Portal de Conhecimento sobre Critérios,
Estratégias e Ferramentas de Teste de Aplicações
Web

Marcella Letícia de Souza Costa

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 28 de janeiro de 2008

Assinatura: _____

Portal de Conhecimento sobre Critérios, Estratégias de Ferramentas de Teste de Aplicações *Web*

Marcella Letícia de Souza Costa

Orientador: *Prof. Dr. José Carlos Maldonado*

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação — ICMC/USP, como parte dos requisitos para obtenção do título de Mestre em Ciências de Computação e Matemática Computacional.

USP - São Carlos
Janeiro/2008

*“E você aprende que realmente pode suportar...
que realmente é forte, e que pode ir muito mais longe
depois de pensar que não se pode mais.”
(William Shakespeare)*

Agradecimentos

Agradeço aos meus pais, que, mesmo longe, estão sempre perto, torcendo por mim. E vibrando por cada uma de minhas conquistas, sejam elas grandes ou pequenas. Amo vocês, com todo meu coração.

Aos meus irmãos: Renata, pela força presente e incentivo durante esses anos que estou longe de casa. Lorena, pelo exemplo de dedicação e esforço, em que eu me inspiro pra continuar seguindo em frente. Victor, pela coragem, por enfrentar a vida sempre com tranquilidade, me mostrando exemplos de como encarar as adversidades.

À toda minha família, pelo carinho, apoio, incentivo, momentos de descontração, e pela alegria que me dão e o prazer de compartilhar essa vitória com eles. Em especial à minha tia e madrinha, Dalva, por oferecer abrigo, comida, cuidados, sempre que eu precisei, e também a minha “família adotiva” em Rio Claro, que me ajudaram nessa caminhada.

Ao meu orientador, Prof. José Carlos Maldonado, pela orientação na condução deste trabalho, por oferecer um pouco de sua experiência e conhecimento, além de paciência e incentivo, para que eu pudesse alcançar o objetivo proposto.

Ao prof. Henrique Rozenfeld, e à equipe do Núcleo de Manufatura Avançada, pela oportunidade oferecida, de adquirir novos conhecimentos e compartilhar experiências, auxiliando, inclusive, nas idéias para o desenvolvimento deste trabalho.

Aos meus amigos, os “velhos” amigos, de tantas aventuras, de quem sinto tanta falta, mas que continuam comigo, pela amizade fortalecida, que persiste apesar da distância e do tempo que passa. Em especial à Jñane, ao Alexander e à Lidiane, que estão sempre me apoiando, e dizendo que eu vou conseguir, mesmo quando nem eu acredito nisso.

Aos meus amigos, os “novos” amigos, que eu conheci em São Carlos, e que estiveram e estão comigo, me apoiando e me dando coragem pra seguir em frente. Agradeço à Alê, pela calma, paciência e especialmente, prestatividade, me ajudando sempre que eu precisei. Sem ela, estar longe de casa seria muito mais difícil. Agradeço à Camila, um exemplo de esforço, dedicação e honestidade, em quem eu posso confiar e pedir apoio quando precisar.

Aos funcionários e professores do ICMC/USP, que ofereceram suporte para a realização do trabalho.

À FAPESP, pelo apoio financeiro.

Resumo

O advento da tecnologia da Internet, juntamente com a *World Wide Web*, popularizaram e permitiram uma grande expansão e demanda por aplicações *web*. Por executarem em um ambiente heterogêneo e complexo, as aplicações *web* apresentam várias características que as diferenciam dos sistemas tradicionais. As particularidades dessas aplicações tornam o fator qualidade essencial para o sucesso dessas aplicações. Alguns requisitos de qualidade, como usabilidade, confiabilidade, interoperabilidade e segurança devem, então, ser validados. Para assegurar a qualidade desejada, são necessárias executar as atividades de Verificação, Validação e Teste (VV&T). Dentre elas, as mais utilizadas são as atividades de teste. Os critérios, estratégias e ferramentas de teste precisam ser identificados e avaliados para se estabelecer uma relação entre os custos e benefícios entre elas, a fim de guiar a escolha de cada uma durante os testes de aplicações *web*. A realização de estudos experimentais para realizar essa análise beneficia tanto a academia, nas atividades de ensino e pesquisa, como a indústria, nas atividades de seleção e aplicação de critérios, estratégias e ferramentas de teste de aplicações *web*. Para que os resultados obtidos estejam acessíveis, é proposto o desenvolvimento de um portal de conhecimento que disponibilize de maneira sistemática o conhecimento obtido sobre critérios, estratégias e ferramentas de teste de aplicações *web*. Esse portal tem como objetivo oferecer às organizações um ambiente compartilhado de conhecimento, a fim de proporcionar a existência de um ciclo de criação, troca, retenção e reuso do conhecimento.

Abstract

The development of internet technology, along with the World Wide Web, made popular and allowed a demand boom and expansion of web applications. Web applications present some characteristics that differentiate them from the traditional systems, since they execute in a heterogeneous and complex environment. The particularities of these applications demand quality as an essential factor for success. In this way, quality requirements, like usability, reliability, interoperability and security must be validated. To assure the desired quality, it's necessary to execute the activities of Verification, Validation and Test (VV&T), where the most used are the testing activities. The testing criteria, strategies and tools need to be identified and evaluated to establish a relation between the costs and benefits, in order to guide the choice of each one during the tests of web applications. The carrying out of experimental studies benefits the academy, in the activities of education and research, and the industry, in the activities of selection and application of criteria, strategies and tools of testing web applications. Moreover, these results need to be accessible to make available information to the interest people. This work presents the development of a knowledge portal, to facilitate knowledge sharing in a systematic way, with information of test criteria, strategies and tools for validating web applications. This portal intends to offer to the organizations and academic institutions a shared environment of web application test knowledge, in order to constitute a knowledge creation, change, retention and reuse cycle.

Sumário

Abstract	i
1 Introdução	1
1.1 Contexto	1
1.2 Motivação e Relevância	4
1.3 Objetivos	5
1.4 Organização do Trabalho	6
2 Revisão Bibliográfica	7
2.1 Considerações Iniciais	7
2.2 Portais de Conhecimento	7
2.2.1 Definição de Portais	9
2.2.2 Classificação de Portais	10
2.2.3 Características e Funcionalidades de Portais	13
2.2.4 Desenvolvimento de Portais	16
2.3 Teste de Aplicações Web	22
2.3.1 Técnicas e Critérios de Teste	24
2.3.1.1 Técnica Funcional	24
2.3.1.2 Técnica Estrutural	25
2.3.1.3 Técnica Baseada em Erros	29
2.3.2 Estratégias de Teste	30
2.3.3 Ferramentas de Teste	34
2.4 Ontologias	35
2.4.1 Características e Uso de Ontologias	36
2.4.2 Ontologia de Teste - OntoTest	38
2.5 Considerações Finais	45
3 Portal de Conhecimento: Definição de Requisitos, Modelagem e Arquitetura	47
3.1 Considerações Iniciais	47
3.2 Definição do Ambiente	48
3.3 Processo de Desenvolvimento	49
3.3.1 Revisão sistemática - Processo de Desenvolvimento de Portais Web	50
3.3.2 Estudo dirigido - Processo de Desenvolvimento de Aplicações Web	54

3.3.3	Comparação entre os processos	58
3.3.4	Processo WAE	60
3.4	Requisitos	62
3.5	Modelagem	67
3.6	Arquitetura	73
3.7	Considerações Finais	74
4	Aspectos de Implementação e Operacionais	75
4.1	Considerações Iniciais	75
4.2	Funcionalidades	75
4.3	Implementação	77
4.4	Codificação	83
4.5	Cenários de Uso	88
4.6	Considerações Finais	92
5	Conclusão	93
5.1	Considerações Gerais	93
5.2	Contribuições	94
5.3	Trabalhos Futuros	95
	Referências	106

Lista de Figuras

2.1	Arquitetura de um Portal Corporativo	17
2.2	Componentes de um Portal	18
2.3	Meta-modelo de uma página web	27
2.4	Modelo de Testes de uma Aplicação Web	33
2.5	OntoTest (Estrutura Principal)	39
2.6	Sub-Ontologia de Processos de Teste	41
2.7	Sub-Ontologia de Procedimentos de Teste	41
2.8	Sub-Ontologia de Passos de Teste	42
2.9	Sub-Ontologia de Artefatos de Teste	43
2.10	Sub-Ontologia de Recursos de Teste	44
3.1	Cenários de Uso - Administrador e Visitante	67
3.2	Cenários de Uso - Especialista e Usuário Cadastrado	68
3.3	Diagrama de Atividades - Administrador	69
3.4	Diagrama de Atividades - Visitante	69
3.5	Atividades - Especialista I	70
3.6	Atividades - Especialista II	70
3.7	Atividades - Especialista III	70
3.8	Atividades - Especialista IV	70
3.9	Diagrama de Atividades - Usuário Cadastrado	71
3.10	Diagrama UX	72
3.11	Diagrama de Classes	72
3.12	Arquitetura do Portal de Conhecimento	73
4.1	Modelo ER (Estrutura Principal)	78
4.2	Modelo ER Processos de Teste	78
4.3	Modelo ER Procedimentos de Teste	79
4.4	Modelo ER Passos de Teste	79
4.5	Modelo ER Artefatos de Teste	80
4.6	Modelo ER Recursos de Teste	81
4.7	Arquitetura do ClassGenerator	82
4.8	Classe Conexão	84
4.9	Classe BD	85
4.10	Classe Básica	85

4.11 Classe Fachada	85
4.12 Código de Interface de Cadastro de Processo de Teste	86
4.13 Código de Interface de Consulta	87
4.14 Cadastro de Processo	89
4.15 Seleção de Processo	89
4.16 Cadastro de Passo	90
4.17 Módulo de Administração do EzPublish	90
4.18 Página Inicial do EzPublish	91
4.19 Projeto de Fórum	91

Lista de Tabelas

2.1	Gerações dos portais públicos (Eckerson, 1999b)	10
2.2	Gerações dos portais corporativos (Eckerson, 1999b)	11
2.3	Requisitos mínimos de um portal corporativo	15
2.4	Modelo de Portal Geral (Smith, 2004)	21
2.5	Mapeamento Características X Modelos	22
3.1	Resultados da Revisão Sistemática	53
3.2	Resultados da Revisão Sistemática	54
3.3	Resultados do Estudo Dirigido	58
3.4	Fases Processos de Desenvolvimento - Portais	61
3.5	Fases Processos de Desenvolvimento - Aplicações web	61

Introdução

1.1 Contexto

Segundo Toledo (2002), um aspecto de grande importância para as organizações é fazer melhor uso da informação existente, com a finalidade de desenvolver estratégias com foco em inovação, criação de conhecimento e sua conversão em produtos e serviços. O conhecimento é, então, um recurso estratégico primário para as empresas, com pesquisadores e gestores esforçando-se para descobrir maneiras de como acumular esse recurso de forma eficaz e geri-lo para produzir novos conhecimentos.

Para que as pessoas possam fazer bom uso do conhecimento e realizar a troca de experiências obtidas em uma determinada área, é necessário um conjunto de práticas que permitam adquirir, organizar e comunicar esse conhecimento, ou seja, é importante realizar a gerência eficaz dos dados encontrados em diversas fontes, como bases de dados, artigos, documentos, páginas web, entre outras (Alavi e Leidner, 1999) *apud* Benbya et al. (2004).

Uma tendência crescente como solução para a gerência do conhecimento das organizações e para suprir as necessidades de acesso às informações é a utilização de portais *web* (Devedzic, 2005). Portais podem ser definidos como pontos de acesso que realizam a integração entre pessoas, aplicações e serviços para o gerenciamento do conhecimento existente acerca de determinado assunto de interesse; eles permitem que as informações sejam disseminadas de maneira integrada e, além disso, permitem que as pessoas possam compartilhar o conhecimento existente em torno dessa área de interesse (Dias, 2001).

Os portais podem ser classificados de acordo com o contexto de sua utilização em públicos ou corporativos. Os portais públicos têm como objetivo atrair potenciais consumidores aos produtos e serviços que eles oferecem. Já os portais corporativos incluem funções que possibilitam a identificação, captura, armazenamento, recuperação e distribuição de grandes quantidades de informação de múltiplas fontes, internas e externas, para os indivíduos e equipes de uma instituição (Dias, 2001).

Dentre os portais corporativos, existem os denominados *Portais de Conhecimento*. Esses portais organizam o acesso a grandes acervos de conteúdo e permitem a cooperação entre pessoas. Além disso, permitem o acesso a pessoas especializadas em determinadas áreas, oferecendo meios de comunicação e troca de experiências entre os indivíduos (Murray, 1999) *apud* (Dias, 2001).

Um dos tipos de Portais de Conhecimento são os denominados *Portais de Comunidades*, considerados como uma plataforma na *Internet* que oferece serviços de comunicação e informações de interesse para uma comunidade em particular e, possivelmente, para um público mais geral; possuem as mesmas características dos portais de conhecimento, mas não estão relacionados estritamente a uma organização específica ou instituição, e por isso, não existem tantas restrições quanto ao acesso à informação (Devedzic, 2005).

Os *Portais de Pesquisa*, dentre os portais de comunidade, são oferecidos por grupos de pesquisa, universidades, institutos de educação, entre outros, e possuem as informações mais recentes em importantes tópicos de pesquisa e políticas públicas. Eles podem assegurar canais de comunicação rápidos e eficientes pelo apoio e sustentação dos relacionamentos pessoais e contatos, e para construir uma base de dados crescente de informações aos participantes da comunidade. Dessa forma, esses portais podem servir como ponto de acesso às informações para o diálogo entre a comunidade científica, o governo e o público em geral (Devedzic, 2005).

Nesse sentido, a escolha da área de conhecimento que será estruturada e armazenada dentro do portal deve estar de acordo com o interesse de grupos de pesquisa. Com o advento da tecnologia da Internet, juntamente com a *World Wide Web*, houve uma mudança significativa em diversos segmentos, como os setores de negócios, comércio, indústrias, bancos e outros, inclusive em nossas vidas pessoais e profissionais. Essa mudança ocorre pela natureza da *web*, que oferece acesso com disponibilidade permanente e uniforme, independentemente da localização física de dados, informações e conhecimento (Kappel et al., 2004).

As aplicações *web* - sistemas executados no ambiente da *World Wide Web* (*WWW*)-aumentam, assim, em importância e popularidade. Com isso, pode-se observar que elas vêm evoluindo para satisfazer as necessidades dos usuários, tornando-se cada vez mais complexas (Ginige e Murugesan, 2001a; Reis, 2004).

As aplicações *web* mais simples possuem páginas estáticas, cujo conteúdo não pode ser modificado e, portanto, com uma navegação mais simplificada entre as páginas. Além disso, elas são sistemas *stand-alone*, cujo desempenho não é considerado requisito essencial. Nas aplicações mais complexas, existe a possibilidade do usuário interagir com o conteúdo apresentado nas páginas. Essas aplicações são mais complexas também por serem integradas a outros sistemas e bases de dados, exigindo, em geral, alto desempenho (Ginige e Murugesan, 2001a).

Independentemente do fato das aplicações *web* serem simples ou complexas, elas possuem as características de estarem em ambientes heterogêneos, executando sobre diversas plataformas e sistemas operacionais e com diversas versões e tipos de navegadores, o que as tornam diferentes de aplicações tradicionais que executam em computadores desktop. Além disso, essas aplicações são fortemente dirigidas a conteúdo, sendo importante a forma com que esse conteúdo está disponibilizado, aumentando a necessidade de introdução de hipermídias (áudio, vídeo, etc.), e a necessidade de elas estarem permanentemente disponíveis (Xu et al., 2005).

Toda essa variedade no escopo das aplicações *web*, bem como suas particularidades, tornam o fator qualidade essencial para o sucesso dessas aplicações. Assim, requisitos de qualidade como usabilidade, confiabilidade, interoperabilidade, desempenho e segurança devem ser analisados (Lucca et al., 2002).

As atividades de Verificação, Validação e Testes (VV&T), conduzidas ao longo do processo de desenvolvimento das aplicações, auxiliam para essa análise, de forma a garantir que os requisitos estão sendo atendidos. Dentro desse conjunto de atividades, o teste é uma das mais utilizadas, com o objetivo de assegurar que o software execute as funções para o qual foi planejado e que não faça nada além daquilo que tenha sido exigido (Myers, 2004).

Os testes devem ser conduzidos de maneira sistemática e bem planejada, com a utilização de técnicas, critérios e estratégias que avaliem os aspectos funcionais e operacionais da aplicação de maneira rigorosa (Myers, 2004; Vincenzi, 2004).

Entretanto, os testes tradicionais, em geral, não são adequados frente às diferenças existentes nas aplicações *web*. Na literatura, existem proposições de técnicas e estratégias de teste, que são empregadas levando em consideração as peculiaridades das aplicações *web* (Liu et al., 2000c,d; Lucca et al., 2002; Ricca e Tonella, 2002; Xu et al., 2005).

Outro ponto importante para os testes de aplicações *web* é a automatização das atividades. A existência de ferramentas que realizam essa função reduz as probabilidades de erros e aumentam a eficiência e abrangência dos testes (Maldonado et al., 2004). Para as aplicações *web*, alguns esforços no desenvolvimento de ferramentas de testes podem ser identificados, sendo que, no mercado, existem produtos disponíveis para satisfazer a necessidade de automatizar os testes, desenvolvidas por várias empresas e/ou instituições

(Compuware, 2006; Mercury Interactive, 2006; MicroQuill, 2006; ParaSoftbreak, 2006; RadView, 2006; Rational, 2006a; Watchfire, 2006).

Atualmente, um grande número de empresas desenvolvedoras de software estão em processo de transição para o contexto de aplicações *web*, e, à medida que mais empresas adotarem tal abordagem, maior será a demanda por técnicas, estratégias e ferramentas de teste para essas aplicações (Domingues, 2005).

Sendo assim, é necessário realizar a análise e comparação entre as técnicas, estratégias e ferramentas de teste existentes e realizar estudos que quantifiquem sua efetividade (Briand e Labiche, 2004). Com esse objetivo, podem ser realizados estudos experimentais, que fornecem informação objetiva e quantificável, a fim de auxiliar na escolha de uma tecnologia em particular, com a construção de uma base de conhecimento com os resultados obtidos (Doria, 2001; Höhn, 2005; Zelkowitz e Wallace, 1998).

Os estudos experimentais fazem uso de pacotes de laboratório, que especificam a maneira como o experimento foi realizado (Doria, 2001). Os pacotes de laboratório documentam o processo de experimentação e caracterizam a infra-estrutura experimental para que sejam realizadas replicações. As replicações oferecem maior credibilidade aos resultados obtidos, complementando o estudo original e modelando o objeto de estudo para um contexto experimental diferente do original (Basili et al., 1999; Doria, 2001).

