“Processo de Desenvolvimento e Testes para Aplicações SIG Web”

Arturo Henry Torres Zenteno

Trabalho Final de Mestrado Profissional em Computação

Instituto de Computação
Universidade Estadual de Campinas

Processo de Desenvolvimento e Testes para Aplicações SIG Web

Arturo Henry Torres Zenteno

Janeiro de 2006

Banca Examinadora:

Profª. Drª. Eliane Martins (Orientadora)
Instituto de Computação – UNICAMP

Profª. Drª. Fátima Pires
Coordenadoria Geral da Universidade – UNICAMP

Profª. Drª. M. Cecília C. Baranauskas
Instituto de Computação – UNICAMP

Prof. Dr. Edmundo Madeira (Suplente)
Instituto de Computação – UNICAMP

UNIDADE BC
Nº CHAMADA:
T/UNICAMP Z46p
V. EX.
TOMBO BCCL 75727
PROC 16P-129-03
C D X
PREÇO 11,00
DATA 20-02-03
BIB-ID 424579

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP
Bibliotecária: Maria Júlia Milani Rodrigues – CRB8a / 2116

Zenteno, Arturo Henry Torres
Z46p Processo de desenvolvimento e testes para aplicações SIG Web / Arturo Henry Torres Zenteno -- Campinas, [S.P. :s.n.], 2007.

Orientadores : Eliane Martins; Maria José Escalona Cuaresma
Trabalho final (mestrado profissional) - Universidade Estadual de Campinas, Instituto de Computação.

1. Engenharia de software. 2. Sistema de informação geográfica. 3. Software - Desenvolvimento. I. Martins, Eliane. II. Escalona Cuaresma, Maria José. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Título em inglês: Development and testing process for GIS Web applications.

Palavras-chave em inglês (Keywords): 1. Software engineering. 2. Geographic information systems. 3. Software – Development.

Área de concentração: Engenharia de Computação

Titulação: Mestre em Computação

Banca examinadora: Profa. Dra. Fátima Pires (IC-UNICAMP)
Profa. Dra. Maria Cecília C. Baranauskas (IC-UNICAMP)
Prof. Dr. Edmundo Madeira (IC-UNICAMP)
Profa. Dra. Eliane Martins (IC-UNICAMP)

Data da defesa: 12/01/2006

Programa de Pós-Graduação: Mestrado em Computação

Processo de Desenvolvimento e Testes para Aplicações SIG Web

Este exemplar corresponde à redação final do Trabalho Final devidamente corrigida e defendida por Arturo Henry Torres Zenteno e aprovada pela Banca Examinadora.

Campinas, 12 de janeiro de 2006.

Eliane Martins

Profª. Drª. Eliane Martins

(Orientadora)

Profª. Drª. María José Escalona Cuaresma

(Co - Orientadora)

Trabalho Final apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para obtenção do título de Mestre em Computação na área de Engenharia de Computação.

2008025047

Alex

Prof. Dr. ALEXANDRE XAVIER FALCÃO
Coordenador do Mestrado Profissional
Instituto de Computação - UNICAMP
Matr. 27.180-2

TERMO DE APROVAÇÃO

Trabalho Final Escrito defendido e aprovado em 12 de janeiro de 2006, pela Banca Examinadora composta pelos Professores Doutores:

Profa. Dra. Maria de Fátima Ridolfi Ordini Pires da Silva
CGU

Profa. Dra. Maria Cecília Calani Baranauskas
IC - UNICAMP

Profa. Dra. Eliane Martins
IC - UNICAMP

© Arturo Henry Torres Zapicos, 2006.
Todos os direitos reservados.

© Arturo Henry Torres Zenteno, 2006.
Todos os direitos reservados.

Aos meus pais, Ricardo e Juana, por todo o apoio que sempre me deram, não me deixando desanimar por maiores que fossem os meus desafios.

"*Confia os teus cuidados ao Senhor, e ele te susterá; jamais permitirá que o justo seja abalado*". (Sal. 55:22)

Agradecimentos

Agradeço a Deus por ter me dado a oportunidade de realizar este trabalho. Nada seria possível sem Ele.

Aos meus pais, Ricardo e Juana, por todo o apoio que sempre me deram, não me deixando desanimar por maiores que fossem os meus desafios.

À minha irmã, Verónica, pelo apoio continuo e aos meus sobrinhos Elyanne e André, os quais adoro como meus filhos.

À minha eterna Cesi, que soube como me dar forças, motivação e acolhida quando eu mais precisei em toda minha etapa de mestrado.

À Professora Doutora Eliane Martins, minha eterna gratitude para ela pela dedicação na orientação deste trabalho. Ela foi meu apoio constante no decorrer do meu trabalho.

À Professora Doutora María José Escalona Cuaresma, da Universidade de Sevilha, Espanha, pela dedicação na co-orientação deste trabalho. Ela foi uma grão fonte de inspiração na complementação do meu trabalho.

Ao Professor Doutor Ricardo da Silva Torres, pelo auxílio no esclarecimento de dúvidas e pelas sugestões.

À Professora Doutora Claudia Bauzer Medeiros, pelo conhecimento adquirido no projeto WebMaps e por todo o apoio.

A meus "irmaos", Roger e Lena, por tudo o que aprendemos e moramos juntos.

A os meus amigos da B2-A na Moradia da Unicamp: Sócrates, Elmer, Edson, Helmo, Rául, Victor, Héctor, Fabricio. Passei momentos inesquecíveis com eles.

A os meus amigos de futebol da Moradia da Unicamp, e ao meu amigo Haroldo pelos jogos noturnos de futebol no Decatlon.

Ao grupo "Sariri Huayra", pelos momentos inesquecíveis na difusão da nossa cultura peruana.

A os meus amigos Jairo Moraes e Bruno Abreu, por todo o incentivo e pela torcida.

Ao grupo de estudos de testes, pelas sugestões e confraternidade. Foi um prazer e uma honra ter trabalhado com todos vocês.

Ao grupo de estudos do LIS, entre eles ao meu amigo Javier, por a excelente qualidade acadêmica.

Ao grupo de estudos dirigido pelo Professor Doutor Yuzo Iano da FEEC. Foi um prazer ter assistido as suas reuniões.

Aos funcionários do Instituto de Computação da Unicamp, sempre prestativos. De maneira especial para Claudia, meu agradecimento para ela pelo apoio incondicional.

Aos meus amigos peruanos em Campinas: Ana Beatriz, Ana Cecília, Baldo, Carlos, Carlitos, Charo, David, Delia, Edwin, Elvis, Fanny Roxana, Fany Verónica, Fernando, Flor, Gladys, Juan, Juan Carlos, Harold, Haroldo, Herbert, Justo, Mario, Melchor, Mónica, Pascual, Robinson, Rosalinda, Raúl, Roxana, Vladimir, Wilbert e Yuri, pela amizade.

Aos meus amigos em Arequipa: Alvaro, Cintia, Christian, Érika, Fernando, Gerardo, Herly, Juana, Milagros, Nadia, Nando, Oliver, Omar e Reynaldo, pela amizade.

Resumo

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. Neste contexto, um SIG Web é um sistema que provê diferentes serviços SIG de análise e visualização de dados espaciais através da Web.

O propósito deste trabalho é propor um processo de desenvolvimento, e um processo de teste de desempenho para aplicações SIG Web. O processo de desenvolvimento proposto integra artefatos do NDT (*Navigational Development Techniques*) e da Semiótica Organizacional. 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.

Os processos propostos, de desenvolvimento e teste de desempenho, foram aplicados 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.

Abstract

Geographic Information System (GIS) is software directed to the management of geo-referenced data. GIS are systems used to store, to analyze and to manipulate geographic data, or either, data that represent objects and phenomenon where the geographic localization is an inherent characteristic. In this study, Web GIS is a system that provide different services of GIS, such as analysis and visualization of geographic data through the Web.

The purpose of this work is to consider a development and performance testing processes for Web GIS applications. The development process proposed integrates artifacts of NDT (Navigational Development Techniques) and the Organizational Semiotics. The process of performance testing proposed deals with the inherent characteristics of Web GIS applications, such as great volume of data and load of simultaneous users, supplying a set of well detailed test activities. These activities are integrated in a robust process, where the starting point was identification of the critical use cases that the process needs to be tested. For this, we assumed that the requirements of the system had been correctly specified.

The processes proposed of development and performance testing had been applied in the evaluation of the Web Maps system, a Web GIS application whose goal is helping its users in the agricultural planning from interest regions.

Sumário

INTRODUÇÃO **1**

1.1 CONTEXTO **1**

1.2 MOTIVAÇÃO **2**

1.3 OBJETIVOS **3**

1.4 ESTRUTURA DO TEXTO **3**

ENGENHARIA WEB **5**

2.1 A EVOLUÇÃO DA WEB **5**

2.2 ENGENHARIA WEB **6**

2.2.1 DEFINIÇÕES **6**

2.2.2 ARQUITETURA **7**

2.2.3 CLASSIFICAÇÃO **8**

2.3 SIG WEB **9**

2.3.1 SISTEMAS DE INFORMAÇÃO GEOGRÁFICA **9**

2.3.3 ARQUITETURA DE SISTEMAS DE INFORMAÇÃO GEOGRÁFICA **10**

2.3.2 SISTEMAS DE INFORMAÇÃO GEOGRÁFICA NA WEB **12**

2.4 CONSIDERAÇÕES FINAIS **13**

TESTE EM APLICAÇÕES WEB **15**

3.1 TESTE **15**

3.2 FASES DE TESTE **16**

3.3 PROCESSO DE TESTE **17**

3.4 TESTES WEB **19**

3.5 TESTES SIG WEB **21**

3.6 CONSIDERAÇÕES FINAIS **21**

PROCESSO DE DESENVOLVIMENTO PARA SIG WEB **23**

4.1 CONCEITOS PRELIMINARES **23**

4.1.1 OOHDM **23**

4.1.2 SEMIÓTICA ORGANIZACIONAL 24
4.1.3 NDT 25
4.2 PROCESSO PROPOSTO 27
4.2.1 ENGENHARIA DE REQUISITOS 30
4.2.2 ANÁLISE 30
4.2.3 MODELO DE INTERFACE ABSTRATA E IMPLEMENTAÇÃO 31
4.2 CONSIDERAÇÕES FINAIS 31
PROCESSO DE TESTE DE DESEMPENHO EM SIG WEB 33
5.1 TESTES DE DESEMPENHO 33
5.2 PROCESSOS DE TESTE DE DESEMPENHO EM APLICAÇÕES WEB 34
5.2.1 PROCESSO DE TESTE DE CARGA / ESTRESSE 35
5.2.2 PROCESSO DE TESTE DE VOLUME 36
5.3 PROCESSO DE TESTE DE DESEMPENHO EM APLICAÇÕES SIG WEB 36
5.3.1 PROCESSO PROPOSTO DE TESTE DE CARGA / ESTRESSE 36
5.3.2 PROCESSO PROPOSTO DE TESTE DE VOLUME 37
5.3.3 PROCESSO PROPOSTO DE TESTE DE DESEMPENHO 38
5.4 CONSIDERAÇÕES FINAIS 40
ESTUDO DE CASO PROCESSO DE DESENVOLVIMENTO PARA SIG WEB 41
6.1 WEBMAPS 41
6.2 ENGENHARIA DE REQUISITOS 43
6.3 ANÁLISE 50
6.3 MODELO DE INTERFACE ABSTRATA E IMPLEMENTAÇÃO 56
6.4 CONSIDERAÇÕES FINAIS 57
ESTUDO DE CASO TESTE DE DESEMPENHO EM SIG WEB 59
7.1 WEBMAPS 59
7.2 APLICAÇÃO DO PROCESSO 60
7.2.1 PLANEJAR TESTE DE DESEMPENHO (FASE 0) 61
7.2.2 PLANEJAR TESTE DE CARGA (FASE 1A) 62

7.2.3 PLANEJAR TESTE DE VOLUME (FASE 1B) 63

7.3 RESULTADOS OBTIDOS 64

7.2 CONSIDERAÇÕES FINAIS 67

CONCLUSÕES E TRABALHOS FUTUROS 69

8.1 CONTRIBUÇÕES DO TRABALHO 69

8.2 TRABALHOS FUTUROS 70

***ANEXO A – GUIA DE REFERÊNCIA DO NDT* 75**

***APÊNDICE A – DOCUMENTO DE REQUISITOS DO SISTEMA (DRS) - NDT* 99**

IDENTIFICAÇÃO E DEFINIÇÃO DOS OBJETIVOS DO SISTEMA 99

IDENTIFICAÇÃO E DEFINIÇÃO DOS REQUISITOS DE ARMAZENAMENTO DE INFORMAÇÃO 99

IDENTIFICAÇÃO E DEFINIÇÃO DOS ATORES 103

IDENTIFICAÇÃO E DEFINIÇÃO DOS REQUISITOS FUNCIONAIS 105

IDENTIFICAÇÃO E DEFINIÇÃO DOS REQUISITOS DE INTERAÇÃO 107

VALIDAÇÃO DOS REQUISITOS 109

***APÊNDICE B – DOCUMENTO DE ANÁLISE DO SISTEMA (DAS) – NDT* 111**

REALIZAÇÃO DO MODELO CONCEITUAL 111

REALIZAÇÃO DO MODELO NAVEGACIONAL 118

Lista de Figuras

Figura 1: Estrutura do texto. **4**

Figura 2: Tempo levado para chegar a 30% da população americana [MUR00]. **6**

Figura 3: Arquitetura Web [SAU05]. **8**

Figura 4: Arquitetura de Sistemas de Informação Geográfica [CCH+96]. **11**

Figura 5: Topologia de um SIG Web. **13**

Figura 6: Fases de Teste no Processo de Desenvolvimento. **17**

Figura 7: Processo de Teste. **18**

Figura 8: Teste de Desempenho em aplicações SIG Web. **21**

Figura 9: Processo de desenvolvimento do OOHDM. **24**

Figura 10: Relação entre os modelos do NDT. **26**

Figura 11: Ciclo de vida do NDT. **27**

Figura 12: Processo proposto. **28**

Figura 13: Processo geral. **29**

Figura 14: Análise no NDT. **30**

Figura 15: Fases no processo de desenvolvimento vs Tipos de teste (modelo V). **34**

Figura 16: Processo de teste de desempenho em aplicações Web. (a) Processo proposto por Molinari [MOL03] e (b) processo proposto por Menascé [MV02]. **35**

Figura 17: (a) Adaptação apartir do processo proposto por Molinari [MOL03]
(b) Adaptação do processo proposto por Menascé [MV02]. **37**

Figura 18: Processo proposto de teste de carga / estresse. **37**

Figura 19: Processo proposto de teste de volume. **38**

Figura 20: Processo de teste de desempenho em aplicações SIG Web. **39**

Figura 21: Arquitetura do sistema WebMaps. **42**

Figura 22: Modelo de Atores. **45**

Figura 23: Diagrama de casos de uso [BBS+05]. **47**

Figura 24: Modelo conceitual. **51**

Figura 25: Modelo navegacional básico. **52**

Figura 26: Modelo de ontologia. **54**

Figura 27: Organização de Conteúdo [BBS+05]. **55**

Figura 28: Modelo navegacional final. **56**

Figura 29: Interface Abstrata. **57**

Figura 30: Processo de teste de desempenho em aplicações SIG Web. **60**

Figura 31: Processo proposto de teste de carga / estresse. **62**
Figura 32: Processo proposto de teste de volume. **63**
Figura 33: Interface de resultados no Apache JMeter. **66**

Lista de Tabelas

Tabela 1: Fases de Teste no Processo de Desenvolvimento. **16**
Tabela 2: OBJ – 01. **43**
Tabela 3: OBJ – 02. **43**
Tabela 4: OBJ – 03. **43**
Tabela 5: RA – 04. **44**
Tabela 6: AT – 02. **45**
Tabela 7: AT – 03. **45**
Tabela 8: Matriz de incompatibilidade de atores. **46**
Tabela 9: RF – 01. **48**
Tabela 10: FR – 03. **49**
Tabela 11: PV – 03. **49**
Tabela 12: Matriz de rastreabilidade. **50**
Tabela 13: CL - 01. **51**
Tabela 14: NO - 01. **53**
Tabela 15: Caso de uso - Cadastrar propriedade. **61**
Tabela 16: Relação na fase 2 entre o processo de carga e volume. **63**
Tabela 17: Instância do processo proposto. **64**

Capítulo 1

Introdução

O primeiro capítulo deste texto se inicia explicando o contexto do trabalho. Em seguida se explica a motivação para a execução do trabalho. Depois para tornar mais claros os resultados esperados, são traçados os objetivos. Finalmente, a estrutura do texto é detalhada para melhor guiar a leitura.

1.1 Contexto

Na atualidade as aplicações Web alcançaram uma grande importância devido à rápida difusão da Internet. Além disso, a relevância econômica das aplicações Web aumenta a importância de controlar e melhorar sua qualidade.

Como conseqüência, uma alta demanda emergiu para metodologias e ferramentas para garantia da qualidade de sistemas baseados em Web [RT01]. Esta demanda de metodologias e ferramentas para sistemas baseados na Web cria o que é chamado de "Engenharia para a Web" (Web Engineering).

Os aplicativos para a Web são diferentes de outras categorias de software e têm características exclusivas: são dirigidos a conteúdo, estão em constante evolução, têm curto prazo de desenvolvimento, dentre outras. Dentre este tipo de aplicativos para Web temos as aplicações SIG (Sistemas de Informação Geográfico) Web. As aplicações SIG 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 [ARO89, BUL94].

Para a construção de software de qualidade é preciso seguir um processo de desenvolvimento maduro. Atualmente existem diferentes modelos de processos para construir aplicações Web, porém para SIG's Web a carência destes processos é notório.

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 uma navegação adequada para estes tipos de usuários. Neste cenário, precisa-se de um processo de desenvolvimento que lide com essas características.

Além disso, para se garantir a qualidade dos sistemas precisa-se da realização de testes ao longo do processo de desenvolvimento. Assim como na Engenharia de Software, na Engenharia para a Web as atividades de teste também visam encontrar erros. De fato, os testes nos aplicativos para a Web, dentre eles os SIG's Web, são um desafio ainda maior; pois estes aplicativos podem executar em diferentes browsers, sistemas operacionais e plataformas de hardware.

Molinari mostra a relação entre risco de teste versus requisito de qualidade em aplicações Web [MOL03, capítulo 7], 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. No contexto dos SIG's Web, este requisito é ainda mais crítico tendo em vista o grande volume de dados gerenciados. Sendo assim, se faz necessário um processo de teste de desempenho que valide este requisito.

1.2 Motivação

Servem como motivação para o presente trabalho:

- ✓ A importância atual das aplicações Web e o surgimento das aplicações SIG Web. Esses sistemas permitem o desenvolvimento de aplicações em que relacionamentos espaciais são importantes. Além do que, as áreas de aplicação destes sistemas são diversas, e abrangem desde telecomunicações, planejamento urbano ou arquitetura à indústria do turismo e entretenimento. Outra área de aplicação importante aqui no Brasil é a agricultura.
- ✓ Embora existam diversas metodologias para desenvolvimento de aplicações Web, percebe-se a falta de modelos de processos de desenvolvimento utilizados em aplicações SIG Web.
- ✓ A importância da aplicabilidade dos testes nos sistemas de software para assegurar sua qualidade e a necessidade de guias de teste para aplicações Web e SIG Web.

- ✓ Falta de um processo de teste de desempenho específico para aplicações SIG Web.

1.3 Objetivos

O trabalho relatado neste texto tem dois objetivos principais. O primeiro é a concepção de uma proposta de processo metodológico de desenvolvimento para aplicações SIG Web. O segundo objetivo é propor um processo de teste de desempenho para aplicações SIG Web.

Há ainda outros objetivos secundários.

- ✓ Usar o projeto WebMaps para validar as propostas mencionadas nos objetivos principais. O WebMaps é um projeto interdisciplinar em andamento no Instituto de Computação da Unicamp suportado com o financiamento de CNPq, que tem como objetivo permitir o monitoramento e planejamento de safras agrícolas no Brasil.
- ✓ Gerar o Documento de Requisitos do Sistema (DRS) e o Documento de Análise do Sistema (DAS) na aplicação do processo de desenvolvimento no WebMaps.
- ✓ Apresentar um guia de referencia da técnica NDT (Navigational Development Techniques) [ESC04], útil na fase de engenharia de requisitos e a análise para sistemas de software baseados na navegação.
- ✓ Integrar alguns artefatos da Semiótica Organizacional [LIU00] no processo de desenvolvimento do WebMaps.
- ✓ Mostrar a utilidade de ferramentas livres na automatização dos testes.

1.4 Estrutura do texto

A estrutura do restante do texto está ilustrada na Figura 1, esta sugere dois caminhos alternativos para a leitura do texto. O caminho da esquerda contém o processo de desenvolvimento de aplicações SIG Web e o caminho da direita, o processo de testes de desempenho em aplicações SIG Web.

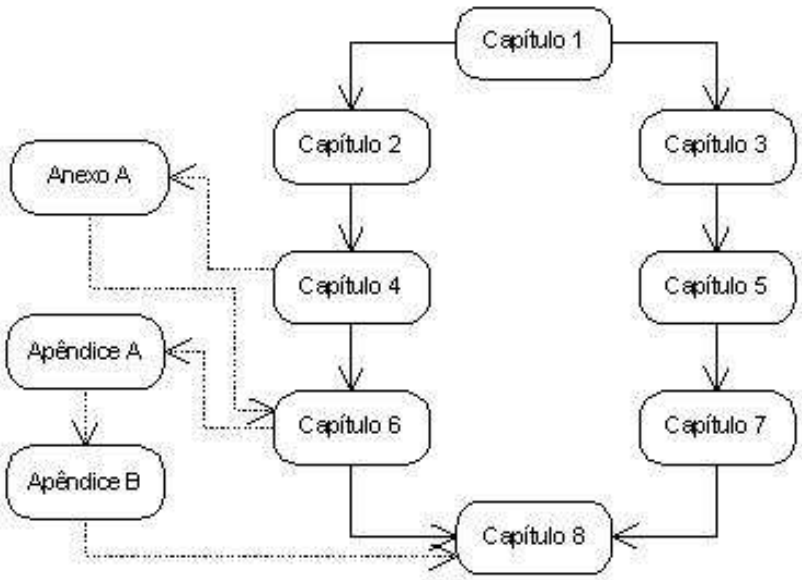


Figura 1: Estrutura do texto.

A estrutura do texto está formada pelos seguintes capítulos:

- ✓ No Capítulo 2, Aplicações Web,
- ✓ No Capítulo 3, Testes Web,
- ✓ No Capítulo 4, Processo de desenvolvimento para SIG Web,
- ✓ No Capítulo 5, Processo de teste de desempenho em SIG Web,
- ✓ No Capítulo 6, Estudo de caso – Processo de desenvolvimento,
- ✓ No Capítulo 7, Estudo de caso – Processo de teste de desempenho,
- ✓ No Capítulo 8, Conclusões e Trabalhos Futuros,
- ✓ No Anexo A, Guia de referência do NDT,
- ✓ No Apêndice A, Documento de Requisitos do Sistema (DRS) – WebMaps,
- ✓ No Apêndice B, Documento de Análise do Sistema (DAS) – WebMaps.

Capítulo 2

Engenharia Web

Os aplicativos para a Web são diferentes de outras categorias de software e têm características exclusivas: são dirigidos a conteúdo, estão em constante evolução, têm curto prazo de desenvolvimento, dentre outras. O grande desafio passa a ser adaptar as técnicas existentes para uso na Web, criando assim o que é chamado de "Engenharia para a Web". Este capítulo apresenta os principais conceitos Web e um estudo das técnicas que vêm sendo adotadas pelos desenvolvedores de sistemas para a Web e para aplicações SIG Web.

2.1 A evolução da Web

A Web foi inicialmente concebida com o intuito de compartilhar informações científicas entre alguns poucos cientistas. O conteúdo era estático e apenas textual, não havia imagens, sons, animações ou conteúdo gerado dinamicamente para cada usuário. A interação era limitada, a navegabilidade era fácil, alto desempenho era desejável, mas não essencial, os *sites* eram desenvolvidos por apenas uma pessoa ou um pequeno grupo [BRE02].

Mas a Web evoluiu, assim Murugesan destaca que a Internet levou apenas quatro anos para estar em 30% dos lares americanos (veja-se Figura 2). É um tempo bem curto quando comparado a outros produtos: o telefone levou 40 anos, o rádio levou 35 anos, o videocassete demorou 20 anos, a televisão 26 anos e o próprio computador levou 19 anos [MUR00] [GIN01].

Figura 2: Tempo levado para chegar a 30% da população americana [MUR00].

Além disso, o escopo e a complexidade das aplicações Web foram aumentando. Pequenos serviços foram cedendo espaço para grandes aplicativos e com isso também aumentou a complexidade dos projetos e as dificuldades de desenvolvimento, manutenção e gerenciamento. Tudo feito com pouca disciplina, sem preocupação com técnicas e métodos padronizados ou maneiras de controlar a qualidade.

A atitude predominante era: "Vamos fazer rápido, não há tempo para planejar", e acabou gerando aplicativos com grande probabilidade de ter problemas como baixo desempenho e falhas. Na Web, problemas como esses são ainda mais graves que no software tradicional, pois como afirma Nielsen: a distância entre um *site* e seus concorrentes é sempre de apenas poucos "cliques" [NIE00].

A falta de planejamento, projetos mal feitos e a falta de gerenciamento acabam tendo conseqüências sérias. Segundo Ginige e Murugesan, 84% dos sistemas entregues não atendem às necessidades do cliente; 79% dos projetos são entregues com atrasos e 63% têm custo maior que o orçamento previsto. Mais de 50% dos sistemas prontos são de baixa qualidade e faltam funcionalidades necessárias [GM01].

Como resultado, desenvolvedores e usuários começaram a se preocupar com a maneira como sistemas Web complexos são criados, bem como com seus níveis de desempenho, qualidade e integridade. E aí surge a Engenharia para a Web.

2.2 Engenharia Web

2.2.1 Definições

Segundo Pressman, sistemas e aplicativos da Web são caracterizados por disponibilizar grande quantidade de conteúdo e funcionalidade para grande população de usuários. A Engenharia para a Web é, portanto, o processo utilizado para criar aplicativos

Web de alta qualidade. A Engenharia para Web não é igual à Engenharia de Software tradicional, mas ambas compartilham muitos conceitos e princípios fundamentais, com ênfase nas mesmas técnicas de gerenciamento e atividades. Há pequenas diferenças na maneira como essas atividades são conduzidas, mas a filosofia que dita uma abordagem disciplinada para o desenvolvimento de um sistema de computador é a mesma. [PRE01, capítulo 29]

Ao mesmo tempo em que adota muitos princípios da Engenharia de Software, a Engenharia para a Web incorpora novas abordagens, metodologias, ferramentas, técnicas e normas para atender aos requisitos exclusivos dos sistemas para a Web. Também é um erro achar que o desenvolvimento de aplicativos para Web é apenas a criação de páginas utilizando HTML (*HyperText Markup Language*). Assim como é equivocado achar que desenvolvimento para a Web envolve apenas a manipulação de diversas mídias e criação de conteúdo. O desenvolvimento para a Web é uma mistura de publicações impressas e desenvolvimento de software, entre marketing e computação, entre comunicações internas e relações externas, e entre arte e tecnologia. [GIN01] [MUR00]

2.2.2 Arquitetura

A Figura 3 mostra o modelo em camadas genérico da arquitetura de uma aplicação Web [SAU05]. As camadas são as seguintes:

Camada de Apresentação: Neste caso o Cliente é o Navegador utilizado pelo usuário, quer seja o Internet Explorer, quer seja o Netscape Navigator, ou outro navegador qualquer.

Camada Web: Passa informações para o Servidor Web. A interface pode ser composta de páginas HTML, ASP, ou qualquer outra tecnologia capaz de gerar conteúdo para o Navegador. Com isso, alterações na camada de Apresentação são feitas diretamente no servidor Web, sendo que estas alterações estarão, automaticamente, disponíveis para todos os Clientes. Com isso não existe a necessidade de reinstalar a aplicação em todos os computadores da rede cada vez que uma alteração for feita na camada de Apresentação. Fica muito mais fácil garantir que todos estão tendo acesso à versão mais atualizada da aplicação. A única coisa que o cliente precisa ter instalado na sua máquina, é o navegador adequado. O acesso ao Banco de dados é feito através do Servidor de aplicações.

Camada de Aplicação: São as regras do negócio, as quais determinam de que maneira os dados serão utilizados. Esta camada está no Servidor de Aplicações. Desta maneira, quando uma regra do negócio for alterada, basta atualizá-la no Servidor de Aplicações. Após a atualização, todos os usuários passarão a ter acesso a nova versão, sem que seja necessário reinstalar o programa em cada um dos computadores da rede. É

importante observar que, ao centralizar as regras do negócio em um Servidor de Aplicações, estamos facilitando a tarefa de manter a aplicação atualizada.

Camada de Dados: Nesta camada temos o Servidor de Banco de dados, no qual reside toda a informação necessária para o funcionamento da aplicação.

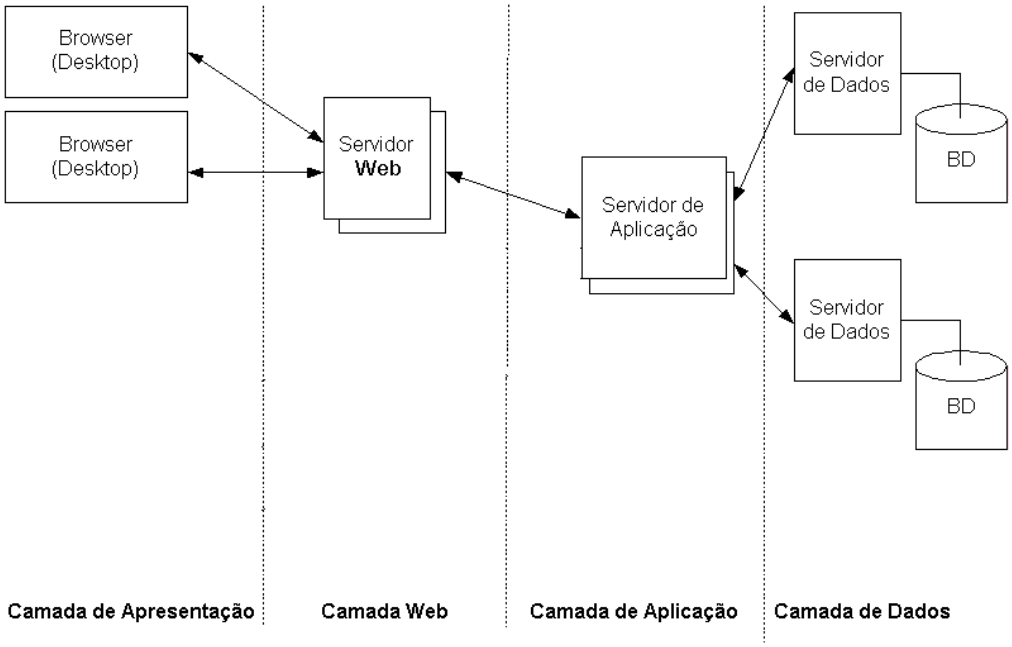


Figura 3: Arquitetura Web [SAU05].

