Teste de Desempenho em Aplicações SIG Web

Arturo H. Torres-Zenteno[1], Eliane Martins[1], Ricardo da S. Torres[1],
Maria J. Escalona Cuaresma[2]

[1] Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
Caixa Postal 6176, 13084-971 – Campinas – SP – Brasil.
{arturo.zenteno,eliane,rtorres}@ic.unicamp.br
[2] Departamento de Linguagens e Sistemas Informáticos – Universidade de Sevilla
Av. Reina Mercedes s/n. 401012 – Sevilla – Espanha.
mjescalona@us.es

Resumo. Este artigo propõe um modelo de processo de teste de desempenho para aplicações SIG Web. O modelo considera os casos de uso mais críticos ou de maior risco quanto ao desempenho de um sistema para a criação de cenários de testes. Além disso, prevê a utilização de ferramentas livres para automatização de etapas do processo de avaliação. O modelo foi aplicado ao projeto WebMaps, que é uma aplicação SIG Web cuja finalidade é auxiliar seus usuários no planejamento agrícola a partir de regiões de interesse. Os resultados preliminares obtidos indicam que os testes foram úteis na identificação de problemas da arquitetura preliminar do sistema.

1 Introdução

Um Sistema de Informação Geográfica (SIG) é um software voltado para o gerenciamento de dados geo-referenciados. Os SIG's são sistemas automatizados usados para armazenar, analisar e manipular dados geográficos, ou seja, dados que representam objetos e fenômenos em que a localização geográfica é uma característica inerente e indispensável para analisá-los [1, 2].

Neste contexto, um SIG Web é um sistema que provê diferentes serviços SIG de análise e visualização de dados espaciais através da Web [3]. O desenvolvimento de sistemas deste tipo é complexo e deve considerar requisitos pouco encontrados em aplicações tradicionais: por exemplo, o grande volume de dados gerenciado (convencionais e geo-referenciados), a diversidade de usuários no ambiente Web e processamento concorrente de requisições. Neste cenário, um dos meios utilizados para se garantir a qualidade destes sistemas é a realização de testes [4] ao longo do processo de desenvolvimento.

O propósito deste artigo é propor um modelo de processo de teste para aplicações SIG Web a partir dos casos de uso mais críticos ou de maior risco quanto ao desempenho. A realização de testes de desempenho é fundamental para assegurar que estas aplicações sejam adequadas quando da sua implantação e uso.

O processo proposto de teste de desempenho lida com as características inerentes das aplicações SIG Web, grande volume de dados e carga de usuários simultâneos,

fornecendo um conjunto de atividades de teste bem detalhadas. Estas atividades são integradas em um processo robusto, onde o ponto de partida é identificação dos casos de uso críticos que precisam ser testados. Para isso se supõe que os requisitos do sistema foram corretamente especificados.

O modelo de processo de teste de desempenho foi aplicado na avaliação do sistema WebMaps, que é uma aplicação SIG Web cuja finalidade é auxiliar seus usuários no planejamento agrícola a partir de regiões de interesse. Estes testes foram feitos de acordo com cenários reais construídos baseando-se em estatísticas de uso reais no Sistema de Monitoramento Agrometereológico [5]. Os resultados preliminares obtidos indicam que os testes foram úteis na identificação de problemas da arquitetura preliminar do WebMaps antes da sua implantação.

O artigo inicia com uma descrição geral dos testes de desempenho, assim como, a sua identificação dentro de um processo de desenvolvimento (seção 2). Em seguida, a seção 3 detalha os tipos de testes de desempenho em aplicações Web, mostrando alguns processos de teste existentes. Estas abordagens servem como base para a especificação da proposta do processo de teste de desempenho para aplicações SIG Web (seção 4). A seção 5 apresenta um caso de estudo onde o processo de teste de desempenho proposto é usado na avaliação de um SIG Web real. Por fim, a seção 6 apresenta as conclusões e trabalhos futuros.

2 Testes de desempenho

A realização de testes tem como finalidade assegurar a qualidade do sistema, sendo seu objetivo principal encontrar erros. Os testes também servem para se obter medidas de requisitos não funcionais do software, tais como confiabilidade ou desempenho, usando-se técnicas estatísticas apropriadas [6].

Entre os diferentes tipos de testes de sistema existentes [7], destacam-se os testes de desempenho. A Fig. 1 mostra uma adaptação do modelo "V", indicando as diferentes fases do desenvolvimento de sistemas associadas com os tipos de teste existentes [8]. Embora este modelo de testes seja discutível [8], ele será utilizado neste artigo para fim didático.

A realização de testes tem por objetivo determinar se o desempenho do sistema integrado é adequado, de acordo com os requisitos do sistema. Testes de desempenho podem ser realizados ao longo do desenvolvimento, mas somente quando todos os componentes são finalmente integrados é que se pode ter uma medida real do seu desempenho [9]. Note que, na Fig. 1, o teste de desempenho se encontra dentro dos testes de sistema, associados à fase de Detalhamento de Requisitos.

