classificado como mediano para crítico uma vez que a finalidade do SIG é atender uma instituição de ensino que é o CEULP/ULBRA.

5. COBERTURA DO TESTE

Foram testados todos os cenários representados pelos casos de testes. cobrindo assim o nível de stress suportado apenas por acessos simultâneos.

6. ANÁLISE DE DEFEITOS

Para a análise dos resultados dos testes com os casos com carga de 100, 500, 1000 usuários foi considerado os relatórios de resultados gerados pela ferramenta JMeter e anotações relacionados ao acompanhamento do ambiente de teste por meio do Htop. Para qualificação foi considerado a pontuação Apdex (Índice de Desempenho da Aplicação), contido nos relatórios do Jmeter onde esse é um método que mede a satisfação que um usuário referente ao cenário de teste. Esse método considera que a pontuação de Apdex devem estar entre 1, onde é considerado Satisfatório, 0,5, onde é Tolerável, e 0, como Frustrante.

Com isso, foi notado que em apenas um dos casos de testes aplicado o resultado Apdex ficou ligeiramente acima da pontuação tolerável com os demais ficando abaixo. Sendo que na carga de 1000 usuários teve a pior índice, de modo que mediu quase zero. E também foi a carga que proporcionou o colapso que derrubou a aplicação possuindo a maior variação de erros dentre os testes.

Em relação a erros, o mais presente em quase todos os casos de testes foi o erro *500/Internal Server Error* que indica a ocorrência de mau funcionamento do SIG Extensão, podendo se tratar de uma componente ou função que possa está gerando os erro com relação ao volume de carga de requisição usado nos testes de estresse. É um problema comum em todos os cenários de testes foi o alto consumo de recurso, principalmente de processamento, sendo que o consumo de memória RAM se manteve estabilizado com pequenas variações, porém o SIG chega a consumir quase 1GB deste recurso.

6.1 Análise Individual dos Resultados dos Casos de Testes

6.1.1 Análise do Caso de Testes 001 (50 usuários em simultâneos)

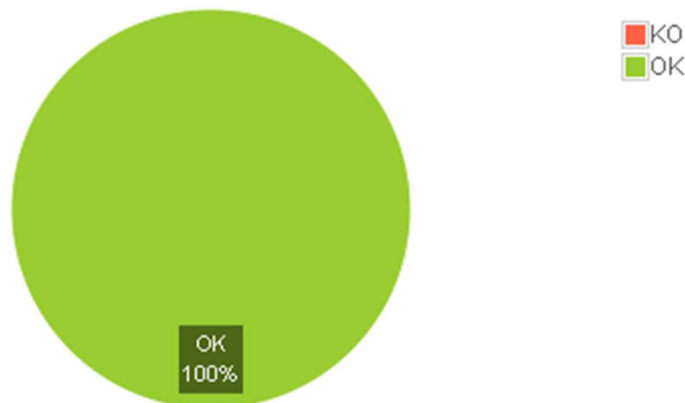
O SIG com uma carga de estresse de 50 usuários em solicitação simultâneos manteve estável e sem recorrência de erro. Com o primeiro caso de teste sobre o SIG tendo um tempo médio de resposta de um minuto e os índice de desempenho da aplicação (Apdex) com média de 0.594, sendo considerado como tolerável. Conforme representado junto com as requisições na Figura 1.1

Figura 1.1 - APDEX Caso de Teste 001 (50 usuários)

Apdex [▲]	T (Toleration threshold) [◆]	F (Frustration threshold) [◆]	Label [◆]
0.594	500 ms	1 sec 500 ms	Total
0.267	500 ms	1 sec 500 ms	Requisição HTTP-0
0.508	500 ms	1 sec 500 ms	Requisição HTTP
0.858	500 ms	1 sec 500 ms	HTTP Request
0.872	500 ms	1 sec 500 ms	Requisição HTTP-1

Com isso, pode ser obtido o seguinte gráfico: Figura 1.2, sendo KO (knockout) Nocaute nas solicitações (Erros) e OK para as Requisições bem sucedidas.

Figura 1.2 - Resumos de Solicitações - Caso de Teste 001 (50 usuários)



Com isso pode ser observado que a carga de estresse de 50 usuários simultâneos, não compromete o SIG Extensão. Mais essa carga já apresenta uma significativa quantidade de consumo significativo de recursos da máquina. Sendo que o sistema chega a consumir quase 1GB de RAM e usando em média 87% da capacidade de processamento do processador durante os momentos de pico do teste referente às solicitações.