A partir da identificação dos critérios, estratégias e ferramentas, e dos resultados obtidos a partir dos estudos experimentais, é constituída a base de conhecimento sobre a área de teste de software. Essa base de conhecimento, que reúne os conteúdos mais relevantes da área de teste, deve ser estruturada a partir de uma representação formal dos conceitos dessa área, constituída por um vocabulário específico usado para descrever os conceitos, e com um conjunto de suposições explícitas referentes ao significado pretendido para os termos usados no vocabulário (Barbosa et al., 2006; Gruber, 1993).

Sendo assim, é importante fazer o uso de ontologias, definidas como uma especificação formal e explícita de uma conceitualização compartilhada (Gruber, 1993). Com isso, elas oferecem as definições de conceitos e seus relacionamentos para uma área específica, facilitando o compartilhamento e reuso de conhecimento existente. Com uma ontologia estruturada dentro de um portal, são localizadas de forma mais fácil e eficaz as informações relevantes dentro do assunto de interesse, provendo uma gama maior de informações.

1.2 Motivação e Relevância

A rápida difusão das tecnologias da internet está produzindo um crescimento significativo na demanda pelo estudo das aplicações *web*. Para essas aplicações, é fundamental que haja a satisfação de requisitos como os de usabilidade, confiabilidade, interoperabilidade e segurança (Lucca et al., 2002)

Como forma de garantir que esses requisitos sejam alcançados, torna-se relevante realizar estudos de técnicas, critérios, estratégias e ferramentas envolvidos nas atividades de teste, que aumentem a confiabilidade das aplicações *web*.

O grupo ICMC-USP/SP possui três trabalhos que pretendem ampliar o horizonte de conhecimento a respeito dos testes de aplicações *web*: dois de mestrado, para o desenvolvimento de um *Ambiente de Apoio ao Uso de Padrões de Software e Requisitos de Teste no Desenvolvimento de Aplicações Web* (Chan, 2005), e para a *Avaliação de Métodos de Desenvolvimento de Aplicações Web* (Bianchini, 2005), e uma tese de doutorado para a *Definição e Análise de Recursos de Teste e Validação de uma Aplicação Web* (Domingues, 2005),

O conhecimento obtido por meio dos estudos realizados deve estar acessível a todos os interessados. Várias tecnologias que facilitam o compartilhamento do conhecimento têm emergido recentemente, mas, em geral, elas estão distribuídas espacialmente por diversas plataformas tecnológicas, o que torna difícil a localização da informação desejada e a integração entre as ferramentas para o compartilhamento do conhecimento (Gwebu, 2000).

Assim, é importante prover uma tecnologia que permita o compartilhamento de informações e ainda possibilidades de colaboração entre as pessoas de maneira integrada (Terra e Gordon, 2002).

Os portais permitem que o conhecimento seja criado, transferido e reutilizado para atender às necessidades dos usuários, pelo fato de organizar de forma simples os dados, informações e conhecimento. Além disso, permitem a interação entre pessoas que compartilham interesses comuns em uma arquitetura integrada e por meio de um único ponto de acesso.

Dessa forma, este trabalho visa a disponibilizar os estudos em relação aos testes de aplicações *web*, com o desenvolvimento de um portal de conhecimento que permita a difusão do conhecimento obtido nessa área. Os objetivos deste trabalho são descritos na próxima seção.

1.3 Objetivos

Diante do cenário apresentado nas seções anteriores, este trabalho tem como objetivo principal contribuir para a utilização do conhecimento relacionado ao teste de aplicações *web* por meio de um **Portal de Conhecimento**, com uma estrutura bem definida e um conjunto de funcionalidades que permitem a agregação, evolução e disseminação do conhecimento associado a critérios, estratégias e ferramentas de teste para aplicações *web*.

Para isso, foram estabelecidas as estruturas de armazenamento das informações, com a utilização de uma ontologia para a área de testes, e a estrutura navegacional do portal, além dos mecanismos que permitam a disponibilização do conhecimento obtido. Tal conhecimento é obtido a partir da identificação das propostas existentes na literatura, tanto do estado da arte quanto estado da prática sobre os testes de aplicações *web*.

Adicionalmente, o portal de conhecimento, com o uso contínuo, objetiva dar apoio à identificação de possíveis melhorias para os critérios, estratégias e ferramentas disponibilizados, a fim de oferecer subsídios ao trabalho de Domingues (2005), que tem como uma de suas proposições definir estratégias de teste para as aplicações *web*.

1.4 Organização do Trabalho

Nas seções anteriores, foram apresentados o contexto no qual o trabalho está inserido, a motivação para que ele seja realizado, e os seus objetivos. O restante do documento está organizado da seguinte maneira: no Capítulo 2 é mostrada a pesquisa realizada que serviu como fundamentação teórica para o desenvolvimento deste trabalho. No Capítulo 3 é apresentada a definição dos requisitos para o desenvolvimento do portal, bem como sua modelagem e arquitetura, que guiaram a construção do portal. No Capítulo 4 são mostrados os aspectos de implementação do portal, com a descrição das funcionalidades existentes no ambiente e sua implementação, além dos aspectos operacionais de uso do portal. Por fim, no Capítulo 5 é apresentada a conclusão do trabalho e são também propostos trabalhos futuros.

Revisão Bibliográfica

2.1 Considerações Iniciais

Conforme mencionado no capítulo anterior, o objetivo deste trabalho é realizar o desenvolvimento de um ambiente que possua uma base de conhecimento relacionada aos testes de aplicações *web*, que ofereça informações relevantes nessa área e possibilite a aquisição, compartilhamento e disseminação de conteúdos à comunidade acadêmica e industrial.

Neste capítulo, é apresentada uma revisão bibliográfica das principais pesquisas associadas ao trabalho, estando organizado da seguinte maneira: na Seção 2.2 são apresentados os principais conceitos e características associadas aos portais. Na Seção 2.3 é mostrada uma visão geral dos conceitos que envolvem os testes de aplicações web. Por fim, na Seção 2.4 são apresentados os conceitos e utilizações de ontologias, bem como uma ontologia para a área de teste de software.

2.2 Portais de Conhecimento

A preocupação em relação ao conhecimento não é um assunto recente. Desde o período da Grécia Antiga, grandes filósofos e pensadores têm discutido a respeito desse assunto, propondo definições e conceitos sobre o mesmo, além de meios para adquirir e transmitir o conhecimento (Benbya et al., 2004).

Embora esse interesse não seja atual, recentemente, uma parcela crescente da população trabalha exclusivamente com as diversas formas de “conhecimento”. Por isso, pode-se falar do surgimento da denominada “Sociedade do Conhecimento”, caracterizada por um ambiente de mudanças rápidas, com fatores como a globalização do mercado, o aumento da complexidade dos produtos e a turbulência dos cenários competitivos (Garavelli et al., 2002; Terra e Gordon, 2002).

O conhecimento passa então a ser considerado um recurso estratégico primário para as empresas, com pesquisadores e gestores esforçando-se para descobrir maneiras de como acumular esses recursos de forma eficaz e geri-los para produzir novos conhecimentos (Toledo, 2002).

A área da gestão de conhecimento é ampla, complexa e em constante desenvolvimento, gerando, com frequência, divergência ou confusão e uma diversidade de conceitos (Toledo, 2002). Alavi e Leidner (1999) *apud* (Benbya et al., 2004) consideram que a Gestão do Conhecimento pode ser entendida como um processo sistêmico e organizacionalmente especificado de adquirir, organizar e comunicar conhecimento a fim de que outros possam fazer uso dele para serem mais efetivos e produtivos.

Em Terra e Gordon (2002), a gestão do conhecimento é considerada como o esforço para melhorar o desempenho humano e organizacional por meio da facilitação de conexões significativas. Com a gestão adequada, pode-se garantir que todos dentro da organização tenham acesso ao conhecimento da organização, quando, onde e na forma de que eles necessitam. Além disso, o processo para esses indivíduos codificarem parte de seu conhecimento e/ou colaborarem com outros torna-se mais simples, já que a gestão pode ajudar e motivar que detentores de conhecimentos importantes os compartilhem.

A gestão do conhecimento possui alguns desafios que têm como causas: (i) grandes volumes de informação que estão sendo criados, armazenados e distribuídos; (ii) incrível velocidade com que o conteúdo de conhecimento está mudando; e (iii) transformação contínua do local de trabalho. Assim, há uma necessidade cada vez maior por sistemas que disponibilizem a fonte de conhecimento ou informação correta para o usuário, onde quer que ele esteja, de maneira virtual e instantânea. Nesse sentido, as novas tecnologias baseadas nos padrões da Internet facilitam a troca de informações entre as organizações e as possibilidades de colaboração entre as pessoas, em modos síncronos ou assíncronos, independentemente da localização física (Terra e Gordon, 2002).

Embora outras tecnologias estejam disponíveis para atender a esse propósito, o uso de uma tecnologia que integre uma variedade de aplicações e ferramentas pode ser útil para desenvolver e implementar de forma eficaz a gestão do conhecimento. Assim, os portais - ponto de acesso integrado a informações, serviços e pessoas - podem ser considerados como uma solução que abrange as funcionalidades de outros componentes, integrando características que oferecem às organizações um rico e complexo espaço compartilhado de

informação para criação, troca, retenção e reuso de conhecimento (Benbya et al., 2004; Frappaolo, 2002; Terra e Gordon, 2002).

Considerando a relevância apresentada pela tecnologia dos portais, serão apresentados a seguir a definição de portais e sua classificação, bem como algumas características básicas que devem estar presentes nos portais e considerações sobre o seu desenvolvimento.

2.2.1 Definição de Portais

Os portais são considerados uma infra-estrutura que, por meio de um único ponto de acesso ou ponto de entrada, realiza a integração entre pessoas, aplicações e serviços, permitindo a reunião, gerência, compartilhamento e utilização de informação - estruturada (documentos textuais, páginas, etc.) ou não estruturada (vídeos, imagens, sons, etc.) - disponível em uma variedade de fontes, como bases de dados de aplicações. Além disso, os portais oferecem os serviços de forma segura, personalizada e customizável, proporcionando a visualização de informações relevantes de acordo com o perfil dos usuários (Benbya et al., 2004; Raol et al., 2002; Smith, 2004; Watson e Fenner, 2000).

No início da internet, o termo portal era associado aos mecanismos de busca, com o objetivo de facilitar o acesso às informações contidas em vários documentos dispersos pela internet. Esse termo era utilizado, então, para descrever aplicações baseadas na *web* que ofereciam acesso organizado aos recursos disponíveis, por meio da utilização de recursos como pesquisas *booleanas* e navegação associativa entre *links* (Dias, 2001; Smith, 2004; Toledo, 2002). Para reduzir o tempo de busca para encontrar informações relevantes e ajudar usuários inexperientes, alguns *sites* de busca passaram a disponibilizar novas funcionalidades e utilizar conceitos de categorias, agrupando *sites* e documentos em grupos pré-definidos de acordo com seu conteúdo. Posteriormente foram implementadas funções de integração, tais como *chats* em tempo real, comunidades de interesse, listas de discussão, personalização de conteúdo definido pelo usuário e acesso direto a conteúdos especializados e comerciais (Dias, 2001; Toledo, 2002).

O significado de portal passou a ser aplicado a sistemas que diferem em termos de capacidade e complexidade (Benbya et al., 2004; Smith, 2004). A partir da variedade de serviços apresentados pelos portais, é possível realizar uma classificação, a fim de relacionar os portais aos objetivos que eles atendem. Essa classificação é apresentada na próxima seção.

2.2.2 Classificação de Portais

Segundo Dias (2001), os portais podem ser classificados de acordo com o *contexto* de sua utilização, ou seja, de acordo com o público alvo, e segundo as *funções* que eles desempenham. A seguir são detalhadas as duas classificações:

1. **Quanto ao contexto:** classificação relacionada ao contexto de utilização, atendendo a grupos de usuários diversos e com propósitos distintos para cada grupo de usuários.
 - (a) *Portais Públicos:* têm como objetivo atrair usuários por meio de uma única interface para aumentar a probabilidade do estabelecimento de comunidades virtuais que potencialmente comprarão o que os anunciantes do *site* têm a oferecer. É um tipo de portal que estabelece uma relação unidirecional com os visitantes;
 - (b) *Portais Corporativos:* evolução do uso das *intranets* com a incorporação de funções que possibilitem a identificação, captura, armazenamento, recuperação e distribuição de grandes quantidades de informação de múltiplas fontes, internas e externas, para os indivíduos e equipes de uma instituição.

Tanto os portais públicos quanto os corporativos apresentam uma evolução nos serviços disponibilizados, dividida em gerações. Nas Tabelas 2.1 e 2.2 são apresentadas a evolução dos dois tipos de portal, para cada uma das gerações (Eckerson, 1999b):

Tabela 2.1: Gerações dos portais públicos (Eckerson, 1999b)

Geração	Categoria	Características
Primeira	Referencial	Mecanismo de busca, com catálogo hierárquico de conteúdo da <i>web</i> . Cada entrada do catálogo contém uma descrição do conteúdo e um <i>link</i> para o conteúdo
Segunda	Personalizado	O usuário, por meio de um identificador e uma senha, pode criar uma visão personalizada do portal, conhecida como “Minha Página”. Essa visão mostra apenas as categorias que interessam a cada usuário. O portal pode avisar ao usuário sempre que um novo conteúdo for adicionado às categorias por ele assinaladas.
Terceira	Interativo	O portal incorpora aplicativos, tais como correio eletrônico, <i>chats</i> e listas de discussão para que o usuário possa interagir com o portal e com seu provedor de conteúdo. Os usuários podem selecionar as aplicações que querem utilizar por meio de suas páginas personalizadas.

Tabela 2.2: Gerações dos portais corporativos (Eckerson, 1999b)

Geração	Categoria	Características
Primeira	Referencial	Mesmas características dos portais públicos da primeira geração, enfatizando a gerência de conteúdo, disseminação das informações e o suporte à decisão
Segunda	Personalizado	Além das características dos portais públicos de segunda geração, os usuários podem publicar documentos nos repositórios corporativos, que podem ser visualizados por outros usuários. Essa geração privilegia a distribuição personalizada de conteúdo.
Terceira	Interativo	Incorpora aplicativos que melhoram a produtividade das pessoas e equipes, como correio eletrônico, calendários, agendas, fluxos de atividades, etc. Essa geração adiciona o caráter colaborativo ao portal, provendo múltiplos tipos de serviços interativos
Quarta	Especializado	Portais baseados em funcionalidades profissionais, para gerência de atividades específicas nas organizações (vendas, finanças, recursos humanos, etc.). Envolve a integração de outros aplicativos para que os usuários possam executar transações, ler, gravar e atualizar dados corporativos, incorporando ainda outras possibilidades como o comércio eletrônico

2. **Quanto à função:** classificação relacionada às funções que o portal desempenha. Essas funções estão relacionadas aos portais corporativos, uma vez que a função dos portais públicos é, em geral, somente a de atrair potenciais consumidores.

(a) *Portais com ênfase em suporte à decisão*

Têm como objetivo auxiliar os executivos, gerentes e analistas de negócios a acessar as informações corporativas para a tomada de decisões de negócios;

i. Portais de Informação ou de Conteúdo

Portais que organizam grandes acervos de conteúdo a partir dos temas ou assuntos nele contidos, conectando as pessoas às informações;

ii. Portais de Negócio

Têm como função tornar disponíveis informações como relatórios, pesquisas, documentos textuais, etc., sendo o ponto de partida central para os aplicativos de gerência de conteúdo e para o processamento das decisões corporativas, conectando os usuários a informações estruturadas e não estruturadas;

iii. Portais de suporte à decisão

Permitem que os usuários organizem e encontrem informações corporativas no conjunto de sistemas que constituem a cadeia produtiva de informações de negócios. Utilizam mecanismos de busca inteligentes e aplicações analíticas para capturar informações armazenadas em bases de dados operacionais, em *dataware house* corporativos e até mesmo em sistemas

externos. A partir dessas informações geram relatórios e análises de negócio para serem distribuídos eletronicamente aos diversos níveis de tomada de decisão na empresa.

(b) *Portais com ênfase em processamento cooperativo*

Esses portais lidam tanto com informações geradas dentro da cadeia produtiva tradicional, armazenadas e manipuladas por aplicativos corporativos, como informações geradas por grupos ou indivíduos fora dessa cadeia.

i. Portais cooperativos ou para processamento cooperativo

Provêem acesso a informações geradas por indivíduos ou grupos, informações geralmente estruturadas, personalizadas e sob a forma de textos, memorandos, gráficos, mensagens de correio eletrônico, páginas *web* e arquivos multimídia;

ii. Portais de especialistas

Meios de comunicação e troca de experiências entre pessoas especializadas em determinadas áreas do conhecimento, por meio de comunicação em tempo real, educação à distância e manutenção de cadastro automática de especialistas, para relacionar e unir pessoas com base em sua habilidade e experiência.

(c) *Portais de suporte à decisão e processamento cooperativo*

Portais que aliam as funcionalidades tanto dos portais com ênfase em suporte à decisão quanto dos portais com ênfase em processamento cooperativo, conectando os usuários não somente às informações existentes, mas também às pessoas necessárias para a realização dos negócios da empresa.

i. Portais de conhecimento

Segundo Murray (1999) *apud* (Dias, 2001), o portal de conhecimento é um ponto de convergência dos portais de informações, cooperativos e de especialistas, sendo capaz de implementar o que os demais portais implementam, oferecendo funções customizadas e conteúdo personalizado de acordo com as atividades de cada usuário, para o público em geral;

ii. Portais de informações empresariais

Aliam as características do portal cooperativo e do portal de suporte à decisão, com a integração de dados estruturados e não estruturados, fornecendo acesso às informações institucionais a partir de uma interface individualizada, para colaboradores de uma empresa.

Em Devedzic (2005), é definido, adicionalmente a essa classificação, os denominados *Portais de Comunidades*, que servem ao interesse de uma comunidade em particular e

possivelmente a um público mais geral. Esses portais são em geral mais abertos ao público já que eles não estão relacionados a uma organização específica ou instituição e permitem o acesso à informação de maneira menos restrita. Dentre os portais de comunidades estão os portais de pesquisa, que são oferecidos por grupos de pesquisas, universidades, gerentes de instituições de ensino, entre outros grupos, que possuem conhecimento sobre tópicos importantes em pesquisa.

Os portais podem, assim, ser projetados para diversos propósitos e, para atendê-los, devem possuir um conjunto de funcionalidades e serviços, apresentados na próxima seção.

2.2.3 Características e Funcionalidades de Portais

Nesta seção, são apresentadas funcionalidades e características referentes aos portais corporativos, que abrangem as características dos portais de conhecimento - objetivo de desenvolvimento deste trabalho - conforme classificação apresentada na seção anterior.