Molinari [10] mostra a relação entre risco de teste versus requisito de qualidade em aplicações Web, ou seja, quais os requisitos de qualidade que mais comprometem o correto funcionamento destes sistemas caso não sejam corretamente testados. Segundo Molinari, o desempenho é o requisito de qualidade que deveria ter mais cuidado em aplicações Web. Esta importância é ainda mais crítica em aplicações SIG Web tendo em vista o grande volume de dados gerenciados.

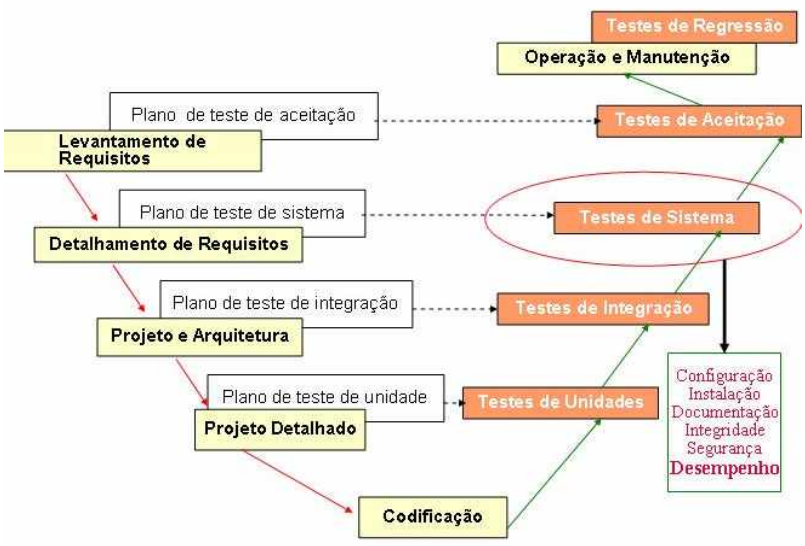

Fig. 1. Fases no processo de desenvolvimento vs Tipos de teste (modelo V).

3 Processos de Teste de Desempenho em Aplicações Web

Como parte dos testes de desempenho tem os testes de carga (testam o desempenho do sistema levando-se em conta uma carga de usuários simultâneos reais), testes de estresse (testam o desempenho do sistema considerando o número máximo de usuários simultâneos que pode suportar) e os testes de volume (testam a quantidade de dados que o sistema pode gerenciar) [9]. A seguir é descrito os processos de teste de cada um deles.

3.1 Processo de Teste de Carga / Estresse

Vários processos de teste de desempenho para aplicações web vêm sendo propostos recentemente [10, 11, 12, 13]. A Fig. 2 mostra dois destes processos de teste de desempenho: o de Molinari [10] que apresenta o processo em 5 fases (a); e o de Menascé [13] que possui 7 fases (b). Estes processos são aplicáveis tanto para teste de carga como estresse.

Molinari inicia o processo com o planejamento de teste de carga / estresse (Fase 1), cujo objetivo principal é identificar e projetar quais funcionalidades poderiam ser testadas no sistema. Já na Fase 2, se procede a especificação e criação de usuários virtuais. Uma vez criados os usuários virtuais, um cenário de teste (Fase 3) correspondente a esta quantidade de usuários é especificado. O cenário é executado na Fase 4. Por fim, a Fase 5 compreende a análise do sistema testado.

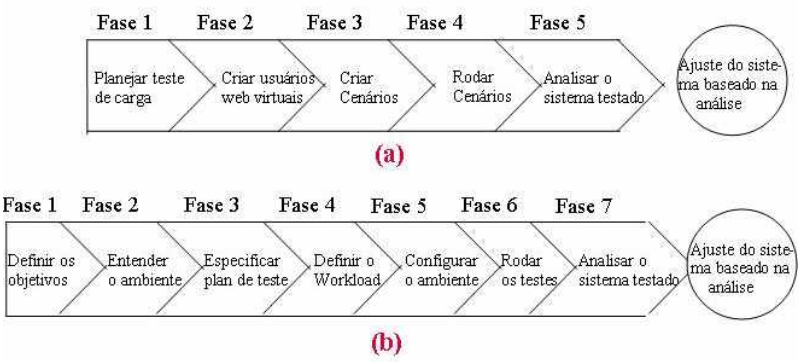

Fig. 2. Processo de teste de desempenho em aplicações Web. (a) Processo proposto por Molinari [10] e (b) processo proposto por Menascé [13].

Menascé apresenta fases mais abrangentes. Para ele o processo se inicia com uma fase de definição de objetivos do teste de desempenho (Fase 1). Logo na Fase 2 estuda-se o ambiente, ou seja, a infraestrutura (servidores, serviços web), software (sistemas operacionais, middleware e aplicações), as conexões de rede e os protocolos de rede presentes no sistema. A Fase 3 é a mesma da Fase 1 de Molinari. A Fase 4, definição do *Workload*, é equivalente à fase de criação de cenários e à fase de criação de usuários virtuais de Molinari. Depois, na Fase 6, as ferramentas de teste são instaladas e configuradas. Nas Fases 7 e 8, os testes são executados e o sistema é analisado, respectivamente. Tanto o processo de Molinari quanto o de Menascé é finalizado com o ajuste do sistema, a partir da análise dos resultados dos testes de desempenho.