6.1.2 Análise do Caso de Testes 002 (100 usuários em simultâneos)

Ao aplicar uma carga de solicitações de 100 usuários em simultâneos o sistema SIG já apresentou um pequeno aumento do consumo de recursos em comparação ao teste com carga de 50 usuários, e com mais de 10% de erros nas requisições. O tempo de duração do

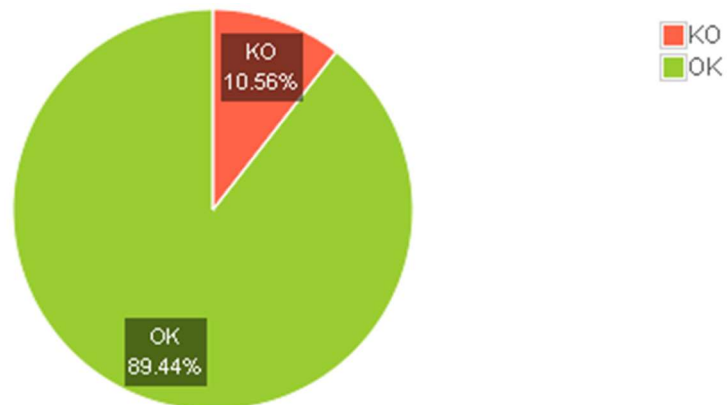
teste foi de aproximadamente 2 minutos e com índice de desempenho da aplicação (apdex) com média de 0.334, ou seja, menor que com a carga do Caso de Teste 001. Conforme representado junto com as requisições na Figura 2.1.

Figura 2.1 - APDEX Caso de Teste 002 (100 usuários)

Apdex [▲]	T (Toleration threshold) [◆]	F (Frustration threshold) [◆]	Label [◆]
0.334	500 ms	1 sec 500 ms	Total
0.130	500 ms	1 sec 500 ms	Requisição HTTP-0
0.203	500 ms	1 sec 500 ms	Requisição HTTP
0.480	500 ms	1 sec 500 ms	HTTP Request
0.645	500 ms	1 sec 500 ms	Requisição HTTP-1
0.750	500 ms	1 sec 500 ms	HTTP Request-1
1.000	500 ms	1 sec 500 ms	HTTP Request-0

Ao final do teste, foi identificado a ocorrência de 10,56 % de erros com uma taxa de sucesso de 89,44% de das requisições do teste. Representado pelo gráfico da figura 2.2, sendo KO (knockout) Nocaute nas solicitações (Erros) e OK para as Requisições bem sucedidas.

Figura 2.2 - Resumos de Solicitações - Caso de Teste 002 (100 usuários)



Sendo assim identificado o tipo, a quantidade e a porcentagem de representativa dos erros conforme a Tabela de Erros:

TABELA DE ERROS			
Tipo de Erro	Quantidade	% dos erros	% em toda a amostra
Error 500: Internal Server Error	236	86,13%	9,10%
Error 400: Bad Request	38	13,87%	1,46%

Como pode ser notado, apesar de apresentar erros o consumo de recursos chegou a ser superior a 1GB de RAM e usando em média 90% da capacidade de processamento do processador durante os momentos de pico com isso não deixando margem podendo ocasionar em lentidão.

6.1.3 Análise do Caso de Testes 003 (500 usuários em simultâneos)

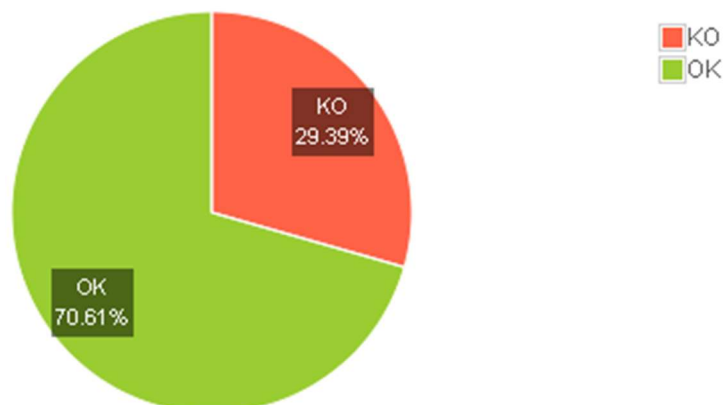
A carga de teste de 500 usuários realizando requisições em simultâneos no SIG Extensão, levou cerca de 13 minutos e com isso o número de erro e de consumo de recursos aumentou consideravelmente em comparação a carga de 100 usuários. E com quase 30% de erros em relação às requisições e com índice de desempenho da aplicação (apdex) com média de 0.228, onde pode ser considerado que é quase totalmente frustrante. Conforme representado junto com as requisições na Figura 3.1.