2.2.3 Classificação

Dentre as principais aplicações existentes na Engenharia Web pode-se citar as seguintes [CCU05] (a lista não pretende ser exaustiva) :

(a) Website: Aplicação web que publica informações bem específicas sobre um único tema. Exemplo: Website sobre Química do Estado Sólido.

(b) Portal: Aplicação web que publica grandes volumes de informações de uma organização e presta serviços. Exemplo: Portal Unicamp que publica conteúdos sobre ensino pesquisa e extensão em várias áreas do conhecimento.

(c) Hotsites promocionais: Pequena aplicação web compostas de uma ou duas páginas para divulgação de produtos e eventos.

(d) Sistemas de feiras virtuais: Aplicações web de serviços e negócios on-line.

(e) Museus virtuais: Aplicações web de serviços culturais on-line.

(f) Sistemas de controle de vôos: Aplicações web que dão suporte e controlam reservas, torres de controle, lista de espera, reservas on-line, entre outros.

(g) Bibliotecas de bases de dados integradas: Aplicações web que gerenciam o acesso a bases de dados bibliográficos hospedadas em servidores distintos.

(h) Tradutores on-line: Aplicações web de serviços de bases de dados de dicionários que oferecem tradução on-line.

(i) Livros on-line (e-livro): Aplicações web que gerenciam estruturas de dados cujo conteúdo são livros digitalizados disponíveis para leitura on-line.

(j) Sistemas de ensino a distância: Aplicações Web que gerenciam ambientes de ensino a distância a partir de conteúdos inseridos por instrutores.

(k) Sistemas de comércio eletrônico: Aplicações web de serviços de vendas on-line.

(l) Sistemas de informação: Aplicações Web que gerenciam informações a partir de dados gerados por usuários.

(m) Sistemas de informação geográfica (SIG Web): Aplicações Web que gerenciam informações a partir de dados geográficos. A proxima seção trata este tipo de sistemas.

2.3 SIG Web

2.3.1 Sistemas de Informação Geográfica

Os Sistemas de Informação Geográfica – SIG – 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 à informação e indispensável para analisá-la [ARO89, BUL94].

SIG's comportam diferentes tipos de dados e aplicações em várias áreas do conhecimento. A título de exemplo podemos citar: otimização de tráfego, controle cadastral, gerenciamento de serviços de utilidade pública, demografia, cartografia, administração de recursos naturais, monitoramento costeiro, controle de epidemias, planejamento urbano. A utilização de SIG facilita a integração de dados coletados de fontes heterogêneas, de forma transparente para o usuário final. Os usuários não estão restritos aos especialistas em um domínio específico – cientistas, gerentes, técnicos, funcionários de administração de diversos níveis e o público em geral vêm usando tais sistemas com frequência cada vez maior [CCH+96].

Dentre as referências genéricas sobre SIG merecem destaque especial os dois volumes de Geographical Information Systems, editados por D. J. Maguire, M. F. Goodchild e D. W. Rhind [MGR93], e a série de relatórios técnicos do National Center for Geographic Information Analysis – NCGIA [NCG89]. Outros exemplos são as referências [ANT91, ARO89, PM90]. No Brasil, devem ser lembrados os textos [AS93, PAR94, TEX92]. As referências [GUN90, NCG89, THI93] contêm diretrizes gerais para pesquisa em SIG.

2.3.3 Arquitetura de Sistemas de Informação Geográfica

Numa visão abrangente, pode-se considerar que um SIG tem os seguintes componentes: interface com usuário; entrada e integração de dados; funções de processamento; visualização e plotagem; e armazenamento e recuperação de dados. A Figura 4 [CCH+96] indica o relacionamento entre estes componentes. Cada sistema, em função de seus objetivos e necessidades, implementa estes componentes de forma distinta, mas todos estão usualmente presentes num SIG.

Historicamente, o primeiro tipo de *interface* utilizado foi a linguagem de comandos que, mesmo podendo possuir um grande poder expressivo, torna-se complexa na medida em que o sistema cresce em funcionalidade, o que dificulta o seu uso. Para contornar este problema, muitos sistemas dispõem de facilidade para a criação de macro-comandos. A disponibilidade de ambientes computacionais interativos deu origem a interfaces baseadas em menus. Mais fáceis de operar, estas interfaces tendem a ter menor poder expressivo.

Existem quatro formas principais de *entrada de dados* em SIG: a entrada de dados via caderneta de campo, a digitalização em mesa, a digitalização ótica e a leitura de dados na forma digital, incluindo a importação de dados em outros formatos. Historicamente, muitos levantamentos (topográficos, parâmetros de solos e outros) utilizam cadernetas de campo para armazenar os resultados, que naturalmente devem ser inseridos no sistema após um controle de qualidade. Com o advento do GPS (*Global Positioning System*), tornou-se possível realizar trabalhos de campo com alto grau de precisão e com registro digital direto. A digitalização de mesa ainda é o modo mais utilizado para a entrada de dados a partir de mapas existentes.

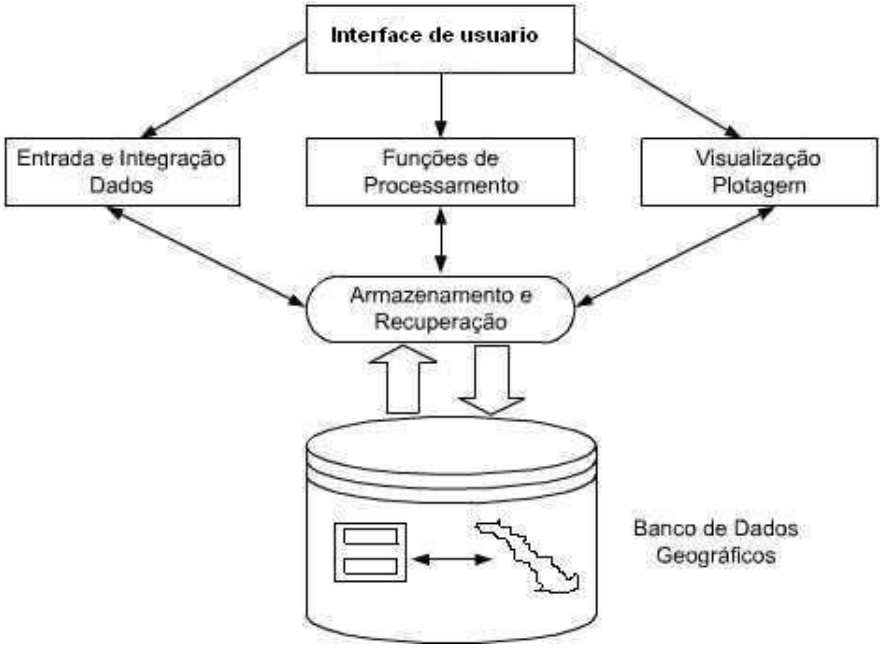


Figura 4: Arquitetura de Sistemas de Informação Geográfica [CCH+96].

Na importação de dados digitais é muito importante aproveitar o investimento já feito, eventualmente por outras instituições, na coleta e armazenamento de dados geográficos. No Brasil, as principais fontes de dados são as bases do IBGE, do INPE e do Centro de Cartografia Automatizada do Exército.

As funções de *processamento* são naturalmente dependentes dos tipos de dados envolvidos. A análise geográfica engloba funções como superposição, ponderação, medidas (área, perímetro), mapas de distância, tabulação cruzada, dentre outras. O processamento digital de imagens envolve funções como retificação, contraste, filtragem, realce e classificação. Modelos numéricos de terreno permitem a geração do modelo a partir de pontos esparsos ou linhas, entre outras funções. Operações sobre redes incluem caminhos ótimos, caminhos críticos e ligação topológica. Já as consultas aos bancos de dados podem ser espaciais ou não.

Os ambientes de *visualização* de um sistema são conseqüências do paradigma adotado para a interface. Quanto à produção cartográfica, alguns sistemas dispõem de recursos gráficos altamente sofisticados de apresentação, englobando a definição de uma área de plotagem, colocação de legendas, textos explicativos e notas de crédito. Já os aparecimentos de padrões "de fato", como o Postscript e o HPGL (do inglês *Hewlett-Packard Graphic Language*), vêm facilitar o desenvolvimento de funções de plotagem.

Os dados de um SIG são geralmente organizados sob a forma de um banco de dados geográficos. Tradicionalmente, os SIG's armazenavam os dados geográficos em arquivos

internos. Este tipo de solução vem sendo substituído pelo uso cada vez maior de SGBD (Sistema de Gestão do Bando de Dados).

Estes componentes se relacionam de forma hierárquica. No nível mais próximo ao usuário, a interface homem – computador define como o sistema é operado e controlado. No nível intermediário, um SIG deve ter mecanismos de processamento de dados espaciais (entrada, edição, análise, visualização e saída). No nível mais interno do sistema, um sistema de gerência de bancos de dados geográficos oferece armazenamento e recuperação dos dados espaciais e seus atributos.

De uma forma geral, as funções de processamento de um SIG operam sobre dados em uma área de trabalho definida pelo usuário. A ligação entre os dados geográficos e as funções de processamento do SIG é feita por mecanismos de seleção e consulta que definem restrições sobre o conjunto de dados, que podem ser espaciais ou não.

2.3.2 Sistemas de Informação Geográfica na Web

Um Sistema de Informação Geográfica reune um conjunto de ferramentas destinadas ao processamento de dados geo-referenciados desde sua coleta até a geração de produtos finais.

Um SIG Web é um sistema que oferece diferentes serviços SIG de análise e visualização de dados espaciais através da Web [DM02]. Um SIG Web se caracteriza basicamente por permitir a visualização e manipulação de informações geográficas através da Internet. Além disso, pode permitir também a realização de pesquisas, simples ou complexas, seleções (diretas ou por parâmetros predefinidos), cálculo de distâncias e áreas, impressão de mapas, entre outros.

A Figura 5 mostra a topologia de um SIG Web. Note que uma característica importante deste tipo de sistemas é a necessidade de um servidor de mapas, ou seja, alguém que forneça os mapas para um posterior processamento e geração de informações no servidor Web e apresentadas ao usuário pela Internet.

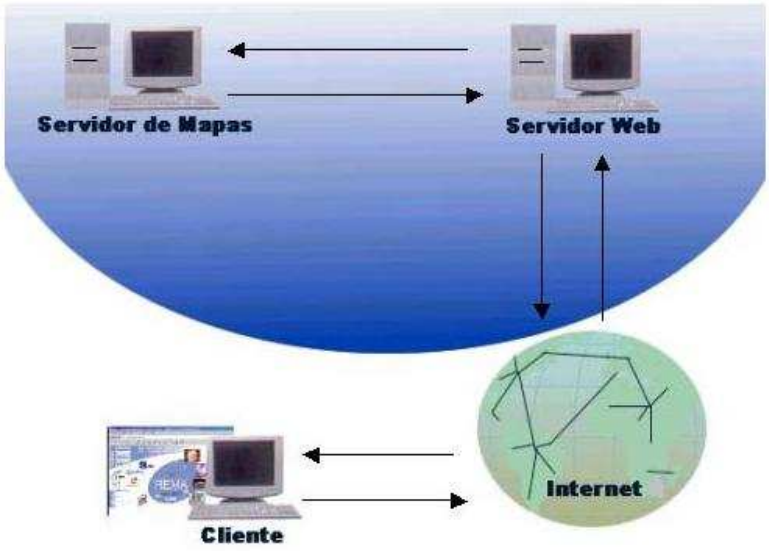


Figura 5: Topologia de um SIG Web.

2.4 Considerações Finais

Neste capitulo foram apresentados os conceitos que envolvem a Engenharia Web, ou seja, sistemas e aplicativos da Web que são caracterizados por disponibilizar grande quantidade de conteúdo e funcionalidade para grande população de usuários.

Foi também apresentada a arquitetura Web através de um modelo em camadas e também a arquitetura de um SIG. Um SIG é um sistema automatizado 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.

Neste contexto, um SIG Web é um sistema que provê diferentes serviços SIG de análise e visualização de dados espaciais através da Web

Mostrou-se uma classificação das aplicações existentes na Web onde o objetivo principal foi situar os SIG Web.

Por fim, a seção 2.3 terminou com a definição dos SIG Web e mostrou-se sua topologia. As características importantes desta topologia foram a presença de um servidor de mapas e o uso da Internet para a disponibilização das informações.

Capítulo 3

Teste em aplicações Web

A maioria dos processos de software incorpora de alguma forma, atividades de teste em suas definições. O objetivo do teste é revelar a presença de erros. Esse tipo de atividade deveria ser aplicado em todo o ciclo de vida do software. Este capítulo inicia com os conceitos principais na atividade de teste e termina com as principais atividades de teste nos processos de software para aplicações SIG Web.

3.1 Teste

Embora os testes tenham um importante papel na aferição da qualidade de um produto e sua execução de forma incompleta ou inadequada possa causar muitos problemas, eles ainda são deixados de lado por algumas organizações. Freqüentemente, atrasos nas atividades de desenvolvimento são compensados através da redução ou eliminação das atividades de teste. De uma forma resumida, a atividade de teste permite a realização de melhorias no processo utilizado, ao mesmo tempo em que auxiliam a garantir a qualidade do produto sendo desenvolvido.

"O teste consiste em executar o programa com a intenção de encontrar erros (bugs)". [MYE79]

Existe uma série de requisitos existentes em uma especificação que só podem ser testados através da execução de um programa. Existem basicamente duas maneiras de se construírem testes baseados na execução: método de *caixa branca* e método de *caixa preta*.

No método caixa branca, também chamado de teste estrutural, são testados os caminhos lógicos através do software, fornecendo casos de teste que exercitam conjuntos específicos de condições e/ou garantem que todos os caminhos independentes dentro de um módulo tenham sido exercitados pelo menos uma vez. Executa todas as decisões lógicas

para valores falsos ou verdadeiros. Executa todos os laços em suas fronteiras. Exercita as estruturas de dados internas.

Já os testes de caixa preta, também chamados de testes funcionais, são usados para demonstrar que as funções do software são operacionais, que a entrada é adequadamente aceita e a saída é corretamente produzida; que a integridade das informações externas é mantida. É uma atividade complementar aos testes de caixa branca, com a finalidade de descobrir tipos/classes de erros. Procura descobrir erro em funções incorretas ou ausentes, erros de interface, erros nas estruturas de dados, acesso a bancos de dados externos, erros de desempenho e erro de inicialização e término.

3.2 Fases de Teste

Existem diversas fases de teste que podem ser realizadas durante o processo de desenvolvimento do software. Em [GPV+05] é apresentada uma classificação possível e a Tabela 1 mostra uma adaptação sem a Fase de Teste de Implantação.

Tabela 1: Fases de Teste no Processo de Desenvolvimento.

Teste	Fase de realização	Descrição
Testes de unidade	Durante a construção do sistema.	Testam o projeto e o comportamento de cada componente do sistema uma vez construído.
Testes de integração	Durante a construção do sistema.	Testam a correta relação entre os componentes do sistema através de suas interfaces e se eles cumprem com a funcionalidade estabelecida.
Testes de sistema	Depois da construção do sistema.	Testam o sistema comprovando sua funcionalidade e atributos de qualidade. O sistema é testado em um ambiente o mais parecido possível ao ambiente operacional.
Testes de aceitação	Durante a implantação no ambiente de produção.	Avaliam que o sistema cumpre com todos os requisitos indicados e permite que os usuários do sistema dêem sua aceitação.
Testes de regressão	Depois de realizar as mudanças no sistema	O objetivo é comprovar que as mudanças sobre um componente do sistema não geram erros adicionais em outros componentes não modificados.

A Figura 6, mostra como estas fases se acoplam no processo de desenvolvimento do sistema em um modelo conhecido como modelo “V” [MAR99]. Embora este modelo de testes seja discutível, ele é utilizado neste trabalho para fim didático.

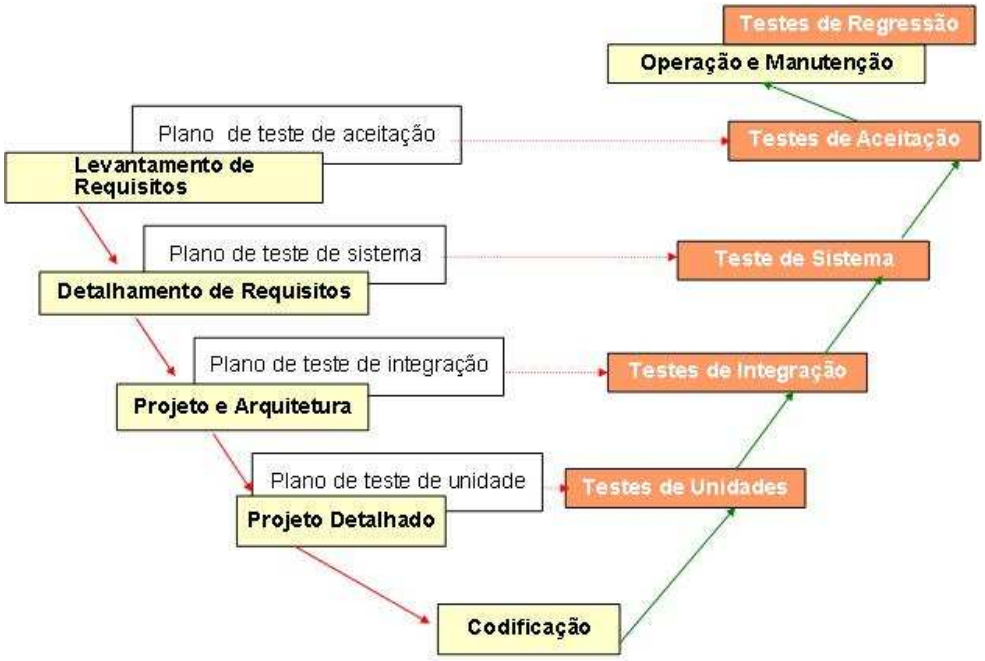


Figura 6: Fases de Teste no Processo de Desenvolvimento.

3.3 Processo de Teste

O processo de cada uma das fases do teste é composto de uma seqüencia de atividades, conforme ilustrado na Figura 7. Essa figura foi adaptada a partir do processo definido em [PAU01], que por sua vez se baseou na norma para testes de unidade do IEEE (IEEE Std 1008 – Std. For Software Unit Testing, 1987). Apesar de terem sido propostas para os testes de unidade, essas atividades são aplicáveis em todas as fases de teste.

As atividades são apresentadas resumidamente a seguir [MAR05].

- ✓ **Planejamento**

O planejamento dos testes enfoca os aspectos gerenciais do processo, que inclui:

(i) Identificação das áreas de risco que devem ser testadas. (ii) Determinação das fontes a serem utilizadas para a elaboração dos casos de testes. (iii) Determinação dos ítens a testar. (iv) Estabelecimento das condições de completeza dos testes para cada ítem a testar. (v) Especificação das condições de término dos testes (término normal e anormal). (vi) Determinação dos recursos necessários (pessoas, hardware, software, ferramentas de teste). (vii) Determinação do cronograma de testes. A saída dessa fase é um conjunto de Planos de Teste, um para cada fase de testes.

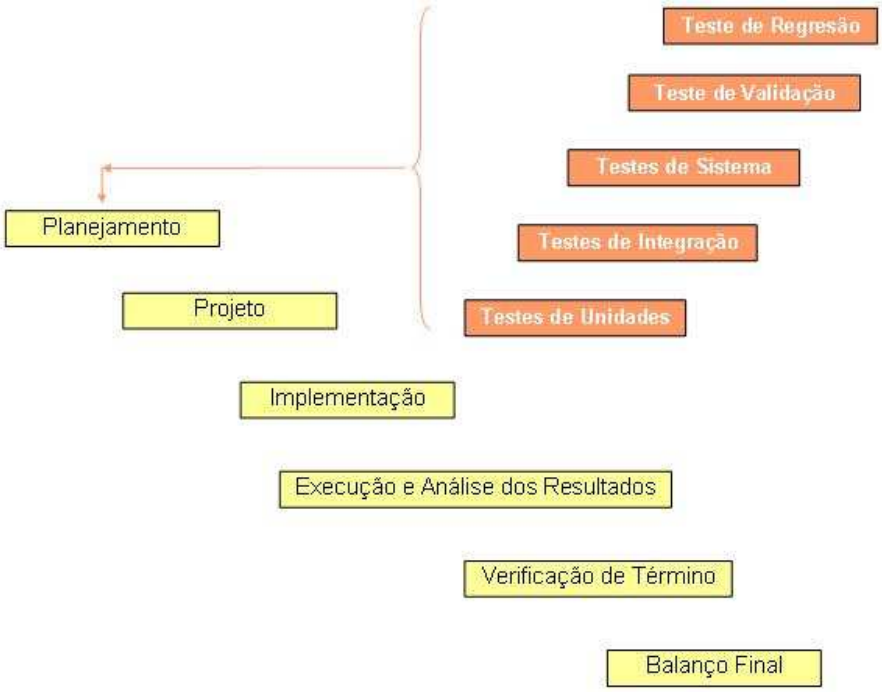


Figura 7: Processo de Teste.

- ✓ **Projeto**

Nesta atividade são obtidos os casos de teste, formados por dados (entradas) de teste e pelas saídas esperadas. Os casos de teste são derivados a partir de um modelo de base [SIE96], sendo que os mais comuns são uma especificação (testes caixa preta) ou o próprio código (testes caixa branca).

O número de testes serão realizados de forma automatizada, os procedimentos e casos de teste podem ser especificados em uma linguagem (ou notação) específica de uma ferramenta de testes. Em junho de 2001 foi iniciada na OMG (Object Management Group), uma proposta de especificação de testes denominada Perfil de Testes UML (UML Testing Profile ou UTP). Empresas como a IBM, Ericsson, Telelogic, entre outras, formaram um consórcio para o desenvolvimento do UTP. O UTP é baseado nos conceitos de UML 2.0, e se baseia em iniciativas já existentes tais como o TTCN-3 (Tree and Tabular Combined Notation) [TTC01].

- ✓ **Implementação**

Nesta atividade prepara-se o ambiente de testes, disponibilizando os recursos previstos nos planos. Os ítens a testar são instalados e configurados bem como as ferramentas de apoio. Os testes podem ser realizados de forma manual ou automática. Caso

sejam automatizados, os procedimentos e casos de teste devem ser escritos na notação específica da ferramenta de testes, ou na mesma linguagem usada para implementar o ítem a testar.

- ✓ **Execução e Análise dos Resultados**

Nesta atividade os testes implementados são executados e os resultados observados são devidamente registrados. Esse registro pode ser feito diretamente, através da captura dos dados de entrada fornecidos a partir de um computador, os quais são armazenados juntamente com as saídas observadas. Pode-se também instrumentar o código, inserindo "sensores" para observar e armazenar o estado do programa.

A análise dos resultados compreende tanto a avaliação da qualidade do software quanto a qualidade dos testes aplicados. Avaliar a qualidade do teste implica em determinar sua eficiência em encontrar falhas no software, bem como o quanto do modelo de base os casos de teste permitiram exercitar.

- ✓ **Verificação de Término**

Essa atividade determina se as condições de completeza dos testes foram satisfeitas. Os resultados da avaliação da qualidade dos testes citada na atividade precedente são úteis nesse caso. O mesmo com relação à qualidade do software.

Caso os resultados não sejam considerados satisfatórios, deve-se projetar testes suplementares.

- ✓ **Balanço Final**

Nessa atividade é feita uma avaliação final dos testes aplicados. O Relatório de Testes é completado com as lições aprendidas, a qualidade do produto, a qualidade final dos testes, entre outros. Pode-se também identificar os artefatos que serão reutilizados em outros testes.

3.4 Testes Web

Assim como na Engenharia de Software, na Engenharia para a Web as atividades de teste também visam encontrar erros.

Pressman apresenta uma abordagem que adota os princípios básicos para o teste de qualquer software e aplica estratégias e táticas análogas àquelas recomendadas para sistemas orientados a objetos [PRE01, capítulo 29]:

(a) O modelo de conteúdo é revisto para descobrir erros. Esta atividade de teste é similar em muitos aspectos com a revisão de documentos impressos. Um site grande pode

utilizar os serviços de um revisor profissional que irá descobrir erros de tipografia e gramática, consistência do conteúdo, as representações gráficas, dentre outros.

(b) O modelo de projeto é revisto para descobrir erros de navegação. Cada cenário é exercitado de acordo com o projeto de arquitetura e navegação. Isto serve para encontrar erros de navegação onde o usuário não consegue chegar ao nó desejado. Além disso, cada elo é testado para garantir que correspondem ao que foi especificado na SNU (do inglês *Semantic Navigation Unit*), subestruturas navegáveis para cada tipo de usuário.

(c) Componentes selecionados passam por um processo de teste de unidade. Nos aplicativos para a Web o conceito de unidade muda. Cada página contém conteúdo, elos (*links*), formulários, scripts, entre outros. Nem sempre é possível testar cada uma dessas características individualmente. Em muitos casos, a menor unidade testável é a página. No software tradicional o teste de unidade é focado em detalhes de algoritmo de um módulo e dos dados que fluem pela interface do módulo. Nos aplicativos para a Web este teste é focado pelo conteúdo, processamento e elos de navegação que estão nas páginas.

(d) A arquitetura é construída e testes de integração são conduzidos. A estratégia para teste de integração depende da arquitetura que foi escolhida. Se for utilizada uma arquitetura linear (seqüência previsível de interações), de grade (o conteúdo pode ser organizado categoricamente) ou hierárquica (o usuário pode navegar por toda a hierarquia, não apenas na vertical, mas também na horizontal) é possível integrar as páginas da mesma maneira que fazemos com software tradicional. Porém, se foram utilizadas arquiteturas combinadas ou estrutura de rede (as páginas são projetadas de modo que possam passar para virtualmente qualquer outra página), o teste de integração passa a ser similar a abordagem usada para orientação a objetos.

(e) O aplicativo já integrado é testado em sua funcionalidade geral e conteúdo fornecido. Assim como na validação de software convencional, o teste de sistema para a Web é focado nas ações do usuário e nas saídas do sistema para o mesmo. Para ajudar na construção de testes de sistema o testador deve se basear em casos de uso.

(f) O aplicativo é implementado em diferentes configurações de ambientes e testado em sua compatibilidade com cada configuração. São definidos todos os prováveis sistemas operacionais, navegadores, plataformas de hardware e protocolos de comunicação. Testes são conduzidos para descobrir erros associados com cada uma das possíveis configurações.

(g) O aplicativo é testado por uma população controlada de usuários. São selecionados usuários que representem cada tipo de usuário que o sistema terá. O aplicativo é testado por estes usuários e os resultados de suas interações são avaliados para encontrar erros de conteúdo e navegação, questões de usabilidade e compatibilidade, bem como questões de desempenho e confiabilidade do aplicativo.

3.5 Testes SIG Web

A aplicabilidade dos testes Web é válida para os sistemas SIG Web, porém existe uma fase que merece maior ênfase, que é a fase de testes de sistema, especificamente no teste de desempenho devido ao tratamento pesado de informações, típico neste tipo de aplicações. A Figura 8 mostra os tipos de teste de desempenho existentes.

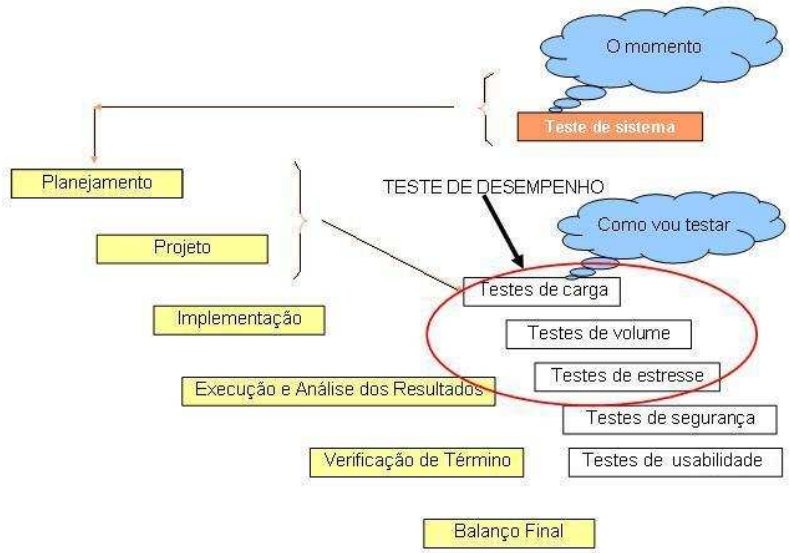


Figura 8: Teste de Desempenho em aplicações SIG Web.

Os testes de carga testam o desempenho do sistema levando-se em conta uma carga de usuários simultâneos real. Os *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 [PRE01, capítulo 18]. Estes conceitos serão mais detalhados no Capítulo 5.

3.6 Considerações Finais

Vimos que o teste é uma atividade fundamental à garantia da qualidade do software. Foram apresentados os objetivos dos testes, sua importância e fases. Os testes são classificados em "caixa branca" quando focam no código fonte do programa para a escolha dos testes a serem executados; e em "caixa preta" quando focam apenas nas funcionalidades que devem ser atendidas pelo software. Mostraram-se as diferentes fases de teste existente e como elas são acopladas em um processo de desenvolvimento de software.

No que se refere aos testes de aplicações Web, vimos que até o presente muitos dos aspectos sistemáticos de testes não têm sido suficientemente investigadas e muitas questões ainda necessitam ser tratadas. Um dos aspectos que tornam os testes dessas aplicações um desafio é a diversidade de tecnologias envolvidas na construção das mesmas.

No referente às aplicações SIG Web, se mostrou a importância dos testes de desempenho, os quais serão tratados com maior ênfase no Capítulo 5.

Capítulo 4

Processo de Desenvolvimento para SIG Web

Este capítulo apresenta uma proposta de processo de desenvolvimento para aplicações SIG Web. A proposta integra diferentes técnicas: alguns modelos da Semiótica Organizacional [LIU00] e do NDT (do inglês *Navigational Development Techniques*) [ESC04]. O processo baseia-se nos modelos clássicos propostos pelo OOHDM [ROS96] para desenvolvimento Web.

4.1 Conceitos preliminares

Nesta seção apresentam-se os conceitos preliminares que servem como base para a elaboração da proposta. As justificativas da escolha destas técnicas serão dadas na próxima seção.

4.1.1 OOHDM

O OOHDM [ROS96] é talvez uma das propostas mais estudadas. Esta proposta de desenvolvimento web baseia-se no HDM (Hypermedia Design Model) [GSP93], porém dentro do paradigma de orientação a objetos.

A figura 9 mostra o processo de desenvolvimento do OOHDM. Este processo se inicia com a criação de classes conceituais, que representam a estrutura estática do sistema. Em seguida se realiza o modelo de navegação ou modelo navegacional, que é uma visão do modelo conceitual e mostra como se poderá navegar através da informação representada. Na terceira fase se realiza o modelo de interface abstrata, que oferece uma visão de como se apresentará a informação ao usuário. Na última fase se realiza a implementação de acordo com os modelos obtidos.

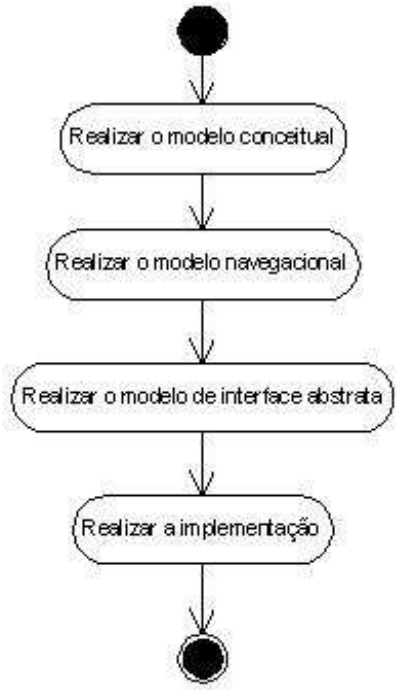


Figura 9: Processo de desenvolvimento do OOHDM.