3.2 Processo de Teste de Volume

O teste de volume avalia a quantidade de dados gerenciados em um sistema Web. O objetivo deste teste é determinar a capacidade do sistema em lidar com o volume de dados especificado nos seus requisitos. Em geral, este tipo de teste usa grandes quantidades de dados, o que permite determinar os limites em que o sistema falha. Além disso, costumam ser utilizados na identicação da carga máxima ou volume de dados que o sistema pode gerenciar em um dado peŕodo de tempo.

A pesquisa realizada até o momento não identificou nenhum processo de teste de volume bem definido que pudesse ser utilizados na avaliação de aplicações web.

4 Processo proposto de Teste de Desempenho em Aplicações SIG Web

Esta seção apresenta uma proposta de processo de teste de desempenho em aplicações SIG Web. Primeiramente, as especificações dos processos de teste de carga / estresse

e de volume são apresentados de maneira isolada (seções 4.1 e 4.2). Em seguida, suas atividades são integradas em um processo único (seção 4.3).

4.1 Processo proposto de Teste de Carga / Estresse

Esta proposta consiste, basicamente, na adaptação dos processos de teste de carga / estresse em aplicações Web vistos na seção 3.1. A Fig. 3(a) mostra que no processo de Molinari se considerou uma inversão da ordem das fases 2 e 3. Esta inversão justifica-se pela importância da criação de cenários para a especificação da quantidade de usuários virtuais que deverão ser criados. Esta inversão permite, portanto, criar diferentes cenários para um mesmo teste de carga / estresse, considerando, por exemplo, diferentes quantidades de usuários virtuais.

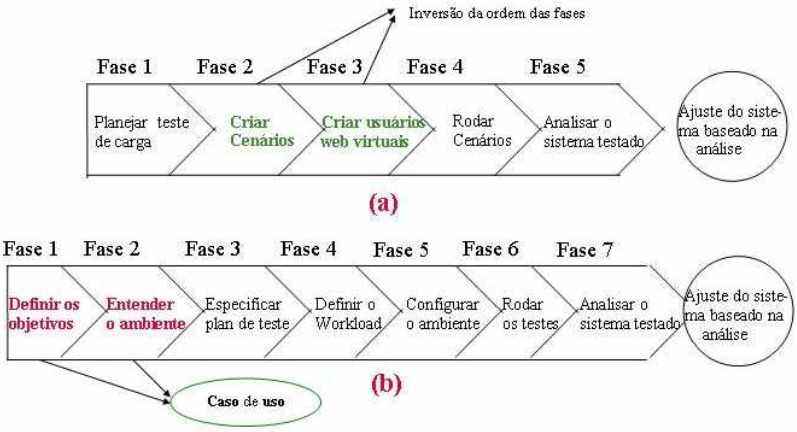

Fig. 3. (a) Adaptação apartir do processo proposto por Molinari [10]
(b) Adaptação do processo proposto por Menascé [13].

A Fig. 3(b) mostra que, no processo de Menascé, as fases 1 e 2 foram consideradas importantes. Estas fases foram agrupadas e consideradas equivalentes à definição dos casos de uso de maior risco.

A etapa de definição de casos de uso foi incorporada ao processo de Molinari modificado (veja seção 4.3).

4.2 Processo proposto de Teste de Volume

Este trabalho apresenta um processo de teste de volume, ilustrado na Fig. 4. Este processo se inicia com uma fase de planejamento, cujo objetivo principal é identificar a finalidade do teste (Fase 1). Já na Fase 2 identificam-se os cenários de volume, ou seja, cenários que envolvem armazenamento de informação em dispositivos de armazenamento primários (unidades de disco rígido), secundários (CD, DVD), em banco de dados ou em algum outro tipo de armazenamento. Estes cenários são criados

na Fase 3. A Fase 4 corresponde ao cálculo do volume de dados que o sistema gerencia para cada cenário de teste. Finalmente, na Fase 5, o sistema testado é analisado levando-se em conta os resultados do teste realizado e, caso necessário, modificado.

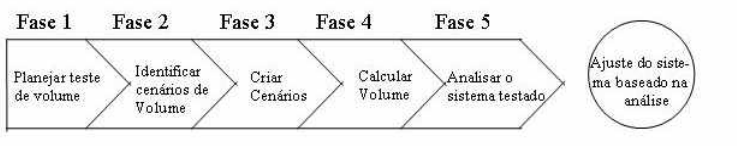

Fig. 4. Processo proposto de teste de volume.

4.3 Processo proposto de Teste de Desempenho

Este trabalhão propõe um processo que integra os testes de carga, estresse e volume na avaliação do desempenho de um sistema SIG Web. A Fig. 5 mostra o modelo proposto, usando a notação UML do diagrama de atividades [14]. Atividades são mostradas através de elipses enquanto setas indicam a ordem nas quais as atividades devem ser realizadas. As setas pontilhadas significam que a relação entre duas atividades não é obrigatória. O processo se inicia com uma fase de planejamento em um nível superior (Fase 0), onde se identifica os casos de uso que serão testados. Como dito anteriormente, a identificação de casos de uso críticos e de grande risco é vital para a realização de um bom teste de desempenho.