Figura 3.1 - APDEX Caso de Teste 003 (500 usuários)

Apdex ▲	T (Toleration threshold) ◆	F (Frustration threshold) ◆	Label ◆
0.228	500 ms	1 sec 500 ms	Total
0.056	500 ms	1 sec 500 ms	Requisição HTTP
0.263	500 ms	1 sec 500 ms	Requisição HTTP-0
0.358	500 ms	1 sec 500 ms	HTTP Request-1
0.384	500 ms	1 sec 500 ms	HTTP Request
0.605	500 ms	1 sec 500 ms	Requisição HTTP-1
0.736	500 ms	1 sec 500 ms	HTTP Request-0

Ao final do teste, foi identificado a ocorrência de 29,39% de erros com uma taxa de sucesso de apenas 70,61% das requisições do teste. Representado pelo gráfico da figura 3.2, sendo KO (knockout) Nocaute nas solicitações (Erros) e OK para as Requisições bem sucedidas.

Figura 3.2 - Resumos de Solicitações - Caso de Teste 003 (500 usuários)



Como pode ser visto, a carga de teste aplicada sobre o SIG Extensão apresentou novos erros mais com o erro *500/Internal Server Error* sendo o que mais ocorre. Outro fator identificado foi o consumo de recursos, que pode ser considerado um problema recorrente uma vez que consumo por parte da aplicação foi aproximadamente a 1GB de RAM mas ainda se manteve estável em comparação aos casos de testes anteriores e usando em média 93,8% da capacidade do processador durante os momentos de pico. Com isso gerando uma alta concorrência pelo processamento refletindo em lentidão na aplicação.

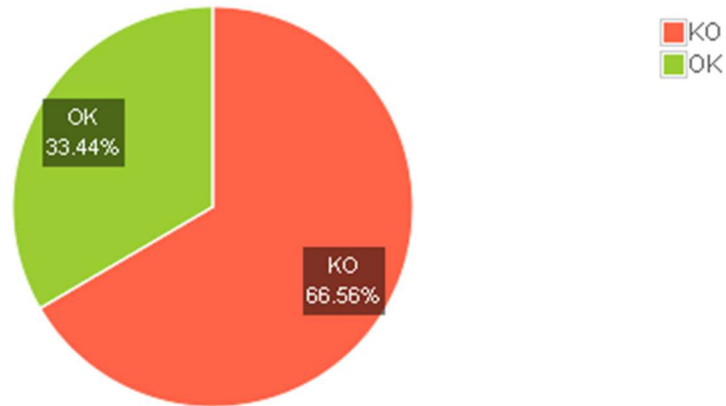
Sendo assim identificado o tipo, a quantidade e a porcentagem de representativa dos erros conforme a Tabela de Erros:

TABELA DE ERROS			
Tipo de Erro	Quantidade	% dos erros	% em toda a amostra
Error 500: Internal Server Error	2026	61,51%	18,08%
Error 403: Forbidden	573	17,40%	5,11%
Non HTTP response code: java.net.SocketException/Non HTTP response message: Connection reset	336	10,20%	3,00%
Non HTTP response code: org.apache.http.conn.HttpHostConnectException/Non HTTP response message: Connect to 127.0.0.1:8000 [V127.0.0.1] failed: Tempo esgotado para conexão (Connection timed out)	239	7,26%	2,13%
Error 400: Bad Request	120	3,64%	1,07%

6.1.4 Análise do Caso de Testes 004 (1000 usuários em simultâneos)

A carga de teste de 1000 usuários realizando requisições em simultâneos no SIG Extensão ocasionou no aumento do volume e aparição de novos erros. Além da carga provocar uma sobrecarga tanto do sistema como dos recursos o que ocasionou em um colapso derrubando a aplicação. Com isso, na execução do caso de teste, proporcionou 66,56% de *KO* (*knockout*) de requisições, sendo a maior taxa entre os casos de testes aplicados neste trabalho. Conforme apresentado pela figura 4.1.