Não existe uma definição padronizada sobre quais serviços e funcionalidades devem estar presentes em um portal (Toledo, 2002). A maioria das organizações que procura por portais pretende alcançar dois objetivos específicos de funcionalidades: uma maneira de reunir informações de recursos de dados dispersos e torná-los disponíveis, e usar uma interface comum que permita que os usuários realizem suas ações, como buscar e acessar documento e interagir com pessoas (Watson e Fenner, 2000).

Vários autores mencionam as características e funções (também denominadas serviços) relacionadas aos portais (Benbya et al., 2004; Eckerson, 1999a; Raol et al., 2002; Smith, 2004; Terra e Gordon, 2002; Watson e Fenner, 2000). As características mostradas nesta seção são as de Benbya et al. (2004), Eckerson (1999a) e Watson e Fenner (2000), pois as demais apresentam pequenas variações em relação às anteriores.

Em Benbya et al. (2004), foi realizado um estudo experimental com o objetivo de identificar as características, benefícios e desafios relacionados aos portais. O estágio de coleta de dados envolveu três fases: identificar os principais “atores” do mercado de portal corporativo; as soluções e características propostas por eles; e os clientes que já haviam adotado tais soluções. Após essa coleta, foram reunidas as funções-chave dos portais, suas descrições e os seus benefícios em potencial.

A partir dos atores identificados, foram selecionados oito casos de estudo que possuíam implementações de sucesso dos portais corporativos, em diferentes níveis de implementação e com cobertura ampla na indústria (área de telecomunicações, automotiva, farmacêutica, produtos de consumo, alta-tecnologia e micro-eletrônica).

Com esse estudo, foi observado que algumas características e funcionalidades são comuns, na maioria dos casos, podendo ser divididas em três categorias: **capacidade núcleo**, **capacidades de apoio** e **webservices**, descritas a seguir (Benbya et al., 2004):

- **Capacidades núcleo:** capacidades que apóiam o processo de geração, armazenamento, distribuição e aplicação de conhecimento.
 - *Taxonomia:* também denominados esquemas de classificação ou esquemas de categorização, que se referem a agrupamentos de itens similares em tópicos ou pacotes, os quais podem ser agrupados em hierarquias. Os benefícios incluem busca, apoio, navegação, controle e mineração de dados, gerência de esquemas, entrega de informação e personalização;
 - *Publicação:* facilidade que apóia a criação, autorização e inclusão de conteúdo e, também, a habilidade para publicar documentos em formatos alternativos, como HTML, PDF, XML, etc. em coleções de conteúdo do portal;
 - *Busca:* capacidade essencial para encontrar as informações que estão dispersas em múltiplos repositórios a todos os interessados;
 - *Personalização:* habilidade dos usuários modificarem suas próprias interfaces e especificarem suas preferências, e a habilidade do sistema de usar tais informações dinamicamente para entregar conteúdos aos usuários;
 - *Integração:* habilidade para apresentar uma visão unificada de informações corporativas. Essa visão integra informações de diferentes repositórios organizacionais ao invés de ter informações corporativas espalhadas por meio de múltiplas fontes dentro da organização;
 - *Colaboração:* habilidade para a criação de uma comunidade compartilhada.
- **Capacidades de Apoio:** ferramentas necessárias para o bom funcionamento dos portais.
 - *Segurança:* habilidade para permitir acesso seguro a uma grande variedade de recursos com controles de acesso;
 - *Profiling:* técnica que permite que sejam enviadas informações personalizadas ao usuário de acordo com seu perfil;
 - *Escalabilidade:* capacidade de expansão do sistema para apoiar um número crescente de usuários.
- **Web services:** são aplicações que possuem uma grande abrangência de serviços oferecidos aos usuários. Esses serviços vão desde a inclusão de notícias até facilidades de serem realizadas negociações com os consumidores.

Eckerson (1999a) define um conjunto mínimo de quinze funcionalidades que um portal corporativo deve possuir, dentre essas funcionalidades, sendo algumas comuns às apresentadas acima. As diferenças estão nas características de: facilidade, acesso dinâmico,

roteamento inteligente, arquitetura baseada em servidor, distribuição, API's (Interface de Programação de Aplicação) e manutenibilidade. Na Tabela 2.3, são listados os requisitos de Eckerson (1999a) e a equivalência às características apresentadas por Benbya et al. (2004).

Tabela 2.3: Requisitos mínimos de um portal corporativo

Característica	Descrição	Benbya et al. (2004)
Fácil para usuários	Apoiar as necessidades de usuários, atendendo a critérios de usabilidade, como o critério de ser intuitivo ao uso	Sem equivalente
Classificação e Busca	Usar metáfora intuitiva para classificar os objetos, descrições e propriedades e um poderoso mecanismo de busca	Busca
Publicação e Subscrição	Permitir que os usuários publiquem objetos e especificar quem pode acessá-los	Publicação
Conectividade universal	Conectar múltiplos e heterogêneos repositórios de dados	Integração
Acesso dinâmico a informações	Acessar de forma dinâmica relatórios criados por ferramentas de gerência de documentos e <i>business-intelligence</i>	Sem equivalente
Roteamento inteligente	Direcionar relatórios/documentos de forma inteligente aos indivíduos, como parte de um <i>workflow</i>	Sem equivalente
Integrar conjunto de ferramentas	Oferecer capacidades de consulta, relatório e análise de modo integrado	Integração
Arquitetura baseada em servidor	Apoiar, ao menos, uma arquitetura em três camadas, para suportar pedidos concorrentes de milhares de usuários	Sem equivalente
Serviços distribuídos	Ter serviços distribuídos por múltiplos servidores a fim de melhorar o balanceamento de carga	Sem equivalente
Administração de permissão flexível	Permitir que os administradores definam permissões para indivíduos, grupos e regras	<i>Profiling</i>
Integrar interfaces externas	Oferecer interfaces para se comunicar com outras aplicações	Integração
Interface Programável	Ter pública a interface programável (API), a fim de ser chamado por outros aplicativos	Sem equivalente
Segurança	Apoiar serviços de segurança para proteger a informação corporativa e prevenir contra acessos não autorizados	Segurança
Implantação e Manutenção de baixo custo	Ser fácil de instalar, configurar e manter (manutenibilidade)	Sem equivalente
Customização e Personalização	Criar aparências distintas de acordo com as preferências dos usuários e permitir que estes personalizem o portal	Personalização

Adicionalmente, Watson e Fenner (2000) citam a característica de *auditoria*, que significa a habilidade de rastrear o uso, acesso a informação, modificações e atualizações nas

informações estruturadas e não-estruturadas do portal. Essa característica proporciona a capacidade de haver gerência do conteúdo disponibilizado no portal.

Reunindo essas características e funcionalidades, deve ser definido como o portal será desenvolvido, ou seja, quais aspectos devem ser estudados a fim de que sejam modelados e projetados os elementos do portal. Na próxima seção, serão abordados alguns aspectos relevantes no escopo do desenvolvimento de portais.

2.2.4 Desenvolvimento de Portais

Para desenvolver os portais corporativos e usá-los de forma efetiva como espaços de trabalho de informação e conhecimento compartilhados, Toledo (2002) considera relevante que três aspectos estejam bem embasados e sedimentados: *(i)* entendimento de como os trabalhadores do conhecimento buscam, estruturam, compartilham e contextualizam informações e experiências requeridas para o desempenho de suas funções diárias; *(ii)* necessidade de haver um modelo robusto de portal do conhecimento; e *(iii)* utilização de uma abordagem sistêmica para o desenho e especificação do portal que seja dirigida não só pelos dados ou fluxos de dados, mas também por uma avaliação completa da estratégia da organização, política, cultura, pessoas e processos que precisam ser suportados

Em relação ao primeiro aspecto - obter o entendimento sobre como os trabalhadores compartilham o conhecimento - Friedman (2002) *apud* (Toledo, 2002) destaca a importância da formação de um vocabulário comum de algumas definições básicas das questões relacionadas aos problemas do conhecimento e seus impactos na eficácia de uma organização. Dentre elas, quatro definições tem maior relevância:

- **Trabalhador do conhecimento:** pessoa que cria valor usando conhecimento e habilidade para avaliar e processar informação;
- **Fluxo de conhecimento:** modo pelo qual o conhecimento é criado, capturado, validado e comunicado e o processo pelo qual o conhecimento passa de elemento de capital humano (pessoal) para capital estruturado (da firma);
- **Processo de trabalho:** uma combinação do fluxo de conhecimento e do fluxo de trabalho (etapas de trabalho executadas em um processo de negócio), provendo o contexto de conhecimento necessário para o trabalho ser executado;
- **Desempenho do trabalhador:** resultado do encadeamento entre os fluxos de trabalho e de conhecimento.

O segundo aspecto está relacionado ao modelo que deve ser utilizado para definir a estrutura do portal. Um modelo é uma representação abstrata do sistema que está sendo

desenvolvido (Conallen, 1999). Ele permite estruturar os componentes que vão atender às características e serviços do sistema, bem como as propriedades externamente visíveis e as relações entre esses componentes.

Há vários modelos diferentes para se estruturar portais, como em Terra e Gordon (2002), Smith (2004), The Delphi Group (2001), Aneja et al. (2000), White (1999) e Shilakes e Tylman (1999). A descrição dos três primeiros é realizada a seguir.

Em Terra e Gordon (2002), os componentes que atendem ao conjunto de funcionalidades de um portal são agrupados da seguinte maneira, conforme apresentado na Figura 2.1:

Figura 2.1: Arquitetura de um Portal Corporativo (Terra e Gordon, 2002)

- **Camada de Apresentação e Personalização:** define como os usuários visualizam e customizam a informação que é disponibilizada ou acessada pelo portal. Nesse sentido, são utilizados os perfis dos usuários para que as informações que eles considerem mais relevantes no contexto de sua atividade sejam disponibilizadas;
- **Solução de busca:** determina quão fácil será para os usuários encontrarem informações relevantes baseadas em um conjunto de critérios de busca. As soluções de busca têm-se desenvolvido para integrar várias fontes de informação, tornando-se assim mais sofisticadas, intuitivas e adaptáveis às diferentes necessidades organizacionais. Inclui os mecanismos de taxonomia, ou seja, classificação da informação, e de buscas para que as informações sejam encontradas;
- **Aplicações web:** soluções sofisticadas de plataformas de portais oferecem uma série de recursos que tornam muito mais fácil integrar as aplicações web com aplicações tradicionais, sem maiores necessidades de ajustes. Entre elas, estão as API's

(*Application Programming Interfaces* - Interfaces de Programação de Aplicações), que oferecem interfaces padronizadas para efetuar a comunicação entre sistemas diferentes. Essas API's permitem acesso à informação estruturada e não estruturada. Dentro da arquitetura proposta, os recursos de integração são chamados de *conectores*.

Além desses componentes principais, estão presentes componentes que se integram aos serviços do portal, como as ferramentas de colaboração, sistemas de gerenciamento de conteúdo e sistemas externos.

Em The Delphi Group (2001) existem nove componentes básicos para o modelo da arquitetura de portais, ilustrados na Figura 2.2:

- **Integração:** provê a estrutura de acesso às fontes de informação internas e externas, fazendo com que elas estejam disponíveis no portal. Essa camada é similar à camada de aplicações *web* do modelo anterior, em relação à utilização dos conectores que realizam a integração entre as informações.
- **Categorização:** componente que tem como objetivo estabelecer padrões para contextualizar a informação disponível no portal por meio de camadas inter-relacionadas de significado. Esse componente está incorporado na camada de solução de busca, equivalente à taxonomia, do modelo anterior, que procura oferecer meios de facilitar que os usuários encontrem informações;

Figura 2.2: Componentes de um Portal
(The Delphi Group, 2001)

- **Mecanismos de Busca:** oferece facilidades para a localização de informações específicas nos diversos repositórios da instituição. Essa camada equivale ao componente de busca da camada de Solução de Busca do modelo de Terra e Gordon (2002);
- **Publicação e Distribuição:** componente que tem como objetivo apoiar a criação, autorização e inclusão de conteúdo em vários formatos;
- **Suporte a Processos:** apoiar processos de administração de negócio eletrônico, como por exemplo, o acompanhamento de *status* de mensagens, apoio a transações, disparar eventos de cobrança, entre outras;
- **Colaboração:** permite interações entre os usuários do portal, possibilitando, por exemplo, a redução do tempo requerido para atividades de consumidores e a melhora do relacionamento com os acionistas. No modelo de Terra e Gordon (2002) existem as ferramentas de colaboração integradas que equivalem a esse componente;
- **Personalização:** permite a customização do conteúdo do portal por meio da configuração de interfaces, da definição de *layout* de apresentação e da eliminação de conteúdo indesejado para o acesso às informações relevantes. As camadas de apresentação e de personalização do modelo de Terra e Gordon (2002) incorporam as características desse componente;
- **Apresentação:** responsável pelo paradigma de um único ponto de acesso, com a integração dos elementos que o usuário acessa de forma consistente, visualmente agradável e que faça sentido ao mesmo. Equivalente à camada de apresentação de Terra e Gordon (2002);
- **Ciclo de Aprendizado:** componente que tem a preocupação com a eficácia contínua do portal, que identifica mudanças de necessidade de informação de diversos tipos de profissionais e no provimento dos ajustes e atualizações com a rapidez necessária. Esse componente não apresenta equivalente no modelo de Terra e Gordon (2002).

Smith (2004) propõe um Modelo de Portal Geral (GPM - *General Portal Model*) com uma infra-estrutura que identifica e organiza as capacidades básicas do portal. Essa arquitetura é dividida em três grupos, reunindo as sete camadas do modelo:

- *Camadas de apresentação ou Camadas de interface de processo:* duas camadas relacionadas ao acesso ao portal pelas aplicações, que funcionam como clientes, servidores ou pontos. Essa camada é similar à camada de apresentação presente em Terra e Gordon (2002) e The Delphi Group (2001).

- Identificação de Processo: auxilia os processos que usam o mecanismo de troca de dados, oferecendo uma conexão ponto a ponto entre um ou mais processos que mandam ou recebem mensagens. Como a camada possui todas as identidades de processos, ela facilita o acesso, registro e monitoramento de desempenho. Essa camada é similar a camada de suporte a processos do modelo apresentado por The Delphi Group (2001);
- Transformação: essa camada usa esquemas de conteúdos e apresentações para garantir que os dados apresentados e trocados pelos processos possam ser compatíveis uns com os outros em termos de tipo, tamanho e formato, realizando as transformações necessárias nos dados. Além disso, ela também implementa regras que descrevem o que fazer se o conteúdo não puder ser apresentado no formato necessário. Essa camada adiciona funcionalidades às camadas de publicação e distribuição de The Delphi Group (2001), permitindo que conteúdos de vários formatos sejam publicados e estejam disponíveis em formatos intercambiáveis;
- *Camadas de acesso a recursos ou Camadas de recuperação de recursos*: três camadas consideradas o coração das funcionalidades de portais. Elas determinam requisitos de recursos e os tornam disponíveis para um domínio comum, baseado em metadados, encontrando recursos que satisfaçam os requisitos e coordenando as interações entre os nós. As três camadas seguintes facilitam as buscas realizadas dentro dos recursos do portal, apresentando equivalentes nos componentes de busca dos modelos anteriores.
 - Identificação de Recurso: implementa um entendimento comum de capacidades de recursos, distinguindo como as camadas acima os utilizam e como eles podem ser obtidos pelas camadas abaixo;
 - Localização de Recurso: coordena o trabalho de localizar os recursos cujas identificações foram fornecidas pela camada de identificação de recurso. Dada uma descrição de recursos, ela determina o endereço desse recurso;
 - Associação de Recurso: implementa uma conexão entre um recurso no servidor e o cliente, mantendo a interação separada uma da outra.
- *Camadas de acesso à rede ou Camadas de interface de rede*: duas camadas que implementam os detalhes de acesso à rede das camadas de recuperação de recursos.
 - Segurança: camada que funciona como um portão entre os processos do portal e a rede. Verifica a identificação dos processos que estão ativos e sua autorização para acesso aos recursos, realizando funções de *firewall* para o portal. Essa camada não é representada nos dois modelos anteriores;

- Acesso à rede: camada que cria um canal entre os nós dos processos e os dados para transmissão por meio da rede. Realizam a integração entre os dados e os sistemas presentes no portal, sendo similar às camadas de integração de The Delphi Group (2001) e de aplicações *web*, de Terra e Gordon (2002).

Na Tabela 2.4 estão representadas as sete camadas do Modelo de Portal Geral, com sua respectiva descrição:

Tabela 2.4: Modelo de Portal Geral (Smith, 2004)

Camada	Descrição
Camada de Interface de Processo	
Identificação de Processo	Interfaces do Portal com processos cliente e servidor; Regras de acesso de processos clientes e servidor ao portal; Definição de apresentação e esquema de conteúdo
Transformação	Baseado no esquema de apresentação, transforma <i>streams</i> de dados para que sejam adequados aos seus destinos
Camada de Descoberta de Recursos	
Identificação de Recurso	Baseado nas descrições dos conteúdos solicitados, identifica necessidades de aplicações para atender as solicitações
Localização de Recurso	Localiza aplicações necessárias, talvez em vários servidores
Associação de Recurso	Associa recursos individuais para a camada de localização de recurso
Camada de Interface de Rede	
Segurança	Regras de acesso ao portal pela rede
Acesso a rede	Oferece uma interface uniforme para os serviços de rede aos processos do portal, ocultando a implementação de rede desses processos

Segundo Smith (2004), esse modelo pode apoiar a criação de uma comunidade de aplicações para servir às necessidades de um usuário particular, acomodando diferentes modelos de interação e arquiteturas.

Na Tabela 2.5 estão representadas as características identificadas na Seção 2.2.3 com os modelos representados nesta seção, a fim de permitir o mapeamento entre o que os modelos propõem e as características que são atendidas por eles.

Além dos dois aspectos mostrados no decorrer desta seção (estruturação da informação e modelo de portal), é importante observar o atendimento ao terceiro aspecto: a abordagem sistêmica que guiará o desenvolvimento e evolução do portal. Para isso, deve ser realizada a escolha de um processo bem definido com um conjunto de passos que guiem a construção do portal. Os estudos realizados sobre processos e a seleção de um deles para o desenvolvimento de portais estão descritos no Capítulo 3.