O teste de desempenho pode conter um ou mais testes de carga, estresse e volume. Tanto para os testes de carga como para os testes de estresse se considerou as modificações descritas na seção 4.1. Na parte A da Fig. 5, essas adaptações são apresentadas.

O processo começa com uma fase de planejamento do teste de carga ou estresse seguido da fase de criação de cenários. É importante notar que no planejamento dos testes da Fase 1 podem ser definidos um ou mais cenários. Uma vez criados o(s) cenário(s) correspondentes a um teste de carga ou estresse criam-se os usuários web virtuais. Na Fase 4 são executados o(s) cenário(s), enquanto que na última fase (Fase 5), o sistema testado é analisado.

A parte B da Fig. 5 mostra o processo proposto para a realização de testes de volume descrito na seção 4.2. Além disso, mostra como os cenários de teste de volume podem se baseiar nos cenários de teste de carga ou estresse caso tenham sido criados. Os testes de volume se iniciam na Fase 1 (planejamento), onde se define o objetivo do teste. Logo na Fase 2 identificam-se o(s) cenário(s) de volume. Esta identificação considera os cenários criados no proceso paralelo de carga ou estresse, ou seja, aqueles cenários que envolvem interação com banco(s) de dados ou com algum tipo de armazenamento de informação. Uma vez identificados, os cenários são criados (Fase 3). Um teste de volume pode conter um ou mais cenários e para cada cenário se determina o volume de dados que suporta (Fase 4). Na Fase 5, estes dados são usados na análise do sistema quanto ao volume de dados permitido. Nesta fase

pode-se fazer também uma projeção do volume de dados ao longo do tempo dependendo do planejamento na Fase 1.

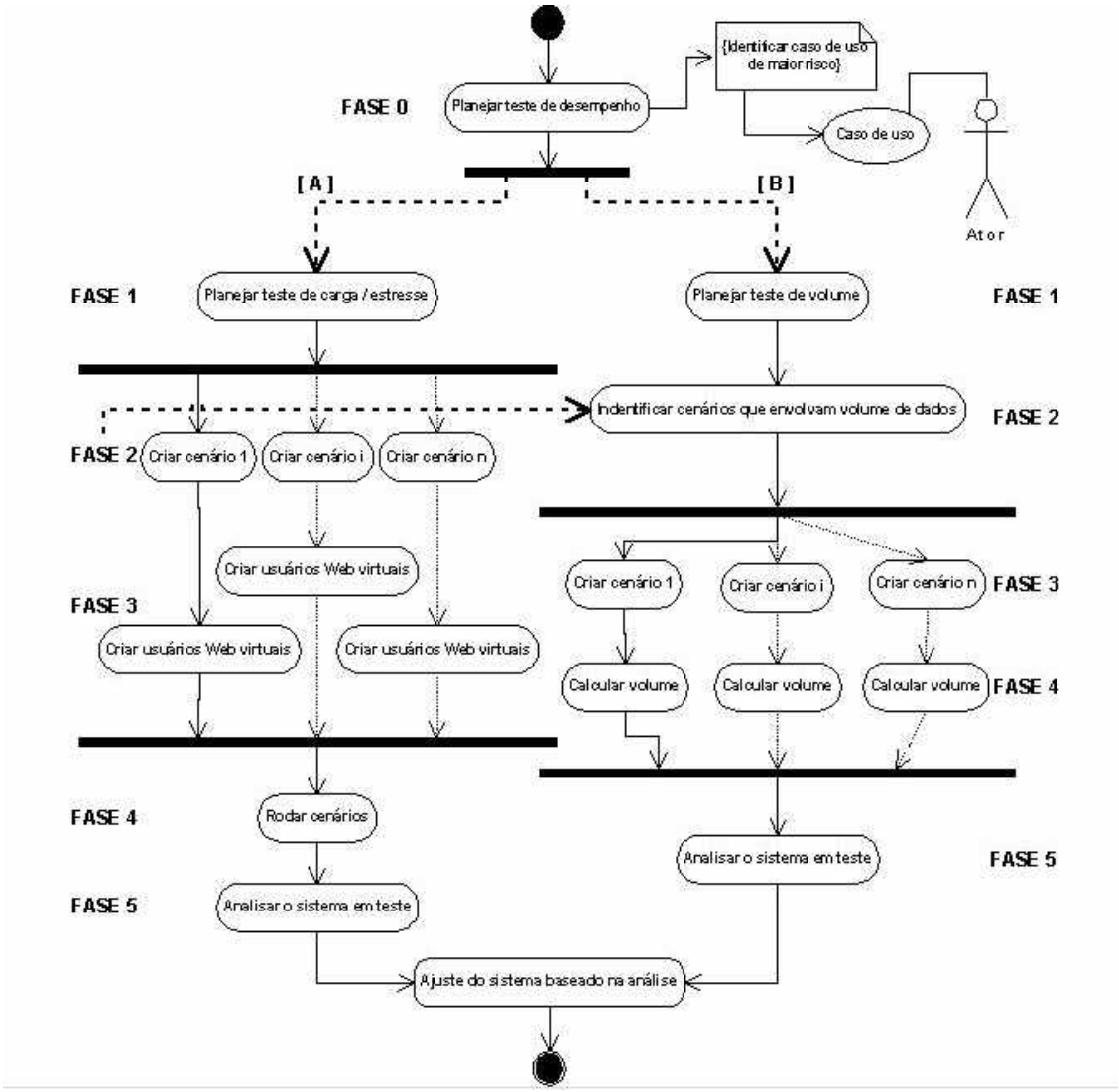

Fig. 5. Processo proposto de teste de desempenho para aplicações SIG Web.