Figura 4.1 - Resumos de Solicitações - Caso de Teste 004 (1000 usuários)



E ainda, foi verificado que o sistema teve sua pior pontuação Apdex sendo de 0,061, praticamente próximo de 0 (zero), onde, no método Apdex é considerado frustrante para os usuários. Representado pela figura 4.2.

Figura 3.1 - APDEX Caso de Teste 004 (1000 usuários)

Apdex ▲	T (Toleration threshold) ◆	F (Frustration threshold) ◆	Label ◆
0.061	500 ms	1 sec 500 ms	Total
0.012	500 ms	1 sec 500 ms	Requisição HTTP
0.034	500 ms	1 sec 500 ms	HTTP Request
0.161	500 ms	1 sec 500 ms	Requisição HTTP-0
0.326	500 ms	1 sec 500 ms	HTTP Request-1
0.356	500 ms	1 sec 500 ms	Requisição HTTP-1
0.457	500 ms	1 sec 500 ms	HTTP Request-0

Com isso, foi possível identificar uma alteração considerável, onde neste caso de teste, a qual o erro mais recorrente sendo de conexão recusada (*Connection refused*) seguido de erros relacionados a falhas de resposta, conexão e entre outros. Como pode ser identificado o tipo, a quantidade e a porcentagem de representativa dos erros conforme a Tabela de Erros:

TABELA DE ERROS			
Tipo de Erro	Quantidade	% dos erros	% em toda a amostra
Non HTTP response code: org.apache.http.conn.HttpHostConnectException/ Non HTTP response message: Connect to 127.0.0.1:8000 [V127.0.0.1] failed: Conexão recusada (Connection refused)	10167	77,01%	51,26%
Non HTTP response code: org.apache.http.NoHttpResponseException/Non HTTP response message: 127.0.0.1:8000 failed to respond	1013	7,67%	5,11%
Error 403: Forbidden	687	5,20%	3,46%
Non HTTP response code: java.net.SocketException/Non HTTP response message: Connection reset	604	4,58%	3,04%
500/Internal Server Error	539	4,08%	2,72%

Non HTTP response code: org.apache.http.conn.HttpHostConnectException/ Non HTTP response message: Connect to 127.0.0.1:8000 [V127.0.0.1] failed: Tempo esgotado para conexão (Connection timed out)	136	1,03%	0,69%
400/Bad Request	55	042%	0,28%
Non HTTP response code: java.net.SocketException/Non HTTP response message: Conexão fechada pela outra ponta (connect failed)	1	001%	0,01%

Outro problema recorrente, apesar de manter padrão o consumo aproximado de 1GB de RAM, é o alto consumo de recursos de processamento de dados durante os momento de pico, onde, a execução do caso de teste chegou a usar 99,8% da capacidade de processamento do reservado para o ambiente de teste.

6.2 Densidade de Defeitos

A densidade dos defeitos identificados durante a execução dos Testes de Stress sobre o sistema SIG Extensão medindo a densidade por meio de níveis de gravidade e método Apdex.

- Nível de Gravidade (critica, alta, média e baixa);
- Método Apdex (pontuação entre 1 Satisfatório, 0.5 Tolerável, 0 Frustrante)

O **desempenho do sistema**, seguindo o método **Apdex**, chegou a ficar um pouco acima do Tolerável porém os demais casos tendiam a Frustrante.

O **Nível de Gravidade** dos erros, que passaram a aparecer a partir do Caso de Teste 002 com carga de 100 usuários em simultâneos, sendo considerados entre **Médio a Grave** uma vez que corresponde de 10,56% a 66,56% dos erros gerados pelas cargas de estresse aplicadas ao sistema chegando a ocorrer a queda da aplicação.

6.3 Tendências de Defeitos

A tendência dos defeitos mostra que a ocorrência de erros tende a aumentar de acordo com a carga de usuários vai aumentando. O que ocasionou no aparecimento de novos erros quando os demais casos de teste foram aplicados.

Outra tendência é o alto consumo de recursos para atividades com autenticar e navegar dentro da aplicação. O que tende a exigir cada vez mais poder de processamento.

6.4 Idade dos Defeitos

Devido ser a primeira passagem do Testes de Stress, a idade dos Defeitos ainda é indefinido e com isso não se pode concluir o tempo gasto para a correção dos defeitos.