4.1.2 Semiótica Organizacional

Como "doutrina dos signos", Semiótica [LIU00] cobre várias disciplinas facilitando o entendimento sobre as formas que as pessoas usam signos para todos os tipos de propósitos. Todo comportamento organizado é afetado pela comunicação e interpretação de signos por pessoas, individualmente ou em grupos. Assim, Semiótica Organizacional é o estudo de organizações utilizando-se de conceitos e métodos da Semiótica.

A Semiótica Organizacional fornece métodos para compremder o contexto organizacional. Um desses métodos é o MEASUR (Methods for Eliciting, Analysing and Specifying Users' Requirements) [STA97], conjunto de métodos orientados a normas para modelagem e especificação de requisitos de sistemas de negócios para desenvolvimento de software.

Entre os métodos da metodologia MEASUR, existem os Métodos de Articulação de Problema (PAM), Método de Análise Semântica (SAM) e o Método de Análise de Norma (NAM).

Para o interesse deste capítulo apresenta-se o método PAM, que compreende um conjunto de métodos que podem ser aplicados no estágio inicial de um projeto quando alguém encontra um problema vago, complexo. Entre as técnicas que compreendem o PAM temos:

- Análise de *Stakeholders*: Faz uma análise dos atores responsáveis, os clientes, os fornecedores, os parceiros, os concorrentes, os espectadores e os legisladores.
- Quadro de Avaliação: Verifica por que algumas das sistemáticas aplicadas não estão sendo eficientes e quais são os principais problemas destas sistemáticas.
- *Framework* Semiótico: Especifica características no nível do mundo físico, empírico, sintático, semântico, pragmático e mundo social.
- Análise Colateral: Auxilia na análise e articulação dos relacionamentos entre sistemas unitários que compreendem sistemas complexos.

Também foi considerado o uso do método SAM, para elicitar e representar os requisitos num modelo formal e preciso, que é o Diagrama de Ontologia. O Modelo Ontológico delineia um contexto que envolve conceitos e terminologias utilizados no domínio de um problema particular.

Por fim, apresenta-se uma adaptação da Semiótica Organizacional, feita pela equipe do IC [BSS+05], esta adaptação compõe-se dos seguintes métodos:

- Descrição do contexto do problema.
- Descrição dos requisitos funcionais e não funcionais.
- Organização de Conteúdo: Específica a estrutura e organização do conteúdo no sistema.

4.1.3 NDT

O NDT [ESC04] é uma metodologia baseada na navegação e compreende a etapa de Engenharia de Requisitos e a etapa de Análise. A figura 10 mostra a relação entre os seus modelos.

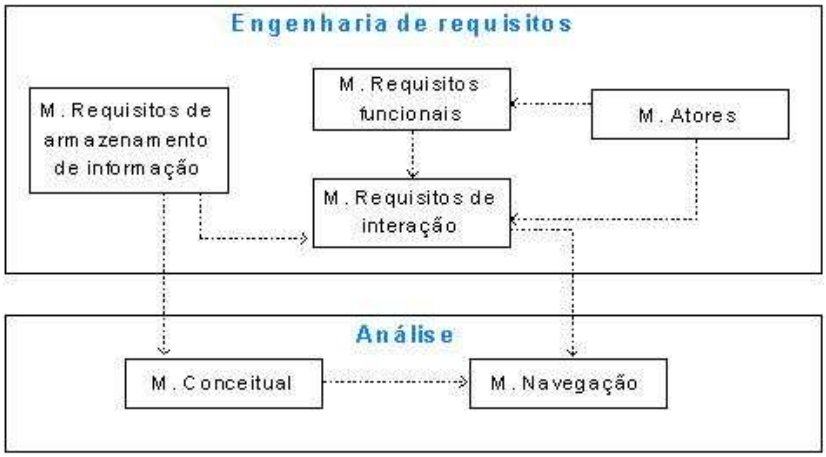


Figura 10: Relação entre os modelos do NDT.

O modelo de requisitos de armazenamento de informação especifica as necessidades de informação do sistema e responde às perguntas: Quais informações o sistema deve armazenar? Com que informação vai trabalhar o sistema?

Os requisitos funcionais especificam as possibilidades funcionais do sistema, e visa responder à pergunta: o que se pode fazer no sistema? Os requisitos não funcionais ao contrário dos funcionais, não expressam nenhuma função a ser implementada em um sistema de informações; eles expressam condições de comportamento e restrições que devem prevalecer. O modelo de atores especifica os tipos de atores que interatuam com o sistema, as incompatibilidades entre atores e possíveis generalizações.

O modelo de requisitos de interação é um modelo importante para um sistema de navegação, pois o sistema não só deve oferecer ao usuário a informação e a funcionalidade que este solicite, mas também o momento oportuno e a forma apropriada de como apresentar esta informação. A Figura 11 mostra o ciclo de vida do NDT, onde os produtos principais das suas fases são do Documento de Requisitos do Sistema (DRS) e o Documento de Análise do Sistema (DAS).

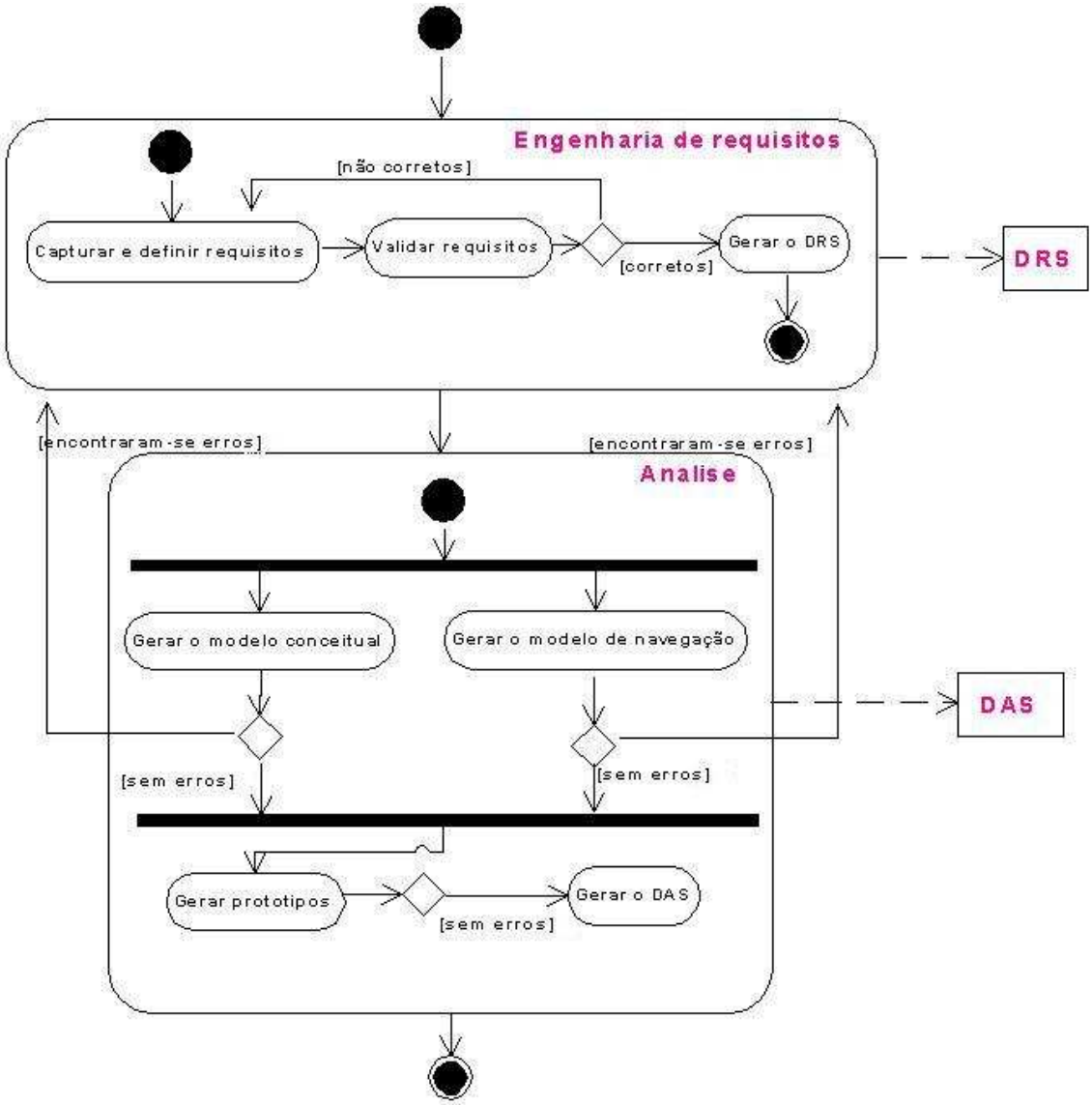


Figura 11: Ciclo de vida do NDT.

4.2 Processo proposto

O processo combina as fortalezas de NDT e da Semiótica Organizacional, balançando a debilidade de uma com a fortaleza da outra.

NDT é uma metodologia baseada na navegação do usuário; ela tem técnicas bem estruturadas durante as fases que ela trata: engenharia de requisitos e análise. NDT é uma metodologia madura, pois tem sido aplicada em diversos projetos de software reais de maneira satisfatória [EPL+05] [TER05] [GEM+04].

Uma desvantagem de NDT é cobrir o processo de desenvolvimento até a fase de análise somente. Outra das desvantagens de NDT é não ter acoplado uma técnica de levantamento de requisitos no seu processo.

Ante as carências de NDT, propõe-se o uso de Semiótica Organizacional como complemento. Semiótica Organizacional tem uma ótima técnica de levantamento de requisitos [BSS+05]. Além disso, provê de técnicas e ferramentas úteis para a fase de projeto. A Semiótica Organizacional é um complemento ideal de NDT.

A Figura 12 mostra a proposta de desenvolvimento para aplicações SIG Web; ela é explicada segundo os modelos clássicos de desenvolvimento web do OOHDM explicados na seção 4.2.1.

A proposta usa estes modelos, porém para chegar a obter estes modelos usa outro processo diferente daquele usado no OOHDM.

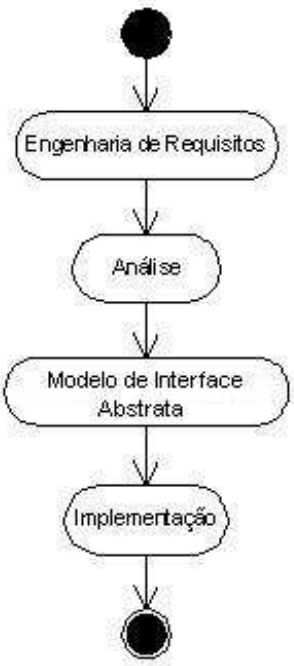


Figura 12: Processo proposto.

De maneira geral, o processo proposto se mostra na Figura 13. Se inicia com uma fase de Engenharia de Requisitos, em seguida a fase de Análise, o modelo de interface abstrata e finalmente a implementação do sistema. Nas seções seguintes se detalha como são complementados os modelos do NDT e da Semiótica Organizacional seguindo este processo geral.

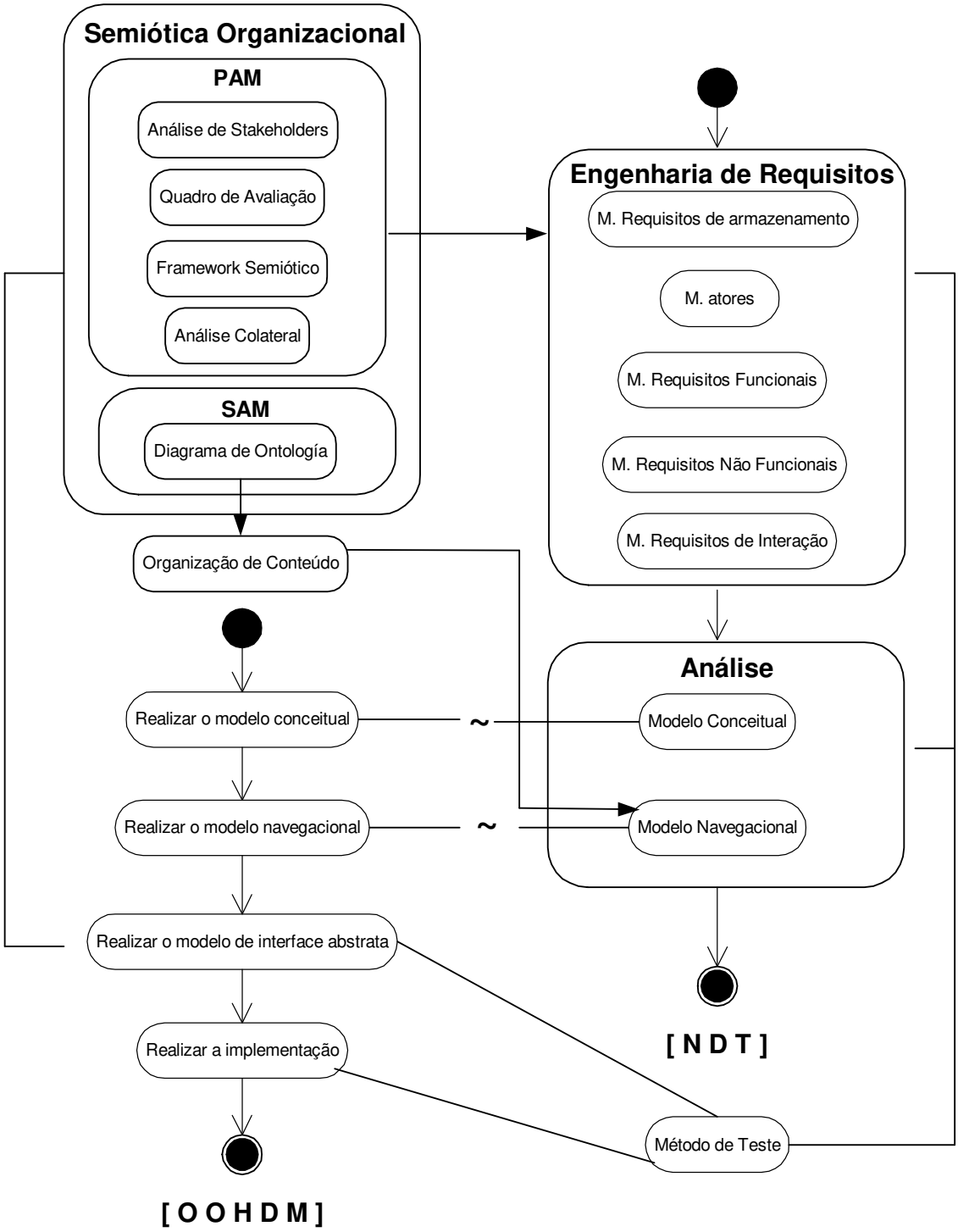


Figura 13: Processo geral.

É importante ressaltar que esta proposta não é destinada unicamente para aplicações SIG Web, pode ser utilizada também em outros tipos de aplicações Web.

A seguir apresenta-se as fases da proposta identificando e especificando o uso de uma ou outra técnica.

4.2.1 Engenharia de Requisitos

A Engenharia de Requisitos tem como objetivo encontrar as reais necessidades do usuário para a futura implementação do software. Pesquisas têm comprovado que muitos projetos de implementação do software têm falhado por problemas de elicitação (do inglês *elicitation*, descobrir, obter informação sobre alguma coisa obscura) dos requisitos do software [BOE81], ou seja, os requisitos obtidos muitas vezes são incompletos ou mal entendidos.

Levando em conta este problema se propõe para a fase de elicitação de requisitos usar a Semiótica Organizacional [LIU00], que usa uma técnica madura para a obtenção de requisitos, MEASUR [STA97]. Esta etapa envolve uma série de *workshops* com todas as pessoas envolvidas, cujo foco é encontrar os atores, requisitos funcionais e não funcionais do sistema considerando questões sociais, políticas, culturais e éticas para compreender o problema alvo [BSS+05].

Obtidos os requisitos funcionais e não funcionais da Semiótica Organizacional, propõe-se a metodologia NDT (Navigational Development Techniques) [ESC04] para enriquecer estes requisitos com o modelo de atores, o modelo de requisitos de armazenamento de informação e o modelo de requisitos de interação.

4.2.2 Análise

Na etapa de Análise, o objetivo é detalhar, estruturar e validar os requisitos, de forma que estes possam ser usados como base para o planejamento detalhado. No NDT a Análise compreende a obtenção de dois modelos: o modelo conceitual e o modelo de navegação. A partir do modelo conceitual se pode gerar um modelo relacional útil para a implementação do banco de dados.

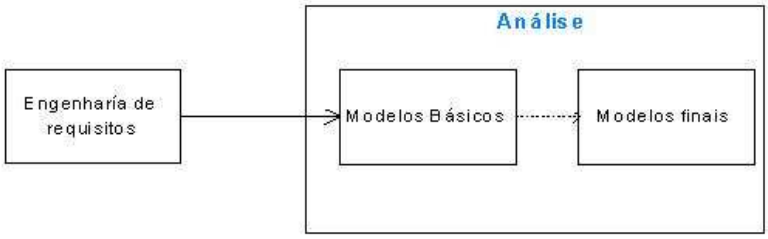


Figura 14: Análise no NDT.

A Figura 14 mostra a Fase de Análise do NDT, onde os modelos básicos devem ser gerados antes dos modelos finais.

O NDT é útil na geração do modelo de navegação para as páginas que requerem procesamento de informação, porém não é muito útil para a geração da organização do conteúdo estático. Para complementar o conteúdo estático no modelo navegacional gerado pelo NDT, se usa a Organização de Conteúdo, proposta e adaptada do Diagrama de Ontologias pela equipe de pesquisadores do Instituto de Computação da Unicamp [BSS+05].

4.2.3 Modelo de interface abstrata e implementação

Na fase do projeto do sistema ou geração do modelo de interface abstrata se faz necessário a elaboração do projeto das páginas. Nesta fase o Web Designer pode idealmente utilizar os produtos da fase de Análise. Ou seja, ele utilizará o modelo navegacional final para gerar os protótipos para o usuário e sua posterior implementação.

4.2 Considerações Finais

Este capítulo apresentou uma proposta de processo de desenvolvimento que pode ser aplicado para aplicações SIG Web. A proposta integra diferentes técnicas: alguns modelos da Semiótica Organizacional [LIU00] e o NDT (Navigational Development Techniques) [ESC04]. O processo baseia-se nos modelos clássicos propostos pelo OOHDM [ROS96] para desenvolvimento Web.

Os produtos das fases da Engenharia de Requisitos e a Análise são: o Documento de Requisitos do Sistema (DRS) e o Documento de Análise do Sistema (DAS), respectivamente. O Anexo A apresenta os passos detalhados do NDT para a geração destes documentos.

Capítulo 5

Processo de Teste de Desempenho em SIG Web

Aplicações SIG Web lidam com um grande volume de dados (convencionais e geo-referenciados) e permitem acesso a um número considerável de usuários simultaneamente (pelo fato de estar na Web). Desta forma, a realização de testes de desempenho é fundamental para assegurar que estas aplicações sejam adequadas quando da sua implantação e uso. O propósito deste capítulo 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.

5.1 Testes de desempenho

A tarefa de teste 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 [PRE01, capítulo 17].

Entre os diferentes tipos de testes de sistema existentes [MYE79], destacam-se os testes de desempenho. No capitulo 3 foram especificadas as diferentes fases de teste. Na Figura 15 são situados os testes de desempenho dentro do processo de desenvolvimento de um sistema (modelo V). Estes testes têm 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 estão integrados é que se pode ter uma medida real do seu desempenho [PRE01, capítulo 18].

Figura 15: Fases no processo de desenvolvimento vs Tipos de teste (modelo V).

Molinari mostra a relação risco de teste versus requisito de qualidade em aplicações Web [MOL03], ou seja, quais os requisitos de qualidade que mais comprometem o funcionamento do sistema caso estes não sejam testados. O desempenho é o requisito de qualidade mais importante em aplicações Web, no entanto para aplicações SIG Web esta importância é ainda mais crítica devido ao volume de dados que são gerenciados.

5.2 Processos de Teste de Desempenho em aplicações Web

Como parte dos testes de desempenho tem-se: os *testes de carga* (testam o desempenho do sistema levando-se em conta uma carga de usuários simultâneos real), *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) [PRE01]. A seguir são descritos os processos de teste de cada um deles.

5.2.1 Processo de Teste de Carga / Estresse

Estudos sobre processos de teste de desempenho em aplicações web [MIL00], [SS00], [MV02] e [MOL03] indicam a importância deste tipo de testes. A Figura 16 mostra dois processos de teste de desempenho para aplicações Web, de Molinari [MOL03] que apresenta o processo em 5 fases e de Menascé [MV02] em 7 fases. 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) onde o objetivo principal desta fase é identificar e projetar quais funcionalidades poderiam ser testadas no sistema. Já na Fase 2 procede-se à especificação e criação de usuários virtuais; uma vez criados os usuários virtuais cria-se um cenário (Fase 3) correspondente a esta quantidade de usuários. Tendo o cenário corretamente especificado, o mesmo é executado na Fase 4. Por fim, a Fase 5 compreende a análise do desempenho do sistema testado.

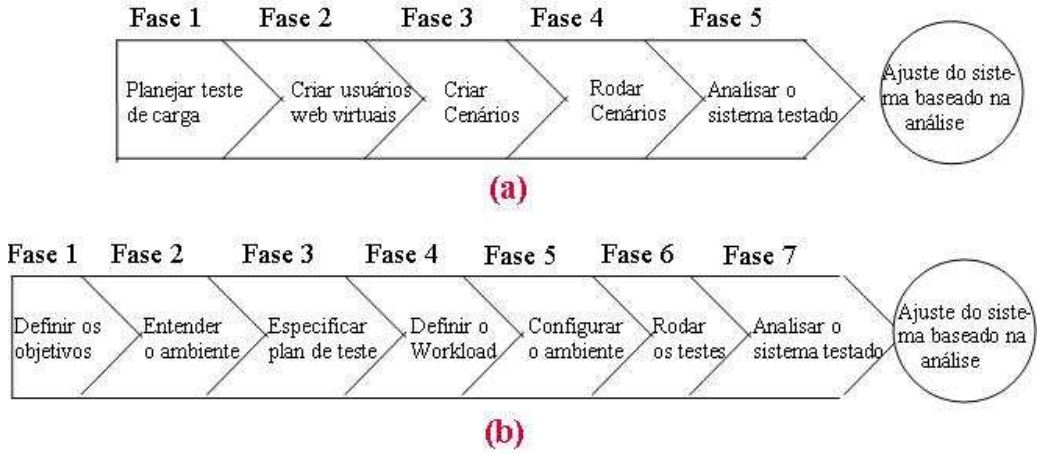


Figura 16: Processo de teste de desempenho em aplicações Web. (a) Processo proposto por Molinari [MOL03] e (b) processo proposto por Menascé [MV02].

Menascé apresenta fases mais abrangentes, para ele o processo 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 a equivalente a Criação de Cenários e a Fase de Criação de usuários virtuais de

Molinari. Depois, na Fase 5, as ferramentas de teste são instaladas e configuradas. Nas Fases 6 e 7, são executados os testes e se analisa o sistema testado, respectivamente.

Ambos os processos são finalizados com o ajuste do sistema, a partir da análise dos resultados dos testes de desempenho.

5.2.2 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 não mais realiza o serviço da forma planejada. Além disso, costumam ser utilizados na identificação da carga máxima o volume de dados que o sistema pode gerenciar em um dado período de tempo.

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

5.3 Processo 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 5.3.1 e 5.3.2). Em seguida, suas atividades são integradas em um processo único (seção 5.3.3).

5.3.1 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 5.2.1. A Figura 17(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.

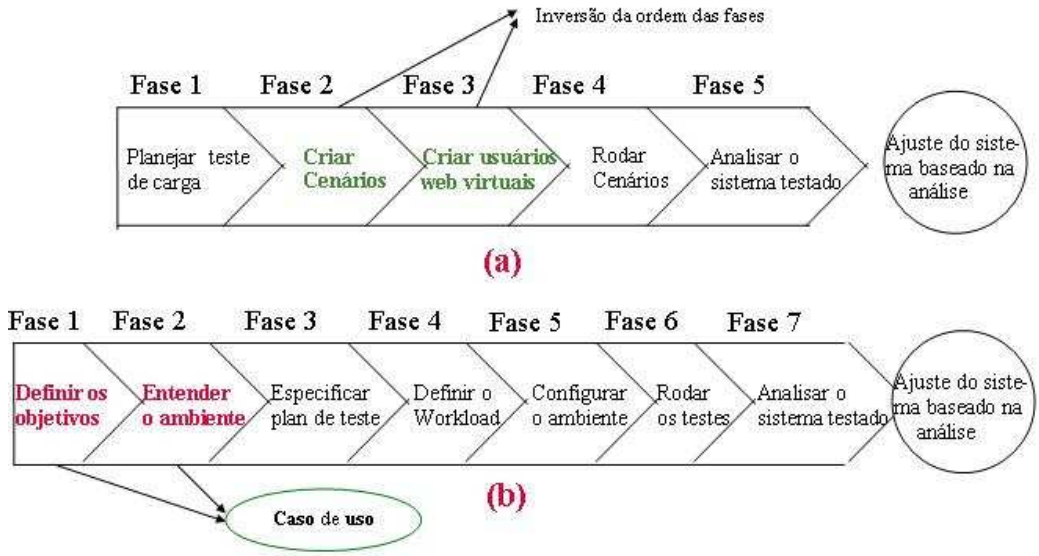


Figura 17: (a) Adaptação apartir do processo proposto por Molinari [MOL03] (b) Adaptação do processo proposto por Menascé [MV02].

A Figura 17(b) mostra que, no processo de Menascé, as fases 1 e 2 foram consideradas importantes. Estas fases foram consideradas equivalentes à definição dos casos de uso de maior risco no sistema. Assim, a Figura 18 apresenta o processo proposto de teste de carga com as mudanças mostradas na Figura 17(a), e considerando os casos de uso de maior risco na fase de planejamento, na Fase 1.

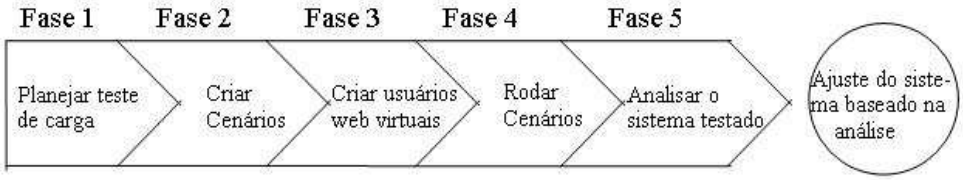


Figura 18: Processo proposto de teste de carga / estresse.

5.3.2 Processo proposto de Teste de Volume

O processo de teste de volume está ilustrado na Figura 19. 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.

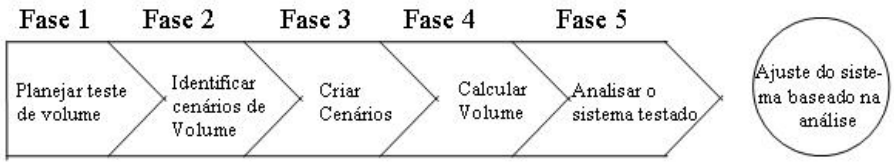


Figura 19: Processo proposto de teste de volume.

5.3.3 Processo proposto de Teste de Desempenho

Foi também definido um processo que integra os testes de carga, estresse e volume na avaliação do desempenho de um sistema SIG Web. A Figura 20 mostra o modelo proposto, usando a notação UML do diagrama de atividades [UML05]. Atividades são mostradas através de elipses enquanto setas indicam a ordem na qual as atividades deverão 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. 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 o processo mostrado na seção 5.3.1. Na parte A da Figura 20, 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 se cria os usuários web virtuais segundo cada cenário criado. Na Fase 4 são executados o(s) cenário(s) para que na última fase (Fase 5), se proceda a análise do sistema testado.

A parte B da Figura 20 mostra o processo proposto para a realização de testes de volume descrito na seção 5.3.2. Além disso, mostra como os cenários de teste de volume podem ser apoiandos pelos cenários de teste de carga ou estresse caso tenham sido criados. Os testes de volume se iniciam na Fase 1 (planejamento), onde se planeja a finalidade do teste; logo na Fase 2 identificam-se o(s) cenário(s) de volume, nesta identificação se avaliam os cenários criados no proceso paralelo de carga ou estresse, 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 de 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 que suporta. Nesta fase pode-se fazer também uma projeção deste volume de dados ao longo do tempo dependendo do planejamento na Fase 1.

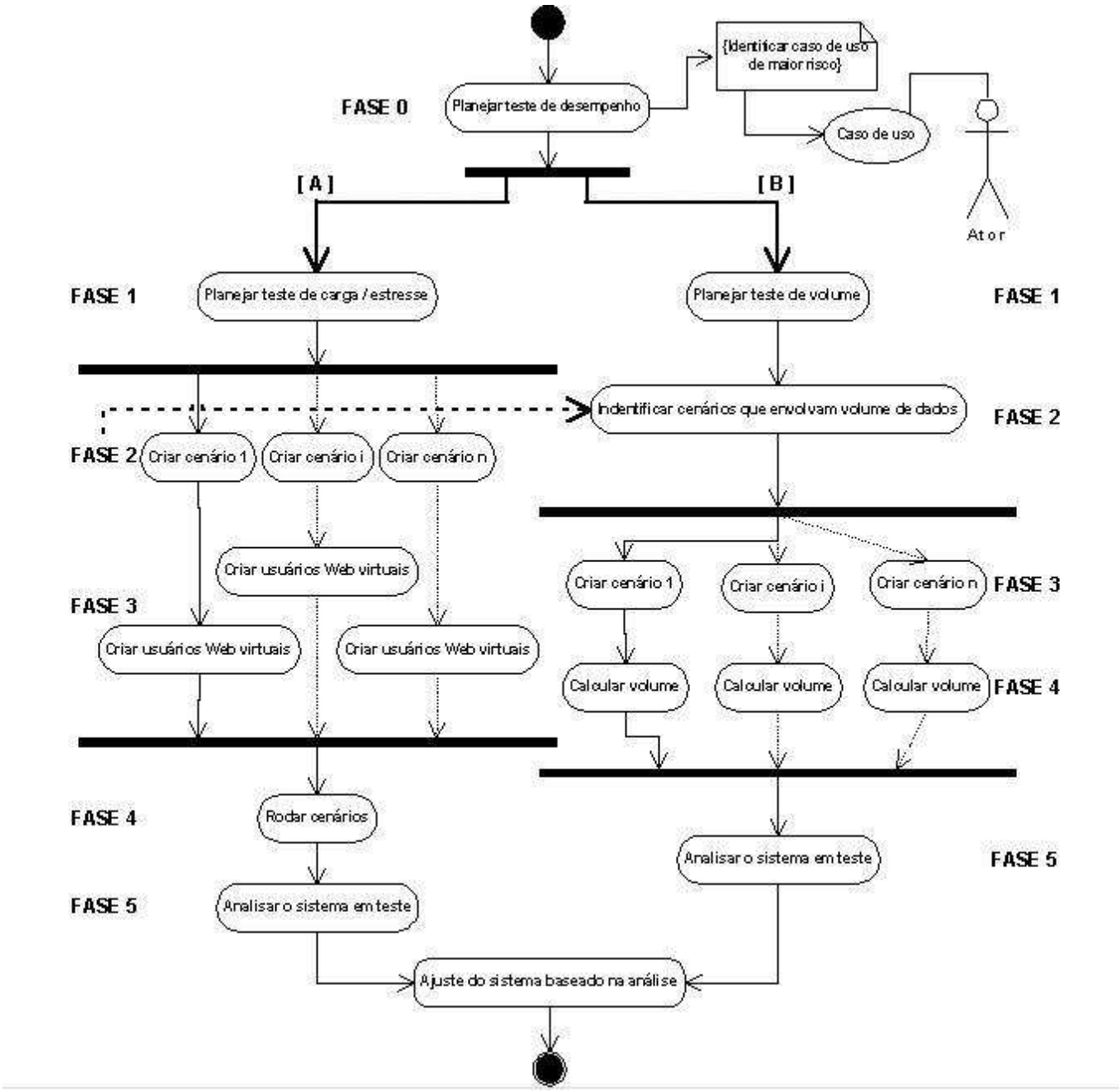


Figura 20: Processo de teste de desempenho em aplicações SIG Web.