A última atividade do processo de teste de desempenho é o ajuste do sistema. Este ajuste requer a colaboração de todos os membros da equipe de testes na análise dos problemas encontrados e na proposta de soluções.

5 Estudo de caso

O processo de teste de desempenho proposto foi usado na avaliação do sistema WebMaps, a partir de em um caso de uso crítico. Esta seção descreve este SIG web, bem como apresenta os resultados preliminares obtidos.

5.1 WebMaps

O WebMaps é um projeto em andamento no Instituto de Computação da Unicamp que tem como objetivo permitir o monitoramento e planejamento de safras agrícolas no Brasil. Este sistema integra, em rede, dados heterogêneos de diversas fontes e alguns serviços que gerenciam estes dados. Assim, pretende-se estabelecer uma plataforma base para a formulação, implementação e avaliação de políticas integradas de planejamento agrícola.

A Fig. 6 mostra o modelo de casos de uso do WebMaps, onde se especifica os requisitos funcionais do sistema. De maneira geral, o sistema deve permitir cadastros e consultas tanto aos seus usuários cadastrados quanto aos não cadastrados.

A versão atual do sistema permite que usuários cadastrem propriedades agrícolas e suas divisões (talhões) indicando as coordenadas geográficas junto com a informação de produtos cultivados. Além disso, permite a realização de consultas referentes à evolução dos plantios ao longo do tempo. Para tanto, o sistema gera para os usuários curvas NDVI (do inglês *Normalized Difference Vegetation Index*) [15], que são curvas que mostram os índices de vegetação por diferença normalizada, ou seja, a quantidade de vegetação em um dado local em um período determinado. Estes índices são obtidos a partir do processamento de imagens de satélite obtidas periodicamente. No momento, o WebMaps trabalha com imagens do tipo MODIS de todo o território brasileiro referentes a 4 anos (2001 até 2004) fornecidas pela NASA [16]. Cada imagem ocupa 109,6 MB e uma nova imagem foi obtida a cada 15 dias.

O processamento das imagens de satélite para geração de curvas NDVI é lento. Sendo assim, optou-se pelo uso de recortes (máscaras) das imagens, relativas a cada propriedade cadastrada, para diminuir o tempo de processamento.

A Fig. 7 apresenta a arquitetura do sistema WebMaps. Ela é composta de uma camada Cliente e uma de Servidor. A parte do Cliente é o navegador que mostra as páginas processadas pela camada servidora. O Servidor é dividido em três grandes módulos, implementados usando Java e a linguangem C: a) módulo responsável pelo cadastro de usuários e propriedades; b) módulo responsável pela realização de consultas; e c) repositórios de dados (Postgresql e disco rígido (HD)).

Do lado do servidor, emcontram-se páginas JSP's, bem como o módulo "Código em C" cuja tarefa é processar as imagens de satélite gerando as máscaras que servem para montar a curva NDVI de interesse para o usuário quando requisitado. A interação entre o código na linguagem C e o código Java é feito mediante o JNI (Java Native Interface). Em [17], Czajkowski apresenta os benefícios do uso de interfaces com códigos nativos no desenvolvimento de aplicações.