Tabela 2.5: Mapeamento Características X Modelos

Caraterísticas	Terra e Gordon (2002)	The Delphi Group (2001)	Smith (2004)
<i>Taxonomia</i>	Solução de Busca	Categorização	-
<i>Publicação</i>	Apresentação e Personalização	Publicação e Distribuição	Apresentação
<i>Busca</i>	Solução de Busca	Mecanismo de Busca	Acesso a Recursos
<i>Personalização</i>	Apresentação e Personalização	Personalização	Apresentação
<i>Integração</i>	Aplicações <i>web</i>	Integração	Acesso à rede
<i>Colaboração</i>	-	Colaboração	-
<i>Segurança</i>	-	-	Segurança
<i>Profiling</i>	Apresentação e Personalização	Personalização	Apresentação
<i>Escalabilidade</i>	-	-	-
<i>Web Services</i>	-	-	-
<i>Facilidade</i>	-	-	-
<i>Acesso Dinâmico</i>	Apresentação e Personalização e Solução de Busca	Apresentação, Personalização e Mecanismo de Busca	Apresentação, Acesso à rede e Acesso a recursos
<i>Arquitetura Servidor</i>	-	-	-
<i>Distribuição</i>	-	Publicação e Distribuição	Acesso a recursos
<i>API's</i>	Aplicações <i>web</i>	Integração	Acesso à rede
<i>Manutenibilidade</i>	-	-	-
<i>Auditoria</i>	-	-	-

2.3 Teste de Aplicações Web

Dentro de um curto período, com o advento da Internet e da *World Wide Web*, as aplicações *web* tornaram-se bastante populares, sendo produzidas em grandes velocidades e com um número cada vez maior de usuários. Isso provoca um impacto significativo em uma ampla variedade de setores, como nos negócios, comércio, indústria, bancos, educação, governo e setores de entretenimento, bem como em nossas vidas pessoais e profissionais (Ginige e Murugesan, 2001b; Xu et al., 2005).

Inicialmente, a *web* foi concebida com o propósito de compartilhar informações científicas entre grupos de pesquisa, com um número pequeno de usuários. Atualmente, elas atendem desde serviços de pequena escala a aplicações complexas distribuídas em diversos sistemas, com uma variação grande em termos de escopo e complexidade (Ginige e Murugesan, 2001b).

As aplicações *web* mais simples possuem páginas com informações textuais, cujo conteúdo não é alterado, e com uma navegação simplificada. Além disso, em geral, são sistemas *stand-alone*, cujo desempenho não é o requisito principal, e são desenvolvidas por

uma única pessoa ou por um grupo pequeno. Já as aplicações mais avançadas apresentam páginas *web* complexas, com conteúdo dinâmico e navegação complexa, com exigência de alto desempenho e necessidade de disponibilidade contínua, sendo desenvolvidas por equipes de especialistas para serem utilizadas em aplicações cruciais (Ginige e Murugesan, 2001b).

As aplicações *web* possuem uma série de particularidades que as diferenciam dos sistemas tradicionais, influenciando nas atividades de teste, com o objetivo de assegurar a qualidade exigida para essas aplicações.

Ainda que técnicas, métodos e ferramentas sejam empregados com a finalidade de reduzir a possibilidade de falhas, erros podem ser introduzidos durante o processo de desenvolvimento de software (Maldonado et al., 2004). Para que seja possível minimizar a ocorrência de erros e riscos associados, várias atividades para a garantia da qualidade de software são conduzidas ao longo de todo o processo de desenvolvimento. Entre essas atividades estão as de *Verificação, Validação e Teste*, comumente denominadas de VV&T. Dentre elas, as atividades de teste são as mais utilizadas, como um elemento importante em qualquer projeto de software, para a garantia da qualidade na revisão final da especificação, projeto e geração de código (Maldonado et al., 2004; Myers, 2004; Pressman, 2005).

Os testes de software têm como objetivo executar os programas a fim de assegurar que o código realize aquilo para o qual foi projetado e não execute nada além daquilo que foi pretendido, além de encontrar um erro ainda não descoberto (Myers, 2004).

Para atingir de maneira satisfatória os objetivos, os testes devem ser conduzidos de maneira sistemática e disciplinada, com a elaboração de técnicas e estratégias que permitam a seleção de um subconjunto do domínio de entrada para avaliar os aspectos funcionais e operacionais de um produto em teste (Myers, 2004; Vincenzi, 2004).

As atividades de teste envolvem basicamente quatro etapas: planejamento de testes, projeto de casos de teste, execução e avaliação dos resultados. Elas ocorrem ao longo do próprio processo de desenvolvimento e são concretizadas, geralmente, em três fases de teste: teste de unidade, teste de integração e teste de sistema. (Maldonado et al., 2004).

Para as aplicações *web*, as atividades de teste se tornam mais complicadas, por conta das características e diferenças dessas aplicações. Assim, é necessária a criação e/ou adequação de técnicas e estratégias que garantam a qualidade das aplicações *web* (Liu et al., 2000a,b; Xu et al., 2005).

Nesta seção, serão apresentados os conceitos das técnicas e critérios de teste de software e propostas delas para o contexto de aplicações *web*, estratégias para conduzir as atividades de teste, de maneira geral, e estratégias específicas para os testes de aplicações *web* e alguns exemplos de ferramentas de teste de aplicações *web*.

2.3.1 Técnicas e Critérios de Teste

As técnicas de teste fornecem diretrizes para projetar os casos de teste de forma a forçar a execução da lógica interna do software, bem como testar os domínios de entrada e os resultados gerados por uma aplicação para descobrir possíveis erros na função, desempenho e comportamento do programa (Pressman, 2005).

Os critérios de teste podem ser utilizados tanto para auxiliar na geração de um conjunto de casos de teste como para auxiliar na avaliação da adequação desses conjuntos. Em geral, os critérios de teste de software são estabelecidos a partir de três técnicas: funcional, estrutural e baseada em erros (Maldonado et al., 2004).

Em Howden (1987) é encontrada uma classificação para o teste em duas formas:

- **Teste baseado em especificação:** refere-se aos testes usados para demonstrar que as funções do software estão operacionais, que a entrada é adequadamente aceita e a saída é corretamente produzida, e que a integridade da informação é mantida (Pressman, 2005);
- **Teste baseado em programa:** baseado em um exame rigoroso dos caminhos lógicos internos do software (Pressman, 2005).

De acordo com essa classificação, tem-se que os critérios da técnica funcional são baseados em especificação e tanto os critérios estruturais quanto os baseados em erros são critérios baseados em programa (Maldonado et al., 2004).

2.3.1.1 Técnica Funcional

O teste funcional, também denominado teste caixa-preta, é um processo que tem a intenção de encontrar as discrepâncias entre o programa e a especificação externa, a qual é uma descrição do comportamento do programa do ponto de vista do usuário (Myers, 2004).

Essa técnica tenta encontrar erros nas seguintes categorias: funções incorretas ou omitidas; erros de interface; erros de estrutura de dados ou acesso à base de dados externa; erros de comportamento ou desempenho e; erros de iniciação e término (Pressman, 2005).

Alguns tipos de técnica funcional são (Myers, 2004):

- **Particionamento de Equivalência:** consiste no particionamento do domínio de entrada de um programa em um número finito de classes de equivalência - identificadas a partir de cada condição de entrada em um ou mais grupos. O objetivo dessa técnica é cobrir um conjunto maior que em outros possíveis casos de teste,

considerando que: (i) se um caso de teste em uma classe de equivalência detecta um erro, todos os outros casos de teste nas classes equivalentes devem encontrar o mesmo erro; e (ii) se um caso de teste não detectar um erro, pode-se esperar que nenhum outro caso de teste irá detectá-lo em outras classes de equivalência;

- **Análise de Valor Limite:** casos de teste que exploram as condições diretamente sobre, acima e abaixo dos limites de entrada e saída das classes de equivalência. Essa técnica difere da anterior em dois aspectos: (i) ao invés de selecionar qualquer elemento da classe de equivalência, exige que um ou mais elementos sejam selecionados de forma que cada limite da classe de equivalência seja submetido ao teste e (ii) ao invés de somente considerar as condições de entrada, os casos de teste projetados consideram o espaço de resultados. A análise de valor limite complementa o particionamento de equivalência;
- **Grafo de causa e efeito:** técnica que tenta explorar combinações de circunstâncias de entrada, as quais não são analisadas pelas técnicas anteriores. Uma causa é uma condição de entrada e um efeito é uma condição de saída ou uma transformação no sistema. As causas e efeitos são relacionados entre si de acordo com a especificação, com a formação do grafo de causa e efeitos, posteriormente convertido em uma tabela de decisão, cujas colunas representam casos de teste.

Segundo Beizer (1990), os critérios da técnica funcional não dependem do código fonte para serem aplicados, mas sim da especificação de requisitos e, portanto, podem ser aplicados para as mais diversas classes de aplicações.

2.3.1.2 Técnica Estrutural

A técnica estrutural, também conhecida como técnica caixa-branca, baseia-se no conhecimento da estrutura interna da implementação para a escolha dos casos de teste (Maldonado et al., 2004). Ao contrário dos testes funcionais, essa técnica implica na inspeção do código fonte e na seleção de casos de teste que exercitem partes do código, e não de sua especificação (Perry e Kaiser, 1990).

Os objetivos dessa técnica são: garantir que todos os caminhos internos de um módulo sejam testados ao menos uma vez; exercitar todas as decisões lógicas em seus lados verdadeiro ou falso; executar os ciclos nos seus limites e dentro de seus intervalos operacionais; e garantir a validade das estruturas internas de dados (Pressman, 2005).

Em geral, os critérios dessa técnica usam uma representação do programa denominada *grafo de fluxo de controle* ou *grafo de programa*. Cada nó (ou vértice) do grafo representa um segmento de comandos executados sequencialmente e cada aresta (arco) representa uma transferência de controle entre os segmentos. Um caminho é uma sequência finita

de dois ou mais nós de tal forma que exista uma aresta entre um nó e o nó subsequente (Maldonado et al., 2004; Myers, 2004).

Os critérios de teste são, em geral, classificados em (Maldonado et al., 2004):

1. **Critérios baseados em fluxo de controle:** utilizam características de controle de execução do programa, como comandos ou desvios, para determinar quais estruturas devem ser exercitadas. Dentre os critérios dessa técnica estão:
 - (a) *Todos-Nós*: cada comando do programa deve ser exercitado ao menos uma vez, assim, a execução do programa deve passar ao menos uma vez em cada nó do grafo;
 - (b) *Todos-Arcos*: cada desvio de fluxo de controle do programa deve ser exercitado pelo menos uma vez, ou seja, cada arco do grafo deve ser exercitado ao menos uma vez;
 - (c) *Todos-Caminhos*: requer que todos os caminhos possíveis do programa sejam executados;
2. **Critérios baseados em fluxo de dados:** utilizam as informações do fluxo de dados para derivar os casos de teste. Essa classe de critérios procura fornecer uma hierarquia entre os critérios Todos-Arcos e Todos-Caminhos, procurando tornar o teste mais rigoroso, já que o teste Todos-Caminhos é, em geral, impraticável. Nesse critérios, são selecionados caminhos de teste de um programa de acordo com a localização das definições e do uso das variáveis do programa (Pressman, 2005). Exemplos dessa classe são:
 - (a) *Critérios de Rapps e Weyuker*: Rapps e Weyuker (1982, 1985) estenderam o grafo de programa em um *Grafo Def-Use* com informações a respeito do fluxo de dados, caracterizando as associações entre pontos do programa onde é atribuído um valor a uma variável (definição) e pontos onde esse valor é utilizado (uso). Nessa família de critérios estão incluídos:
 - i. *Todas-Definições*: cada definição da variável deve ser exercitada ao menos uma vez;
 - ii. *Todos-Usos*: todas as associações entre uma definição de variável e seus subsequentes usos devem ser exercitados por pelo menos um caminho onde a variável não é redefinida.
 - (b) *Critérios Potenciais-Usos* (Maldonado, 1991): os elementos são caracterizados independentemente do uso explícito de uma determinada definição. Assim, se um uso dessa definição pode existir, a potencial associação entre a definição e o potencial-uso é caracterizada e, eventualmente, requerida. Esse critério

procura explorar todos os possíveis efeitos a partir de uma mudança de estado do programa em teste.

3. **CrITÉrios baseados na complexidade:** utilizam informações sobre a complexidade do programa para derivar os requisitos de teste.

Para as aplicações *web*, Ricca e Tonella (2001) propuseram um meta-modelo de uma página *web*, como mostrado na Figura 2.3. Esse modelo possui como entidade central uma **webpage** (página *web*), que contém as informações exibidas ao usuário. A navegação entre as páginas é definida por uma auto-associação da classe *webpage* denominada **link**.

Figura 2.3: Meta-modelo de uma página web (Ricca e Tonella, 2001)

A página *web* pode ser estática ou dinâmica, sendo que o conteúdo da primeira é fixo e o conteúdo da segunda é dinâmico, computado em tempo de execução. Caso a página seja dinâmica, ela depende de um valor de entrada, denominado **use**.

Além disso, cada página pode estar em um ou mais **frames**. A propriedade **target** do **link** indica em que **frame** a nova página deve ser carregada, sendo que o **frame** alvo é a classe associativa **LoadPageIntoFrame**.

As páginas também podem conter um ou mais **forms** (formulários), que possibilitam a entrada de dados, submetidos e enviados por meio do **submit**. A classe associativa

denominada ***ConditionalEdge*** indica que a exibição da página depende de determinado valor para uma variável de entrada, podendo ela ser ou não exibida.

Os modelos gerados são interpretados como grafos, sendo que os nós correspondem aos objetos do modelo e as arestas correspondem às associações entre os objetos.

Com os grafos, é feita a geração de casos de teste a partir da seleção de um conjunto de caminhos no grafo. Os casos de testes podem ser derivados a partir dos testes das aplicações tradicionais (Ricca e Tonella, 2001):

- **Teste de páginas:** cada página do site deve ser visitada ao menos uma vez;
- **Teste de *Hiperlinks*:** cada *link* de cada página deve ser visitado ao menos uma vez;
- **Teste definição-uso:** todos os caminhos de navegação para cada definição de uma variável e que forme dependência devem ser exercitados;
- **Teste todos-usos:** ao menos um caminho de navegação para cada definição de uma variável e que forme dependência deve ser exercitado;
- **Teste todos-caminhos:** cada caminho no site é exercitado em algum caso de teste ao menos uma vez.

Em relação à classe de critérios baseados em fluxo de dados, Liu et al. (2000a,b) propõem o Modelo de Teste de Aplicações *Web* (WATM - *Web Application Test Model*), composto de:

1. **Modelo de Objetos:** representa os componentes da aplicação *web* como objetos, constituídos de atributos e operações. Os atributos podem ser variáveis do programa ou elementos dos documentos e as operações podem ser funções de *script* ou das linguagens de programação da aplicação. Existem três tipos de objetos nesse modelo:
 - (a) *Páginas cliente:* documento HTML ou XML, possivelmente com *scripts* incluídos e visualizados no lado cliente por meio do navegador;
 - (b) *Páginas servidor:* página executada no lado servidor;
 - (c) *Componentes:* podem ser qualquer programa que interage com uma página cliente ou servidor, ou com outros componentes.
2. **Modelo de Estruturas:** diagrama de relação de objetos (ORD - *Object Relation Diagram*), que representa o fluxo de dados entre os objetos do modelo, sendo utilizados quatro grafos de fluxo:
 - (a) *Grafo de fluxo de controle:* fluxo de dados de uma única função;

- (b) *Grafo de fluxo de controle interprocedural*: fluxo de dados em mais de uma função;
- (c) *Grafo de fluxo de controle de objeto*: sequência de invocação de várias funções dentro de um objeto;
- (d) *Grafo de fluxo de controle composto*: fluxo de dados entre páginas interagindo.

Com esse modelo, casos de teste podem ser derivados a partir de três perspectivas:

- **Intra-objeto**: caminhos de teste são selecionados das variáveis que têm ligações definição-uso dentro de um objeto;
- **Inter-objeto**: os caminhos de teste são selecionados para as variáveis que têm uma associação definição-uso entre os objetos;
- **Inter-cliente**: como existe a possibilidade de haver variáveis sendo compartilhadas entre vários clientes, pode haver associações definição-uso entre essas páginas. Assim, os casos de teste devem ser derivados a partir das interações de dados entre os clientes.

2.3.1.3 Técnica Baseada em Erros

A técnica baseada em erros utiliza informações sobre os tipos de erros mais frequentes no processo de desenvolvimento de software para derivar os requisitos de teste (Barbosa et al., 2000). Os principais critérios utilizados durante a realização de teste baseado em erros são: (i) Semeadura de Erros e (ii) Análise de Mutantes.

A semeadura de erros utiliza estimativas de probabilidade para verificar possíveis erros existentes no programa. No critério de Análise de Mutantes são identificados desvios sintáticos mais comuns, por meio da aplicação de pequenas transformações sobre o programa original, gerando os denominados programas mutantes. Esse critério tenta determinar um conjunto de casos de teste que consiga revelar as diferenças entre o programa original e seus mutantes (Maldonado et al., 2004).

Lee e Offutt (2001) descrevem uma abordagem baseada em mutação para o teste de aplicações *web* baseadas em XML. Essa abordagem preocupa-se na confiabilidade com que os dados são transferidos entre os componentes por meio do XML, mais especificamente, a corretude funcional de suas interações.

Convém ressaltar que as técnicas de teste devem ser vistas como complementares e a questão está em como utilizá-las para extrair as vantagens de cada uma, por meio de uma estratégia de teste que leve a uma atividade de teste de boa qualidade, ou seja, eficaz e de baixo custo (Maldonado et al., 2004).

2.3.2 Estratégias de Teste

As estratégias de teste de software integram os métodos de projetos de caso de teste e fornecem um roteiro para descrever os passos a serem conduzidos como parte do teste, quando esses passos devem ser realizados e depois executados (Pressman, 2005).

Esses passos devem ser executados no decorrer do processo de desenvolvimento. Em geral, são executados nas seguintes fases (Pressman, 2005):

- **Teste de unidade:** processo de testar individualmente o menor bloco construtor (componente) de um programa, sendo um modo de gerenciar os elementos combinados de teste, de forma a facilitar as tarefas de depuração e introduzir paralelismo ao processo de teste, pois apresenta a possibilidade de testar simultaneamente múltiplos blocos (Myers, 2004);
- **Teste de integração:** é uma técnica sistemática para construir a estrutura do programa, a fim de conduzir testes para descobrir erros associados às interfaces entre os componentes. Cada vez que forem adicionados novos componentes ao sistema como parte do teste de integração, devem ser conduzidos os denominados *testes de regressão*, para efetuar a reexecução de algum subconjunto de testes para garantir que as modificações não propagaram efeitos colaterais indesejáveis (Pressman, 2005);
- **Teste de sistema:** é o processo mais difícil e mal-compreendido do processo de teste. Ele não tem como objetivo testar as funções do sistema completo - como os testes funcionais - mas sim compará-lo aos seus objetivos originais (Myers, 2004). O teste de sistemas é uma série de diferentes testes cuja finalidade é exercitar por completo o sistema com o objetivo de verificar se o software, juntamente com outros elementos (por exemplo, hardware, pessoas e banco de dados), se combinam adequadamente e se a função e desempenho globais desejados pelo usuário são atingidos (Pressman, 2005).