Como última atividade em todo processo de teste de desempenho realiza-se o ajuste do sistema. Este ajuste requer a colaboração de todos os membros da equipe de testes de

desempenho. Cada um deles tem um papel importante na análise e ajuste do sistema de acordo com a sua especialidade. É necessária uma conversa entre os membros da equipe para analisar os problemas e propor soluções que tenham como objetivo melhorar o sistema.

5.4 Considerações Finais

Este capítulo apresentou a definição de um processo para realização de testes de desempenho em aplicações SIG Web. Também se apresentou uma proposta de processo de teste de volume. Este processo pode ser usado tanto para aplicações Web quanto para aplicações SIG Web.

O processo proposto inclui a realização de testes de carga, estresse e volume, a partir de casos de uso de maior risco.

O processo proposto de teste de desempenho lida com as características inerentes às aplicações SIG Web, quais sejam, 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 que lida paralelamente com os testes de carga / estresse e volume, 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.

Capítulo 6

Estudo de caso Processo de desenvolvimento para SIG Web

Este capítulo apresenta a aplicação do processo metodológico para aplicações SIG Web enunciado no capítulo 4. Para a aplicação do processo foi usado como estudo de caso, o projeto WebMaps.

6.1 WebMaps

O WebMaps é um projeto interdisciplinar em andamento no Instituto de Computação da Unicamp suportado com o financiamento de CNPq, onde também participam a CEPAGRI (Centro de Pesquisas Meteorológicas e Climáticas Aplicadas à Agricultura) e a Feagri (Faculdade de Engenahria Agricola). O projeto tem como objetivo permitir o monitoramento e planejamento de safras agrícolas no Brasil. Tem envolvidos especialistas de diversas áreas incluindo: processamento de imagens, engenharia de software, banco de dados, geo-processamento, estudos agro-ambientais e especialistas em interação humano-computador. 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 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-se a realização de consultas referentes à evolução dos plantios ao longo do tempo. O sistema gera para o usuário curvas NDVI (do inglês Normalized Difference Vegetation Index) [CPT05], que são curvas que mostram os índices de vegetação por diferença normalizada, ou seja, mostram a quantidade de vegetação 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 (do inglês *Moderate Resolution Imaging Spectroradiometer*) de todo o território brasileiro referentes a 4 anos (2001 até 2004) fornecidas pela NASA [MOD05]. Cada imagem ocupa 109,6 MB e uma nova imagem é 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 Figura 21 apresenta a arquitetura do sistema WebMaps. Ela é composta de uma camada Cliente e uma de Servidor. A parte do Cliente contém o navegador que mostra as páginas processadas pela camada servidora. O Servidor é dividido em três grandes módulos, implementados usando Java e a linguagem 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 (*Hard Disk*)).

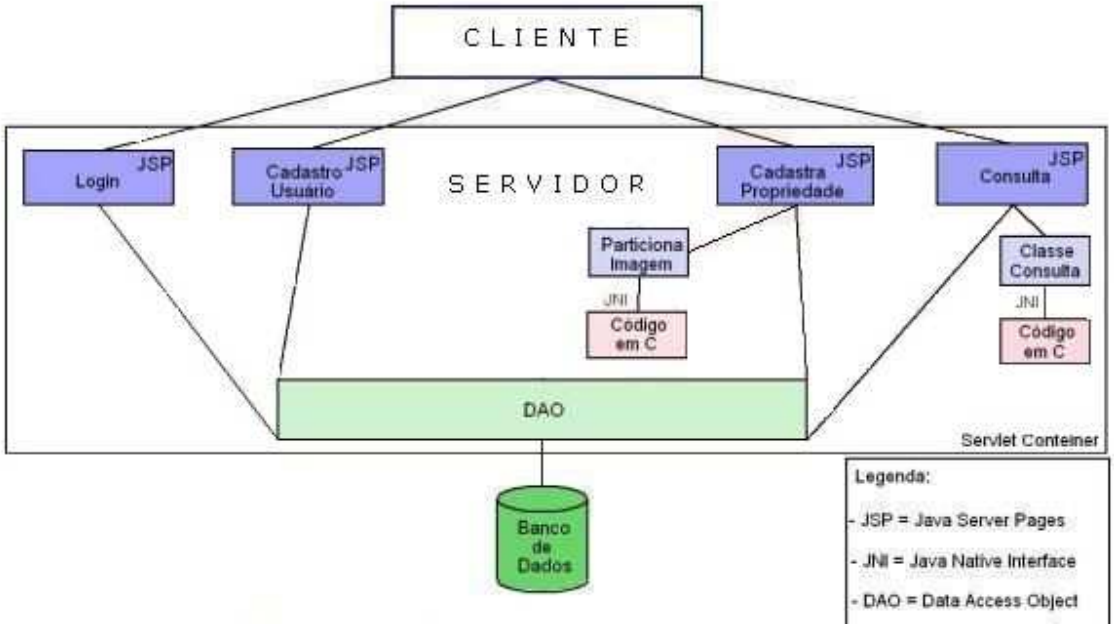


Figura 21: Arquitetura do sistema WebMaps.

Do lado do servidor, encontram-se páginas JSP, 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). Czajkowski apresenta os benefícios do uso de interfaces com códigos nativos no desenvolvimento de aplicações [CDW01].

A seguir se mostra cada uma das fases do processo proposto no Capítulo 4 aplicadas no desenvolvimento do WebMaps.

6.2 Engenharia de Requisitos

Esta fase se iniciou com a aplicação da Semiótica Organizacional [Liu, 2000]. No levantamento de requisitos foram realizados diferentes *workshops*, que foram conduzidos por facilitadores e *stakeholders* de forma participativa. Os stakeholders são aquelas pessoas envolvidas no projeto. Os *workshops* contaram com 17 participantes de diferentes áreas, incluindo processamento de imagens, engenharia de software, banco de dados, geo-processamento, estudos agro-ambientais, especialistas em interação humano-computador, representantes dos usuários, desenvolvedores de sistemas de software, entre outros. Os artefatos da Semiótica Organizacional foram usados como ferramentas de comunicação durante os *workshops*. Em [BBS+05] se apresenta maiores detalhes do uso da Semiótica Organizacional no WebMaps.

A fase de Engenharia de Requisitos usa os padrões do NDT [ESC04] para cada um dos seus modelos descritos na seção 4.1.3 do Capítulo 4. No decorrer deste capítulo não se apresentará todos os padrões, pois o objetivo é mostrar a aplicação do processo e não mostrar muito detalhe. O leitor pode recorrer aos apêndices A e B, Documento de Requisitos do Sistema (DRS) e Documento de Análise do Sistema (DAS), respectivamente, se desejar ver a especificação completa da geração de todos os padrões.

Os primeiros padrões do NDT consistem na identificação e definição dos objetivos do sistema. O intuito desta tarefa é conhecer o escopo do problema. Os objetivos do WebMaps são representados nas Tabelas 2, 3 e 4.

Tabela 2: OBJ – 01.

OBJ - 01	Fornecer dados e informações de uma determinada região geográfica.
Descrição	O sistema deverá fornecer: dados e informações, curvas ou imagens NDVI, de acordo a sua região geográfica de interesse.

Tabela 3: OBJ – 02.

OBJ - 02	Trabalhar com dados geo-refenciados
Descrição	O sistema deverá trabalhar com imagens de satélite periódicas, úteis para a geração das curvas NDVI.

Tabela 4: OBJ – 03.

OBJ – 03	Adequar o sistema ao perfil do usuário
Descrição	O sistema deverá permitir diferentes tipos ou níveis de permissões para os usuários.

Uma vez identificados os objetivos, o NDT propõe identificar os requisitos do sistema. Dentre os tipos de requisitos do sistema, encontra-se a identificação e definição dos requisitos de armazenamento de informação, que especifica a informação que deve armazenar o sistema. Na Tabela 5 mostra-se um exemplo de requisito de armazenamento de informação: uma propriedade. Especificam-se os objetivos associados assim como seus dados específicos.

Tabela 5: RA – 04.

RA – 04	Propriedade	
Objetivos associados	OBJ – 1, OBJ – 02	
Descrição	O sistema deverá armazenar as informações correspondentes a uma propriedade.	
Dados específicos	**Nome e descrição**	**Natureza**
	Código	Númerico
	Nome	Texto
	Descrição	Texto
	Coordenadas	Geo-referenciado
	Tipo	Texto
	Talhão	**RA – 06** Cardinalidade: 1..n
	Máscara	**RA – 03** Cardinalidade:1..1
	MBB (do inglês *Minimum Bounding Box*)	**RA – 05** Cardinalidade: 1..n
	Usuário	**RA – 01** Cardinalidade: 1..n
	Consulta	**RA – 07** Cardinalidade: 0..n

Uma vez especificados os requisitos de informação, precisa-se determinar os atores que interagem com o sistema. Todo ator se refere a um papel de interação com o sistema. A Figura 22 apresenta os diferentes atores no WebMaps.

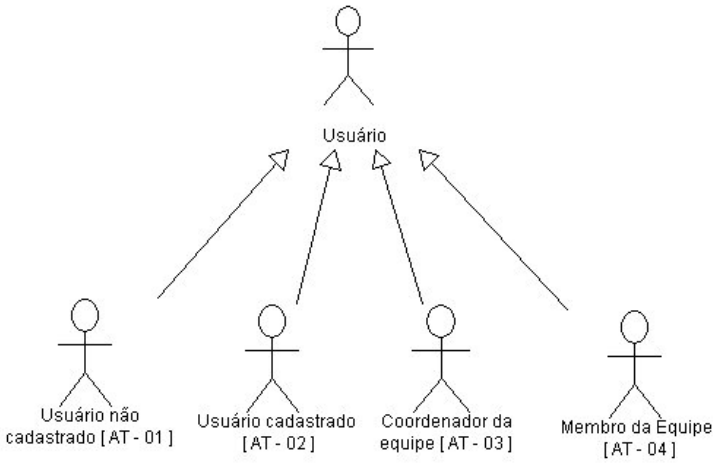


Figura 22: Modelo de Atores.

A Tabela 6 e a Tabela 7 mostram a especificação do ator usuário cadastrado e coordenador da equipe, respectivamente, seguindo o padrão proposto pela NDT. Como se mencionou anteriormente, a totalidade de padrões se encontra nos apêndices A e B.

Tabela 6: AT – 02.

AT – 02	Usuário cadastrado
Objetivos associados	OBJ – 3
Descrição	O sistema deverá prever o tratamento dos usuários que pertencem ao grupo de usuários cadastrados e que se refere a aquela pessoa que acessa no sistema e faz as consultas de acordo as coordenadas de sua região de interesse.

Tabela 7: AT – 03.

AT – 03	Coordenador da equipe
Objetivos associados	OBJ – 3
Descrição	O sistema deverá prever o tratamento dos usuários que pertencem ao grupo de usuários cadastrados e que se refere a aquela pessoa que ingressa no sistema como coordenador da equipe e tem privilégios para fazer mudanças no sistema.

Tendo especificados os atores, analisa-se a incompatibilidade entre atores. A definição de incompatibilidade é essencial, pois, cada papel incompatível pode definir uma interface completamente diferente dos outros papeis. A Tabela 8 mostra a incompatibilidade dos usuários do WebMaps; nela pode-se notar por exemplo que, o ator

cadastrado (AT - 02) tem incompatibilidade com o ator coordenador da equipe (AT - 03), ou seja, um usuário cadastrado não pode cumprir o papel do coordenador da equipe.

Tabela 8: Matriz de incompatibilidade de atores.

ATORES	AT – 01	AT – 02	AT – 03	AT – 04
AT – 01	-	X	X	X
AT – 02		-	X	
AT – 03			-	X
AT – 04				-

Legenda: : simetrico : compatível - : não tem sentido X : incompatível

A identificação e a elaboração da matriz é realizada pela equipe de analise. É útil identificar a incompatibilidade de atores pois define as navegabilidades existentes no sistema. Se todos os atores fossem compatibles, deve existir um único modelo navegacional.

Nas atividades anteriores foram definidos o que se armazena no sistema e quem pode fazer uso desta informação. Mas também se precisa saber o que se pode fazer com a informação e as possibilidades funcionais, ou seja, serviços que o sistema pode prestar a seus usuários.

Para a captura e definição das necessidades funcionais, o NDT propõe duas técnicas: a utilização do diagrama de casos de uso e os diagramas de informação textual, especificada mediante padrões, que aclara o seu significado.

A Figura 23 apresenta o modelo de casos de uso. Este modelo foi gerado graças às atividades realizadas nos *workshops* da Semiótica Organizacional no levantamento de requisitos. Mostra-se a interação dos tipos de usuário com o sistema.

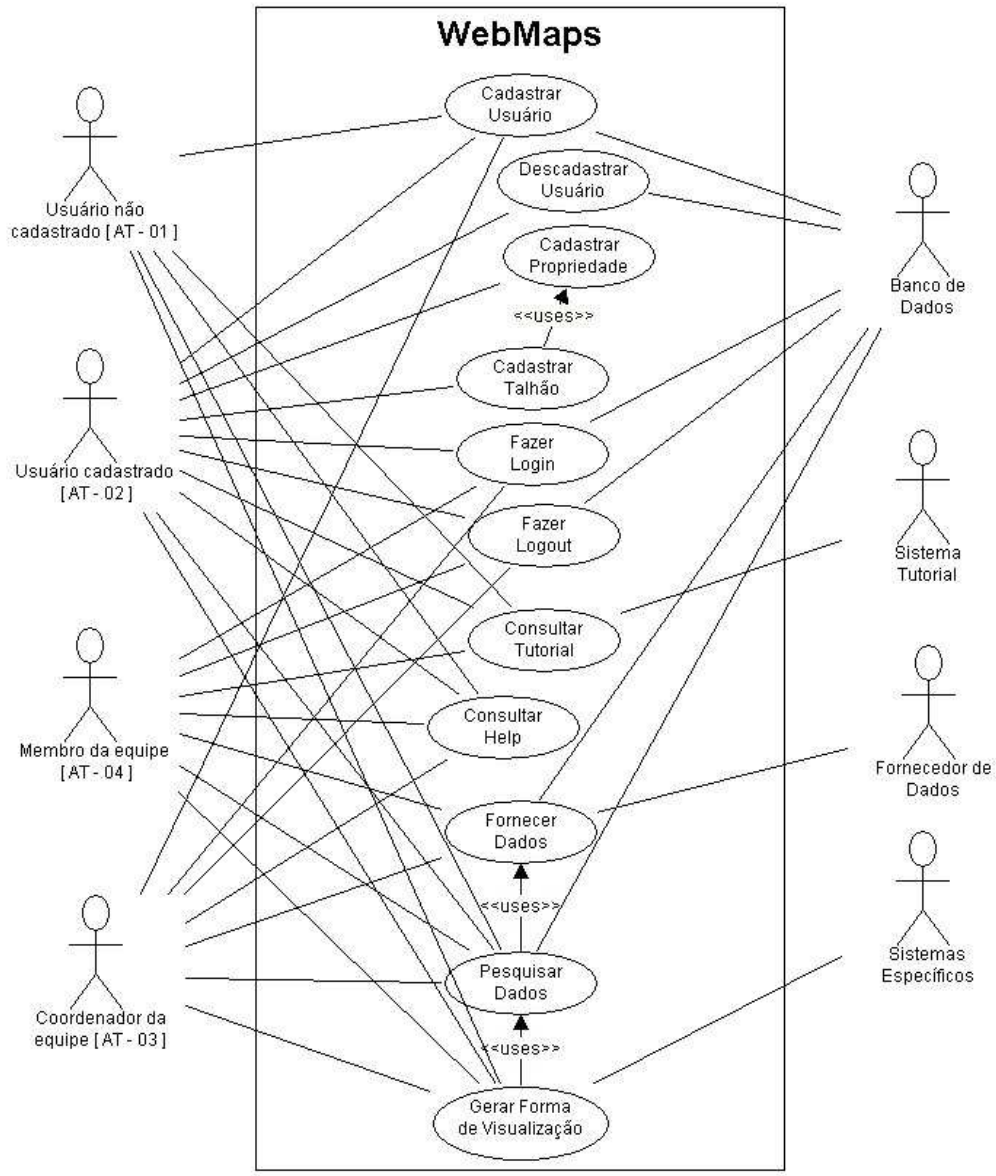


Figura 23: Diagrama de casos de uso [BBS+05].

A Tabela 9 mostra a especificação do caso de uso “cadastrar propriedade” representada segundo o padrão proposto pelo NDT.

Tabela 9: RF – 01.

RF – 01	Cadastrar propriedade	
Objetivos Asociados	OBJ – 01 OBJ – 02	
Descrição	O sistema deverá comportar-se tal e como se descreve no seguinte caso de uso e que representa a possibilidade de que o sistema ofereça para o usuário cadastrar sua propriedade ou região de interesse.	
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 escolher um tipo de acceso privado para sua propriedade, ele ainda pode dar acesso a ela a algum usuário cadastrado específico.

Neste ponto, já se conhece a informação que o sistema armazena, quem vai usar e as possibilidades funcionais que este oferece. Porém, em sistemas navegacionais isto não é suficiente. O sistema deve fornecer a informação e funcionalidade no momento apropriado e da forma apropriada. A ordem em que é mostrada a informação é também um aspecto Todos estes elementos formam os requisitos de interação. Os requisitos de interação estão representados por dois padrões: os critérios de recuperação ou frases e os protótipos de visualização.

A Tabela 10 mostra um exemplo do primeiro padrão, um critério de recuperação de informação. Apresentam-se as frases de recuperação associadas aos atores.

Tabela 10: FR – 03.

FR – 03	Recuperação de uma curva NDVI	
Corpo	**Descrição**	**Atores**
	O conceito RA – 07.Usuário deve ser exatamente _____	AT – 02: Usuário cadastrado
	O conceito RA – 07.Propriedade deve ser exatamente _____	AT – 02: Usuário cadastrado
	O conceito RA – 07.Talhão deve ser exatamente _____	AT – 02: Usuário cadastrado
	O conceito RA – 07.Data inicial deve conter a seguinte cadeia _____	AT – 02: Usuário cadastrado
	O conceito RA – 07.Data final deve conter a seguinte cadeia _____	AT – 02: Usuário cadastrado

Já especificadas as frases, se define os protótipos de visualização. Os protótipos de visualização permitem especificar as possibilidades de navegação que existem no sistema. A Tabela 11 mostra o protótipo de visualização na recuperação ou consulta de uma curva NDVI explicada na seção 6.1. Os requisitos funcionais RF-01, RF-02 e RF-03 estão definidos no apêndice A. Nos campos Protótipos de entrada e saída especifica-se a navegação.

Tabela 11: PV – 03.

PV – 03	Recuperação de uma curva NDVI
Atores	AT – 02: Usuário cadastrado.
Descrição	O sistema deverá permitir a visualização dos dados que se mostram a continuação e a navegação especificada e que representa a informação que se mostra a consulta realizada pelo usuário.
Frases	FR – 03
Funcionalidade Asociada	RF – 03, RF – 02, RF – 01
Informação visualizada	RA – 03.Arquivo
Protótipos de saída	-
Protótipos de entrada	PV – 02

Uma vez identificados os requisitos precisa-se validá-los. A realização da matriz de rastreabilidade é uma técnica que permite validar os resultados obtidos na especificação de

requisitos. A Tabela 12 mostra a matriz de rastreabilidade do nosso caso de estudo. Todos os padrões mostrados na tabela encontram-se especificados no apêndice A.

A geração desta tabela sirve para identificar se todos os objetivos estão sendo cobertos; ou seja, se uma columna não tem uma indicação significa que ainda esse objetivo não está sendo abordado. Esta matriz pode ser gerada automaticamente usando a ferramenta case da metodologia (NDT-Tool).

Tabela 12: Matriz de rastreabilidade.

	OBJ – 1	OBJ – 2	OBJ – 3
RA – 01	*	*	*
RA – 02	*	*	
RA – 03	*	*	
RA – 04	*	*	
RA – 05	*	*	
RA – 06	*	*	
RA – 07	*	*	*
AT – 01			*
AT – 02			*
AT – 03			*
AT – 04			*
RF – 01	*	*	
RF – 02	*	*	
RF – 03	*	*	
FR – 01	*	*	
FR – 02	*	*	
FR – 03	*	*	
PV – 01	*	*	
PV – 02	*	*	
PV – 03	*	*	

Legenda: * : o requisito associado corresponde ao objetivo.

Aqui termina a fase de Engenharia de Requisitos. O produto final desta fase é o Documento de Requisitos do Sistema (DRS). Este documento, como foi mencionado anteriormente, está especificado no Apêndice A.

6.3 Análise

O processo de análise tem como objetivo obter os modelos de análise: o modelo conceitual, que representa a estrutura estática do sistema, e o modelo de navegação, que representa a estrutura de navegação do sistema.

A realização destes modelos no NDT se faz de uma forma sistemática a partir dos requisitos. Tomando como ponto de partida os resultados da fase anterior, o NDT propõe uma serie de processos que permitem gerar estes modelos. O detalhe desses processos encontra-se no Anexo A.

O modelo conceitual se compõe de um diagrama único. Ele vem representado por dois elementos: o diagrama de classes conceituais, que se representa usando a notação de UML [UML05], e o dicionário de dados, que descreve de maneira textual o significado das classes e das associações do diagrama anterior. A Figura 24 apresenta o diagrama conceitual do sistema.

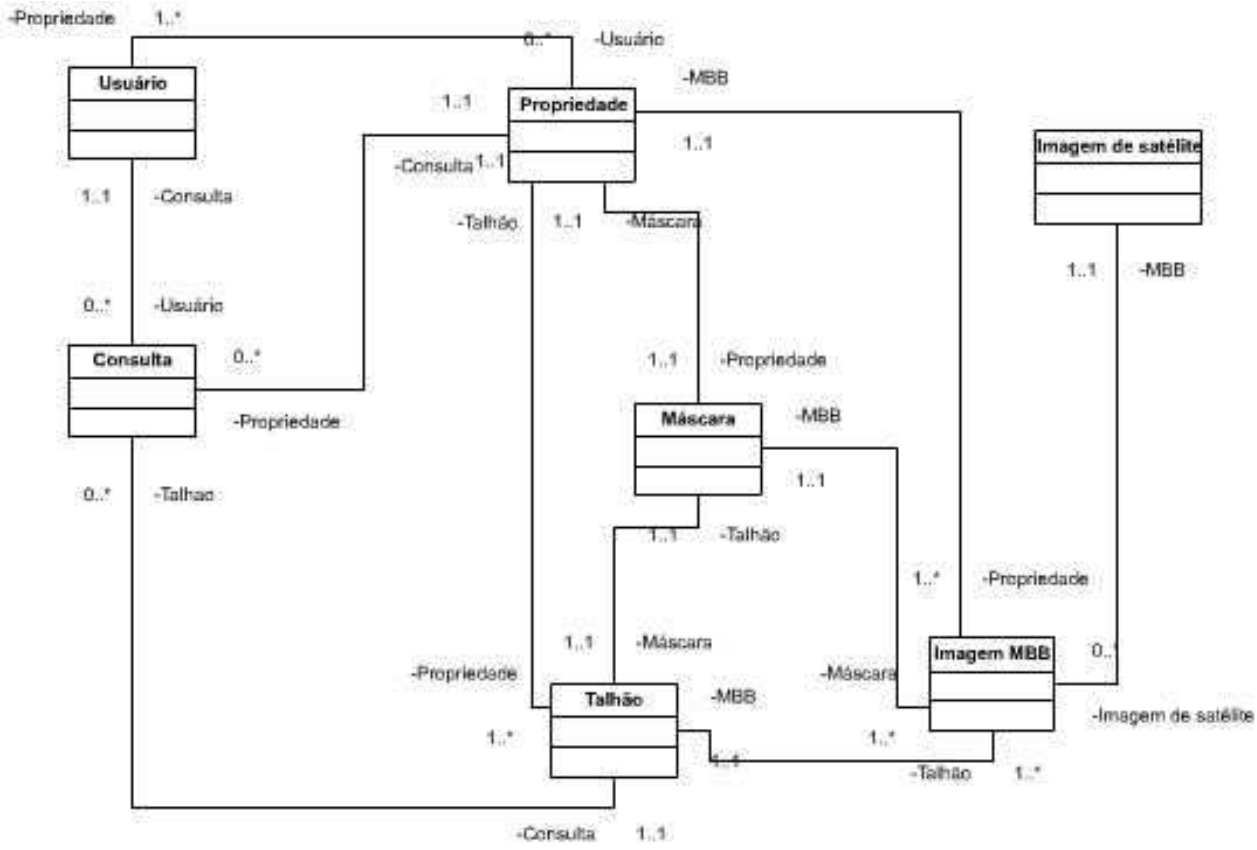


Figura 24: Modelo conceitual.

A Tabela 13 mostra o padrão para a descrição textual da classe usuário. Existe também um padrão para a descrição das associações. A totalidade dos padrões está no Apêndice B, Documento de Análise do Sistema (DAS).

Tabela 13: CL - 01.

CL – 01	Usuário		
Requisito	RA – 01.Usuário		
Descrição	O sistema deverá armazenar as informações correspondentes ao usuário		
Atributos	**Descrição**	**Significado**	**Dato Específico**
	Código usuário: CLn-01	Armazena informação sobre o código que identifica de maneira únivoca a cada usuário	RA – 01.Código usuário
	Nome: Texto	Armazena o nome do usuário.	RA – 01.Nome
	Login: Texto	Armazena o login do usuário.	RA – 01.Login
	Password: Texto	Armazena o password do usuário.	RA – 01.Password
	E – mail: Texto	Armazena o e-mail do usuário.	RA – 01.E-mail
	Endereço: Texto	Armazena o endereço do usuário.	RA – 01.Endereço
	Bairro: Texto	Armazena o bairro do endereço do usuário.	RA – 01.Bairro
	Complemento: Texto	Armazena o complemento do endereço do usuário.	RA – 01.Complemento
	Tipo de usuário: Texto	Estabelece a informação sobre o tipo de usuário.	RA – 01.Tipo de usuário
	Cidade: Texto	Armazena a cidade do usuário.	RA – 01. Cidade
	Estado: Texto	Armazena o estado do usuário.	RA – 01. Estado
Estado	Básico		

Terminada a especificação e criação da estrutura estática do sistema, procede-se a especificação do modelo navegacional. O modelo navegacional representa a forma de navegação da informação conceitual. Este modelo é uma visão do modelo conceitual, e representa os elementos que vão aparecer na navegação e as relações entre eles. O modelo navegacional pode ser muito diferente dependendo do papel do usuário que ingresse no sistema. Como se mencionou anteriormente, o NDT propõe uma série de processos que permitem gerar este modelo. O detalhe desse processo encontra-se no Anexo A. A Figura 25 mostra o modelo navegacional para o ator “usuário cadastrado”.

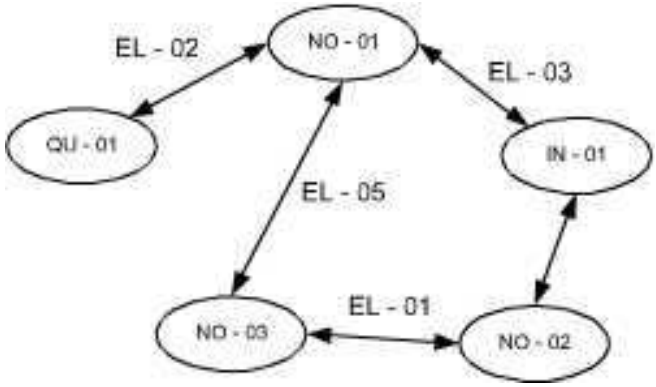


Figura 25: Modelo navegacional básico.

Como nas fases anteriores, os elementos que aparecem no modelo navegacional são descritos em NDT mediante padrões. A Tabela 14 mostra o padrão para a especificação do nó de recuperação de propriedades. Veja que nesta tabela aparece a relação do protótipo de visualização com o nó. Os elos (*links*) também são especificados em padrões do NDT (veja-se apêndice B).

Tabela 14: NO - 01.

NO – 01	Recuperação de propriedades	
Protótipo	PV – 01	
Atores em estudo	AE – 02	
Descrição	O sistema deverá permitir a visualização dos dados que se mostram a continuação e a navegação especificada e que representa a informação que se mostra ao usuário sobre suas propriedades cadastradas.	
Atributos	**Descrição**	**Protótipo**
	RA – 04.Nome	PV – 01
Estado	Básico	

Depois de realizar o modelo navegacional básico de uma maneira sistemática de acordo ao processo do NDT, precisa-se fazer uma revisão do modelo. Percebeu-se que o modelo navegacional gerado envolve uma navegação de informações dinâmicas ou aquelas que requerem algum tipo de processamento no servidor, mas se sentiu a falta de um modelo de navegação da informação ou conteúdo estático. Então, para complementar essa falta, a Semiótica Organizacional fornece um artefato que cobre essa necessidade. A Figura 26 mostra a relação do modelo de ontologia com o conteúdo do sistema. Estes modelos foram gerados pelo pessoal de interface homem – computador do projeto WebMaps [BBS+05].