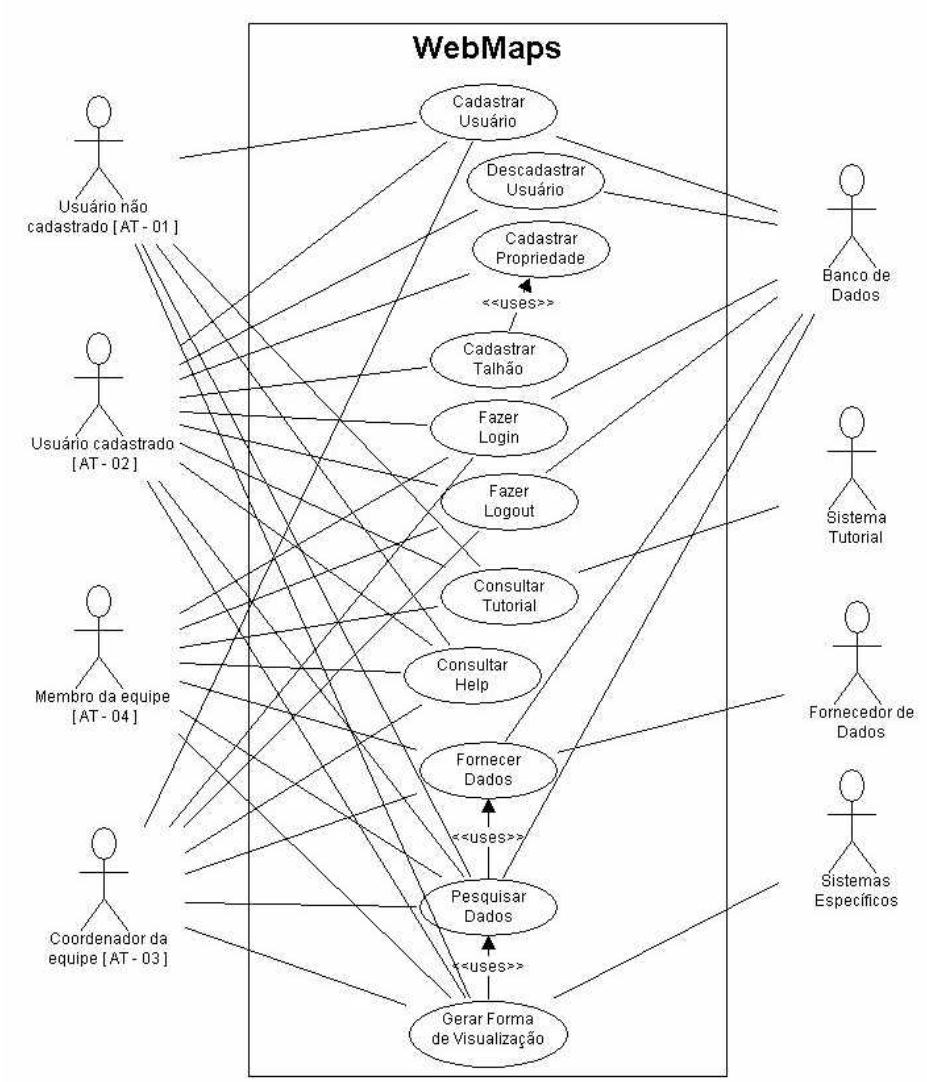

Fig. 6. Modelo de casos de uso do sistema WebMaps.

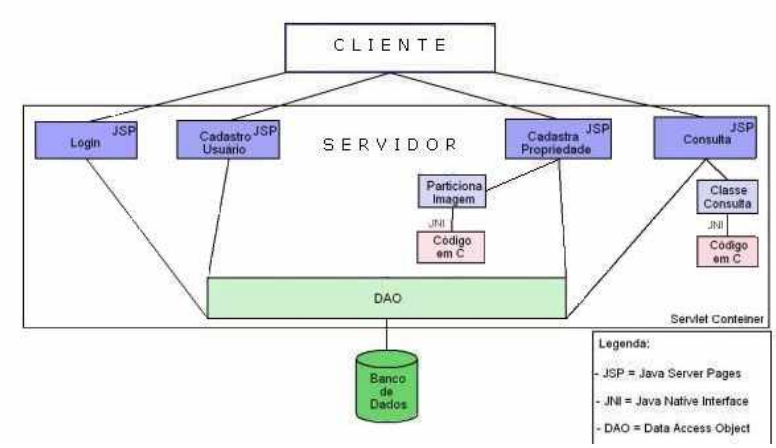

Fig. 7. Arquitetura do sistema WebMaps.

5.2 Aplicação da proposta

A primeira tarefa é identificar o caso de uso de maior risco no sistema. Neste caso, o módulo que requer maior processamento no sistema WebMaps é o módulo de cadastro de propriedade, já que é responsável pelos recortes das imagens de satélite.

A Tabela 1 mostra a especificação deste caso de uso com o padrão de requisitos funcionais do NDT (Navigational Development Techniques) [18].

Tabela 1. RF – 01: Requisito funcional de cadastro de propriedade.

RF – 01	Cadastrar propriedade	
Objetivos Associados	OBJ – 01: Fornecer dados e informações de uma determinada região geográfica. OBJ – 02: Trabalhar com dados geo-refenciados.	
Descrição	O cadastro de uma propriedade no sistema WebMaps deve seguir as especificações descritas nos itens subsequentes.	
Atores	**Ator do caso de uso**	**Ator do sistema**
	Usuário cadastrado	AT – 02
Seqüência normal	**Passo**	**Ação**
	1	O usuário solicita a realização de um cadastro de uma propriedade.
	2	O sistema mostra a tela de cadastro de propriedade.
	3	O usuário introduz o nome da sua propriedade.
	4	O usuário introduz as coordenadas geográficas da sua propriedade.
	5	O usuário introduz o tipo de acesso de sua propriedade (público / privado).

	6	O usuário pede ao sistema cadastrar sua propriedade.
	7	O sistema faz o cadastro da propriedade.
Pós-condição	São feitos os recortes das imagens de satélite e é gerada a máscara correspondente a esta propriedade.	
Exceções	**Passo**	**Ação**
	3	É opcional que o usuário introduza uma descrição de sua propriedade.
	5	Caso o usuário escolha um tipo de acceso privado para sua propriedade, há ainda a possibilidade de permitir acesso para usuários cadastrados específicos.

A Tabela 2 ilustra a aplicação da proposta de testes de desempenho para o sistema WebMaps, considerando o caso de uso mais crítico para o WebMaps: o cadastro da propriedade.