As categorias de teste de sistema explicadas a seguir não precisam ser aplicadas a todos os programas, mas aplicadas de acordo com o que foi exigido para o sistema e com os objetivos do teste (Myers, 2004):

- *Teste de Facilidade:* teste para determinação de que cada função requerida pelo sistema, listada em seus objetivos, estejam implementadas;
- *Teste de Volume:* tem como objetivo mostrar se o programa pode manipular os volumes de dados especificados nos objetivos;
- *Teste de Stress:* testa o comportamento do sistema frente a um volume de dados manipulados em um curto período de tempo;

- *Teste de Usabilidade*: tenta encontrar erros relacionados à interação entre o homem e o computador;
- *Teste de Segurança*: é o processo que deriva casos de teste que tentam acessar de forma indevida o sistema e subverter as verificações de segurança do sistema, para encontrar erros nos mecanismos de proteção;
- *Teste de Desempenho*: testa o desempenho do sistema durante a execução das funções;
- *Teste de Armazenamento*: assegurar que os objetivos de armazenamento (como em memória secundária e arquivos temporários) estão sendo cumpridos;
- *Teste de Configuração*: tem como objetivo assegurar que as configurações da aplicação com dispositivos de hardware e componentes externos à aplicação, como sistemas de bases de dados, estejam corretas;
- *Teste de Compatibilidade/Conversão*: assegurar que objetivos de compatibilidade e conversão em relação aos dados manipulados pela aplicação estão sendo cumpridos;
- *Teste de Instalabilidade*: a instalação de um sistema é parte do seu processo de desenvolvimento, assim, esse teste tem como objetivo assegurar que o sistema não apresenta erros durante esse processo;
- *Teste de Confiabilidade*: testa os requisitos específicos de confiabilidade do sistema, como, por exemplo, haver um tempo médio entre as falhas que o sistema pode apresentar por tempo de uso;
- *Teste de Recuperação*: tem como objetivo testar como a aplicação se recupera diante de erros de programação, falhas de hardware e erros de manipulação de dados;
- *Teste de Serviços/Manutenibilidade*: tem como objetivo testar as características de manutenção da aplicação, como facilidade de entendimento do código, incluindo a qualidade da lógica interna, e procedimentos de manutenção;
- *Teste de Documentação*: teste que utiliza a documentação do usuário (manual do usuário) para assegurar características como clareza e acurácia;
- *Teste de Procedimentos*: para alguns sistemas não completamente automatizados, esse teste assegura que a documentação de procedimentos realizados de forma manual são adequadas.

Para as aplicações *web*, deve ser analisado o que pode ser considerado como a menor unidade a ser testada, pois pode ser tanto os componentes internos de uma página, como *scripts* e formulários, que interagem entre si, quanto a própria página, levando em conta

o fato de já haver uma combinação associada a seus componentes internos (Lucca et al., 2002).

Lucca et al. (2002) propõem como unidade da aplicação *web* uma página - cliente ou servidor - para criar os casos de testes funcionais e ,então, realizar os testes de integração.

As páginas cliente e servidores são testadas a partir de uma *tabela de decisões*, estrutura que representa as páginas a serem testadas. Essa tabela é composta de duas partes: seção de condição, que contém as variáveis de entrada e combinações das entradas, juntamente com as possíveis ações de entrada e estado da página antes da execução da ação; e seção de ação, que contém os resultados das ações executadas nas páginas.

Com a tabela de decisões, os casos de teste são produzidos com as técnicas de grafo de causa e efeito e/ou pelo particionamento de equivalência das condições de entrada, como explicados na Seção 2.3.1. Para o teste de integração, Lucca et al. (2002) propõem os seguintes passos:

1. De acordo com o modelo de teste, é construído um grafo cujos nós representam as páginas (cliente ou servidor) com seus componentes e os *links* representam as arestas, com associações entre os nós;
2. A cada *link* é associado um peso proporcional ao número de parâmetros que são passados de uma página a outra;
3. O grafo é transformado em um grafo acíclico (não contém ciclos);
4. É estabelecida uma ordem entre os nós e se realiza o teste de integração das páginas com base nesse grafo.

Liu et al. (2000a,b) propõem uma estratégia a ser realizada em cinco níveis, que utiliza o modelo WATM apresentado na Seção 2.3.1:

- **Nível de Função:** usado para testar funções individuais para as variáveis que têm associação def-uso limitado a uma única função;
- **Nível de *Cluster* de Funções:** usado para testar um conjunto de funções dentro de um objeto para as variáveis cuja associação def-uso envolve mais de uma função dentro do conjunto;
- **Nível de Objeto:** usado para testar um objeto que tem variáveis com associações def-uso que podem ser mudadas por diferentes sequências de invocação de funções;
- **Nível de *Cluster* de Objetos:** usado para testar um conjunto de objetos cujas variáveis e associações def-uso atravessam os objetos do conjunto;

- **Nível de Aplicação:** usado para testar as aplicações cujas variáveis e associações def-uso estão entre os vários clientes que podem acessar a aplicação.

Em Xu et al. (2005), os testes de aplicações *web* são realizados com o foco nas especialidades da *web*, como os numerosos usuários, as estruturas distribuídas e as funções dinâmicas e interativas.

Inicialmente, é feito o modelo da aplicação *web*, mostrado na Figura 2.4, que representa os elementos da aplicação. Esse modelo é composto de: **modelo de objeto**, que descreve os atributos e operações de um único objeto; **modelo de relações interativas**, que representam as associações entre os objetos do modelo; e o **modelo de arquitetura**, que descreve as estruturas e conteúdos da aplicação inteira.

Figura 2.4: Modelo de Testes de uma Aplicação Web
(Xu et al., 2005)

A partir desses três sub-modelos são realizados os seguintes testes:

- **Testes de unidade**, com a utilização do modelo de objetos e as informações de nome de variáveis, tipo e escopo de valor dos objetos a serem testados;
- **Testes de propriedades dinâmicas**, que utilizam o modelo de interação para testar as interações entre os objetos do modelo;
- **Testes de integração**, que utiliza o modelo de arquitetura para realizar testes como o de visitas concorrentes de usuários e para testar a compatibilidade entre os servidores e clientes da aplicação

Outras estratégias para conduzir os testes de aplicações *web* como em Elbaum et al. (2003); Lucca e Penta (2003); Sneed (2004) são propostas. Cada uma apresenta maneiras diferentes para realizar de forma sistemática os testes, mas todas com o objetivo de fazê-los

de uma forma bem planejada e alcançar os objetivos de obter uma aplicação com maior qualidade.

Para auxiliar na condução das atividades de teste, além das técnicas, critérios e estratégias, é necessária a utilização de ferramentas que automatizem o processo de testes. Na próxima seção, algumas ferramentas existentes para efetuar os testes das aplicações *web* são citadas.

2.3.3 Ferramentas de Teste

As atividades de teste são muito propensas a erros, e se forem conduzidas de forma manual, se tornam improdutivas (Maldonado et al., 2004). Sendo assim, a efetividade de um processo de teste pode depender bastante de ferramentas que sejam utilizadas para apoiar o processo (Lucca et al., 2002).

As ferramentas de teste, normalmente, automatizam algumas tarefas exigidas durante o processo, como a geração e execução dos casos de teste e a avaliação de seus resultados, complementando os esforços manuais de teste (Lucca et al., 2002; Nguyen, 2001).

Alguns tipos de teste, como o teste de desempenho, seriam impraticáveis sem o auxílio de ferramentas que poderiam simular as ações de milhares de usuários. O valor de cada ferramenta varia de acordo com a necessidade, cronograma e restrições associadas ao sistema sob teste (Nguyen, 2001).

Além disso, os benefícios da utilização das ferramentas podem auxiliar na condução dos testes de regressão, pois cada caso de teste pode ser reexecutado automaticamente a cada nova versão da aplicação *web* gerada e os resultados podem ser comparados com os obtidos dos testes anteriores (Ricca e Tonella, 2001).

Abaixo serão listados alguns tipos de ferramentas de teste para aplicações *web* identificados, bem como alguns exemplos delas (Nguyen, 2001).

Teste de Navegabilidade e Validadores de HTML: são analisadores estáticos. Verificam erros em tags HTML, a compatibilidade de navegadores, e a existência de *links* incorretos dentro de uma determinada aplicação. Exemplos de Ferramentas: Watchfire Linkbot (Watchfire, 2006), ParaSoft SiteRuler (ParaSoftbreak, 2006).

Teste de Desempenho e Carga: essas ferramentas geram scripts de teste por meio da simulação de acesso de vários usuários a uma determinada aplicação. Assim, obtêm parâmetros de execução nas páginas do servidor para assegurar que a resposta é efetuada em tempo aceitável. Dessa forma, é determinado como os dados respondem a velocidade de conexão, tempo de carga, intensidade de carga e uso contínuo. Exemplos de Ferramentas: Envive Prophecy (Envive, 2006), Mercury Interactive LoadRunner (Mercury Interactive, 2006), RadView WebLoad (RadView, 2006), Rational Performance Studio (Rational, 2006a).

Testes de Funcionalidade e Regressão: asseguram a corretude funcional da aplicação de acordo com as especificações. Realizam a verificação de sintaxe HTML, fazem os testes de *link*, de formulário, de *applets*, *scripts*, componentes, etc. O teste de regressão é necessário para verificar se as modificações efetuadas na aplicação afetaram sua funcionalidade. Exemplos de Ferramentas: JStudio SiteWalker (JStudio, 2006), jWebUnit (JWebunit, 2006).

Detecção de Erros de Execução: durante a execução do programa, essas ferramentas procuram por operações inválidas e/ou errôneas que são requisitadas pela aplicação, como erros de leitura, falhas de memória, entre outros. Exemplos de Ferramentas: MicroQuill HeapAgent (MicroQuill, 2006), ParaSoft Insure++ (ParaSoft, 2006), Rational Purify (Rational, 2006b), Compuware BoundsChecker (Compuware, 2006).

Teste de Segurança de Sites Web: detectam e analisam as características de segurança da rede ou de uma aplicação web. Exemplos de Ferramentas: NetScan Tools (Northwest Performance Software, Inc., 2006), Surfincheck Firewall (Finjan Software, 2006).

Além dessas, existem outras ferramentas de teste relacionadas a aplicações *web*, como ferramentas de teste de interfaces gráficas (*Segue SilkTest* (Segue, 2006), *Rational VisualTest* (Rational, 2006c), *eTester* (RSW Software, 2006)), de análise de logs (*Deep Log Analyzer* (DeepSoftware, 2006) e *HTTPD Log Analyzers* (UPPSALA University, 2006)), entre outros, que podem auxiliar as atividades de teste dessas aplicações.

2.4 Ontologias

Ontologia é uma especificação formal e explícita de uma conceitualização compartilhada (Gruber, 1993). Um conceitualização se refere a um modelo abstrato de algum fenômeno no mundo, com o objetivo de identificar os conceitos relevantes deste fenômeno. Ser explícito significa que o tipo de conceitos usados e as suas restrições de uso são definidos explicitamente. Formal se refere ao fato de que a ontologia deve ser legível pelo computador (Fensel, 2003).

As ontologias podem ser usadas como artefatos de engenharia, constituídos por um vocabulário específico, com um conjunto de suposições referentes ao significado dos termos do vocabulário. Usualmente, esse conjunto de suposições tem a forma de uma teoria lógica de primeira ordem, no qual os termos do vocabulário aparecem como nomes de predicados unários ou binários, chamados de conceitos e de relações, respectivamente (Barbosa et al., 2006). Sendo especificações formais e consensuais que oferecem um entendimento compartilhado de um domínio, a ser comunicado por pessoas e sistemas, elas trazem dois importantes benefícios (Fensel, 2003):

- a definição de semânticas formais da informação, permitindo o processamento da informação pelo computador; e
- a definição de semânticas do mundo real, que torna possível relacionar conteúdo processável pelo computador com significado para seres humanos baseado em terminologias consensuais.

A seguir, são apresentadas algumas características das ontologias e suas possíveis utilizações, bem como uma ontologia para o domínio de teste de software.

2.4.1 Características e Uso de Ontologias

As ontologias oferecem um entendimento compartilhado sobre domínios específicos, que podem ser transmitidos entre pessoas e sistemas de aplicação. São estruturas formais que apóiam o compartilhamento do conhecimento e o seu reuso. Dessa maneira, elas podem ser usadas para representar explicitamente a semântica de informações estruturadas e semi-estruturadas, capacitando o apoio automatizado para adquirir, manter e acessar a informação, ou seja, facilitando a gerência do conhecimento (Fensel, 2003).

Em Gruber (1993) é proposto um conjunto preliminar de critérios para projeto de ontologias, a fim de determinar que características as ontologias devem ter para facilitar o compartilhamento e interoperabilidade de aplicações que as utilizam:

- **Clareza:** uma ontologia deve comunicar efetivamente o significado dos termos definidos, de maneira objetiva;
- **Coerência:** uma ontologia deve ser coerente, ou seja, deve expressar inferências que são consistentes com suas definições;
- **Extensibilidade:** uma ontologia deve ser projetada a fim de antecipar o uso de vocabulário compartilhado;
- **Mínimo compromisso ontológico:** para permitir que sejam definidas tão poucas suposições quanto possíveis sobre o mundo a ser modelado, permitindo que as especializações e instanciações da ontologia sejam definidas com liberdade; e
- **Mínima influência de codificação:** a conceitualização deve ser especificada sem depender de uma codificação particular. A dependência deve ser minimizada para poder ser implementada com diferentes sistemas e estilos de representação.

Segundo Guarino (1997) as ontologias podem ser descritas em três diferentes níveis de abstração, dependendo de sua aplicabilidade, sendo classificadas em:

- **Ontologia de Nível Topo:** descreve conceitos gerais que são independentes de um domínio específico;
- **Ontologia de Domínio e de Tarefa:** descreve respectivamente os conceitos relacionados a um domínio específico, ou uma tarefa genérica ou atividade por especialização dos termos definidos na ontologia de nível de topo;
- **Ontologia de Aplicação:** descreve os conceitos dependendo de uma ontologia de domínio e de uma ontologia de tarefa particulares, que são muitas vezes especializações das duas ontologias relacionadas (domínio e tarefa).

Para a formalização de uma ontologia, Gruber (1993) definiu cinco componentes: *conceitos*, que podem representar qualquer coisa em um domínio, como uma tarefa, uma função, uma estratégia, etc.; *relações*, que representam um tipo de interação entre os conceitos no domínio, sendo a cardinalidade sempre n:n; *funções*, que são um caso especial de relações, sendo a cardinalidade n:1; *axiomas*, que são as sentenças que são sempre verdadeiras; e *instâncias*, utilizadas para representar os elementos do domínio.

De acordo com Duarte e Falbo (2000), as ontologias possuem os seguintes propósitos:

- refletir e melhorar a compreensão sobre o domínio, para as pessoas envolvidas no processo de desenvolvimento de uma ontologia;
- auxiliar para a obtenção de um consenso no entendimento sobre uma área de conhecimento, considerando que em geral, para uma área de conhecimento, diferentes especialistas têm entendimento diferenciado sobre os conceitos envolvidos, o que leva a problemas na comunicação. Ao se construir uma ontologia, essas diferenças são explicitadas e busca-se um consenso sobre seu significado e sua importância;
- uma vez que haja uma ontologia sobre uma determinada área de conhecimento desenvolvida, uma pessoa que deseje aprender mais sobre essa área não precisa se reportar sempre a um especialista: ela pode estudar a ontologia e aprender sobre o domínio em questão.

Segundo Swartout et al. (1997), uma ontologia é a estrutura fundamental para uma base de conhecimento, já que pode ser considerada como um conjunto de termos hierarquicamente estruturados. Sendo assim, duas bases de conhecimento que são construídas a partir de uma ontologia podem se comunicar mais facilmente, por compartilharem o mesmo “entendimento” dos conceitos utilizados.

As ontologias têm sido aplicadas para descrever uma variedade de domínios de conhecimento, como em medicina, engenharia e direito (grubner). No campo da engenharia de

software, algumas ontologias podem ser identificadas, como em (April et al., 2006; Balushi et al., 2006; Falbo et al., 1997).

Em Duarte e Falbo (2000), é apresentada uma ontologia para a área de qualidade de software, com o objetivo de apoiar o entendimento dos conceitos dessa área e, consequentemente, a definição de programas de qualidade e a construção de ferramentas de apoio. Balushi et al. (2006) definem uma ontologia de qualidade, com o objetivo de ajudar a assegurar que requisitos de qualidade, relacionados a esse domínio, sejam especificados durante a elicitacão de requisitos. Em April et al. (2006), é mostrado o desenvolvimento de um sistema baseado em conhecimento que auxilie a encontrar as melhores práticas de um modelo de maturidade de manutenção de software com o uso de uma ontologia relacionada a essa área.

Em Villela et al. (2001), é mostrado o ambiente ADSOrg, com o propósito de possibilitar que desenvolvedores de software encontrem, dentro da estrutura organizacional, os profissionais mais adequados à realização de uma atividade ou à solução de um problema. Esse ambiente foi desenvolvido com o uso de ontologias de organização, do domínio e de Engenharia de Software para descrever as organizações que desenvolvem e mantêm software.

A ontologia de organização fornece os conceitos e atributos relacionados à estrutura e aos processos organizacionais, possibilitando a descrição da organização e permitindo especificar, por exemplo, o contexto em que um item de conhecimento ou habilidade é necessário. A ontologia do domínio define o vocabulário para representar o conhecimento de um domínio de aplicação, expressando conceitos e fatos deste domínio. Por fim, a ontologia de Engenharia de Software define o vocabulário que orienta o registro e a distribuição de conhecimento sobre engenharia de software no ADSOrg (Villela et al., 2001).

O uso de ontologias pode auxiliar no desenvolvimento de sistemas baseados em conhecimento. Considerando o objetivo deste trabalho, para o desenvolvimento de um portal com conhecimentos relacionados a área de teste de aplicações, é importante o uso de uma ontologia que defina os conceitos existentes para essa área. Para isso, foi estudada a OntoTest, descrita em Nakagawa (2005) e Barbosa et al. (2006), ontologia desenvolvida por pesquisadores do ICMC-USP/SC com o objetivo de conceitualizar e formalizar os termos e relacionamentos existentes sobre teste de software. Essa ontologia é abordada na próxima seção.

2.4.2 Ontologia de Teste - OntoTest

A OntoTest foi desenvolvida pelos pesquisadores do ICMC/USP, para dar apoio à aquisição, organização, reuso e compartilhamento de conhecimento do domínio de Teste de Software Nakagawa (2005).

No nível mais alto da ontologia, denominado *Ontologia Principal de Teste de Software*, são definidos os principais conceitos e suas relações, associados ao teste de software, conforme a Figura 2.5.

Figura 2.5: OntoTest (Estrutura Principal)

Nos sub-níveis, os conceitos específicos da ontologia principal são refinados e tratados com mais detalhes (*processo de teste*, *fase de teste*, *procedimento de teste*, *passo de teste*, *artefato de teste* e *recurso de teste*). É possível perceber que a estrutura da OntoTest é flexível para o reuso e para ser integrada. Dessa maneira, dependendo da aplicação, ela pode ser usada completa, somente com o nível principal, ou com algum de seus sub-níveis. As sub-ontologias são descritas a seguir.