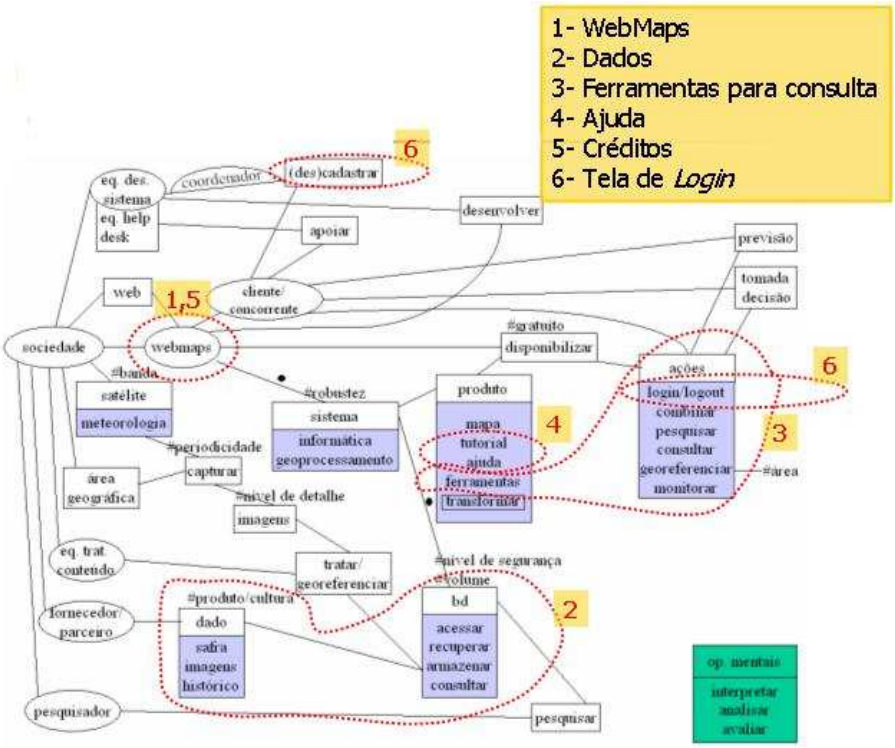


Figura 26: Modelo de ontologia [BBS+05].

O modelo de ontologia delineia um contexto que envolve conceitos e terminologias usadas no domínio do problema. A Figura 27 mostra a organização de conteúdo do WebMaps. Maiores detalhes sobre a geração destes artefatos encontra-se em [BBS+05].

Grupo	Tópico
1- WebMaps	- O que é WebMaps
	- Destaques
	- Questões Legais
	- Privacidade
	- Propriedade intelectual
	- Uso da imagem
	- Contratos
	- Idiomas
	- Mapa do *Site*
	- *Links*
	- Publicações
	- Equipe
2- Dados	- Banco de dados
	- Metadados
	- Bandas / imagens geoprocessadas
	- Bandas / imagens processadas
	- Dados históricos
	- NDVI / NOAA / Satélites
	- Dados temporais
	- Evolução temporal
	- Alimentação de dados
3- Ferramentas para consulta	- Comparações
	- Consulta
	- Informações por produto
	- Informações por área
	- Interpretação de dados amostrados

Grupo	Tópico
4- Ajuda	- *Help*
	- FAQ
	- Tutoriais de uso do sistema
	- Tutoriais por usuário
	- Escolas 1°. e 2°. graus
	- Exemplos de uso
	- Vocabulário / glossário
	- *Yellow Pages*
	- *Wizards*
5- Créditos	- Créditos / fontes
	- Clientes
	- Parceiros
	- Fornecedores
	- Cooperativas
	- Exportadores
	- Órgãos governamentais
6- Tela de *Login*	- Acesso restrito para pesquisadores do projeto
	- Acesso restrito para usuários cadastrados
	- Cadastro (indicando região / fazendas por coordenadas)
	- Usuário
	- Diferentes níveis de usuário
7- teclas de atalho como funcionalidades	

Figura 27: Organização de Conteúdo [BBS+05].

Graças ao produto gerado pela Semiótica Organizacional, são especificados os menus principais do sistema. Com essas informações pode-se construir o modelo de navegação final. A Figura 28 mostra esse modelo: note que o modelo de navegação básico mostrado na Figura 25 é parte do modelo de navegação final. O modelo de navegação básico associa-se aos menus principais do sistema. Os "nós" relacionados a páginas

estáticas especificadas na Figura 27, não foram colocadas no modelo final para fins de clareza; só foram levados em conta os menus principais.

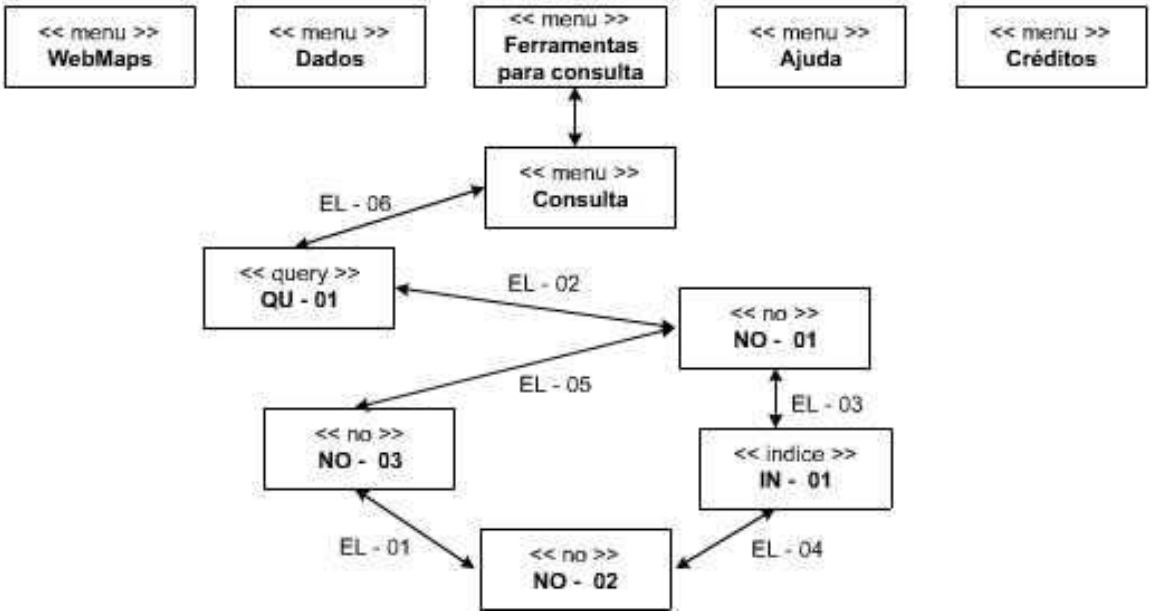


Figura 28: Modelo navegacional final.

6.3 Modelo de Interface Abstrata e Implementação

Nesta etapa o projeto WebMaps teve entre sua equipe de desenvolvimento uma Web Designer que elaborou as interfaces abstratas. Como foi comentado na seção 6.1, a linguagem de desenvolvimento é Java.

A Figura 29 mostra a implementação de uma interface abstrata. Cabe assinalar que o sistema ainda não está no ar. Atualmente o WebMaps tem implementado o modelo de navegação para um usuário cadastrado e não cadastrado. Esta implementação cobre as necessidades de consultas geográficas, que trabalham com mapas de satélite. Por se tratar de um SIG, precisa-se de processamento de grandes volumes de dados. No capítulo 7 apresenta-se a aplicação do teste de desempenho no WebMaps. Estes testes têm por objetivo determinar se o desempenho do sistema integrado é adequado de acordo com os requisitos do sistema (Fase de Engenharia de Requisitos).

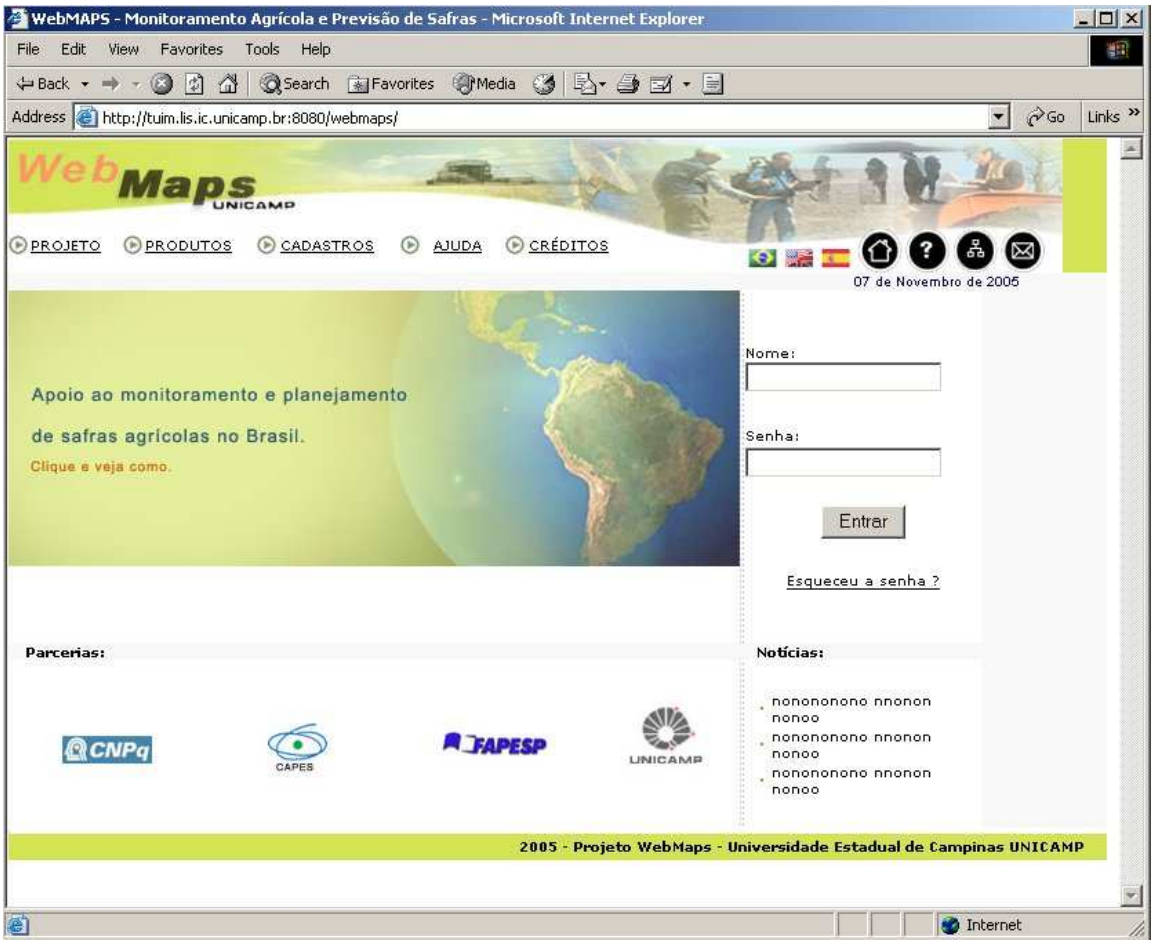


Figura 29: Interface Abstrata.

6.4 Considerações Finais

Este Capítulo apresentou a aplicação do modelo de processo proposto no Capítulo 4, processo para desenvolvimento de aplicações SIG Web.

Foi usado como estudo de caso o WebMaps, O WebMaps é um projeto interdisciplinar em andamento no Instituto de Computação da Unicamp suportado com o financiamento de CNPq, que tem como objetivo permitir o monitoramento e planejamento de safras agrícolas no Brasil. Este capítulo também mostrou que já existe uma aplicação protótipo funcionando e que avalia o processo proposto.

Capítulo 7

Estudo de caso Teste de desempenho em SIG Web

O processo de teste de desempenho proposto foi aplicado no projeto WebMaps. Este capítulo descreve este SIG web, bem como a aplicação da proposta baseado num caso de uso crítico do sistema e, além disso, se apresentam os resultados preliminares com o apoio de uma ferramenta livre, Apache JMeter [JME05].

7.1 WebMaps

O WebMaps é um projeto interdisciplinar em andamento no Instituto de Computação da Unicamp suportado com o financiamento de CNPq, 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 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.

Mais detalhes pode ser visto na Seção 6.1 do Capítulo 6.

7.2 Aplicação do processo

Nesta seção se apresenta a aplicação do processo proposto de teste de desempenho para aplicações SIG Web, especificado na seção 5.3.3, no sistema WebMaps. Para facilitar a leitura, a Figura 30 mostra novamente o processo proposto. Nas seções seguintes são detalhadas cada uma dessas atividades.

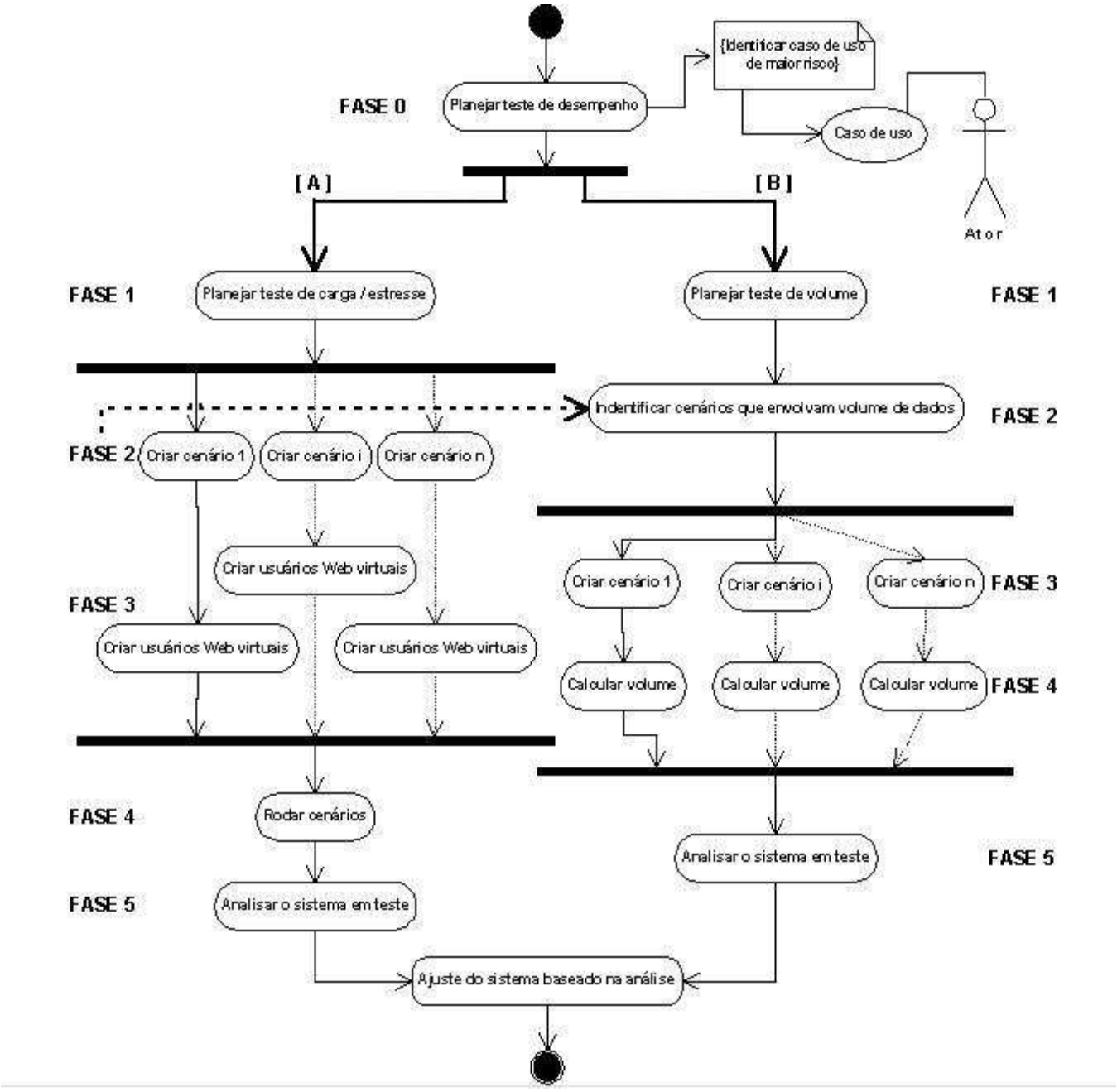


Figura 30: Processo de teste de desempenho em aplicações SIG Web.

7.2.1 Planejar teste de desempenho (FASE 0)

A primeira tarefa é definir o objetivo do teste de desempenho e principalmente o que vai ser testado. Sendo assim, deve-se 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 como foi explicado na seção 6.1, este é responsável pelos recortes das imagens de satélite sendo uma das tarefas mais críticas no sistema devido ao processamento que esta tarefa envolve. A Tabela 15 mostra a especificação deste caso de uso com o padrão de requisitos funcionais do NDT (*Navigational Development Techniques*) [ESC04]. Todos os requisitos funcionais do sistema estão especificados no Documento de Requisitos do Sistema (DRS); pode-se encontrá-lo no apêndice A deste trabalho.

Tabela 15: Caso de uso - Cadastrar propriedade.

RF – 01	Cadastrar propriedade	
Objetivos Asociados	OBJ – 01 OBJ – 02	
Descrição	O sistema deverá comportar-se como se descreve no caso de uso abaixo, que permite que o usuário cadastre sua propriedade ou região de interesse.	
Atores	**Ator do caso de uso**	**Ator do sistema**
	Usuário cadastrado	AT – 02
Sequencia 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
Poscondição	São feitos os recortes das imagens de satélite, sendo 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 aceso privado para sua propriedade, ele ainda pode dar acesso a ela a outro usuário cadastrado específico.

7.2.2 Planejar teste de carga (FASE 1A)

A Figura 31 apresenta o processo de teste de carga / estresse, especificado na seção 5.3.1.; ele é um sub-processo do teste de desempenho mostrado na Figura 33. Esta figura é mostrada novamente aqui com objetivo didático, para facilitar o entendimento do que será explicado a seguir. Como produto da Fase 1, se considerou testar o sistema simulando os usuários entrando na página principal da aplicação e entrando na página de cadastro de propriedade, dado que este foi o caso de uso escolhido. Só podem fazer cadastro de propriedade aqueles usuários que fizeram cadastro no sistema.

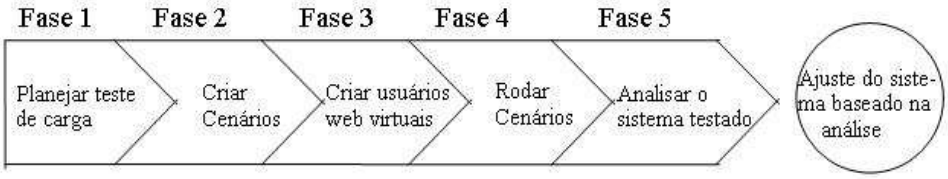


Figura 31: Processo proposto de teste de carga / estresse.

A Fase 2 precisa de dados reais para a especificação correta dos cenários de teste. Estes testes foram feitos de acordo com cenários construídos baseando-se em estatísticas de uso reais no Sistema de Monitoramento Agrometereológico, Agritempo [AGR05]. Este sistema teve até 200 usuários reais conectados simultâneamente. Então, o primeiro cenário especificado é o seguinte:

Cenário 1 de carga: Uma carga de 200 usuários virtuais a cada 10 segundos até chegar a 1000 usuários na página de cadastro de propriedades.

De acordo com o planejamento do teste, especificado na Fase 1, se considerou também o cenário seguinte:

Cenário 2 de carga: Uma carga de 300 usuários virtuais a cada 10 segundos até 3000 usuários na página principal da aplicação.

As Fases 3, 4 foram apoiadas pela ferramenta JMeter [JME05]. A Fase 5 corresponde a uma análise do sistema testado. Estas fases serão especificadas na seção 7.3.

7.2.3 Planejar teste de volume (FASE 1B)

A Figura 32 apresenta o processo de teste de volume, especificado na seção 5.3.2.; ele é também um sub-processo do teste de desempenho mostrado na Figura 33.

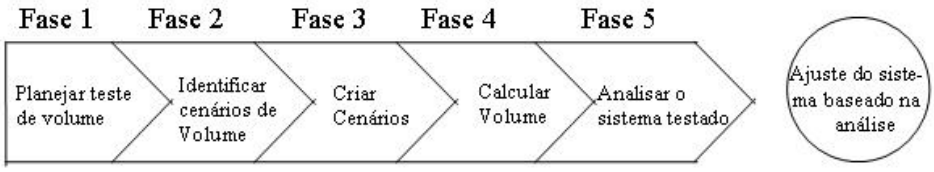


Figura 32: Processo proposto de teste de volume.

Como produto da Fase 1, planejamento do teste de volume, o objetivo do teste é avaliar o comportamento do sistema referente ao volume associado ao caso de uso de cadastro de propriedade. Na Fase 2, se identifica o cenário 1 do teste de carga da seção 7.2.2. A Tabela 16 mostra a identificação desse cenário de teste de volume já que envolve armazenamento de informação tanto no banco de dados quanto em arquivos.

Tabela 16: Relação na fase 2 entre o processo de carga e volume.

TESTE DE CARGA [A]			**TESTE DE VOLUME [B]**	
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 é considerado como cenário de volume já que envolve o armazenamento de informação tanto no banco de dados quando no arquivo.

Assim, na Fase 3 é especificado da maneira seguinte:

Cenário 1 de volume: Testar o volume ou quantidade de armazenamento de informação gerenciado quando 200 usuários cadastram simultaneamente uma propriedade.

Na fase 4, se faz o cálculo de volume. No momento, o WebMaps trabalha com 92 imagens do tipo MODIS de todo o território brasileiro, referentes a 4 anos (2001 até 2004) fornecidas pela NASA [MOD05]. Cada imagem ocupa 109,6 MB e uma nova imagem foi obtida a cada 15 dias. O processamento das imagens de satélite é lento. Sendo assim, optou-se pelo uso de recortes (máscaras) das imagens, relativas a cada propriedade cadastrada, para diminuir o tempo de processamento. No cadastro de uma propriedade se fazem os recortes de todas as imagens de satélite de acordo as coordenadas geográficas da

propriedade. A quantidade de armazenamento para o cadastro de uma propriedade é aproximadamente 1 MB (92 recortes) e para 200 usuários seria uma quantidade aproximada de 200 MB.

Com estes resultados, na Fase 5, se faz a análise do sistema, aqui, por exemplo, se pode fazer uma projeção do comportamento do sistema para uma dada quantidade de anos.

7.3 Resultados obtidos

A Tabela 17 apresenta a aplicação da proposta de testes de desempenho para o sistema WebMaps, considerando o caso de uso escolhido.

O exemplo mostrado na tabela é o resultado da aplicação de cada uma das fases mostradas do processo proposto de testes de desempenho para aplicações SIG Web na seção 5.3 do Capítulo 5. Veja 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, graças a uma boa especificação do cenário 1 do processo de carga, a especificação do cenário no processo de teste de volume ficou mais fácil (veja-se seção 7.2.3).

Tabela 17: Instância do processo proposto.

<table>
<tr><th colspan="6">TESTE DE DESEMPENHO</th></tr>
<tr><td>Fase 0</td><td colspan="5">Testar o sistema com o objetivo de avaliar o desempenho do caso de uso: Cadastro de propriedade.</td></tr>
<tr><th colspan="3">TESTE DE CARGA [A]</th><th colspan="3">TESTE DE VOLUME [B]</th></tr>
<tr><td>Fase 1</td><td colspan="2">Testar o sistema simulando os usuários entrando na página principal da aplicação e entrando na página de cadastro de propriedade.</td><td>Fase 1</td><td colspan="2">Fazer o teste de volume do cadastro de propriedades.</td></tr>
<tr><td>Fase 2</td><td>Cenário 1: Uma carga de 200 usuários virtuais a cada 10 segundos até 1000 usuários na página de cadastro de propriedades.</td><td>Cenário 2: Uma carga de 300 usuários virtuais a cada 10 segundos até 3000 usuários na página principal da aplicação.</td><td>Fase 2</td><td colspan="2">O cenário 1 é considerado como cenário de volume já que envolve o armazenamento de informação tanto no banco de dados quando no arquivo.</td></tr>
<tr><td>Fase 3</td><td>Criação de usuários virtuais apoiada pelo JMeter</td><td>Criação de usuários virtuais apoiada pelo JMeter</td><td>Fase 3</td><td>Cenário1: relacionado ao Cenário 1 do processo [A] Quantidade de armazenamento de informação para 200 usuários cadastrando uma propriedade.</td><td></td></tr>
<tr><td>Fase 4</td><td colspan="2">Para rodar os cenários se fez uso do JMeter.</td><td>Fase 4</td><td>Cálculo de volume para um usuário:
Criação de 92 recortes ~ 1 Mb.
Para 200 usuários ~200 Mb.</td><td></td></tr>
<tr><td>Fase 5</td><td colspan="2">Análise do sistema</td><td>Fase 5</td><td colspan="2">Análise do sistema. Aqui, por exemplo, se pode fazer uma projeção do comportamento do sistema para uma determinada quantidade de anos.</td></tr>
</table>

A Fase 4 do processo de carga poder ser feito de duas formas: manual ou automática. A forma manual requer um coordenador e vários usuários reais. A tarefa é

complicada já que precisa de uma boa coordenação. No entanto, a forma automática é menos complicada, já que pode-se criar usuários virtuais. Esta automatização pode ser apoiada por ferramentas livres de teste; algumas delas estão especificadas em [CDW01].

A Figura 33 mostra uma tela da ferramenta Apache JMeter [JME05] usada nas Fases 3 e 4 do cenário 1 do processo de carga A no exemplo da seção anterior. Esta ferramenta é usada tanto no ambiente acadêmico [GPV+05] como no comercial como apoio para os testes de desempenho e representa de maneira gráfica o tempo de resposta do sistema sobre os cenários montados na Tabela 17.

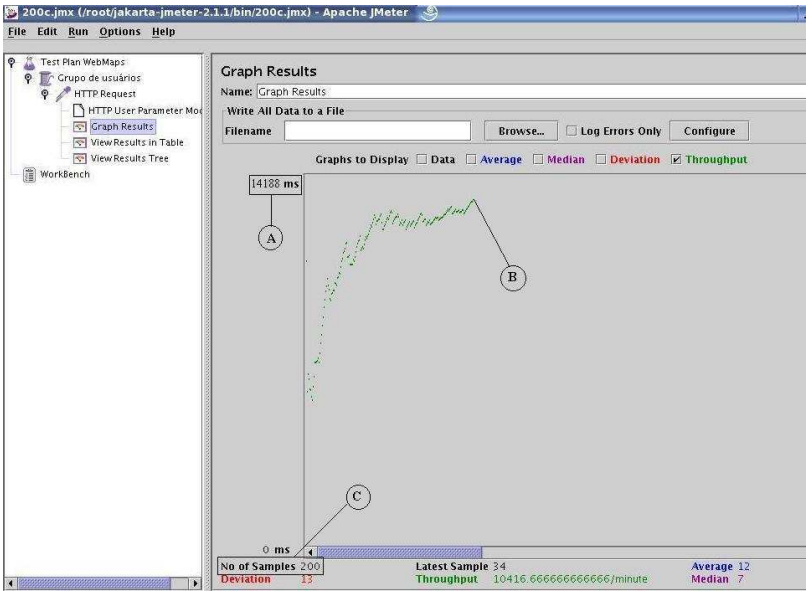


Figura 33: Interface de resultados no Apache JMeter.

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. A curva mostra a variação do tempo de resposta a cada usuário virtual que entra no sistema. 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 é crítico considerando-se que 95% das petições dos usuários são abortadas caso o tempo de resposta de suas transações excedem a 6 segundos no servidor Web [MV02, capítulo 1].

Também foram realizados testes de estresse. Verificou-se que o servidor WebMaps aborta quando (na média) 267 usuários estão cadastrando uma propriedade simultaneamente. O servidor do WebMaps considerado nos experimentos foi um computador Intel(R) Pentium(R) 4 CPU 2.40GHz, 1Gb de memoria RAM e com sistema operacional Linux, distribução SUSE Linux.

Finalmente o ajuste do sistema baseado na análise permite adaptar o sistema e tomar medidas necessárias junto à equipe de testes de desempenho e à equipe de desenvolvimento para melhorar o desempenho.

Estes resultados são muito úteis, pois o estudo dos tempos de resposta forneceu os indicativos suficentes para revisar e analisar a arquitetura. Uma possibilidade de ajuste do sistema seria a proposta de melhoramento de infra-estrutura em processamento como, por exemplo, o uso de clusters (computadores com processamento distribuído) ou a não realização de processamento on-line no cadastro de propriedades, entre outros.

7.2 Considerações Finais

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.

Estes testes foram feitos de acordo com cenários reais construídos baseando-se em estatísticas de uso reais do Sistema de Monitoramento Agrometereológico, Agritempo, atualmente em uso na Embrapa. Os resultados preliminares obtidos indicam que os testes foram úteis na identificação de problemas da arquitetura preliminar do WebMaps antes da sua implantação.

O conjunto de testes realizado permitiu a identificação de problemas de desempenho no sistema que resultaram na revisão da sua arquitetura. Uma possibilidade é que as imagens sejam tratadas não mais como arquivos, mas como consultas a BD. A revisão da arquitetura está ainda em estudo.

Capítulo 8

Conclusões e Trabalhos futuros

8.1 Contribuções do trabalho

Este trabalho propôs um processo de desenvolvimento para aplicações SIG Web. O processo de desenvolvimento proposto integrou artefatos do NDT (Navigational Development Techniques) [ESC04] e da Semiótica Organizacional [LIU00].

Também se propôs um processo de teste de desempenho para aplicações SIG Web. 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 único, onde o ponto de partida é a identificação dos casos de uso críticos que precisam ser testados. Para isso se supõe que os requisitos do sistema foram totalmente especificados.

Os processos propostos, de desenvolvimento e de teste de desempenho, foram aplicados no sistema WebMaps, que é uma aplicação SIG Web cuja finalidade é auxiliar seus usuários no planejamento agrícola a partir de regiões de interesse.

A aplicação da proposta do processo de desenvolvimento permitiu obter um protótipo do WebMaps. Seguindo as diretrizes do processo foram criados o Documento de Requisitos do Sistema (DRS) e o Documento de Análise do Sistema (DAS).

A realização dos testes de desempenho aplicando o processo proposto foi muito importante já que permitiu reconhecer problemas na arquitetura do sistema.

8.2 Trabalhos futuros

A modelagem conceitual dos dados geográficos está sendo estudada para sua incorporação no processo de desenvolvimento de aplicações SIG Web.

Um outro trabalho já em andamento trata da utilização de um método ágil para o desenvolvimento do WebMaps. Pretende-se avaliar os resultados obtidos com a utilização de ambos os processos.

Com referência aos testes, trabalhos em andamento incluem um processo de testes mais abrangente para aplicações SIG Web. Esta metodologia genérica incluirá, além dos testes de desempenho, testes de unidade, testes de integração, dentre outros.

Referências

[ABC+91] J. ANTENUCCI, K. BROWN, P. CROSWELL, M. KEVANY, and H. ARCHER. Geographical Information Systems – a Guide to the Technology. Van Nostrand Reinhold, 1991.

[AGR05] AGRITEMPO. http://www.agritempo.gov.br, última visita: setembro 2005.

[ARO89] S. ARONOFF. Geographic Information Systems. WDL Publications, Canada, 1989.

[AS93] E. ASSAD and E. SANO. Sistema de Informações Geográficas: Aplicações na Agricultura. Edição Embrapa / CPAC.

[BEM+01] BRISABOA, N.R., ESCALONA, M.J., MEJÍAS, M., PLACES, A.S.,RODRÍGUEZ, F.J., TORRES, J. Sistema de consulta vía Web para el Instituto Andaluz dePatrimonio Histórico. II Jornadas de bibliotecas digitales. pp. 199-116. Ciudad Real, España. Noviembre 2001.

[BOE81] B. BOEHM, Software Engineering Economics, Prentice-Hall.

[BRE02] Fabricio A. BREVE. Engenharia para a Web. Universidade de São Carlos.

[BSS+05] M. Cecilia C. BARANAUSKAS, Juliano SCHIMIGUEL, Carlos A. SIMONI, Cláudia B. MEDEIROS. Guiding the Process of Requirements Elicitation with a Semiotic-based Approach – A Case Study, HCI – International 2005, Las Vegas, Nevada, USA.