Este exemplo aplica cada uma das fases mostradas do processo de teste de desempenho para aplicações SIG Web apresentadas na seção 4.3. Vale ressaltar a relação existente entre o cenário 1 do processo de carga A e o cenário 1 do processo de volume. Esta relação é importante porque a boa especificação do cenário 1 no processo de carga (ou estresse) falicita a identificação e posterior especificação do seu correspondente cenário no processo de teste de volume.

A Fase 2, para o processo de teste de carga ou estresse, precisa de dados reais para a especificação correta dos cenários de teste. Estes testes foram feitos de acordo com cenários reais construídos baseando-se em estatísticas de uso reais no Sistema de Monitoramento Agrometereológico, Agritempo [4]. Este sistema, um dos mais utilizados pelos usuários-alvo do WebMaps, obteve no máximo 200 usuários reais conetados simultâneamente.

Tabela 2. Exemplo de aplicação da proposta.

TESTE DE DESEMPENHO				
Fase 0	Testar o sistema com o objetivo de avaliar o desempenho do caso de uso: cadastro de propriedade.			
TESTE DE CARGA [A]			**TESTE DE VOLUME [B]**	
Fase 1	Testar o sistema simulando os usuários entrando na página principal da aplicação e entrando na página de cadastro de propriedade.		**Fase 1**	Fazer o teste de volume do módulo de cadastro de propriedades.
Fase 2	**Cenário 1**: Uma carga de 200 usuários virtuais a cada 10 segundos até 1000 usuários na página de cadastro de propriedades.	**Cenário 2:** Uma carga de 300 usuários virtuais a cada 10 segundos até 3000 usuários na página principal da aplicação.	**Fase 2**	O cenário 1 do teste de carga é identificado como o cenário de volume, já que envolve o armazenamento de informação tanto no banco de dados quando no HD (disco rígido).

Fase 3	Criação de usuários virtuais apoiada pelo JMeter.	Criação de usuários virtuais apoiada pelo JMeter	**Fase 3**	**Cenário1:** *relacionado ao Cenário 1 do processo [A]* Quantidade de armazenamento de informação para 200 usuários cadastrando uma propriedade.
Fase 4	Para rodar os cenários se fez uso do JMeter.		**Fase 4**	Cálculo de volume para um usário: Criação de 92 recortes ~ 1 Mb. Para 200 usuários ~200 Mb.
Fase 5	Análise do sistema, aqui, se analisa as curvas obtidas no JMeter.		**Fase 5**	Análise do sistema, aqui, por exemplo, se pode fazer uma projeção do comportamento do sistema para o volume de dados esperado ao longo do tempo.

A Fase 4 do processo de carga poder ser feito de duas formas: manual ou automática. A forma manual é complexa do ponto de vista de gerenciamento, pois se precisa de um coordenador. No entanto, a forma automática é mais efetiva para a criação de usuários virtuais. Esta automatização pode ser apoiada por ferramentas livres de teste, algumas delas estão especificadas em [19]. Os experimentos aqui reportados utilizaram a ferramenta JMeter [20, 21]. As fases 4, 5 e 6 são discutidas na seção seguinte.

5.3 Resultados obtidos

O servidor do WebMaps considerado nos experimentos é um computador Intel(R) Pentium(R) 4 CPU 2.40GHz, 1Gb de memoria RAM e com sistema operacional Linux, distribução SUSE Linux.

A Fig. 8 mostra uma tela da ferramenta JMeter versão 2.1 [20] usada na Fase 3 e 4 do cenário 1 do processo de carga A considerando o caso de uso descrito na seção anterior. Esta ferramenta apresenta de maneira gráfica o tempo de resposta do sistema sobre os cenários montados na Tabela 1.

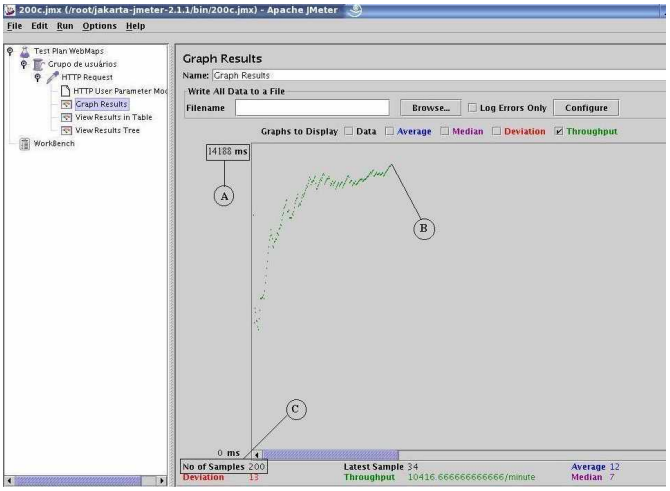

Fig. 8. Interface de resultados no Apache JMeter.