1. **Processo de Teste** (*TestingProcess*): os processos de teste são baseados em uma tecnologia de desenvolvimento e um paradigma de desenvolvimento, além de um procedimento de teste. Modelos de ciclo de vida de teste são usados como referência na definição de um processo de teste, estabelecendo macro passos e a relação de dependência entre eles;
2. **Fase de teste** (*TestingPhase*): os testes podem ser divididos em três fases incrementais: unidade de teste, que explora cada unidade separadamente para assegurar que os aspectos algorítmicos estão corretamente implementados; testes de integração, que identificam falhas relacionadas a interfaces entre as unidades; e os testes de sistema, que têm a finalidade de assegurar que o software e os outros elementos do sistema estão adequadamente combinados e as suas funcionalidades e desempenho estão de acordo com a especificação. Além disso, os dados coletados durante os testes também regem uma regra fundamental para manutenção. Os testes de regressão (reteste do software que foi modificado) pretendem assegurar que as modificações

ainda estão de acordo com os requisitos especificados e os códigos não modificados não foram afetados;

3. **Procedimento de Teste** (*TestingProcedure*): podem ser categorizados em métodos, guias e técnicas. As técnicas, por exemplo, são investigadas para realizar os testes de uma maneira sistemática de acordo com bases teóricas. A idéia é selecionar sub-testes de um domínio de entrada, preservando a probabilidade de falhas não cobertas. Exemplos de técnicas são funcionais, estruturais, baseadas em erros e baseada em estados. Cada técnica primária estabelece um ou mais critérios de teste, que podem ser categorizados como critérios de seleção ou critérios de adequação. Diferentes estratégias podem ser estabelecidas dependendo dos procedimentos de teste, passos de teste e fases consideradas;
4. **Passos de teste** (*TestingStep*): quando um teste é realizado, um conjunto de passos essenciais devem ser considerados: os testadores devem ser capazes de planejar os testes, construir os casos de teste, executar os testes, e analisar e avaliar os resultados do teste. Além disso, o passo de teste pode ser visto como uma composição de atividades de teste;
5. **Artefato de teste** (*TestingArtifact*): um artefato de teste pode ser visto como um artefato de entrada ou de saída, dependendo de como for usado (consumido ou produzido por um passo de teste). Além disso, cada passo de teste pode envolver um número de diferentes artefatos, como documentos de teste, diagramas de teste, casos de teste, requisitos de teste, drivers e stubs, além de artefatos sob teste;
6. **Recurso de teste** (*TestingResource*): de acordo com as sub-ontologias anteriores, um passo de teste pode ser visto como uma primitiva de transformação, onde as entradas e saídas do passo de teste correspondem a um artefato de teste. Entretanto, outros elementos são necessários para a execução de um passo de teste, denominados recursos de teste. Esse recurso pode ser um recurso humano, como um testador ou um gerente de teste; um recurso de hardware ou de software, como uma ferramenta de teste ou um sistema de apoio. Recursos de hardware e software são caracterizados em termos de um ambiente de teste, que pode ser usado para automatizar os procedimentos de teste. As ferramentas de teste, que oferecem apoio automatizado para realizar os testes são um tipo especial de recursos de teste, classificadas em primárias, organizacionais e de apoio, além de ferramentas de propósito geral.

Essas sub-ontologias estão representadas, respectivamente, pelas Figuras 2.6, 2.7, 2.8, 2.9 e 2.10. Mais detalhes dos conceitos, bem como os axiomas que definem os relacionamentos entre os conceitos e restrições dessa ontologia estão descritos em Barbosa et al. (2006).

Figura 2.6: Sub-Ontologia de Processos de Teste

Figura 2.7: Sub-Ontologia de Procedimentos de Teste

Figura 2.8: Sub-Ontologia de Passos de Teste

Figura 2.9: Sub-Ontologia de Artefatos de Teste

Figura 2.10: Sub-Ontologia de Recursos de Teste

2.5 Considerações Finais

Os portais têm uma série de especificidades que os tornam apropriados para disseminar o conhecimento em uma área. Para desenvolver um portal, é importante atender a um conjunto mínimo de características que satisfaçam às necessidades do público que irá utilizá-lo. Dessa maneira, é importante a definição do público alvo e a definição de que informações, para uma área de interesse, serão armazenadas.

Diversas iniciativas para a adequação de critérios, estratégias para o teste de aplicações *web* vêm sendo propostas. Além disso, pode-se observar o desenvolvimento de ferramentas que permitem apoiar e complementar os esforços de teste.

Com a relevância dos testes de aplicações *web*, é importante haver formas de armazenamento do conteúdo disponível, para facilitar o acesso e a aquisição dessas informações, e para tanto, é realizada a proposta e implementação do *Portal de Conhecimento sobre Critérios, Estratégias, Técnicas e Ferramentas de Teste de Aplicações Web* que reúne o conhecimento adquirido nessa área.

Para tanto, a compreensão do domínio de teste precisa ser representada por um formalismo declarativo, com a existência de um conjunto de objetos, denominados *universo de discurso*, e suas relações descritas e refletidas num vocabulário com o qual os sistemas podem representar esse conhecimento (Gruber, 1993).

Essa representação, realizada por meio de ontologias, permite que o conhecimento possa seja manipulado, auxiliando na padronização, comunicação e interoperabilidade, tanto para agentes humanos como para agentes computacionais, auxiliando no desenvolvimento de sistemas de conhecimento.

O uso de uma ontologia para determinado domínio auxilia na compreensão de seus conceitos e relacionamento, e além disso, como ela é representada de maneira formal, seguindo regras claras de definições desse conceitos e relacionamentos, ela pode ser utilizada como fundamento para o desenvolvimento de uma base de dados.

Para o domínio de teste de software, a OntoTest, definida em Barbosa et al. (2006) define os conceitos da área, com os axiomas e instâncias destes conceitos, podendo ser aplicada em sistemas que pretendam efetuar a gerência adequada do conhecimento existente nessa área.

No próximo capítulo, são mostradas as principais características, requisitos, modelagem e arquitetura definida para a implementação do portal.

Portal de Conhecimento: Definição de Requisitos, Modelagem e Arquitetura

3.1 Considerações Iniciais

Para apoiar a busca de conhecimento sobre os testes de aplicações *web* e sua constante evolução, é necessária a disponibilização de um ambiente com funcionalidades que permitam aos usuários encontrar, armazenar e gerar novos conhecimentos. Sendo assim, o objetivo principal deste capítulo é mostrar o desenvolvimento do **Portal de Conhecimento de Ferramentas, Estratégias, Técnicas e Critérios de Teste de Aplicações Web**, que possui módulos que auxiliam no ciclo de criação, troca, retenção e reuso do conhecimento.

Neste capítulo são mostradas as definições de requisitos e do ambiente definido para o Portal. Na Seção 3.2 é mostrada a definição do ambiente. Na Seção 3.3, são mostrados os estudos realizados para a seleção do processo de desenvolvimento do Portal. Na Seção 3.4 os requisitos definidos para o desenvolvimento do Portal são discutidos. Na Seção 3.5 é mostrada a modelagem do portal. Na Seção 3.6 é mostrada a arquitetura do Portal de Conhecimento, e, por fim, na Seção 3.7 são feitas as considerações finais deste capítulo.

3.2 Definição do Ambiente

O Portal de Conhecimento em Testes de Aplicações Web tem como objetivo principal fornecer à comunidade acadêmica (pesquisadores e estudantes) e à indústria (empresas de desenvolvimento de software) um conjunto de conhecimento obtido acerca dos testes de aplicações *web*, como ferramentas, técnicas e critérios de teste que possam auxiliar para execução das atividades nessa área.

O Portal deve permitir que os usuários incluam informações a respeito de ferramentas, técnicas e critérios de teste de aplicações *web* e possam realizar consultas sobre o que existe no portal. Além disso, o Portal deve permitir a colaboração entre os usuários do sistema para que eles possam incluir suas observações acerca da utilização desses elementos e estabelecer meios de comunicação para que os usuários discutam entre si os assuntos relevantes nesse cenário.

O ambiente do Portal é constituído por cinco módulos: *módulo de conhecimento*, *módulo de colaboração*, *módulo de comunicação*, *módulo de consulta* e *módulo de administração*.

- **Módulo de Conhecimento:** o módulo de conhecimento está relacionado à OntoTest apresentada na Seção 2.4.2. Nesse módulo, as informações sobre processos, técnicas, critérios e estratégias de teste são cadastradas, seguindo a estrutura definida pela ontologia. Dessa forma, o usuário armazena os dados de maneira relacionada, havendo uma hierarquia e associação entre os objetos que deve ser respeitada para facilitar a aquisição do conhecimento;
- **Módulo de colaboração:** nesse módulo, os usuários inserem dados que são associados a cada um dos processos existentes na base de dados do portal. Esses dados, denominados “Material de Apoio”, contêm informações sobre o uso do processo e atividades realizados pelos usuários, como artigos em que o processo foi mencionado, casos de uso de ferramentas, manuais de procedimentos, pacotes de laboratório referentes aos estudos experimentais de ferramentas de teste, entre outros. Esse módulo permite que os usuários possam conhecer melhor cada um dos elementos da OntoTest, já que reporta maiores informações sobre cada um por meio do material relacionado;
- **Módulo de comunicação:** este módulo permite a comunicação entre os usuários do portal, viabilizando o acesso aos especialistas no domínio de teste de software, e aos demais usuários do sistema, como forma de fomentar e estimular a utilização dos recursos oferecidos pelo portal;

- **Módulo de Consulta:** neste módulo, os usuários têm acesso às informações existentes na Base de Dados do Portal, podendo executar as buscas sobre o conhecimento desejado de acordo com chaves de busca para recuperar os itens que foram cadastrados, tanto pro meio do módulo de conhecimento como por meio do módulo de colaboração. Além disso, eles podem realizar as buscas de informações presentes no módulo de comunicação para recuperarem dados informais presentes nas trocas de informações entre os usuários do Portal realizadas nesse módulo;
- **Módulo de Administração:** inclui as funcionalidades necessárias para as tarefas administrativas, como a gerência de usuários e operações de manutenção do portal.

A integração desses módulos permite que o usuário acesse o universo de conhecimento existente no portal referente ao domínio dos testes de software. Para o acesso, foi realizada uma distinção entre os perfis dos usuários, que possuem permissões de acesso diferenciadas, segundo esse perfil. Para isso, os usuários são identificados em quatro grupos distintos:

- **Visitante:** usuário que acessa o portal para obter informações sobre o que ele contém, e, possivelmente, solicitar o cadastro para o acesso aos módulos internos do portal.
- **Usuário Cadastrado:** usuário que já tem o cadastro no Portal e tem acesso ao módulo de colaboração e módulo de comunicação do Portal, podendo, portanto, ver as informações existentes e fazer a comunicação com os demais usuários do Portal.
- **Especialista:** esse usuário tem permissão para acessar os módulos de conhecimento, colaboração e comunicação, podendo, portanto, adicionalmente ao usuário cadastrado, incluir as informações sobre ontologias de processos de teste.
- **Administrador:** esse usuário pode acessar o módulo de administração do Portal.

Todos os usuários têm acesso ao módulo de consulta. A partir da definição dos módulos que compõem o Portal, é possível realizar o seu desenvolvimento, seguindo um processo que possibilite uma abordagem sistemática para sua construção. Na próxima seção é mostrado o processo de desenvolvimento adotado para a construção do portal.

3.3 Processo de Desenvolvimento

Para construir o portal, é necessário seguir um processo de desenvolvimento sistematizado, a fim de que ele seja implementado de forma a atender aos objetivos aos quais ele foi proposto.

Nesta seção, são mostrados dois estudos realizados com o objetivo de identificar processos de desenvolvimento existentes na literatura, a fim de definir qual o processo mais adequado para ser utilizado no desenvolvimento do *Portal de Conhecimento*:

(i) Revisão sistemática para identificar processos que fossem utilizados especificamente para portais web; e (ii) Estudo dirigido para identificar processos e métodos de desenvolvimento mais comumente utilizados em desenvolvimentos de aplicações web.

Além disso, é mostrada a comparação entre os resultados do estudo e o processo selecionado a partir dessa comparação.

3.3.1 Revisão sistemática - Processo de Desenvolvimento de Portais Web

Uma revisão sistemática é um meio de identificar, avaliar e interpretar as pesquisas disponíveis e relevantes direcionadas a uma questão de pesquisa em particular, tópico de uma área ou um fenômeno de interesse, com um grau de rigor maior em sua condução. Ao contrário das revisões convencionais da literatura, onde o pesquisador não segue um processo definido para sua condução, uma revisão sistemática é executada de maneira formal de acordo com uma estratégia de pesquisa bem definida (Kitchenham, 2004).

Tendo em vista a adoção de um processo de desenvolvimento de portais *web* como forma de sistematizar o desenvolvimento do portal e obter maior qualidade no produto final, foram revisados os processos de desenvolvimento de portais *web* aplicados para seu desenvolvimento, por meio de uma revisão sistemática.

Segundo Kitchenham (2004), uma revisão sistemática deve seguir um protocolo com um conjunto de passos que estão divididos em três fases principais: o *planejamento*, em que é identificada a necessidade de se conduzir a revisão e é desenvolvido o protocolo de revisão que será adotado; a *condução*, em que é realizada a identificação da pesquisa, a seleção dos estudos primários, a avaliação da qualidade desses estudos, a extração dos dados existentes nos estudos e a síntese dos dados; e por fim, o *relato* da revisão, em que os resultados obtidos são comunicados para seu uso efetivo.

No planejamento da revisão sistemática de desenvolvimento de portais *web*, foi identificada a necessidade de se conduzir um levantamento dos processos de desenvolvimento de portais citados na literatura técnica, para se obter um panorama geral de quais os processos e/ou abordagens de desenvolvimento utilizados para a construção de portais em diversos domínios de aplicação. Para o protocolo de revisão desenvolvido durante essa fase, foi seguida a estrutura definida em Kitchenham (2004), conforme mostrado a seguir:

1. Formulação da questão:

- (a) Foco: Em vista da crescente popularização dos portais, é necessário que se tenha um panorama geral de quais os processos e/ou abordagens de desenvolvimento utilizadas para a construção de portais em diversos domínios de aplicação.
- (b) Qualidade e Amplitude
 - Problema: Adoção de um processo de desenvolvimento de portais *web* como forma de sistematizar o desenvolvimento e obter maior qualidade no produto final;
 - Questão: Quais os processos de desenvolvimento de portais *web* aplicados no contexto do desenvolvimento desse tipo de aplicação?
 - Palavras chave e sinônimos
 - Web portal, portal system;
 - Development, project, engineering;
 - Process, method, methodology, technique, approach, framework.
 - Intervenção: Processo de desenvolvimento de portais;
 - Controle: Para essa revisão sistemática, será utilizado como controle o subconjunto de processos de desenvolvimento de portais *web* identificados anteriormente por meio de revisões *ad hoc* de teses, dissertações e artigos;
 - Resultados: Diferentes abordagens para desenvolver portais;
 - População: Projetos de desenvolvimento de portais;
 - Aplicação: Identificar e selecionar processos de desenvolvimento de portais *web* para utilização de um deles para o desenvolvimento do portal de testes e ferramentas de teste de aplicações *web*;
 - Abordagem Experimental: Será considerado qualquer tipo de abordagem experimental que tiver sido empregada para justificar a adoção do processo de desenvolvimento utilizado.

2. Seleção de fontes

- (a) Definição do critério de seleção de fontes
 - Máquinas de busca automáticas com consulta de artigos via *web*;
 - Buscas por palavras chave e com possibilidades de combinação de palavras.
- (b) Linguagens: O inglês será utilizado para os estudos por conta de sua universalidade.
- (c) Identificação das fontes
 - Métodos de Busca: buscas automáticas via *web*;

- *String* de Busca
 - (“*portal*”) and (“*development*” or “*engineering*” or “*software*”) and (“*process*” or “*method*” or “*methodology*” or “*technique*” or “*approach*” or “*project*” or “*framework*”).
 - Adaptação da *string* de busca por sintaxe e limitações de cada máquina de busca;
 - As strings de busca devem estar presentes no título ou no resumo do artigo.
- Lista de Fontes
 - Kluwer journals;
 - IEEE conferences, IEEE journals;
 - Elsevier journals;
 - ACM conferences, ACM journals.

3. Seleção de estudos

(a) Definição dos estudos

- Critérios de Inclusão e Exclusão
 - Textos completos disponíveis na *web*;
 - Os artigos devem aplicar ou referenciar o processo utilizado para o desenvolvimento do portal.
- Avaliação da qualidade dos estudos: os artigos devem apresentar a descrição detalhada ou uma referência ao processo de desenvolvimento em cada uma das fases propostas.
- Definição dos tipos de estudos
- Procedimentos para seleção de estudos
 - Execução das strings de busca em cada máquina de busca;
 - Documentação dos artigos encontrados;
 - Seleção inicial dos estudos com a leitura de resumos/abstracts e introdução para incluir ou excluir o estudo com base nos critérios de inclusão e exclusão;
 - Documentação dos artigos incluídos e excluídos com justificativa;
 - Leitura dos textos completos dos artigos incluídos;
 - Avaliação da qualidade dos estudos com base no critério de qualidade estabelecido;

- Revisão da seleção dos estudos, com a revisão incremental a partir de revisões parciais que procurem priorizar os estudos por meio de índices de citação e avaliação da qualidade dos estudos.
- (b) Estratégia de extração de informação: Os estudos selecionados durante a execução do processo de avaliação da qualidade dos estudos primários terão suas informações extraídas com os itens do Formulário de Extração de Dados.
- (c) Sumarização dos resultados: Os resultados serão tabulados. Não será realizada nenhuma meta-análise.

A partir desse protocolo de revisão, a execução da revisão sistemática de processo de desenvolvimento de portais *web* foi conduzida nas máquinas de busca escolhidas. Obteve-se, então, o primeiro conjunto de artigos, para que fosse realizada a fase de seleção inicial, em que eram lidos os *abstracts* de cada artigo. A partir dessa leitura, foi reduzido o conjunto inicial, e foram lidos os artigos completos, para avaliação da relevância e aderência ao tema de pesquisa, com a seleção final dos artigos dos quais seriam extraídos os dados. Os resultados obtidos estão sumarizados na Tabela 3.1.