[BUL94] G. BULL. Ecosystem Modelling with GIS. Environmental Management, 18(3):345-349, 1994.

[CCH+96] G. CÂMARA, Marco A. CASANOVA, Andrea S. HEMERLY, Geovane Cayres MAGALHÃES, Claudia María Bauzer MEDEIROS. Anatomia de Sistemas de Informação Geográfica, 10 Escola de Computação, Campinas 1996.

[CCU05] CCUEC, Centro de Computação - Unicamp. Projetos de Aplicações Web. http://www.ccuec.unicamp.br/treinamento_int2004/webpro/pesquisa/pesquisa_oprojeto.html, última visita: novembro 2005.

[CDW01] 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

[CPT05] CPTEC, http://satelite.cptec.inpe.br/htmldocs/ndvi/ndvi_fram.htm, última visita: novembro 2005.

[DM02] 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

[EPL+05]M.J. Escalona, A. Martín-Pradas, L.F de Juan, D. Villadiego, J.J. Gutierrez. El Sistema de Información de Autoridades del Patrimonio Histórico Andaluz Proceedings of V Jornadas de Bibliotecas Digitales (JBiDi 2005). ISBN: 84-9732-453-6 Granada, Spain. September, 2005

[ESC04] M.J. ESCALONA. 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

[GB90] O. GUNTHER e A. BUCHMAN. Research Issues in Spatial Databases. ACM Sigmod Record, 19(4):61-68, 1990.

[GEM+04] J.J. GUTIERREZ, M.J. ESCALONA, M. MEJIAS, J. TORRES. Aplicando técnicas de testing en sistemas para la difusión Patrimonial. Actas del V Congreso Nacional de Turismo y Tecnologías de la Información y las comunicaciones (TURITEC'2004). ISBN: 84-608-0181-0. pp. 237-252. Málaga, Spain. October, 2004.

[GIN01] GINIGE, Athula. "Engineering A Better Website". University Of Western Sydney. Austrália, Outubro de 2000. <http://aeims.uws.edu.au/talks/eng-web-sites.pdf>

[GM01] GINIGE, Athula e MURUSEGAN, San. "Web Engineering: An Introduction". IEEE Multimedia. Janeiro-Março 2001.

[GPV+05] GUTIERREZ 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. pp.208-219 Madrid, Spain.

[GSP93] GARZOTTO F., SCHWABE D. and PAOLINI P. HDM-A Model Based Approach to Hypermedia Application Design. ACM Transactions on Information System, 11 (1), pp 1-26. Enero, 1993.

[JBR99] JACOBSON, I., BOOCH, G., RUMBAUGH, J. The Unified Software Development Process. Ed. Addison-Wesley, 1999

[JME05] Apache JMETER, http://jakarta.apache.org/jmeter/, última visita: novembro 2005.

[KOC01] KOCH, N. Software Engineering for Adaptative Hypermedia Applications. Ph. Thesis, FAST Reihe Softwaretechnik Vol(12), Uni-Druck Publishing Company, Munich. Germany. 2001.

[LIU00] K. LIU. "Semiotics in Information Systems Engineering". Cambridge University Press, Cambridge.

[MAR99] B. MARICK. New Models for Test Development. Reliable Software Technologies.Testing Foundations.

[MAR05] MARTINS, Eliane . "Apostila sobre Testes de Software".

[MGR93] D. MAGUIRE, M. GOODCHILD, e D. RHIND. Geographical Information Systems – volume I. John Wiley and Sons, 2 edição, 1993.

[MIL00], MILLER E.: WebSite Loading and Capacity Analysis. Software Research, Inc. San Francisco, USA.

[MOD05] MODIS, Moderate Resolution Imaging Spectroradiometer, http://modis.gsfc.nasa.gov/, última visita: setembro 2005.

[MOL03] MOLINARI, Leonardo. "Testes de Software: Produzindo Sistemas Melhores e Confiáveis", Editora Érica Ltda, São Paulo.

[MUR00] MURUGESAN, San. "Web Engineering For Sucessful Web Application Development". University Of Western Sydney. Austrália, Outubro de 2000. <http://aeims.uws.edu.au/Talks/Web-ApWeb2000.PDF>

[MV02] Daniel A. MENASCÉ e Virgilio A. F. ALMEIDA. Capacity Planning for Web Services: Metrics, Models, and Methods. Prentice Hall.

[MYE79] MYERS, Glendord J. (1979), The Art of Software Testing, Jhon Wiley & Sons, Inc, Canada.

[NCG89] NCGIA. The Research Plan of the National Center for Geographical Information and Analysis. International Journal of Geographical Information Systems, 3(2):117-136, 1989.

[NIE00] NIELSEN, Jakob. "Designing Web Usability". New Riders Publishing.

[PAU01] PAULA F. W. P. Engenharia de Software. Editora LTC.

[PAR94] E. PAREDES. Sistema de Informação Geográfica – Princípios e Aplicações(Geopro-cessamento). Editora Érica.

[PM90] D. PEUQUET e D. MARBLE. Introdutory Readings in Geographical Informations Systems. Taylor e Francis.

[PRE01] PRESSMAN, Roger S. "Software Engineering: A Practioner's Approach". McGraw-Hill. 5ª edição.

[RIO04] RIOS E. "Gerência de testes em projetos de sistemas", http://www.alats.org.br/eventos/040726/gerencia_de_testes.pdf, última visita: setembro 2004.

[R0S96] G. ROSSI. An Object Oriented Method for Designing Hipermedia Applications. PHD Thesis, Departamento de Informática, PUC-Rio, Brazil, 1996.

[RT01] Filippo RICCA and Paolo TONELLA; Analysis and Testing of Web Applications; Proceedings of the 23rd International Conference on Software Engineering (ICSE'01)

[SAU05] Jacques Philippe SAUVÉ. Introdução e motivação : Arquiteturas em n camadas. http://jacques.dsc.ufcg.edu.br/cursos/j2ee/html/intro/intro.htm, última visita: novembro.

[SIE96] S. SIEGEL. Object-Oriented Software Testing: a Hierarchical Approach. John Wiley & Sons.

[SS00] 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

[STA97] STAMPER, R.K. Organizational Semiotics. In Mingers J. & Stowell F. (eds.) Information Systems: An Emerging Discipline. Mc Graw Hill, London, 267-283.

[TER05] J. TORRES, M.J. ESCALONA, A.M.REINA. NIDO: Navegación e interacción con el usuario en el desarrollo de sistemas de información Web: Métodos, Técnicas y Herramientas. Proceeding of V Jornadas de Seguimiento de Proyectos en Tecnologías Informáticas (CEDI 2005). pp.296-303. Granada, Spain. September, 2005.

[THI93] J. THIEMAN. The International Directory Network and Connected Data Information System for Research in the Earth and Space Sciences. In Proc. of the ACM SIGMOD conference, pages 475-478.

[TMC92] A. TEXEIRA, E. MORETTI, e A. CRISTOFOLETTI. Introdução aos Sistemas de Informação Geográfica. Edição de Autor.

[TTC01] ETSI ES 201 873-1: The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language. V2.1.0 (2001-10); também ITU-T standard Z.140.

[VGE+05] D. VILLADIEGO, J.J. GUTIERREZ, M.J. ESCALONA, J. TORRES, M. MEJÍAS. Una aplicación real de NDT en un sistema para el reconocimiento, declaración y calificación del grado de minusvalía.Proceedings of Mundo Internet 2005. Vol. II. pp.116-123; Madrid, Spain. April, 2005.

[UML05] UML , Unified Modeling Language. http://www.uml.org/, última visita: setembro.

Anexos

Anexo A – Guia de referência do NDT

1. Introdução

Este anexo explica as atividades e tarefas das fases que cobrem o NDT (Navigational Development Techniques) [ESC04]: Engenharia de Requisitos e Análise; se especificam também os resultados que se deve obter com o NDT.

2. Engenharia de Requisitos em NDT

NDT é uma proposta que dá muita importância a esta etapa. Ela defende a realização de um grande esforço na tarefa de captura, definição e tratamento de requisitos, com a idéia de aplicar uma base adequada e correta, com o menor número de erros possíveis. Na Figura A1, mostra-se mediante um diagrama de atividades, o processo de engenharia de requisitos proposto por NDT. Os diferentes requisitos do sitema são definidos. NDT contempla 5 tipos de requisitos:

- ✓ Requisitos de armazenamento de informação, que por sua vez se dividem em dois: requisitos de armazenamento, ou seja, a informação que vai usar o sistema.
- ✓ Requisitos de atores. Neste grupo de requisitos se definem as funções dos usuários que interatuam com o sistema, assim como as relações que se estabelecem entre eles.
- ✓ Requisitos funcionais, que vão representar as necessidades dos usuários em termos das funcionalidades que o sistema oferece.
- ✓ Requisitos de interação, que definem como cada usuário interage com o sistema e como este pode navegar através do mesmo. Os requisitos de interação se definirão mediante as Frases e os Protótipos de Visualização.
- ✓ Requisitos não funcionais, que vão englobar outros requisitos não tratados anteriormente. Além do levantamento dos requisitos, NDT propõe realizar uma validação dos mesmos. Se ao validar os requisitos não se detectam erros ou incongruências, se gerará o documento de requisitos do sistema (DRS), que é o resultado da fase e que, além de ser o produto da mesma, é a base para a realização da Análise.

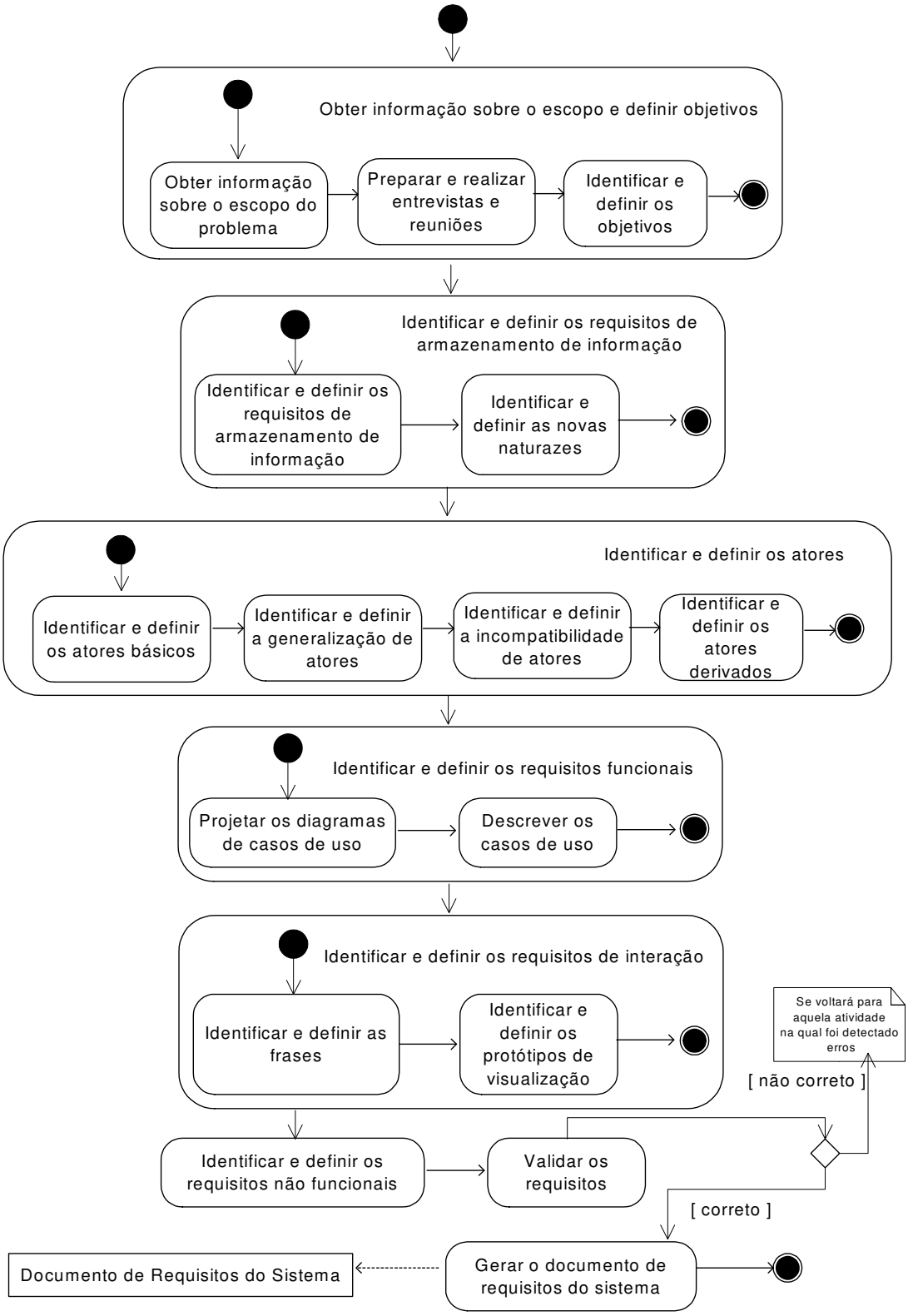


Figura A1. Processo da Engenharía de Requisitos em NDT.

A seguir se apresentam com detalhe as atividades e tarefas que as compõem:

3. Atividade 1 - Obter informação sobre o contexto do trabalho e definir objetivos.

Na primeira atividade da Engenharia de Requisitos, a equipe de desenvolvimento deve fazer uma aproximação ao escopo do sistema. Deve-se estabelecer o vocabulário a utilizar, os usuários e clientes que vão participar no projeto e os objetivos do mesmo.

É necessário destacar quais são os objetivos pretendidos do sistema. Como foi comentado, o fluxo de especificação de requisitos do NDT gera o documento de requisitos do sistema (DRS). Neste documento, entre outras informações, se inclui a definição dos objetivos detectados durante esta fase. Para definir estes objetivos se deve fazer uso de um padrão com uma estrutura similar mostrada na Tabela A1. Na definição do padrão existem frases ou palavras que devem aparecer sempre. Outras, pelo contrário, devem ser completadas pelo usuário. Estas últimas são inerentes aos caracteres "<" e ">".

Tabela A1: Padrão para a definição dos objetivos.

OBJ - <id>	**<nome descritivo>**	
Versão*	**<número da versão atual>**	**<data da versão atual>**
Autores*	**Nome do autor: <nome do autor>** **Cargo: <cargo do autor>** **Organização: <organização do autor>**	
	...	
	Nome do autor: <nome do autor> **Cargo: <cargo do autor>** **Organização: <organização do autor>**	
Fontes*	**Nome da fonte: <nome da fonte>** **Cargo: <cargo da fonte>** **Organização: <organização da fonte>**	
	...	
	Nome da fonte: <nome da fonte> **Cargo: <cargo da fonte>** **Organização: <organização da fonte>**	
Descrição	**O sistema deverá <descrição do objetivo a cumprir>**	
Subobjetivos*	- **OBJ-<X>: <nome do subobjetivo>** - **...**	
Importância*	**<importância do objetivo>**	
Urgência*	**<urgência do objetivo>**	
Estado*	**<estado do objetivo>**	
Estabilidade*	**<estabilidade do objetivo>**	
Comentarios*	**<comentarios adicionais do objetivo>**	

4. Atividade 2 - Identificar e definir os requisitos de armazenamento de informação.

Uma vez identificados os objetivos, NDT propõe identificar os requisitos que o sistema deve cumprir. NDT, como foi comentado, cobre cinco tipos diferentes de requisitos. Nesta atividade se identificam os primeiros deles: os requisitos de armazenamento de informação. Os requisitos de armazenamento se definem mediante dois elementos: requisitos de

armazenamento de informação e as naturezas.

4.1. Tarefa 2.1 - Identificar e definir os requisitos de armazenamento de informação

Nesta tarefa se determinam todas as necessidades de armazenamento que se detectam durante a realização das entrevistas. A idéia essencial dos requisitos de armazenameto de informação é dar resposta a perguntas como: que informações devem armazenar o sistema? Com que informação vai trabalhar o sistema?. O requisitos de armazenamento também aparecem no documento de requisitos do sistema e, novamente faremos uso do padrão (ver Tabela A2) que estrutura os dados que se recolhem dos mesmos.

Tabela A2: Padrão para definir os requisitos de armazenamento de informação.

RA-<id>	**<nome descritivo do requisito>**	
Versão*	**<número da versão>**	**<data da versão>**
Autores*	**Nome do autor: <nome do autor>** **Cargo: <cargo do autor>** **Organização: <organização do autor>**	
	...	
	Nome do autor: <nome do autor> **Cargo: <cargo do autor>** **Organização: <organização do autor>**	
Fontes*	**Nome da fonte: <nome da fonte>** **Cargo: <cargo da fonte>** **Organização: <organização da fonte>**	
	...	
	Nome da fonte: <nome da fonte> **Cargo: <cargo da fonte>** **Organização: <organização da fonte>**	
Objetivos associados	- **OBJ-<X>: <nome do objetivo>** - **...**	
Descrição	**O sistema deberá armazenar a informação referente a <conceito relevante>. Especificamente:**	
Dados específicos	Nome e descrição	Natureza
	<nome do dado>:<breve descrição do dado>	**<natureza do dado> [<Cardinalidade:cardinalidade]**
	...	
	<nome do dado>:<breve descrição do dado>	**<natureza do dado> [<Cardinalidade:cardinalidade]**
Intervalo temporal*	**<intervalo temporal do requisito>**	
Importância*	**<importância do requisito>**	
Urgência*	**<urgência do requisito>**	
Estado*	**<estado do requisito>**	
Estabilidade*	**<estabilidade do requisito>**	
Comentarios*	**<comentarios adicionais do requisito>**	

4.2. Tarefa 2.2 - Identificar e definir as novas naturezas

Uma natureza define o escopo de um dado específico. Permite delimitar o conjunto de valores e detalhes que tem o dado especifico. Em NDT se contempla também a definição de novas naturezas. A Tabela A3 mostra o padrão para as naturezas.

Tabela A3: Padrão para definir as novas naturezas.

NA-<id>	**<nome descritivo da natureza que está sendo definida>**	
Versão*	**<número da versão>**	**<data da versão>**
Autores*	**Nome do autor: <nome do autor>** **Cargo: <cargo do autor>** **Organização: <organização do autor>**	
	...	
	Nome do autor: <nome do autor> **Cargo: <cargo do autor>** **Organização: <organização do autor>**	
Fontes*	**Nome da fonte: <nome da fonte>** **Cargo: <cargo da fonte>** **Organização: <organização da fonte>**	
	...	
	Nome da fonte: <nome da fonte> **Cargo: <cargo da fonte>** **Organização: <organização da fonte>**	
Objetivos associados	- **OBJ-<X>: <nome do objetivo>** - **...**	
Descrição	**Esta natureza representa <descrição da natureza>**	
Dados específicos*	Campo	Natureza
	<nome do dado>:<breve descrição do dado>	**<natureza do dado> [<Cardinalidade:cardinalidade]**
	...	
	<nome do dado>:<breve descrição do dado>	**<natureza do dado> [<Cardinalidade:cardinalidade]**
Escopo*	**<escopo dos valores da natureza>**	
Restrições*	**<restrições que devem ter os dados da natureza>**	
Presentação*	**<descrição da maneira de como se representam os dados da natureza>**	
Importância*	**<importância da natureza>**	
Urgência*	**<urgência da natureza>**	
Estado*	**<estado da natureza>**	
Estabilidade*	**<estabilidade da natureza>**	
Comentarios*	**<comentarios adicionais da natureza>**	

5. Atividade 3 - Identificar e definir os atores

Esta atividade se baseia em quatro tarefas nas quais se definem os atores básicos, a derivação entre atores, a incompatibilidade e a generalização.

5.1. Tarefa 3.1 - Identificar e definir os atores básicos de sistema

O objetivo final desta tarefa é determinar os atores básicos que interagem com o sistema, de maneira que em tarefas e atividades posteriores se possa definir o sistema em base desta classificação. Um ator básico é todo ator que se identifica de forma individual atendendo a algum tipo de critério ou ponto de vista no momento de interagir com o sistema. Utiliza-se um novo padrão, na Tabela A4, para a definição de atores.

Tabela A4: Padrão para definir atores.

AT-<id>	**<nome descritivo do ator>**	
Versão*	**<número da versão>**	**<data da versão>**
Autores*	**Nome do autor: <nome do autor>** **Cargo: <cargo do autor>** **Organização: <organização do autor>**	
	...	
	Nome do autor: <nome do autor> **Cargo: <cargo do autor>** **Organização: <organização do autor>**	
Fontes*	**Nome da fonte: <nome da fonte>** **Cargo: <cargo da fonte>** **Organização: <organização da fonte>**	
	...	
	Nome da fonte: <nome da fonte> **Cargo: <cargo da fonte>** **Organização: <organização da fonte>**	
Objetivos associados	- **OBJ-<X>: <nome do objetivo>** - **...**	
Classificação*	**Este é um dos possiveis roles dentro do sistema quando se faz uma classificação dos atores em base a <descrição da classficação>**	
Descrição	**O sistema deverá prever o tratamento dos usuários que pertencem ao grupo descrito como <nome descritivo> e que se refere a pessoas que <descrição do grupo de pessoas que representa>**	
Herda de*	**AT-<id>:<nome do ator do qual herda>**	
Importância*	**<importância da existência do ator>**	
Urgência*	**<urgência da existência do ator>**	
Estado*	**<estado da definição do ator>**	
Estabilidade*	**<estabilidade da definição do ator>**	
Comentarios*	**<comentarios adicionais sobre o ator>**	

5.2. Tarefa 3.2 - Identificar e definir a generalização de atores

Quando se estudam os atores para um sistema nevegacional, em muitos casos, se podem identificar relações de especialização entre atores. Um ator especializado é todo aquele ator que se pode definir a partir dos atores básicos ou de outros atores especializados mediante uma relação de generalização (JBR99). A função associada a um ator especializado herda as funções associadas dos atores aos quais se especializam e podem adicionar semântica espcifica à função que desempenha.

5.3. Tarefa 3.3 - Identificar e definir a incompatibilidade entre atores

Uma vez identificados os atores básicos, incluidas as especializações, se analisa a compatibilidade de funções entre os mesmos atores. Definir a incompatibilidade entre atores será essencial para fases posteriores. Definir a incompatibilidade de funções num sistema na web é essencial visto que cada função incompativel pode chegar a definir uma interface completamente diferente à das outras funções.

5.4. Tarefa 3.4 - Identificar e definir os atores derivados

A última tarefa desta atividade estuda os atores derivados. A função associada a um ator derivado assume as funções correspondentes aos atores que o compôem. A definição dos atores derivados é muito útil para a definição dos requisitos funcionais e de interação.

6. Atividade 4 - Identificar e definir os requisitos funcionais

Nas atividades anteriores se estudou o que se armazena no sistema e quem pode fazer uso dessa informação. Além disso, precisam-se saber também as funções a serem realizadas pelo sistema. Os requisitos funcionais respondem a pergunta de: que tarefas podem ser realizadas com a ajuda do sistema?

6.1. Tarefa 4.1 - Projetar os diagramas de casos de uso

Os diagramas de casos de uso são uma das técnicas mais aceitadas como técnica de definicação de requisitos. Neles aparecem dois elementos importantes, o caso de uso e os atores. Os atores são definidos no NDT na atividade 3.

6.2. Tarefa 4.2 - Descobrir os casos de uso

Propõe-se um novo padrão. Esta tarefa tem por objetivo detalhar cada um dos diagramas da tarefa anterior e descreve-los de forma detalhada e estruturada num padrão similar ao mostrado na Tabela A5.

7. Atividade 5 - Identificar e definir os requisitos de interação

O sistema não só deve oferecer ao usuário toda a informação e funcionalidade que este lhe pede, deve também dar essa informação e funcionalidade no momento apropriado e da forma apropriada. Nesta atividade se define o que se conhece como requisitos de interação. Os requisitos de interação vão ser representados por dois aspectos: os critérios de recuperação ou fase e os protótipos de visualização.

7.1. Tarefa 5.1 - Identificar e definir as frases

A maneira como o usuário recupera a informação é mediante as frases. NDT assume o uso de BNL [BEM+01] para a definição do corpo das frases. Na Tabela A6 se apresenta o padrão para as frases.

Tabela A5: Padrão para definir os casos de uso.

RF-<id>	**<nome descritivo>**	
Versão*	**<número da versão>**	**<data da versão>**
	Nome do autor: <nome do autor> **Cargo: <cargo do autor>** **Organização: <organização do autor>**	
	...	
	Nome do autor: <nome do autor> **Cargo: <cargo do autor>** **Organização: <organização do autor>**	
Fontes*	**Nome da fonte: <nome da fonte>** **Cargo: <cargo da fonte>** **Organização: <organização da fonte>**	
	...	
	Nome da fonte: <nome da fonte> **Cargo: <cargo da fonte>** **Organização: <organização da fonte>**	
Objetivos associados	- **OBJ-<X>: <nome do objetivo>** - **...**	
Descrição	**O sistema deverá se comportar de acordo ao seguinte caso de uso e que representa <descrição do significado da ação que é executada no caso de uso>**	
Precondição	**<precondição do caso de uso>**	
Atores*	Ator do caso de uso	Ator do sistema
	Ator 1	**i. AC-x: <nome do ator>** **ii. ...**
	...	
Sequencia normal	Passo	Ação
	N	**<ação associada>**
	...	
Postcondição*	**<postcondição do sistema>**	
Exceções*	Passo	Ação
	N	**<ação associada á exeção>**
	...	
Rendimento*	Passo	Cota do tempo
	N	**M <unidade>**
	...	
Frequencia esperada*	**<n° de vezes>vezes / <unidade de tempo>**	
Importância*	**<importância do requisito>**	
Urgência*	**<urgência do requisito>**	
Estado*	**<estado da definição do requisito>**	
Estabilidade*	**<estabilidade da definição do requisito>**	
Comentarios*	**<comentarios adicionais do requisito>**	

Tabela A6: Padrão para definir as frases.

FR-<id>	**<nome descritivo da frase>**	
Versão*	**<número da versão>**	**<data da versão>**
Autores*	**Nome do autor: <nome do autor>** **Cargo: <cargo do autor>** **Organização: <organização do autor>**	
	...	
	Nome do autor: <nome do autor> **Cargo: <cargo do autor>** **Organização: <organização do autor>**	
Fontes*	**Nome da fonte: <nome da fonte>** **Cargo: <cargo da fonte>** **Organização: <organização da fonte>**	
	...	
	Nome da fonte: <nome da fonte> **Cargo: <cargo da fonte>** **Organização: <organização da fonte>**	
Objetivos associados	- **OBJ-<X>: <nome do objetivo>** - **...**	
Descrição das frases	Corpo	Frases
	<corpo da frase>	**i. AT-x:<nome do ator>** **ii. ...**
	...	
Importância*	**<importância da existência da frase>**	
Urgência*	**<urgência da existência da frase>**	
Estado*	**<estado da definição da frase>**	
Estabilidade*	**<estabilidade da definição da frase>**	
Comentarios*	**<comentarios adicionais sobre a frase>**	

7.2. Tarefa 5.2 - Identificar e definir os protótipos de visualização

Quando se definem os protótipos de visualização de dados, se faz referência aos dados mostrados a cada um dos atores e a funcionalidade que se lhe associa a cada módulo de apresentação da informação. Além disso, os protótipos de visualização permitem expressar as possiblidades de navegação que existem no sistema. Para recolher no documento de requisitos do sistema os protótipos de visualização de dados se faz uso de padrão cuja estrutrura se descreve na Tabela A7.

8. Atividade 6 - Identificar e definir os requisitos não funcionais

Em todas as etapas descreveram-se todas as necessidades de armazenamento, de funcionalidade e de interação do sistema. Mas precisa-se também definir as restrições que se coloca sobre o sistema, ou seja, os requisitos não funcionais. Para recolher esses requisitos dentro do documento de requisitos, se usará novamente um padrão. A estrutura deste padrão se apresenta na Tabela A8.

Tabela A7: Padrão para definir protótipos de visualização.

PV-<id>	**<nome descritivo do protótipo>**
Versão*	**<número da versão>** | **<data da versão>**
Autores*	**Nome do autor: <nome do autor>** **Cargo: <cargo do autor>** **Organização: <organização do autor>**
	...
	Nome do autor: <nome do autor> **Cargo: <cargo do autor>** **Organização: <organização do autor>**
Fontes*	**Nome da fonte: <nome da fonte>** **Cargo: <cargo da fonte>** **Organização: <organização da fonte>**
	...
	Nome da fonte: <nome da fonte> **Cargo: <cargo da fonte>** **Organização: <organização da fonte>**
Objetivos associados	- **OBJ-<X>: <nome descritivo do objetivo>** - **...**
Ator/es	- **AT-<X>: <nome descritivo do ator>** - **...**
Descrição	**O sistema deverá permitir a visualização dos dados concretos que se mostram a seguir e a navegação que representam <resumo da informação que mostra o protótipo>**
Frases	**[<condição 1>]** **i. FR-x: <nome descritivo da frase>** **ii. FR-y: <nome descritivo da frase> ...**
	...
	[<condição n>] ...
Funcionalidade associada	**[<condição 1>]** **i. RF-x: <nome descritivo do requisito>** **ii. RF-y: <nome descritivo do requisito> ...**
	...
	[<condição n>] ...
Informação visualizada	**[<condição 1>]** **RA-x.<dado especifico>**
	...
	[<condição n>] **RA-z.<dado especifico>**
Protótipos de saída	**[<condição 1>]** **PV-x[(DeVolta, Múltiple)]**
	...
	[<condição n>] **PV-z[(DeVolta, Múltiple)]**
Protótipos de entrada	**[<condição 1>]** **PV-x**
	...
	[<condição n>] **PV-z**
Importância*	**<importância do protótipo>**
Urgência*	**<urgência do protótipo>**
Estado*	**<estado da definição do protótipo>**
Estabilidade*	**<estabilidade da definição do protótipo>**
Comentarios*	**<comentarios adicionais sobre o protótipo>**

Tabela A8: Padrão para definir requisitos não funcionais.

RNF-<id>	**<nome descritivo>**	
Versão*	**<número da versão>**	**<data da versão>**
Autores*	**Nome do autor: <nome do autor>** **Cargo: <cargo do autor>** **Organização: <organização do autor>**	
	...	
	Nome do autor: <nome do autor> **Cargo: <cargo do autor>** **Organização: <organização do autor>**	
Fontes*	**Nome da fonte: <nome da fonte>** **Cargo: <cargo da fonte>** **Organização: <organização da fonte>**	
	...	
	Nome da fonte: <nome da fonte> **Cargo: <cargo da fonte>** **Organização: <organização da fonte>**	
Objetivos associados	- **OBJ-<X>: <nome do objetivo>** - **...**	
Descrição	**O sistema deverá <capacidade do sistema>**	
Importância*	**<importância do requisito>**	
Urgência*	**<urgência do requisito>**	
Estado*	**<estado do requisito>**	
Estabilidade*	**<estabilidade do requisito>**	
Comentarios*	**<comentarios adicionais do requisito>**	

9. Atividade 7 - Validar os requisitos

NDT propõe a criação de uma matriz de rastreabilidade que permita verificar se todos os requisitos do sistema foram definidos com a finalidade de atingir um objetivo e se todos os objetivos são alcançados com a realização de pelo menos, um requisito.