O sistema foi testado com uma quantidade de 200 usuários virtuais simultâneos (ver círculo C) cadastrando uma propriedade que gerava uma máscara de tamanho médio de 131 kb. O círculo A indica o eixo correspondente ao tempo de resposta do sistema, enquanto o círculo B indica o tempo de resposta do último usuário virtual. Segundo os resultados obtidos, o tempo médio de resposta do sistema foi de 14s. Este resultado, segundo Menascé [13], seria crítico, levando-se em conta que 95% das consultas dos usuários são abortadas caso o tempo de resposta de suas transações exceda em 6 segundos.

Também foram realizados testes de estresse. Verificou-se que o servidor WebMaps aborta quando (na média) 267 usuários estão cadastrando uma propriedade simultaneamente.

Assim, estes resultados foram muito úteis, já que forneceram indicativos para revisão da arquitetura do sistema WebMaps. Uma possibilidade de ajuste que ven senndo considerada consiste na melhoria da infra-estrutura de processamento, com, por exemplo, o uso de clusters e/ou a não realização de processamento on-line. Considera-se ainda a possibilidade de revisão do modelo de banco de dados do sistema de modo a evitar a gravação em disco de grande quantidade de imagens.

5 Conclusões

Este artigo apresentou trabalho em andamento que visa a definição de um processo para realização de testes de desempenho em SIG's Web. O processo proposto inclui a realização de testes de carga, estresse e volume, a partir de casos de uso.

O processo de realização de testes de desempenho proposto é parcialmente validado através de experimentos envolvendo o SIG Web WebMaps, atualmente em desenvolvimento no Instituto de Computação da Unicamp. O conjunto de testes

realizado permitiu a identificação de problemas de desempenho no sistema que resultaram na revisão da sua arquitetura.

Trabalhos em andamento incluem a adaptação do processo de testes de desempenho proposto visando à elaboração de uma metodologia genérica para realização de testes em SIG web. Esta metodologia genérica incluirá, além dos testes de desempenho, testes de usabilidade, testes funcionais, dentre outros.

Referências

1. Aronoff S.: Geographic Information Systems. WDL Publications, Canada (1989)
2. Bull G.: Ecosystem Modelling with GIS. Environmental Management. (1994) 18(3):345-349.–438
3. Do-Hyun Kim; Min-Soo Kim: Web GIS service component based on open environment. Geoscience and Remote Sensing Symposium, 2002. IGARSS '02. 2002 IEEE InternationalVolume 6, 24-28 June (2002) Page(s):3346 - 3348 vol.6
4. Filipak Machado C. A.: A importância dos testes no desenvolvimento de sistemas. Compañia de Informática do Paraná, CELEPAR (2003)
5. Agritempo, http://www.agritempo.gov.br, última visita: novembro (2005)
6. Martins, E.: Apostila sobre Testes de Software. Instituto de Computação, UNICAMP, Campinas, Brasil (2005)
7. Myers, Glendord J.: The Art of Software Testing. Jhon Wiley & Sons, Inc, Canada (1979)
8. Marick, B.: New Models for Test Development. Reliable Software Technologies.Testing Foundations. (1999)
9. Pressman, Roger S.: Software Engineering: A practioner's approach. MacGraw Hill (2001)
10. Molinari, L.:Testes de Software: Produzindo Sistemas Melhores e Confiáveis. Editora Érica Ltda, São Paulo (2003)
11. Miller E.: WebSite Loading and Capacity Analysis. Software Research, Inc. San Francisco, USA (2000)
12. Subraya, B.M.; Subrahmanya, S.V.: Object driven performance testing of Web applications. Quality Software, 2000. Proceedings. First Asia-Pacific Conference on 30-31 (2000) Page(s):17 - 26
13. Menascé Daniel A. and Almeida Virgilio A. F.: Capacity Planning for Web Services: Metrics, Models, and Methods. Prentice Hall (2002)
14. UML : Unified Modeling Language, http://www.uml.org/, última visita: novembro (2005)
15. CPTEC, http://satelite.cptec.inpe.br/htmldocs/ndvi/ndvi_fram.htm, última visita: novembro (2005)
16. Modis, Moderate Resolution Imaging Spectroradiometer, http://modis.gsfc.nasa.gov/, última visita: novembro (2005)
17. Czajkowski, G.; Daynes, L.; Wolczko, M.: Automated and portable native code isolation. Software Reliability Engineering, 2001. ISSRE 2001. Proceedings. 12th International Symposium on 27-30 Nov. (2001) Page(s):298 - 307
18. Escalona M. J.: Modelos y técnicas para la especificación y el análisis de la navegación en sistemas software. Tese doutoral. Escuela Técnica Superior de Ingeniería Informática. Universidad de Sevilla, Octubre (2004)
19. Pillat F. R., Ferreira L., Dias A. P.: Ferramentas para automatização de testes, http://dinf.unicruz.edu.br/~pillatt/2004_urcamp.pdf, última visita: novembro (2004)
20. Apache JMeter, http://jakarta.apache.org/jmeter/, última visita: novembro (2005)
21. Gutiérrez J.J., Pineda R., Villadiego D., Escalona M.J., Mejías M. : Pruebas funcionales y pruebas de carga sobre aplicaciones Web. Proceedings of Mundo Internet 2005. Vol. II. (2005) pp.208-219 Madrid, Spain