Tabela 3.1: Resultados da Revisão Sistemática

ID	Referência	Título
1	Gruhn et al. (2002)	Development of an electronic commerce portal system using a specific software development process
2	Hazra (2002)	Building enterprise portals: principles to practice
3	Pettersson e Jarzabek (2005)	Industrial experience with building a <i>web</i> portal product line using a lightweight, reactive approach
4	Corcho et al. (2006)	A platform for the development of semantic <i>web</i> portals
5	Tahboub et al. (2005)	Software engineering and database Web management in Daleelcom design
6	Galligan et al. (2005)	An approach to deliver an early return on investment during the development of a corporate Web portal
7	Li et al. (2005)	A collaborative application portal for the mould industry
8	Fernandes et al. (2005)	Portals as a knowledge repository and transfer tool—VIZCon case study

Com os artigos selecionados, foram extraídas as fases de cada um dos processos de desenvolvimento de portais *web* encontrados. As fases de cada um dos processos estão representadas na Tabela 3.2.

A revisão sistemática completa, com o detalhamento da seleção e execução da revisão, bem como a descrição das fases contidas em cada processo de desenvolvimento citado nas tabelas anteriores será detalhada no Relatório Técnico intitulado “*Processos de Desenvolvimento de Portais Web: Uma Revisão Sistemática*” (Costa e Maldonado, 2008) para melhor análise das informações encontradas durante a revisão.

Tabela 3.2: Resultados da Revisão Sistemática

ID	Fase 1	Fase 2	Fase 3	Fase 4	Fase 5	Fase 6	Fase 7
1	Analisar Requisitos	Selecionar Subsis-temas	Desenvolver Protótipo	Desenvolver Interface	Integrar e testar sistema		
2	Descoberta	Análise de Gaps	Plano de mediação	Solução			
3	Definição Funcional	Definição do Portal	Codificação e Testes				
4	Modelar o domínio	Identificar Tipos de Usuário	Identificar Conteúdo	Identificar Mapa do Site	Implementar Visualiza-ção		
5	Análise de requisitos	Projeto e Implemen-tação	Testes e Checagem de segu-rança				
6	Análise de requi-sitos com usuário	Desenvolver proposta de solução	Desenvolver protótipo	Implementa-ção			
7	Análise de Requisitos	Projeto de arquitetura	Seleção de softwares universais	Desenvolver Programas e Interface	Preencher certificação segura e gerência de usuário	Teste de Sistema	Implementar
8	Definição	Seleção de Funcionali-dade	Seleção de Conteúdo	Estabelecer Relaciona-mento	Validação		

3.3.2 Estudo dirigido - Processo de Desenvolvimento de Aplicações Web

O estudo dirigido de métodos de desenvolvimento de aplicações *web* identificou as fases utilizadas por esses métodos, que pudessem ser aplicadas no desenvolvimento do portal *web*, considerando o fato de que tanto os portais quanto as aplicações *web* possuem como característica estarem sobre a plataforma WWW.

De modo geral, os métodos atuais de desenvolvimento *web* estendem métodos tradicionais de desenvolvimento de software com a adição de algum modelo de navegação, que especifica as características navegacionais de uma aplicação Web. Outro procedimento utilizado é a adaptação de linguagens de modelagem extensíveis, como a UML, ao contexto das aplicações Web. Esses modelos definem visões navegacionais de um sistema e relacionam-se aos grupos de usuários de determinada aplicação (Bianchini, 2006).

Considerando que existe ampla variedade de métodos de desenvolvimento de aplicações *web* (HDM - *Hypermedia Design Model* (Garzotto et al., 1993), OOHDM - *Object Oriented*

Hypermedia Design Method (Rossi, 1996), UWE - *UML-based Web Engineering* (Koch, 2000), OO-H - *Object-oriented hypermedia* (Gómez et al., 2001), WebML - *Web Modelling Language* (Ceri et al., 2000), HMBS/M - *Hypertext Model Based on Statecharts/Method* (Carvalho, 1998; Carvalho et al., 1999), WAE - *Web Application Extension* (Conallen, 2002), OOWS - *Object-Oriented Web Solution* (Fons et al., 2003), W2000 (Baresi et al., 2001)), foi necessário realizar uma escolha de quais seriam estudados para serem utilizados no escopo deste trabalho. Essa escolha foi feita com base no estudo de Gu et al. (2002), realizado para comparar métodos de desenvolvimento de aplicações *web*.

Segundo Gu et al. (2002), um método deve considerar dois aspectos de arquitetura técnica dessas aplicações: a arquitetura funcional e a arquitetura de informação. Para cada um desses aspectos, foram relacionadas características que podem ser adotadas pelos métodos existentes, ou por novos métodos. As características da arquitetura funcional de uma aplicação *web* impõem requisitos aos métodos utilizados, relacionados a uma determinada funcionalidade. As características da arquitetura da informação investigam aspectos de como o conteúdo é gerenciado, estruturado e disponibilizado; modos de visão da informação e questões de apresentação e navegação.

A partir dessas características, foi conduzida a comparação entre os seis métodos seguintes: OO-HDM (Schwabe e Pontes, 1998), WebML (Ceri et al., 2000), UWE (Koch, 2000; Koch e Mandel, 1999), WAE - extensões da UML por (Conallen, 2002), W2000 (Baresi et al., 2001) e HDM-Lite/Autoweb (Fraternali e Paolini, 2000). A comparação resumiu-se em uma lista de checagem, onde foi verificado se cada método abordava completamente, parcialmente ou não abordava as características mencionadas anteriormente, obtendo-se uma pontuação final para cada um dos métodos, dentro do aspecto funcional e dentro do aspecto de informação.

Com base na comparação de Gu et al. (2002), os métodos escolhidos para o estudo dirigido foram os que possuíam as notas mais significativas, sendo eles: *Web Application Extension* (WAE (Conallen, 2002)), *UML Extension for Modeling Web Application* (W2000 (Baresi et al., 2001)) e *Web Modeling Language* (WebML (Ceri et al., 2000)). Além disso, foi adicionado um método selecionado a partir de avaliação realizada no estudo de Bianchini (2006): *Object-oriented hypermedia* (OO-H). A seguir esses métodos serão brevemente explicados.

O **WAE** tem por objetivo estender a notação da UML para apoiar a modelagem das operações e interações de aplicações Web, ele pretende prover um guia sobre a ordem das atividades da equipe de desenvolvimento, especificar artefatos que devem ser desenvolvidos, dirigir as atividades de desenvolvedores individuais e da equipe como um todo, e oferecer critérios para medida e monitoramento dos produtos de projeto e atividades.

Esse método sugere que o processo para o desenvolvimento de aplicações *web* seja dirigido a casos de uso, com a utilização de uma arquitetura central, e realizado de maneira

iterativa e incremental. Além disso, que eles sejam adaptáveis à realidade de cada empresa. Os seguintes modelos são os recomendados por Conallen (2002) para o desenvolvimento de aplicações Web: domínio, casos de uso, análise/projeto, implementação, processo e segurança.

O **W2000** é baseado em dois elementos já existentes: UML, com características que permitem a especificação dos aspectos evolucionais e funcionais do software; e o HDM (Gómez et al., 2001), que provê a facilidade de organizar conteúdos altamente estruturados ou semi-estruturados de maneira não-linear e a facilidade de definir múltiplos caminhos de navegação. Seu projeto compreende cinco atividades: (i) *Análise de requisitos*; (ii) *Projeto de evolução de estados*; (iii) *Projeto de hipermídia*; (iv) *Projeto funcional*; e (v) *Projeto de visibilidade*.

Cada uma dessas atividades produz um modelo, o qual descreve alguns aspectos das aplicações *web*. Uma série dessas atividades pode ser executada em paralelo, e, eventualmente, os projetistas podem reconstruir diversas vezes o mesmo requisito por meio de refinamentos ou modificações para tornar o modelo mais próximo da especificação. Dessa forma, é possível: definir estereótipos e customizações de diagramas; especificar *guidelines* para usar a UML a fim de modelar aspectos dinâmicos e operacionais de aplicações *web*; e refinar os diagramas de casos de uso para descrever em alto nível os requisitos dos usuários, relacionados tanto aos aspectos funcionais quanto aos navegacionais.

O **OO-H** fornece um arcabouço padrão para a captura das propriedades envolvidas na modelagem e implementação de aplicações web. Basicamente, seu processo de modelagem consiste na adição de duas visões que complementam a modelagem conceitual tradicional de um método de desenvolvimento de software compatível com a UML, denominado *OO-Method* (Pastor et al., 1998), e a disponibilização de um ambiente de desenvolvimento automático. Nesse ambiente, são considerados no método OO-H um conjunto de visões que capturam a estrutura do sistema (elementos estáticos), seu comportamento (elementos dinâmicos) e um modelo de compilação que gera fonte de dados e módulos lógicos em um ambiente de implementação.

A **WebML** tem como objetivo oferecer uma notação para especificar aplicações *web* em nível conceitual, preenchendo o *gap* entre a coleta de requisitos e as fases subsequentes no processo de desenvolvimento de *web* sites. Ela oferece uma representação em alto-nível e independente de plataforma de uma aplicação *web*, não representando os detalhes arquiteturais da aplicação, sendo composta pelos seguintes modelos:

1. *Modelo Estrutural*: conteúdo de dados do site, em relação às entidades relevantes que compõem a aplicação e seus relacionamentos;

2. *Modelo de Hipertexto*: descreve um ou mais hipertextos que podem ser publicados no site. Cada hipertexto define uma visão do site (site view), que compreende dois submodelos:
 - *Modelo de composição*: especifica quais páginas compõem o hipertexto e quais unidades de conteúdo formam uma página. As unidades de conteúdo são entidades atômicas que representam um objeto dentro de um hipertexto. Podem ser de seis tipos: unidades de dados, unidades de multi-dados, unidades de índice, unidades de filtro, unidades de rolagem e unidades de direção. As duas primeiras estão relacionadas às informações de objetos, enquanto que as últimas representam modos alternativos de navegar entre o conjunto de objetos. Além disso, existem as unidades de composição, no topo da hierarquia do esquema de estrutura do site, que definem a entidade ou relacionamento no qual o conteúdo de cada unidade é baseado;
 - *Modelo de Navegação*: expressa como as páginas e as unidades de conteúdo estão ligadas para formar o hipertexto. No modelo de navegação são utilizados os links, que podem ser de dois tipos: contextual (baseado no esquema da estrutura), em que o link está associado (carrega) a informações da unidade fonte, sendo assim, o conteúdo da unidade destino depende do conteúdo da unidade fonte; e não-contextual, que independe do conteúdo das unidades fonte e destino.
3. *Modelo de Apresentação*: expressa o *layout* e o gráfico de aparência das páginas. Pode ser específico ou genérico;
4. *Modelo de Personalização*: entidades usuário e grupo que existem nas aplicações. O modelo expressa quais características dessas entidades são usadas para armazenar conteúdo específico pra o grupo ou de forma individualizada. Utilização da linguagem OQL para definir o conteúdo derivado da página baseado no perfil. O conteúdo personalizado pode ser usado na composição das unidades ou na definição das especificações da apresentação.

Com o estudo desses quatro métodos, foram identificadas as fases pertencentes a cada um, sumarizadas na Tabela 3.3.

Os estudos mostrados nesta seção buscaram abranger diferentes processos, tanto no contexto de portais *web* quanto no contexto de aplicações *web*, para que fosse realizada uma análise mais aprofundada nos processos, com a posterior comparação de cada um deles a fim de que um deles fosse adotado no desenvolvimento do portal. A seguir, é mostrada a comparação entre os dados obtidos e qual o processo foi escolhido para a implementação do portal.

Tabela 3.3: Resultados do Estudo Dirigido

Artigo	Fase 1	Fase 2	Fase 3	Fase 4	Fase 5	Fase 6	Fase 7	Fase 8
WAE	Gerenciar Projeto	Obter Requisitos	Experiência do Usuário	Análise	Projeto	Implementação	Teste	Implantação
W2000	Análise de Requisito	Projeto de Evolução de Estados	Projeto Hipermídia	Projeto Funcional	Projeto de Visibilidade			
WebML	Modelagem estrutural	Modelagem de hipertexto	Modelagem de Apresentação	Modelagem de Personalização				
OO-H	Modelagem Conceitual	Modelagem da Navegação e Interface	Implementação					

3.3.3 Comparação entre os processos

Após os resultados obtidos com o estudo detalhado acima, foi realizada uma comparação entre os processos, para analisar as características de cada fase dentro deles e, a partir disso, realizar a escolha do que fosse mais adequado às necessidades do desenvolvimento.

Nos processos encontrados na revisão sistemática, as fases apresentam algumas variações, além de uma diferenciação de nomes, que podem representar o mesmo objetivo para cada processo diferente. A seguir estão agrupadas as fases da tabela anterior que possuem o mesmo objetivo, para que sejam comparados os processos em relação ao conjunto total de fases diferenciadas que eles apresentam.

- **Fase 1** - Análise de Requisitos/Modelagem de Domínio/Definição/Identificar os tipos de usuários/Descoberta/Definição Funcional

Objetivo: esta fase define os requisitos funcionais e não funcionais do portal. Todos eles apresentam essa fase;

- **Fase 2** - Seleção de Conteúdo/Identificar o conteúdo a ser publicado/Identificar o mapa do site do conteúdo do portal

Objetivo: esta fase visa a identificação dos conteúdos que serão disponibilizados no portal aos seus usuários. Somente dois dos artigos apresentam a fase, entretanto ela pode ser agrupada a fase anterior, pois os conteúdos podem ser definidos de acordo com os requisitos;

- **Fase 3** - Seleção de Subsistemas/Seleção de Funcionalidade/ Análise de Gaps/ Seleção de Softwares universais

Objetivo: realizar a busca por ferramentas/soluções já prontas, que podem ser integradas às funcionalidades específicas do portal, de acordo com os requisitos que foram definidos na Fase 1;

- **Fase 4** - Definição do Portal/ Projeto/ Desenvolvimento da proposta de solução/ Projeto do sistema/ Projeto da arquitetura do sistema/ Construção do sistema OO com notação UML

Objetivo: nesta fase, é proposto o projeto do portal, e a solução que deverá ser implementada. A exceção fica por conta do artigo de Pettersson e Jarzabek (2005), que tem a fase denominada “Definição do Portal” que realiza a integração entre os módulos, embora a fase seja para fazer o projeto;

- **Fase 5** - Desenvolvimento de protótipo/Plano de Mediação/Desenvolvimento do protótipo com base na proposta/ Estabelecer relacionamento

Objetivo: esta fase apresenta um protótipo de implementação do portal, para posterior validação.

- **Fase 6** - Solução/ Implementação /Desenvolver todos os programas de aplicação e interfaces do cliente/Integração/ Codificação/ desenvolvimento da interface/ Implementação

Objetivo: a partir do protótipo, a solução do portal é implementada, com as interfaces para os possíveis subsistemas selecionados implementadas;

- **Fase 7** - Testes/Validação do Portal/Preencher certificação segura e gerência de usuário - teste de sistema/ Teste de sistema/ Testes e checagem de segurança.

Objetivo: nesta fase, são realizados os testes e validações do portal, além de testes para verificação de segurança e acesso de usuário.

Para os processos de desenvolvimento de aplicações web, existem algumas variações dentre as fases de cada um:

- **Fase 1** - Gerenciar Projeto/ Projeto de Evolução de Estados

Objetivo: esta fase existe no processo de Conallen (2002) e similar em Baresi et al. (2001), para realizar a gerência de todas as fases do desenvolvimento da aplicação;

- **Fase 2** - Obter Requisitos/ Análise de Requisitos

Objetivo: visa a definição dos requisitos funcionais e não funcionais do portal. É equivalente à Fase 1 apresentada nos processos de desenvolvimento de Portais;

- **Fase 3** - Análise/ Modelagem Estrutural/Modelagem Conceitual

Objetivo: realizar a modelagem do sistema, definindo as estruturas e conceitos principais envolvidos na aplicação. Essa fase é similar à fase 4, em que, para os processos de portais, é proposto o projeto do portal, e a solução que deverá ser implementada;

- **Fase 4** - Projeto de Hipermídia e Projeto de Visibilidade/Modelagem de Hiper-texto e Modelagem da Apresentação e Modelagem de Personalização/Modelagem da Navegação e Interface/Experiência do Usuário

Objetivo: esta fase prevê a modelagem dos objetos de hipermídia, com as características de interface (apresentação) e de navegação, ou seja, as interfaces que serão disponibilizadas para o usuário e as formas de interação existentes. Não possui um equivalente direto dentre os processos de desenvolvimento de portais, embora seja similar à Fase 2, em que são identificados os conteúdos (modelagem de conteúdo) e o mapa de navegação do portal;

- **Fase 5** - Projeto/Projeto Funcional

Objetivo: esta fase apresenta o projeto de implementação da aplicação, apresentando as características funcionais da aplicação. Equivalente à Fase 5 do processo de portais;

- **Fase 6** - Implementação

Objetivo: a partir do projeto, a aplicação é implementada, com as interfaces para os possíveis subsistemas selecionados implementadas. Equivalente a Fase 6 dos processos de portais;

- **Fase 7** - Testes/Validação do Portal/Preencher certificação segura e gerência de usuário - teste de sistema/ Teste de sistema/ Testes e checagem de segurança.

Objetivo: nesta fase, são realizados os testes e validações do portal, além de testes para verificação de segurança e acesso de usuário.

Pode-se perceber que os processos, tanto específicos para os portais quanto os para aplicações web, possuem similaridades para o desenvolvimento. Na Tabela 3.4 são mostradas as fases agrupadas por similaridades e a existência ou não dessa fase dentro do processo, para os processos específicos de portais. Na Tabela 3.5 são mostradas as fases agrupadas, mas para os processos de aplicações web.

3.3.4 Processo WAE

Pode-se perceber, na comparação entre as duas tabelas e com base nas equivalências entre as fase, que o processo definido em Conallen (2002) atende a praticamente todas as fases dentro do processo. Por essa razão, ele foi utilizado para o desenvolvimento do portal de conhecimento.

Além das fases definidas por (Conallen, 2002) listadas na tabela acima, ele ainda propõe uma série de sugestões para o desenvolvimento de software, dentre elas:

Tabela 3.4: Fases Processos de Desenvolvimento - Portais

Artigo	Fase 1	Fase 2	Fase 3	Fase 4	Fase 5	Fase 6	Fase 7
Gruhn et al. (2002)	X		X		X	X	X
Hazra (2002)	X		X		X	X	
Pettersson e Jarzabek (2005)	X			X		X	X
Corcho et al. (2006)	X	X				X	
Tahboub et al. (2005)	X			X		X	X
Galligan et al. (2005)	X			X	X	X	
Li et al. (2005)	X		X	X		X	X
Fernandes et al. (2005)	X	X	X		X	X	X

Tabela 3.5: Fases Processos de Desenvolvimento - Aplicações web

Artigo	Fase 1 -	Fase 2 -	Fase 3 -	Fase 4 -	Fase 5 -	Fase 6	Fase 7
Conallen (2002)	X	X	X	X	X	X	X
Baresi et al. (2001)	X	X		X	X		
Gómez et al. (2001)			X	X			
Ceri et al. (2000)			X	X			

- Prover um guia sobre a ordem das atividades da equipe de desenvolvimento;
- Especificar artefatos que devem ser desenvolvidos;
- Dirigir as atividades de desenvolvedores individuais e da equipe como um todo; e
- Oferecer critérios para medida e monitoramento dos produtos de projeto e atividades.