9.1. Tarefa 7.1 - Criar matriz de rastreabilidade

A criação da matriz de rastreabilidade permite validar os resultados obtidos na especificação de requisitos. Na Tabela A9 se apresenta um exemplo possível desta matriz. O símbolo * indica que o requisito indicado nessa linha serve para alcançar total ou parcialmente o objetivo da coluna.

10. Atividade 8 - Gerar o documento de requisito do sistema

Como resultado do processo de Engenharia de Requisitos se deve gerar o documento de Requisitos do Sistema (DRS). Veja-se sua estrutura na Figura A2.

Tabela A9: Matriz de rastreabilidade.

	OBJ-1	OBJ-2	...	OBJ-n
RA-01	*			
RA-02				*
...				
AT-01	*	*		
AT-02				
…				
RF-01		*		
RF-02		*		
…				
FR-01	*			
FR-02	*			
…				
PV-01	*			
PV-02		*		
…				
RNF-01				
RNF-02				
…				

Página de rosto
Folha de controle de mudanças
Índice
Lista de figuras
Lista de tabelas

1. Objetivo do projeto
2. Participantes
3. Objetivos do sistema
4. Catálogo de requisitos
   - 4.1. Requisitos de armazenamento de informação
     - 4.1.1. Definição dos requisitos de armazenamento de informação
     - 4.1.2. Definição das novas naturezas
   - 4.2. Definição de atores
     - 4.2.1. Definição dos atores básicos
     - 4.2.2. Incompatibilidade de atores
     - 4.2.3. Generalização de atores
     - 4.2.4. Atores derivados
   - 4.3. Requisitos funcionais
     - 4.3.1. Diagramas de casos de uso
     - 4.3.2. Definição dos casos de uso do sistema
   - 4.4. Requisitos de interação
     - 4.4.1. Definição das frases
     - 4.4.2. Definição dos protótipos de visualização
   - 4.5. Requisitos não funcionais
5. Matriz de rastreabilidade

. **Figura A2:** Estructura do documento de requisitos do sistema (DRS).

11. A análise em NDT

O processo da análise em NDT tem como objetivo obter os modelos da análise de um sistema web. NDT assume que para realizar a análise de um sistema web é necessário realizar os modelos:

- ✓ O modelo conceitual, que representa a estrutura estática do sistema.
- ✓ O modelo de navegação, que representa a estructura de navegacção do sistema.
- ✓ O modelo de protótipos do sistema que representa.

O modelo conceitual é formado por um único diagrama. No entanto, o modelo de navegação e o modelo de protótipos podem ser muito diferentes dependendo do papel do usuário que interage com o sistema em um dado momento. Por isso, estes modelos serão adaptados de acordo com o ator em estudo. Um ator em estudo é um papel, ou um conjunto de papeis, do usuário que compartilham a mesma navegação.

A realização destes modelos se faz em NDT de uma maneira sistemática desde os requisitos. Partindo dos resultados da fase anterior, NDT propõe uma série de processos que permitem gerar estes modelos. No entanto, os modelos gerados desde os requisitos devem ser revisados para detectar possiveis erros ou incongruências em fases anteriores, ou para adequá-los melhor à realidade do sistema. As mudanças e modificações possiveis são controlados por NDT.

Desta forma, como se mostra na figura 5, a análise se compoõe de quatro fases. A primeira delas permite obter o modelo conceitual, a segunda o modelo de navegação, a terceira os protótipos e, na última, se gera o documento de análise do sistema, resultado desta fase.

As três primeiras atividades, por sua vez, se dividem em duas tarefas. Nas primeiras, se consiguem os modelos básicos, de maneira sistemática desde os requisitos. Nas segundas, o grupo de analistas pode aplicar sua experiência para melhorar os resultados, mas as mudanças e modificações que estes podem realizar são controladas por NDT.

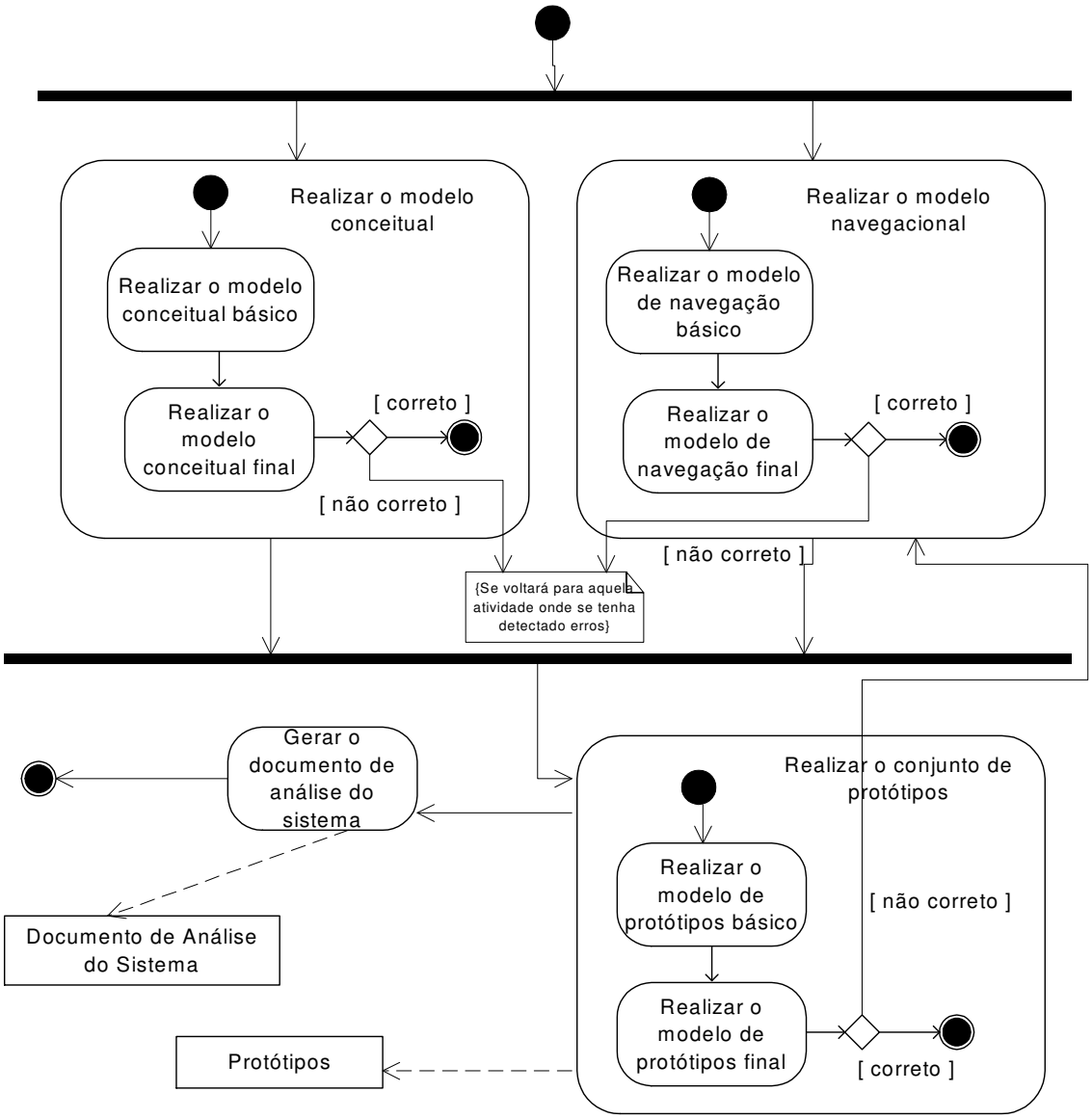


Figura A3: Processo de Análise do NDT.

12. Atividade 1 - Realizar o modelo conceitual

O modelo conceitual representa a estrutura estática do sistema. Permite modelar como se estrutura a informação que maneja o sistema.

Em NDT o modelo conceitual vem representado por dois elementos:

- ✓ O diagrama de classes conceituais, representado usando a notação de UML.
- ✓ O dicionário de dados, que descreve de maneira textual, o significado das classes e as asociaciações de diagrama anterior.

12.1. Elementos usados e nomenclatura

No diagrama conceitual aparecem três tipos de elementos: classes, asociaciações e pacotes, representados segundo UML.

1- **Classes.** NDT considera a definição de UML, uma classe é uma descrição de um conjunto de objetos do sistema que compartilham os mesmos atributos. O padrão para definir as classes em NDT é aquele que se mostra na Tabela A10.

2- **Associações.** Também evoluiu o padrão geral para a definição das asociações o qual é representado na Tabela A11.

3- **Pacotes.** Segundo UML um pacote é um mecanismo que permite organizar elementos em grupos. NDT permite incorporar pacotes no seu modelo conceitual. O padrão é mostrado na Tabela A12.

Tabela A10: Padrão para definir uma classe.

{CL-<id>/ CLn-<id>}	**<nome descritivo da clase>**		
Versão*	**<número da versão>**		**<data da versão>**
Autores*	**Nome do autor: <nome do autor>** **Cargo: <cargo do autor>** **Organização: <organização do autor>**		
	...		
	Nome do autor: <nome do autor> **Cargo: <cargo do autor>** **Organização: <organização do autor>**		
Requisito/ Natureza	- **[RA-x:<nome do requisito>, NA-x:<nome da natureza>]** - **...**		
Herda de*	**[CL-x, CLn-x]:<nome da clase pãe>**		
Descrição	**O sistema deverá armazenar a informação referente ao <conceito relevante>. Especificamente:**		
Atributos	Descrição	Significado	Dado específico
	[visibilidade]nome[multiplicidade] **[:tipo][=valor inicial][{Propriedades}]**	**<significado do atributo>**	**{RA-x.<dado>, NA-x.<campo>}**
	...		
	[visibilidade]nome[multiplicidade] **[:tipo][=valor inicial][{Propriedades}]**	**<significado do atributo>**	**{RA-x.<dado>, NA-x.<campo>}**
Estado	**{básico, final}**		
Comentarios*	**<comentarios adicionais sobre a clase>**		

Tabela A11: Padrão para definir uma associação

AS-<id>	**<nome descritio da associação>**		
Versão*	**<número da versão>**		**<data da versão>**
Autores*	**Nome do autor: <nome do autor>** **Cargo: <cargo do autor>** **Organização: <organização do autor>**		
	...		
	Nome do autor: <nome do autor> **Cargo: <cargo do autor>** **Organização: <organização do autor>**		
Requisitos	- **[RA-x:<nome do requisito>** - **...**		
Descrição	**As classes <CL-x> e <CL-y> se relacionam mediante esta associação que representa <significado da associação>**		
Tipo	**{unidirecional, bidirecional}**		
Classes	Nome da classe	Nome do rol	Cardinalidade
	A.{CL-x, CLn-x}	**<rol da associação para a classe>**	**<cardinalidade da classe>**
	...		
Atributos*	Descrição		Significado
	[visibilidade]nome[multiplicidade][:tipo][{Propriedades}]		**<significado do atributo>**
	...		
Dados específicos	**i. <RA-x>.<dado concreto>** **ii. ...**		
Estado	**{básico, final}**		
Comentarios	**<comentarios adicionais sobre a associação>**		

Tabela A12: Padrão para definir um pacote.

PT-<id>	<nome descritivo do pacote>	
Versão*	<número da versão>	data da versão>
Autores*	Nome do autor: <nome do autor> Cargo: <cargo do autor> Organização: <organização do autor>	
	...	
	Nome do autor: <nome do autor> Cargo: <cargo do autor> Organização: <organização do autor>	
Descrição	<descrição do pacote>	
Comentarios*	<comentarios adicionais sobre o pacote>	

12.2. Tarefa 1.1 - Realizar o modelo conceitual básico

Para conseguir o modelo conceitual básico, se deve partir dos resultados obtidos durante a atividade 2 da Engenharía de Requisitos. Nesta primeira tarefa da análise, usam-se os padrões que definem os requisitos de armazenamento de informação e às novas naturezas e a partir deles se gera o diagrama de classes.

Passo 1 - Desde os requisitos de armazenamento até o diagrama de classes

O primeiro passo do processo gera as classes do sistema que procedem dos requisitos de armazenamento de informação. Basicamente, para cada requisito de armazenamento, RA-x, se gera uma classe, CL-x. Dependendo de se a natureza do dado específico é outro requisito ou não, se atua de maneira diferente. Se a natureza não é outro requisito de armazenamento, o dado se traduz em um atributo da classe.

Passo 2 - Desde as novas naturezas até o diagrama de classes

O seguinte passo do processo inicia nos padrões de definição das novas naturezas. Vão ser umas classes especiais estereotipadas mediante o estereotipo <<type>> de UML no diagrama conceitual.

Passo 3 - Gerar os pacotes

Embora não seja necessário, se no sistema existem novas naturezas, se aconselha para aumentar a clareza, o agrupamento das classes em dois pacotes. O primeiro pacote denominado classes do sistema que contém todas as classes, CL-x, derivadas dos requisitos de armazenamento de informação. Por otro lado, se define um segundo pacote denominado novas naturezas que contém todas as clases, CLn-x, derivadas a partir das novas naturezas do sistema. Se não existem novas naturezas no sistema, este passo é desnecessário.

Passo 4 - Gerar o diccionario de dados

Uma vez terminados os passos anteriores, deve-se complementar o padrão correspondente a cada classe, a cada asociaciação e a cada pacote dos definidos nos passos anteriores.

12.3. Tarefa 1.2 - Realizar o modelo conceitual final

NDT permite fazer modificações sobre este modelo básico. No entanto, estas modificações devem ser controladas.

13. Atividade 2 - Realizar o modelo de navegação

O modelo de navegação ou navegacional em NDT representa:

1- Como se pode navegar através da informação conceitual, pelo que o modelo navegacional vai ser uma visão do modelo conceitual.

2- Que elementos vão aparecer nessa navegação e como vão-se adaptar ao usuário que interage com o sistema.

3- As relações que aparecem entre esses elementos da navegação.

O modelo de navegação em NDT é representado por dois elementos:

- O modelo de classes navegacionais, que se compõe por uma série de diagramas de classes especiais denominadas classes navegacionais, as quais são descritas mediante a nomenclatura de UWE [KOC01].

- O diccionário de dados desse modelo, composto por uma serie de padrões.

13.1. Elementos usados e nomenclatura

Os elementos que aparecem em um modelo de navegação são as classes de navegação, os índices, as rotas guiadas, os menús e os elos (links) de navegação.

1. **Classes de navegação (navegação classe) ou nós**, são descritos de acordo com o padrão mostrado na Tabela A13.

2. **Índices (index),** são descritos de acordo com o padrão da Tabela A14. Os campos acrescentados têm um significado similar ao padrão dos nodos.

3. **Consultas (query),** são descritas de acordo com o padrão da Tabela A15.

Tabela A13: Padrão para definir um nó.

NO-<id>	<nome descritivo do nó>	
Versão*	<número da versão>	<data da versão>
Autores*	Nome do autor: <nome do autor> Cargo: <cargo do autor> Organização: <organização do autor>	
	...	
	Nome do autor: <nome do autor> Cargo: <cargo do autor> Organização: <organização do autor>	
Protótipo	• PV-x: <nome do protótipo> • ...	
Atores em estudo	• AE-x • ...	
Descrição	O sistema deverá mostrar a informação referente ao <conceito relevante>. Especificamente:	
Operações	• [<condição1>] RF-x: <nome descritivo do requisito> RF-y: <nome descritivo do requisito> ... • [<condição2>] ...	
Atributos	**Descrição**	**Protótipo**
	<nome>	PV-x
	...	
	<nome>	PV-y
Estado	{básico, final}	
Comentarios*	<comentarios adicionais sobre o nó>	

Tabela A14: Padrão para definir um índice.

IN-<id>	<nome descritivo do índice>	
Versão*	<número da versão>	<data da versão>
Autores*	Nome do autor: <nome do autor> Cargo: <cargo do autor> Organização: <organização do autor>	
	...	
	Nome do autor: <nome do autor> Cargo: <cargo do autor> Organização: <organização do autor>	
Destinos	{NO-x: <nome descritivo do nó o qual dá entrada o índice>, IN-x: <nome do índice ao qual dá entrada o índice> QU-x: <nome da query ao qual dá entrada o índice> ME-x: <nome do menu ao qual dá entrada o índice> }	
Atores em estudo	• AE-x • ...	
Descrição	O sistema deverá permitir selecionar de uma lista os dados referentes a <descrição da lista>	
Tipo	{índice, rota guiada}	
Estado	{básico, final}	
Comentarios*	<comentarios adicionais sobre o índice>	

. **Tabela A15**: Padrão para definir uma consulta.

QU-<id>	<nome descritivo da query>	
Versão*	<número da versão>	<data da versão>
Autores*	Nome do autor: <nome do autor> Cargo: <cargo do autor> Organização: <organização do autor>	
	...	
	Nome do autor: <nome do autor> Cargo: <cargo do autor> Organização: <organização do autor>	
Destino	{NO-x: <nome descritivo do nó o qual dá entrada o índice>, IN-x: <nome do índice ao qual dá entrada o índice> QU-x: <nome da query ao qual dá entrada o índice> ME-x: <nome do menu ao qual dá entrada o índice> }	
Descrição	O sistema deverá permitir obter informação sobre o usuário a partir das seguintes frases:	
	Corpo	**Atores**
	• <corpo da frase>	• AE-x • ...
	• ...	• ...
Estado	{básico, final}	
Comentarios*	<comentarios adicionais sobre a query>	

4. **Menús**, vem descrito pelo padrão da Tabela A16.

Tabela A16: Padrão para definir um menú.

ME-<id>	<nome descritivo do menu>	
Versão*	<número da versão>	<data da versão>
Autores*	Nome do autor: <nome do autor> Cargo: <cargo do autor> Organização: <organização do autor>	
	...	
	Nome do autor: <nome do autor> Cargo: <cargo do autor> Organização: <organização do autor>	
Destinos	{NO-x: <nome descritivo do nó o qual dá entrada o índice>, IN-x: <nome do índice ao qual dá entrada o índice> QU-x: <nome da query ao qual dá entrada o índice> ME-x: <nome do menu ao qual dá entrada o índice> }	
Atores em estudo	• AE-x • ...	
Estado	{básico, final}	
Comentarios*	<comentarios adicionais sobre o menu>	

5. **Elos de navegação (direct navigability)**, por último, os elos de navegação são descritos de acordo com o padrão da Tabela A17.

Tabela A17: Padrão para definir um elo de navegação.

EL-<id>	<nome descritivo do elo>	
Versão*	<número da versão>	<data da versão>
Autores*	Nome do autor: <nome do autor> Cargo: <cargo do autor> Organização: <organização do autor>	
	...	
	Nome do autor: <nome do autor> Cargo: <cargo do autor> Organização: <organização do autor>	
Origem	{NO-x:<nome do nó>, IN-x:<nome do índice>, QU-x:<nome da query>, ME-x:<nome do menu>}	
Destino	{NO-y:<nome do nó>, IN-y:<nome do índice>, QU-y:<nome da query>, ME-y:<nome do menu>}	
Atores em estudo	• AE-x • ...	
Descrição	O sistema deve permitir a navegação entre os elementos {NO-x, IN-x, QU-x, ME-x} e os elementos {NO-y, IN-y, QU-y, ME-y} e que representa a <descrição do elo>	
Tipo	{bidirecional, unidirecional}	
Estado	{básico, final}	
Comentarios*	<comentarios adicionais sobre o elo>	

13.2. Tarefa 2.1 - Definir os atores em estudo

O sistema de navegação pode variar substancialmente dependendo do ator que em cada momento interatue com o sistema. Nesta primeira tarefa classifica-se e estuda-se os

atores para ver quais deles compartilham um sistema de navegação similar.

A definição dos atores em estudo se faz segundo um processo sistemático. Para explicar este proceso, se define AE como o conjunto de atores em estudo. Para conseguir o conjunto de atores em estudo há que realizar três passos:

1- Começa-se definindo AE como o conjunto vazio.

2- Cada ator do sistema que não herde de nenhum outro ator, é incluído no conjunto AE como um ator em estudo.

3- Dos atores em estudo que se herdam de outro ator:

a. Se existe um protótipo que seja accessivel pelo ator filho e não pelo ator pai, se acrescenta o ator filho ao conjunto de atores em estudo como um ator independente.

b. Senão, o ator filho passa a ser representado pelo ator em estudo que representa o seu pai.

13.3. Tarefa 2.2 - Realizar o modelo de navegação básico

O modelo de navegaçaõ básico tem o mesmo enfoque que o modelo conceitual básico. Pode ser obtido diretamente do documento de requisitos do sistema de uma maneira sistemática utilizando as propriedades de derivação apresentadas.

Entretanto, este processo, à diferença do explicado para gerar o modelo conceitual básico, não se realiza de uma só vez. É necessário aplicá-lo para cada um dos atores em estudo definidos.

Passo 1 - Desde os protótipos de visualização até os nos

No primeiro passo do processo, se geram os nós ou classes de navegação a partir dos protótipos de visualização. Basicamente, cada protótipo gera um nó. Os campos dos nodos se completam com parte da informação que têm os protótipos de visualização.

Passo 2 - Desde os protótipos de visualização até os elos simples

O campo de protótipos de saída e os protótipos de entrada dos protótipos de visualização permitem definir a navegação entre nodos.

Passo 3 - Desde os protótipos de visualização até os índices

Nos padrões dos protótipos de visualização, no campo de protótipos de saída, podem existir dois modificadores. O modificador de volta serve para detectar se o elo é bidirecional ou não, como se viu no passo anterior. No caso do outro modificador, múltiple, se vai usar para detectar a necessidade de criar índice.

Passo 4 - Desde os protótipos de visualização e as frases até as querys

As querys representam pontos de navegação nos quais é necessário obter informação desde o usuário para poder continuar navegando. As querys são uns elementos fáceis de detectar a partir do documento de requisitos. Neste caso, não só se usam os padrões dos prototipos de visualização, também se usam os padrões das frases.

Passo 5 - Desde o modelo de navegação até os menús

O passo para detectar os menús, não se baseia como nos casos anteriores no documento de requisitos. O passo de detectação de menús na verdade se baseia em garantir a qualidade do modelo de navegação final. A inclusão de menús no modelo de navegação vai ter como objetivo de garantir que todas as partes do modelo navegacional sejam alcançaveis desde qualquer outro ponto.

O processo se baseia na detectação das componentes conexas. Com os elementos do modelo navegacional se pode construir um grafo, denominado grafo de navegação. Neste grafo, os nós, os índices e as querys são os vértices e os elos são as aristas. Se se estuda este grafo e se detectam diferentes componentes conexas, significa que há zonas do grafo de navegação separadas, sem elos que as conectem pelo que é necessário criar um menú que as conecte todas.

Passo 6 - Gerar o diccionário de dados

Uma vez gerado o modelo de navegação básica, se deve gerar a primeira versão do diccionário de dados que o descreve.

13.4. Tarefa 2.3 - Realizar o modelo de navegação final

De igual forma que o modelo conceitual, depois de realizar o modelo navegacional de maneira sistemática tal e como se define na tarefa anterior, o analista deve realizar uma série de revisões para ver se o modelo pode ser modificado para adequar-se em maior medida à realidade. Estas revisões podem ocasionar uma série de modificações que, em alguns casos, afectam modelos anteriores.

14. Atividade 3 - Gerar o documento de análise do sistema

Na sua estrutura básica (veja-se Figura A4), o documento de análise do sistema é bastante similar ao documento de requisitos do sistema.

Página de rosto
Folha de controle de mudanças
Índice
Lista de figuras
Lista de tabelas

1. Objetivo do projeto
2. Participantes
3. Objetivos do sistema
4. Modelo conceitual
   - 1.1. Diagrama de classes conceituais
   - 1.2. Dicionário de dados
5. Modelo de navegação
   - 5.1. Definição de atores em estudo
   - 5.2. Modelo de navegação para AE-01
     - 5.2.1. Diagrama de classes de navegação
     - 5.2.2. Dicionário de dados
   - 5.3. Modelo de navegação para AE-02
     - 5.3.1. Diagrama de classes de navegação
     - 5.3.2. Dicionário de dados

Glossário de termos [opcional]
Apêndices y anexos [opcionais]

Figura A4: Estructura do documento de requisitos do sistema (DRS).

Apêndices

Apêndice A – Documento de Requisitos do Sistema (DRS) - NDT

Nesta seção se apresenta o Documento de Requisitos do Sistema (DRS). Todos os padrões do NDT gerados na fase de Engenharia de Requisitos no processo do desenvolvimento do sistema WebMaps são apresentados. Recomenda-se seguir estes padrões com a leitura do Capítulo 6 deste trabalho.

Identificação e definição dos objetivos do sistema

OBJ – 01	Fornecer dados e informações de uma determinada região geográfica.
Descrição	O sistema deverá fornecer dados e informações , curvas ou imagens NDVI, de acordo a sua região geográfica de interesse.

OBJ - 02	Trabalhar com dados geo-refenciados
Descrição	O sistema deverá trabalhar com imagens satelitais periódicas, úteis para a geração das curvas NDVI.

OBJ – 03	Adequar o sistema ao perfil do usuário
Descrição	O sistema deverá permitir diferentes tipos ou níveis de permissões para os usuários.

Identificação e definição dos requisitos de armazenamento de informação

RA – 01	Usuário	
Objetivos asociados	OBJ – 01, OBJ – 03	
Descrição	O sistema deverá armazenar as informações correspondentes ao usuário	
Dados específicos	**Nome e descrição**	**Natureza**
	Código	Númerico
	Nome	Texto
	Login	Texto
	Password	Texto
	E – mail	Texto

	Propriedade	**RA – 04** Cardinalidade: 1..n
	Endereço	Texto
	Bairro	Texto
	Complemento	Texto
	Tipo de usuário	Texto
	Cidade	Texto
	Estado	Texto
	Consulta	**RA – 07** Cardinalidade: 0..n

RA – 02	Imagem de satélite	
Objetivos asociados	OBJ – 01, OBJ – 2	
Descrição	O sistema deverá armazenar as imagens de satélite.	
Dados específicos	**Nome e descrição**	**Natureza**
	Código	Númerico
	Data	Data
	Descrição	Texto
	MBB	**RA – 05** Cardinalidade: 0..n

RA – 03	Máscara	
Objetivos asociados	OBJ – 01, OBJ – 2	
Descrição	O sistema deverá armazenar as máscaras das propriedades e/ou dos talhões para agilizar as consultas.	
Dados específicos	**Nome e descrição**	**Natureza**
	Código	Númerico
	Arquivo	Arquivo
	Data	Data
	Formato	Texto
	Tipo	Texto
	Talhao	**RA – 06** Cardinalidade: 1..1
	Propriedade	**RA – 04** Cardinalidade: 1..1
	MBB	**RA – 05** Cardinalidade: 1..n

RA – 04	Propriedade	
Objetivos asociados	OBJ – 1, OBJ – 02	
Descrição	O sistema deverá armazenar as informações correspondentes a uma propriedade.	
Dados específicos	**Nome e descrição**	**Natureza**
	Código	Númerico
	Nome	Texto
	Descrição	Texto
	Coordenadas	Geo-referenciado
	Tipo	Texto
	Talhão	**RA – 06** Cardinalidade: 1..n
	Máscara	**RA – 03** Cardinalidade:1..1
	MBB	**RA – 05** Cardinalidade: 1..n
	Usuário	**RA – 01** Cardinalidade: 1..n
	Consulta	**RA – 07** Cardinalidade: 0..n

RA – 05	Imagem MBB	
Objetivos asociados	OBJ – 01, OBJ – 2	
Descrição	O sistema deverá armazenar as informações correspondentes aos recortes MBB (Minimum Bounding Box)	
Dados específicos	**Nome e descrição**	**Natureza**
	Código	Númerico
	Arquivo	Image
	Data	Data
	Propriedade	**RA – 04** Cardinalidade: 1..1
	Talhão	**RA – 06** Cardinalidade: 1..1
	Imagem de satélite	**RA – 02** Cardinalidade: 1..1
	Máscara	**RA – 03** Cardinalidade: 1..1

RA – 06	Talhão
Objetivos asociados	OBJ – 01, OBJ – 2

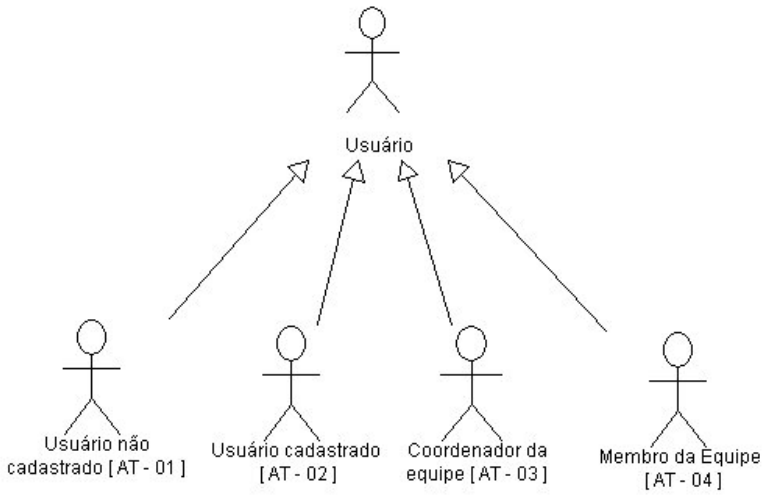
Descrição	O sistema deverá armazenar as informações correspondentes aos talhões que compoem uma propriedade	
Dados específicos	**Nome e descrição**	**Natureza**
	Código	Númerico
	Nome	Texto
	Data inicial	Data
	Data final	Data
	Coordenadas	Geo – referenciado
	Cultura	Texto
	Arquivo	Arquivo
	Máscara	**RA – 03** Cardinalidade: 1..1
	MBB	**RA – 06** Cardinalidade: 1..n
	Propriedade	**RA – 04** Cardinalidade: 1..1
	Consulta	**RA – 07** Cardinalidade: 0..n

RA – 07	Consulta	
Objetivos asociados	OBJ – 01, OBJ – 2, OBJ – 3	
Descrição	O sistema deverá armazenar as informações correspondientes a uma consulta (Curva NDVI).	
Dados específicos	**Nome e descrição**	**Natureza**
	Código	Numérico
	Arquivo	Image
	Data inicial	Data
	Data final	Data
	Descrição	Texto
	Usuário	**RA – 01** Cardinalidade: 1..1
	Propriedade	**RA – 04** Cardinalidade: 1..1
	Talhão	**RA – 06** Cardinalidade: 1..1
	Tipo	Texto
	Status	Booleano

Identificação e definição dos atores

AT – 01	Usuário não cadastrado
Objetivos asociados	OBJ – 3
Descrição	O sistema deverá prever o tratamento dos usuários que não estão cadastrados e que são aquelas pessoas que se conetam ao sistema para fazer consultas de interesse geral.

AT – 02	Usuário cadastrado
Objetivos asociados	OBJ – 3
Descrição	O sistema deverá prever o tratamento dos usuários que pertencem ao grupo de usuários cadastrados e que se refere a aquela pessoa que acessa no sistema e faz as consultas de acordo as coordenadas de sua região de interesse.



Modelo de Atores.

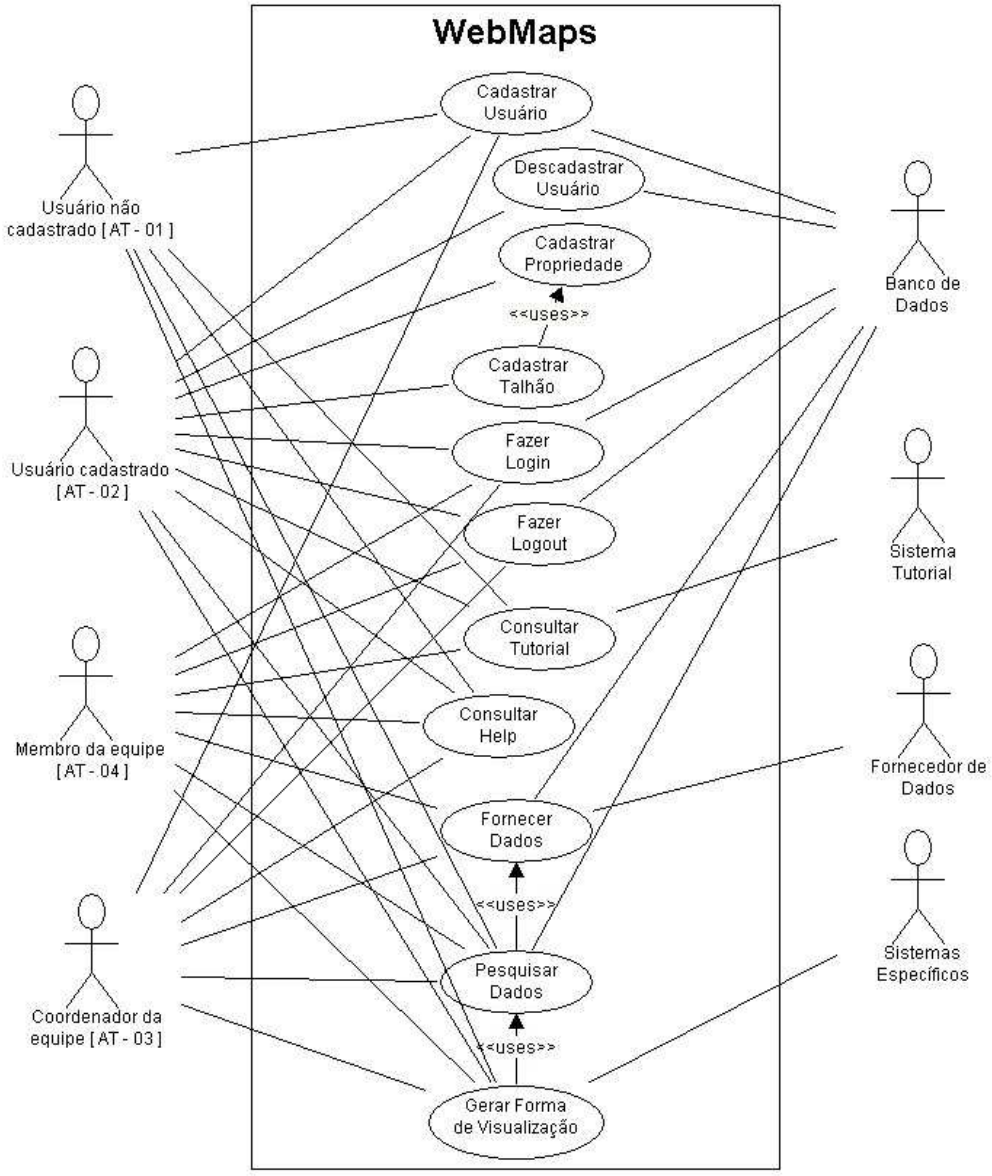
AT – 03	Coordenador da equipe
Objetivos asociados	OBJ – 3
Descrição	O sistema deverá prever o tratamento dos usuários que pertencem ao grupo de usuários cadastrados e que se refere a aquela pessoa que acessa no sistema como coordenador da equipe e tem privilegios para fazer mudanças no sistema.

AT – 04	Membro da equipe
Objetivos asociados	OBJ – 3
Descrição	O sistema deverá prever o tratamento dos usuários que pertencem ao grupo de usuários cadastrados e que se refere a aquela pessoa que acessa no sistema como membro da equipe o qual tem privilegios para fazer mudanças no sistema embora menos que o coordenador do projeto.

ATORES	AT – 01	AT – 02	AT – 03	AT – 04
AT – 01	-	X	X	X
AT – 02		-	X	
AT – 03			-	X
AT – 04				-

Tabela : Matriz de incompatibilidade de atores.

Identificação e definição dos requisitos funcionais



Modelo de casos de uso.

RF – 01	Cadastrar propriedade	
Objetivos Asociados	OBJ – 01 OBJ – 02	
Descrição	O sistema deverá comportar-se tal e como se descreve no seguinte caso de uso e que representa a possibilidade de que o sistema ofereça para o usuário cadastrar sua propriedade ou região de interesse.	
Atores	**Ator do caso de uso**	**Ator do sistema**
	Usuário cadastrado	AT – 02
Sequencia	**Passo**	**Ação**

normal		
	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
Postcondição	São feitos os recortes das imagens de satélite e é gerado 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 escolher um tipo de acceso privado para sua propriedade, ele ainda pode dar acesso a ela a algum usuário cadastrado específico.

RF – 02	Cadastrar talhão	
Objetivos Asociados	OBJ – 01 OBJ – 02	
Descrição	O sistema deverá comportar-se tal e como se descreve no seguinte caso de uso e que representa a possibilidade de que o sistema ofereça para o usuário cadastrar um talhão ou região de interesse em sua propriedade.	
Precondição	Deve existir a propriedade cadastrada da qual faz parte o talhão a cadastrar.	
Atores	**Ator do caso de uso**	**Ator do sistema**
	Usuário cadastrado	AT – 02
Sequencia normal	**Passo**	**Ação**
	1	O usuário solicita a realização de um cadastro de talhão.
	2	O sistema mostra uma tela com uma listagem das propriedades cadastradas pelo usuário.
	3	O usuário seleciona a propriedade da qual faz parte o talhão a cadastrar.
	4	O usuário introduz o periodo, data inicial e data final, do talhão.
	5	O usuário introduz as coordenadas do talhão.
	6	O usuário introduz o tipo de cultura do talhão.
	7	O usuário pede ao sistema cadastrar seu talhão.
	8	O sistema faz o cadastro do talhão.
Postcondição	São feitos os recortes das imagens de satélite e é gerado a máscara	

	correspondente ao talhão.

RF – 03	Pesquisar dados	
Objetivos Asociados	OBJ – 01 OBJ – 02	
Descrição	O sistema deverá comportar-se tal e como se descreve no seguinte caso de uso e que representa a possibilidade de que o sistema ofereça para o usuário pesquisar dados referentes a suas regiões cadastradas.	
Precondição	Deve existir cadastro da propriedade e/ou talhão a consultar.	
Atores	**Ator do caso de uso**	**Ator do sistema**
	Usuário cadastrado	AT – 02
Sequencia normal	**Passo**	**Ação**
	1	O usuário solicita a realização de uma consulta.
	2	O sistema mostra uma tela com uma listagem das propriedades cadastradas pelo usuário.
	3	O usuário seleciona a propriedade da qual faz parte o talhão a consultar.
	4	O usuário introduz o periodo, data inicial e data final, do talhão a consultar.
	5	O usuário introduz o tipo de acesso de sua propriedade (público / privado)
	7	O usuário pede ao sistema fazer a consulta.
	8	O sistema mostra para o usuário a consulta.
Postcondição	O arquivo da consulta é gerado e armazenado.	
Exceções	**Passo**	**Ação**
	4	É opcional que o usuário introduza uma descrição da sua consulta.
	5	Caso o usuário escolher um tipo de acceso privado para sua propriedade, ele ainda pode dar acesso a ela a algum usuário cadastrado específico.

Identificação e definição dos requisitos de interação

FR – 01	Recuperação de propriedades	
Corpo	**Descrição**	**Atores**
	O conceito RA – 04.Usuário deve ser exatamente _____	AT – 02: Usuário cadastrado

FR – 02	Recuperação de talhões	
Corpo	**Descrição**	**Atores**
	O conceito RA – 06.Propriedade deve ser exatamente _____	AT – 02: Usuário cadastrado

FR – 03	Recuperação de uma curva NDVI	
Corpo	**Descrição**	**Atores**
	O conceito RA – 07.Usuário deve ser exatamente _____	AT – 02: Usuário cadastrado
	O conceito RA – 07.Propriedade deve ser exatamente _____	AT – 02: Usuário cadastrado
	O conceito RA – 07.Talhão deve ser exatamente _____	AT – 02: Usuário cadastrado
	O conceito RA – 07.Data inicial deve conter a seguinte cadeia _____	AT – 02: Usuário cadastrado
	O conceito RA – 07.Data final deve conter a seguinte cadeia _____	AT – 02: Usuário cadastrado

PV – 01	Recuperação de propriedades
Atores	AT – 02: Usuário cadastrado
Descrição	O sistema deverá permitir a visualização dos dados que se mostram a continuação e a navegação especificada e que representa a informação que se mostra ao usuário sobre suas propriedades cadastradas.
Frases	FR – 01
Funcionalidade Asociada	RF – 03, RF – 02
Informação visualizada	RA – 04.Nome
Protótipos de saída	PV – 02 (múltiple)
Protótipos de entrada	-

PV – 02	Recuperação de talhões
Atores	AT – 02: Usuário cadastrado.
Descrição	O sistema deverá permitir a visualização dos dados que se mostram a continuação e a navegação especificada e que representa a informação que se mostra ao usuário sobre suas propriedades cadastradas.
Frases	FR – 02
Funcionalidade	RF – 03, RF – 02

Asociada	
Informação visualizada	RA – 06.Nome
Protótipos de saída	PV – 03
Protótipos de entrada	PV – 01

PV – 03	Recuperação de uma curva NDVI
Atores	AT – 02: Usuário cadastrado.
Descrição	O sistema deverá permitir a visualização dos dados que se mostram a continuação e a navegação especificada e que representa a informação que se mostra a consulta realizada pelo usuário.
Frases	FR – 03
Funcionalidade Asociada	RF – 03, RF – 02, RF – 01
Informação visualizada	RA – 03.Arquivo
Protótipos de saída	-
Protótipos de entrada	PV – 02

Validação dos requisitos

	OBJ – 1	OBJ – 2	OBJ – 3
RA – 01	*	*	*
RA – 02	*	*	
RA – 03	*	*	
RA – 04	*	*	
RA – 05	*	*	
RA – 06	*	*	
RA – 07	*	*	*
AT – 01			*
AT – 02			*
AT – 03			*
AT – 04			*
RF – 01	*	*	
RF – 02	*	*	
RF – 03	*	*	
FR – 01	*	*	

FR – 02	*	*	
FR – 03	*	*	
PV – 01	*	*	
PV – 02	*	*	
PV – 03	*	*	

Matriz de rastreabilidade.

Apêndice B – Documento de Análise do Sistema (DAS) – NDT

Nesta seção se apresenta o Documento de Análise do Sistema (DAS). Todos os padrões do NDT gerados na fase de Análise no processo do desenvolvimento do sistema WebMaps são apresentados. Recomenda-se seguir estes padrões com a leitura do Capítulo 6 deste trabalho.

Realização do modelo conceitual

CL – 01	Usuário		
Requisito	RA – 01.Usuário		
Descrição	O sistema deverá armazenar as informações correspondentes ao usuário		
Atributos	**Descrição**	**Significado**	**Dato Específico**
	Código usuário: CLn-01	Armazena informação sobre o código que identifica de maneira únivoca a cada usuário	RA – 01.Código usuário
	Nome: Texto	Armazena o nome do usuário.	RA – 01.Nome
	Login: Texto	Armazena o login do usuário.	RA – 01.Login
	Password: Texto	Armazena o password do usuário.	RA – 01.Password
	E – mail: Texto	Armazena o e-mail do usuário.	RA – 01.E-mail
	Endereço: Texto	Armazena o endereço do usuário.	RA – 01.Endereço
	Bairro: Texto	Armazena o bairro do endereço do usuário.	RA – 01.Bairro
	Complemento: Texto	Armazena o complemento do endereço do usuário.	RA – 01.Complemento
	Tipo de usuário: Texto	Estabelece a informação sobre o tipo de usuário.	RA – 01.Tipo de usuário
	Cidade: Texto	Armazena a cidade do usuário.	RA – 01. Cidade
	Estado: Texto	Armazena o estado do usuário.	RA – 01. Estado
Estado	Básico		

CL – 02	Imagem de satélite		
Requisito	RA – 02.Imagem de satélite.		
Descrição	O sistema deverá armazenar as imagens de satélite.		
Atributos	**Descrição**	**Significado**	**Dato Específico**
	Código imagem de satélite: CLn-02	Armazena informação sobre o código que identifica de maneira únivoca a cada imagem	RA – 02.Código Imagem de satélite

	Data: Data	Armazena a informação da data da imagem.	RA – 02.Data
	Descrição: Texto	Recolhe uma descrição da imagem de satélite.	RA – 02.Descrição
Estado	Básico		

CL – 03	Máscara		
Requisito	RA – 03.Máscara		
Descrição	O sistema deverá armazenar as máscaras das propriedades e/ou dos talhões para agilizar as consultas.		
Atributos	**Descrição**	**Significado**	**Dato Específico**
	Código máscara: CLn-03	Armazena informação sobre o código que identifica de maneira únivoca a cada máscara.	RA – 03.Código máscara
	Arquivo: Arquivo	Armazena o arquivo que contem a máscara.	RA – 03.Arquivo
	Data: Data	Armazena a data em que a máscara foi gerada.	RA – 03.Data
	Formato: Texto	Armazena o formato da máscara.	RA – 03.Formato
	Tipo: Texto	Armazena o tipo de arquivo utilizado para armazenar a máscara.	RA – 03.Tipo
Estado	Básico		

CL – 04	Propriedade		
Requisito	RA – 04.Propriedade		
Descrição	O sistema deverá armazenar as informações correspondentes a uma propriedade.		
Atributos	**Descrição**	**Significado**	**Dato Específico**
	Código propriedade: CLn-04	Armazena informação sobre o código que identifica de maneira únivoca a propriedade.	RA – 04.Código máscara
	Nome: Texto	Armazena o nome da propriedade.	RA – 04.Nome
	Descrição: Texto	Armazena a descrição da propriedade.	RA – 04.Descrição
	Coordenadas: Geo-referenciado	Armazena as coordenadas geográficas da propriedade.	RA – 04.Georeferenciado
	Tipo: Texto	Armazena o tipo de propriedade (público ou privada).	RA – 04.Tipo
Estado	Básico		

CL – 05	Imagem MBB
Requisito	RA – 05.Imagem MBB
Descrição	O sistema deverá armazenar as informações correspondentes um recorte MBB

	(Minimum Bounding Box).		
Atributos	**Descrição**	**Significado**	**Dato Específico**
	Código imagem MBB: CLn-04	Armazena informação sobre o código que identifica de maneira únivoca a um recorte MBB.	RA – 05.Código imagem MBB
	Arquivo: Imagem	Armazena o arquivo que tem a informação da imagem MBB.	RA – 05.Arquivo
	Data: Data	Armazena a data de criação do recorte.	RA – 05.Data
Estado	Básico		

CL – 06	Talhão		
Requisito	RA – 06.Talhão		
Descrição	O sistema deverá armazenar as informações correspondentes a um talhão.		
Atributos	**Descrição**	**Significado**	**Dato Específico**
	Código talhão: CLn-06	Armazena informação sobre o código que identifica de maneira únivoca o talhão.	RA – 06.Código talhão
	Nome: Texto	Armazena o nome do talhão.	RA – 06.Talhão
	Data inicial: Data	Armanzena a data inicial da criação do talhão	RA – 06.Data inicial
	Data final: Data	Armazena a data final da existência do talhão.	RA – 06.Data final
	Coordenadas: Geo-referenciado	Armazena as coordenadas geográficas do talhão.	RA – 06.Georeferenciado
	Cultura: Texto	Armazena o tipo de cultura do talhão.	RA – 06.Cultura
	Arquivo: Arquivo	Armazena a referência do arquivo que contem as coordenadas (opcional)	RA – 06.Arquivo
Estado	Básico		

CL – 07	Consulta		
Requisito	RA – 07.Consulta		
Descrição	O sistema deverá armazenar as informações correspondentes a uma consulta.		
Atributos	**Descrição**	**Significado**	**Dato Específico**
	Código consulta: CLn-07	Armazena informação sobre o código que identifica de maneira únivoca uma consulta.	RA – 07.Consulta
	Arquivo: Imagem	Armazena o arquivo que contem a imagem da consulta.	RA – 07.Arquivo
	Data inicial: Data	Armazena a especificação da data inicial da consulta.	RA – 07.Data inicial

	Data final: Data	Armazena a especificação da data final da consulta.	RA – 07.Data final
	Descrição: Texto	Recolhe uma descrição da consulta	RA – 07.Descrição
	Tipo: Texto	Armazena o tipo de consulta.	RA – 07.Tipo
	Status: Booleano	Armazena o estado de disponibilidade da consulta.	RA – 07.Status
Estado	Básico		

AS – 01			
Requisitos	RA – 01: Usuário RA – 04: Propriedade		
Descrição	As classes CL – 01 e CL – 04 se relacionam mediante esta associação que representa uma referencia à propriedade que é registrada pelo usuário.		
Tipo	Bidirecional		
Classes	**Nome da Classe**	**Nome do Rol**	**Cardinalidade**
	CL – 01	Usuário	0..n
	CL – 04	Propriedade	1..n
Dados Específicos	RA – 01.Propriedade RA – 04.Usuário		
Estado	Básico		

AS – 02			
Requisitos	RA – 01: Usuário RA – 07: Consulta		
Descrição	As classes CL – 01 e CL – 04 se relacionam mediante esta associação que representa uma referencia à propriedade que é registrada pelo usuário.		
Tipo	Bidirecional		
Classes	**Nome da Classe**	**Nome do Rol**	**Cardinalidade**
	CL – 01	Usuário	0..n
	CL – 07	Consulta	1..1
Dados Específicos	RA – 01.Consulta RA – 07.Usuário		
Estado	Básico		

AS – 03			
Requisitos	RA – 02: Imagem de satélite RA – 05: Imagem MBB		
Descrição	As classes CL – 02 e CL – 05 se relacionam mediante esta associação que representa uma referencia à imagem MBB associado a uma imagem de satélite.		
Tipo	Bidirecional		
Classes	**Nome da Classe**	**Nome do Rol**	**Cardinalidade**
	CL – 02	Imagem de satélite	0..n

	CL – 05	MBB	1..1
Dados Específicos	RA – 02.MBB RA – 05.Imagem de satélite		
Estado	Básico		

AS – 04			
Requisitos	RA – 03: Máscara RA – 05: Imagem MBB		
Descrição	As classes CL – 03 e CL – 05 se relacionam mediante esta associação que representa uma referencia à imagem MBB associada a sua máscara.		
Tipo	Bidirecional		
Classes	**Nome da Classe**	**Nome do Rol**	**Cardinalidade**
	CL – 03	Máscara	1..n
	CL – 05	MBB	1..1
Dados Específicos	RA – 03.MBB RA – 05.Máscara		
Estado	Básico		

AS – 05			
Requisitos	RA – 03: Máscara RA – 04: Propriedade		
Descrição	As classes CL – 03 e CL – 04 se relacionam mediante esta associação que representa uma referencia à máscara da propriedade.		
Tipo	Bidirecional		
Classes	**Nome da Classe**	**Nome do Rol**	**Cardinalidade**
	CL – 03	Máscara	1..1
	CL – 04	Propriedade	1..1
Dados Específicos	RA – 03.Propriedade RA – 04.Máscara		
Estado	Básico		

AS – 06			
Requisitos	RA – 03: Máscara RA – 06: Talhão		
Descrição	As classes CL – 03 e CL – 06 se relacionam mediante esta associação que representa uma referencia à máscara do talhão.		
Tipo	Bidirecional		
Classes	**Nome da Classe**	**Nome do Rol**	**Cardinalidade**
	CL – 03	Máscara	1..1
	CL – 06	Talhão	1..1
Dados Específicos	RA – 03.Talhão RA – 06.Máscara		
Estado	Básico		

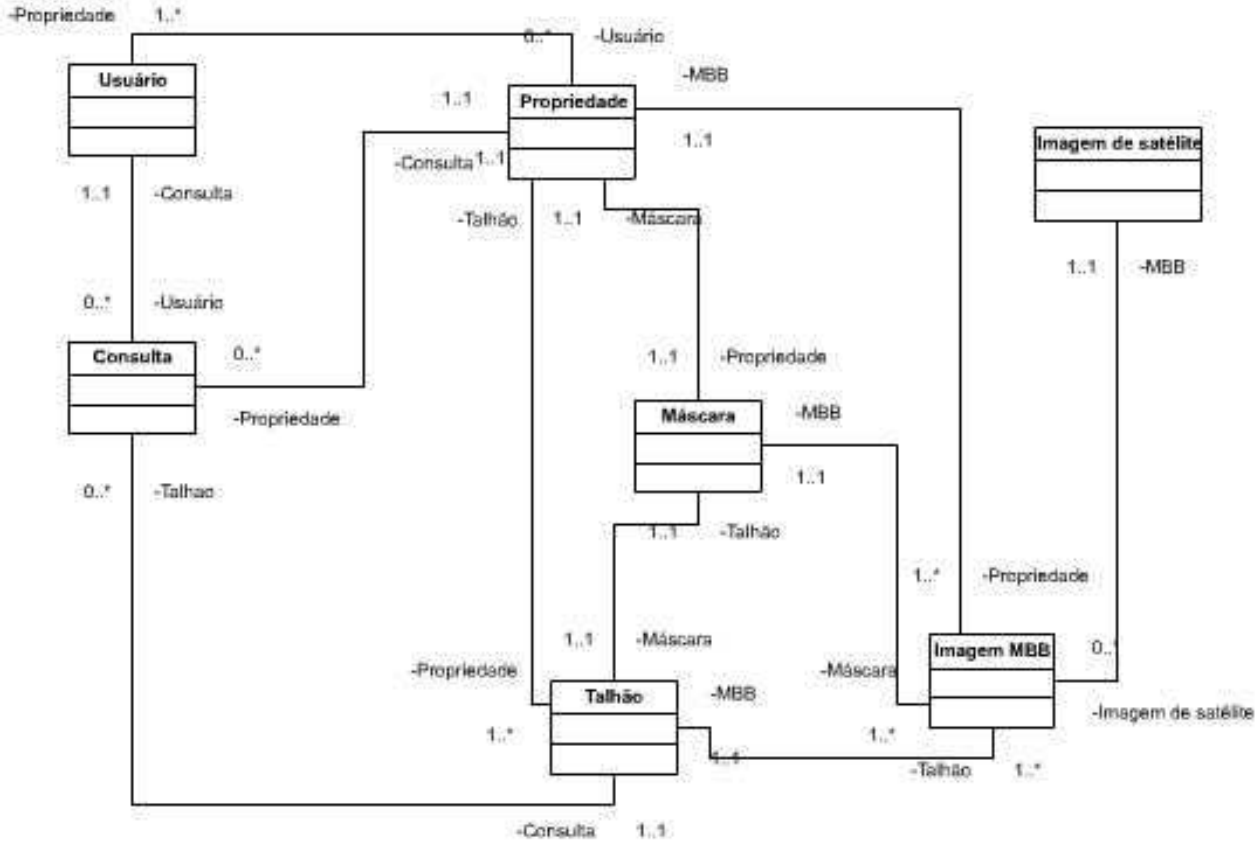
AS – 07			
Requisitos	RA – 04: Propriedade RA – 05: Imagem MBB		
Descrição	As classes CL – 04 e CL – 05 se relacionam mediante esta associação que representa uma referencia às imagens MBB da propriedade.		
Tipo	Bidirecional		
Classes	**Nome da Classe**	**Nome do Rol**	**Cardinalidade**
	CL – 04	Propriedade	1..n
	CL – 05	MBB	1..1
Dados Específicos	RA – 04.MBB RA – 05.Propriedade		
Estado	Básico		

AS – 08			
Requisitos	RA – 06: Talhão RA – 05: Imagem MBB		
Descrição	As classes CL – 05 e CL – 06 se relacionam mediante esta associação que representa uma referencia às imagens MBB do talhão.		
Tipo	Bidirecional		
Classes	**Nome da Classe**	**Nome do Rol**	**Cardinalidade**
	CL – 06	Talhão	1..n
	CL – 05	MBB	1..1
Dados Específicos	RA – 06.MBB RA – 05.Talhão		
Estado	Básico		

AS – 09			
Requisitos	RA – 04: Propriedade RA – 06: Talhão		
Descrição	As classes CL – 04 e CL – 06 se relacionam mediante esta associação que representa uma referencia à propriedade do talhão.		
Tipo	Bidirecional		
Classes	**Nome da Classe**	**Nome do Rol**	**Cardinalidade**
	CL – 04	Propriedade	1..n
	CL – 06	Talhão	1..1
Dados Específicos	RA – 04.Talhão RA – 06.Propriedade		
Estado	Básico		

AS – 10			
Requisitos	RA – 04: Propriedade RA – 07: Consulta		
Descrição	As classes CL – 04 e CL – 07 se relacionam mediante esta associação que representa uma referencia à propriedade referida em uma consulta.		
Tipo	Bidirecional		
Classes	**Nome da Classe**	**Nome do Rol**	**Cardinalidade**
	CL – 04	Propriedade	0..n
	CL – 07	Consulta	1..1
Dados Específicos	RA – 04.Consulta RA – 07.Propriedade		
Estado	Básico		

AS – 11			
Requisitos	RA – 06: Talhão RA – 07: Consulta		
Descrição	As classes CL – 06 e CL – 07 se relacionam mediante esta associação que representa uma referencia ao talhão referido em uma consulta.		
Tipo	Bidirecional		
Classes	**Nome da Classe**	**Nome do Rol**	**Cardinalidade**
	CL – 06	Talhão	0..n
	CL – 07	Consulta	1..1
Dados Específicos	RA – 06.Consulta RA – 07.Talhão		
Estado	Básico		



Modelo conceitual.

Realização do modelo navegacional

ATORES	AT – 01	AT – 02	AT – 03	AT – 04
AE – 01	^			
AE – 02		^		
AE – 03			^	
AE – 04				^

NO – 01	Recuperação de propriedades	
Protótipo	PV – 01	
Atores em estudo	AE – 02	
Descrição	O sistema deverá permitir a visualização dos dados que se mostram a continuação e a navegação especificada e que representa a informação que se mostra ao usuário sobre suas propriedades cadastradas.	
Atributos	**Descrição**	**Protótipo**
	RA – 04.Nome	PV – 01

Estado	Básico

NO – 02	Recuperação de talhões	
Protótipo	PV – 02	
Atores em estudo	AE – 02	
Descrição	O sistema deverá permitir a visualização dos dados que se mostram a continuação e a navegação especificada e que representa a informação que se mostra ao usuário sobre suas propriedades cadastradas.	
Atributos	**Descrição**	**Protótipo**
	RA – 06.Nome	PV – 02
Estado	Básico	

NO – 03	Recuperação de uma curva NDVI	
Protótipo	PV – 03	
Atores em estudo	AE – 02	
Descrição	O sistema deverá permitir a visualização dos dados que se mostram a continuação e a navegação especificada e que representa a informação que se mostra a consulta realizada pelo usuário.	
Atributos	**Descrição**	**Protótipo**
	RA – 03.Arquivo	PV – 03
Estado	Básico	

EL – 01	
Origen	NO – 02: Recuperação de talhões
Destino	NO – 03: Recuperação de uma curva NDVI
Atores em estudo	AE – 02.
Descrição	O sistema deve permitir a navegação entre os elementos NO – 02 e NO – 03
Tipo	Bidirecional
Estado	Básico

IN – 01	
Destino	NO – 02: Recuperação de talhões
Atores em estudo	AE – 02.
Descrição	O sistema deve permitir ao usuário selecionar a opção para chegar ao NO – 02
Tipo	Indice

Estado	Básico

QU - 01		
Destino	PV – 01	
Atores em estudo	AE – 02	
Descrição	O sistema deverá permitir obter informação desde o usuário apartir da frase que se mostra a seguir.	
Atributos	**Descrição**	**Atores**
	O conceito RA – 04.Usuário deve ser exatamente _____	AE – 02
Estado	Básico	

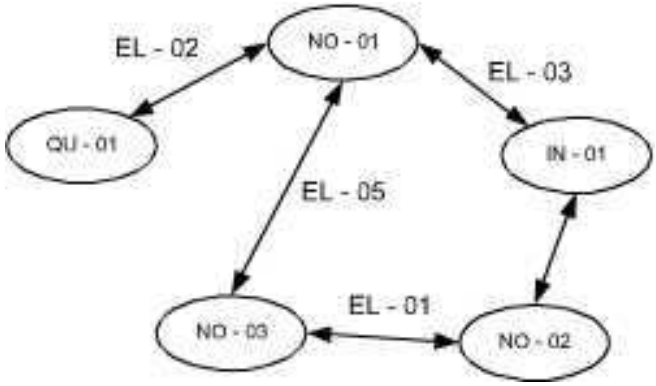
EL – 02	
Origen	QU – 01
Destino	NO – 01: Recuperação de propriedades
Atores em estudo	AE – 02.
Descrição	O sistema deve permitir a navegação entre os elementos QU – 01 e NO – 01
Tipo	Bidirecional
Estado	Básico

EL – 03	
Origen	NO – 01: Recuperação de propriedades
Destino	IN – 01
Atores em estudo	AE – 02
Descrição	O sistema deve permitir a navegação entre os elementos NO – 01 e IN – 01
Tipo	Bidirecional
Estado	Básico

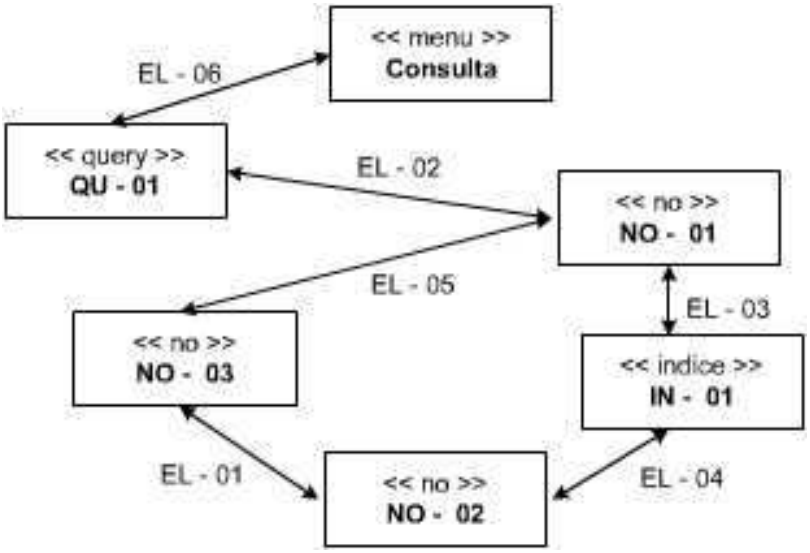
EL – 04	
Origen	IN – 01
Destino	NO – 02: Recuperação de talhão
Atores em estudo	AE – 02.
Descrição	O sistema deve permitir a navegação entre os elementos IN – 01 e NO – 02
Tipo	Bidirecional

Estado	Básico

EL – 05	
Origen	NO – 01: Recuperação de propriedade
Destino	NO – 03: Recuperação de imagem NDVI
Atores em estudo	AE – 02.
Descrição	O sistema deve permitir a navegação entre os elementos NO – 01 e NO – 03
Tipo	Bidirecional
Estado	Básico



Modelo navegacional básico.



Modelo navegacional final.