O processo especificado pelas fases do processo da WAE deve ser também dirigido a casos de uso, utilizado uma arquitetura central para o sistema, deve ser conduzido de forma iterativa e incremental, e também deve ser adaptado à realidade de cada empresa ou pessoa que irá desenvolvê-lo. Nesse aspecto, deve-se tomar cuidado com os seguintes itens para realizar essa adaptação:

- Conjunto da companhia e da organização;
- A natureza da aplicação;
- O nível de habilidades da equipe de desenvolvimento; e
- As prioridades relativas do conjunto de características, tempo para entrega número de defeitos aceitáveis, etc.

Entretanto, uma parte bastante importante nos processos de desenvolvimento de portais, em comparação ao desenvolvimento de aplicações *web* está na presença da Fase 2

dos processos de portais, que só é satisfeita pelos processos específicos de portais. Essa fase representa a parte de seleção de ambientes de apoio, ou seleção de funcionalidades, que estabelece uma etapa em que são escolhidas ferramentas prontas, que possuam funcionalidades importantes para o Portal (como por exemplo, os fóruns) e que devem ser integrados à estrutura do Portal como forma de minimizar o tempo de desenvolvimento do portal e incluir mais funcionalidades.

Por esse motivo, para o desenvolvimento do Portal, foi adicionada ao processo de desenvolvimento da WAE uma fase que reflete essa necessidade de integração a ferramentas prontas, realizando-se um estudo acerca de softwares livres que pudessem ser integrados a ele. O estudo referente a essa seleção de ferramentas foi realizado no desenvolvimento do projeto de *Cluster de Portais do Núcleo de Manufatura Avançado* da Escola de Engenharia de São Carlos (EESC - USP/SC). Nesse estudo, foi realizada uma análise comparativa de ferramentas que pudessem auxiliar na gestão de conhecimento, sendo que foram selecionadas algumas ferramentas, e feito um estudo de suas características. Com base numa série de requisitos e critérios, essas ferramentas foram pontuadas, e as que obtiveram nota maior foram selecionadas. No caso da ferramenta de gestão de conteúdo, o EZPublish (eZSystems, 2007) - sistema de gerência de conteúdos e um *framework* de desenvolvimento - foi a ferramenta com a maior nota, e utilizada para a integração com o portal de conhecimento.

Além disso, é importante observar que em nenhum dos processos, tanto dos específicos para portais quanto os de aplicações é considerada a aplicação de uma ontologia dentro do processo de desenvolvimento. Assim, nenhum deles determina como pode ser estruturado o esquema conceitual das informações e conteúdos a ser utilizado dentro da aplicação. Sendo assim, o processo de desenvolvimento escolhido é estendido com a aplicação da OntoTest, explicada na Seção 2.4.

A utilização dessa ontologia combinada ao processo de desenvolvimento permitiu que o tratamento da informação, no que diz respeito ao armazenamento, compartilhamento e consulta fosse otimizado por haver uma representação formalizada dos conceitos de teste de software, respeitando as definições da OntoTest. Essa representação formal possibilitou o mapeamento da estrutura da OntoTest para a base de dados do portal,

Nas seções seguintes, são mostrados os artefatos produzidos para o desenvolvimento do portal seguindo as fases do processo WAE, adicionando a fase de seleção de ferramentas e o uso da OntoTest como base principal de estrutura de informação.

3.4 Requisitos

Como primeira fase do dentro do processo de desenvolvimento de Conallen (2002), e com base no ambiente definido na seção anterior, os requisitos do sistema são definidos. Para

facilitar essa definição, os requisitos foram divididos para cada módulo dentro do portal. Em seguida, são apresentados os requisitos de cada um.

No módulo de conhecimento, o usuário insere os dados sobre os processos de teste, e os requisitos definidos para esse módulo são os seguintes:

1. O sistema deve permitir o cadastro de ferramentas de teste de aplicações web por meio da inclusão das informações no Formulário de Cadastro de Ferramentas.
 - (a) Nome da Ferramenta;
 - (b) Descrição resumida;
 - (c) Dono/Proprietário;
 - (d) Link deve permitir o direcionamento ao site que disponibiliza a ferramenta;
 - (e) Upload para o caso de ferramentas com no máximo 2Mb que ficarão hospedadas no servidor do portal;
 - (f) Técnica/Método que apóia;
 - (g) Contato;
 - (h) Documentação; e
 - (i) Licença;
2. O sistema deve permitir o cadastro de Processos de Teste de aplicações web por meio da inclusão das informações no Formulário de Cadastro de Processos.
 - (a) Nome;
 - (b) Autor/Proprietário;
 - (c) Descrição Resumida;
 - (d) Fases (unidade, integração, sistema, regressão);
 - (e) Passos, que devem ser selecionados a partir de procedimentos previamente cadastrados, de acordo com a seção B1, item 3
 - (f) Procedimentos, que devem ser selecionados a partir de procedimentos previamente cadastrados, de acordo com a seção B1, item 4;
 - (g) Artefatos Produzidos;
 - (h) Recursos Utilizados;
 - (i) Artigos;
 - (j) Manuais.

3. O sistema deve permitir o cadastro de Passos de Teste de aplicações web, por meio da inclusão das informações do Formulário de Cadastro de Passos de Teste:
 - (a) Nome;
 - (b) Planejamento;
 - (c) Projeto;
 - (d) Execução;
 - (e) Análise;
 - (f) Tipo de Atividade.
4. O sistema deve permitir o cadastro de Procedimentos de teste de aplicações web. Os procedimentos podem ser categorizados em Métodos de Teste, Guias de Teste ou Técnicas de Teste.
5. O sistema deve permitir o cadastro de Atividades de teste de aplicações web.
6. O sistema deve permitir o cadastro de Recursos de teste de aplicações web.
7. O sistema deve permitir o cadastro de Artefatos de teste de aplicações web.
8. O sistema deve permitir a inclusão das informações anteriores seguindo a estrutura definida pela OntoTest ((Barbosa et al., 2006; Nakagawa, 2005)). Dessa forma, haverá um relacionamento bem definido entre cada uma das informações presentes na base de dados do portal.

No módulo de colaboração, os requisitos do Portal são:

1. O sistema deve permitir o cadastro de materiais de apoio relacionados ao estudo/utilização de ferramentas, critérios e estratégias de teste de aplicações web com a inclusão das seguintes informações:
 - (a) Título;
 - (b) Autor;
 - (c) Data;
 - (d) Descrição;
 - (e) Palavras Chave;
 - (f) Classificação, dentre as opções: Manuais, Artigos, Estudos de Caso e Pacotes de Laboratório;
 - (g) Link deve permitir o direcionamento ao site que disponibiliza o material;

- (h) Upload para o caso de ferramentas com no máximo 2Mb que ficarão hospedadas no servidor do portal;
- .
- 2. O sistema deve permitir a submissão do material de apoio segundo um processo de aprovação. O processo de aprovação consiste dos seguintes passos:
 - (a) O usuário submete um conteúdo no portal;
 - (b) O sistema categoriza o novo conteúdo com o status de “aguardando aprovação”, e não permite a visualização por nenhum usuário;
 - (c) O sistema notifica o administrador que um novo conteúdo foi adicionado;
 - (d) O administrador altera o status do conteúdo para “aprovado”, e o sistema permite a visualização para todos os usuários
- ;
- 3. O sistema deve manter um histórico (log) que registre as operações realizadas pelos usuários;
- 4. O sistema deve possuir em cada formulário de cadastro um campo que permita que o usuário possa relacionar o item cadastrado aos demais itens já incluídos na base de dados;
- 5. O sistema deve notificar aos usuários que tenham escolhido receber informação a respeito da inclusão de novos conteúdos.

Para o módulo de comunicação, foi definido o seguinte conjunto de requisitos:

- 1. O sistema deve fornecer meios de colaboração entre os usuários do sistema:
 - (a) O sistema deve permitir que os usuários participem da área do fórum existente no portal;
 - (b) O sistema deve permitir que o usuário escolha os fóruns que quer participar, inserindo comentários, adicionando respostas e criando novos tópicos para discussão;
 - (c) O sistema deve permitir que o usuário envie e-mails aos demais usuários do sistema;
- 2. O sistema deve permitir que os usuários registrem opiniões a respeito do conteúdo disponível e os torne visíveis aos demais usuários.

Para o módulo de consulta, os requisitos abaixo são exigidos:

1. O sistema deve possuir um formulário para a busca que permita a busca por:
 - (a) Palavra chave;
 - (b) Título;
 - (c) Autor;
 - (d) Tipo de Material;
2. O sistema deve ordenar e apresentar os resultados de busca de acordo com os itens de cada conteúdo;
3. O sistema deve permitir que o usuário visualize o conteúdo desejado e permita a realização de download desse conteúdo, de acordo com os critérios de segurança e acesso;
4. O sistema deve permitir que, na visualização de um conteúdo, exista links para os conteúdos relacionados.

Os requisitos para as tarefas administrativas, para o módulo de administração são:

1. O sistema deve permitir o cadastro de grupos de usuário no sistema. Cada grupo possui um nível de acesso diferente para cada parte do portal. Como mostrado na Seção 3.2, os grupos são: usuário cadastrado, especialista e administrador. O visitante não está relacionado a nenhum grupo já que não possui vínculo cadastral com o sistema;
2. O sistema deve permitir o cadastro dos usuários no sistema, com a inclusão das seguintes informações:
 - (a) Dados Pessoais;
 - (b) Dados Profissionais;
 - (c) Grupo;
 - (d) Perfil.
3. O sistema deve permitir o acesso de usuários não cadastrados, que podem visualizar conteúdos não restritos (definidos pelo especialista) do sistema.

3.5 Modelagem

Como próximas fases do processo de Conallen (2002), são realizadas a análise e projeto do sistema, em que são utilizados os diagramas da UML que representam o sistema a partir de visões diferenciadas.

Um dos diagramas para realização da análise, é o Diagrama de Caso de Uso que representa a interação entre os atores do sistema e as atividades que cada um pode realizar. Os diagramas de caso de uso foram divididos em dois cenários.

Na Figura 3.1 está representado o cenário que tem como atores o Administrador e o usuário denominado Visitante, que não possui cadastro dentro do Portal de Conhecimento.

Figura 3.1: Cenários de Uso - Administrador e Visitante

Na Figura 3.2 são representados os atores Especialista e Usuário Cadastrado, que participam dos módulos de colaboração, conhecimento e consulta.

A partir dos casos de uso, foram elaborados os diagramas de atividades que permitem que sejam visualizados os fluxos dos eventos realizados pelos atores dentro do sistema.

Para o Ator Administrador, as atividades realizadas estão representadas na Figura 3.3. Para os visitantes, o fluxo de atividades realizadas está representado na Figura 3.4.

Para o Especialista, o diagrama de atividades foi dividido em quatro, para que fosse possível uma melhor visualização do fluxo de atividades realizado por esse Ator. As atividades estão representadas nas Figuras 3.5, 3.6, 3.7 e 3.8. Por fim, na Figura 3.9 está representado o fluxo de atividades que o usuário cadastrado no Portal pode realizar.

Seguindo no processo de Conallen (2002), para representar a interação entre o usuário e o computador, foi feito o Diagrama de Experiência do Usuário, ou Diagrama UX (User

Figura 3.2: Cenários de Uso - Especialista e Usuário Cadastrado

eXperience). Na Figura 3.10 é mostrado o Diagrama UX que representa as atividades gerais do usuário dentro do Portal.

Por fim, foi feito o diagrama de classes do Portal, com as classes presentes no sistema e o modo como essas classes estão relacionadas, apresentado na Figura 3.11. A classe Usuário representa os atores do sistema, que devem possuir o seu currículo e perfil cadastrados. O ator usuário interage com as classes da OntoTest, representadas pelo Pacote OntoTest. O usuário pode inserir conhecimentos relacionados à ontologia, e, também, inserir comentários tanto sobre os processos existentes quanto aos conhecimentos relacionados. Além disso, os usuários do sistema participam do módulo de colaboração, representada pelo Pacote EzPublish (eZSystems, 2007), que foi integrado à estrutura do Portal, para que algumas de suas características fossem utilizadas como forma de ampliar as funcionalidades oferecidas pelo Portal.

Figura 3.3: Diagrama de Atividades - Administrador

Figura 3.4: Diagrama de Atividades - Visitante

Figura 3.5: Atividades - Especialista I

Figura 3.6: Atividades - Especialista II

Figura 3.7: Atividades - Especialista III

Figura 3.8: Atividades - Especialista IV

Figura 3.9: Diagrama de Atividades - Usuário Cadastrado

Figura 3.10: Diagrama UX

Figura 3.11: Diagrama de Classes

3.6 Arquitetura

Com a definição do ambiente, dos requisitos e da modelagem do portal, é possível determinar a arquitetura que será seguida para a implementação do portal. Essa arquitetura delimita como serão organizadas e integradas as estruturas do portal.

Na Figura 3.12 é representada a arquitetura do portal de conhecimento. Essa arquitetura foi desenhada com base nos modelos propostos detalhados na Seção 2.2.4, para os módulos definidos dentro do portal.

Figura 3.12: Arquitetura do Portal de Conhecimento

Assim, existe a camada de apresentação, denominada como *Interface do Portal de Conhecimento*, na qual os usuários realizam o acesso e visualização das informações contidas no portal.

O portal oferece os serviços de *publicação*, *consulta* e *colaboração*, no qual os usuários podem fazer a inserção dos dados sobre testes de aplicações web, as buscas a esses dados, e realizar as interações entre os usuários do portal.

Para o armazenamento dos dados do portal, existem:

- Dados não estruturados e/ou semi-estruturados, armazenados na base de dados do EzPublish, que além de oferecer o apoio para a colaboração entre os usuários, também foi utilizado como ferramenta como sistema de gerência para esses dados; e

- Dados estruturados, na base de dados da OntoTest, que seguem as restrições de categorização e taxonomia definidas por essa ontologia.

A estrutura do EzPublish deve ser integrada à estrutura da OntoTest para haver o relacionamento entre os dados da ontologia com o material de apoio relacionado a esses dados.

3.7 Considerações Finais

Neste capítulo, foram apresentados os requisitos e a modelagem do portal, com a definição dos módulos do ambiente a ser disponibilizado, suas funcionalidades e a arquitetura que integra os componentes do portal para a implementação das funcionalidades.

No próximo capítulo, serão mostrados alguns aspectos de implementação do Portal, com a apresentação das principais funcionalidades que foram implementadas e sua codificação e também os cenários de uso da implementação, representando os aspectos operacionais oferecidos pelo portal aos usuários.

Aspectos de Implementação e Operacionais

4.1 Considerações Iniciais

A partir da estrutura definida para o portal, e como última parte no processo de desenvolvimento do ambiente, é realizada a sua implementação. Neste capítulo, serão mostrados alguns dos aspectos operacionais do portal, incluindo as funcionalidades que foram implementadas, a linguagem de programação escolhida para sua implementação e exemplos de codificação realizada. Serão também mostrados os aspectos operacionais do portal, com algumas ilustrações do seu uso.

4.2 Funcionalidades

Nesta seção, serão explicadas quais as funcionalidades escolhidas para serem implementadas, devido a sua importância e a priorização feita por limitações do tempo disponível para implementação.

No *módulo de conhecimento*, os cadastros das entidades pertencentes a OntoTest foram implementados, bem como as consultas isoladas de cada entidade. Sendo assim, é possível realizar o cadastro de toda a hierarquia relacionada a um processo de teste. O usuário com permissão para efetuar o cadastro insere as informações relacionadas ao

processo. Em seguida, o usuário pode optar por qual sub-nível da ontologia que ele seguirá cadastrando as informações do processo. Assim, ele efetua o cadastro dos Passos, Recursos, Procedimentos e Artefatos pertencentes ao processo, e continua seguindo nos níveis mais internos da ontologia, dentro de cada sub-ontologia principal. Com isso, temos o cadastro completo das informações pertencentes a um processo, que serão armazenadas na base e disponibilizadas para consulta pelos usuários interessados.

Dentro do **módulo de consulta**, a funcionalidade principal é a consulta de processo e de suas informações, então foi implementada a consulta básica, com a inserção de um critério de busca para ser realizada a busca dentro da base. Entretanto, quando houver um número grande de informações dentro da base de dados, será necessária a existência de buscas mais bem elaboradas, com a definição de mais de um critério de busca que permita a busca mais específica das informações, sem que haja um número muito grande de resultados de busca;

Para o **módulo de administração** foi definida como funcionalidade mais importante a gerência dos usuários. Sendo assim, é importante que os usuários sejam devidamente cadastrados, dentro de cada grupo definido para o Portal. No caso, como explicado no capítulo anterior, os grupos existentes dentro do portal estão relacionados a cada um dos diferentes atores do sistema:

- *Usuário Comum*: usuário com acesso ao módulo de consulta, que pode fazer as pesquisas acessando as informações do módulo de conhecimento, módulo de colaboração e do módulo de comunicação, bem como o cadastro de informações dentro do módulo de colaboração e do módulo de comunicação.
- *Especialista*: Tem acesso de cadastro e consulta tanto no módulo de conhecimento quanto nos módulos de colaboração e comunicação.
- *Administrador*: Tem acesso ao módulo de usuários.

Para que as permissões de acesso a cada um dos módulos do sistema sejam respeitadas, os usuários devem ser cadastrados única e exclusivamente em um grupo, e assim, posteriormente, obter o acesso ao sistema. O único ator que não está relacionado a um grupo é o Visitante, pois o seu acesso ao sistema é apenas às informações públicas, ou seja, ele não possui vínculo de cadastro ao portal.

Para o **módulo de colaboração** e o **módulo de comunicação**, as funcionalidades são pré-definidas, considerando que foi determinado como parte da estrutura do portal o **EZPublish**. Dentro dele, já existem várias funcionalidades, que são adequadas as necessidades dos módulos. Existe o modelo de conteúdo (content model) que foi definido de acordo com a estrutura da Ontotest para armazenar os conteúdos relativos ao modulo

de conhecimento. Além disso, nele existem várias possibilidades de busca, que aumentam o poder de consulta das informações do módulo de conhecimento, bem como o fórum, que é utilizado como parte integrante do módulo de comunicação, facilitando o acesso aos especialistas de domínio e a interação entre todos os usuários do sistema.

Com a demarcação das funcionalidades mais importantes de serem disponibilizadas inicialmente no portal, foi realizada a sua implementação. Na próxima seção são mostrados alguns dos aspectos sobre como foi feito o desenvolvimento das funcionalidades.

4.3 Implementação

A partir da definição de quais seriam as funcionalidades mais importantes a serem implementadas, foi realizada a sua implementação, para que fosse possível demonstrar a utilização do Portal de Conhecimento.

Para isso, foram escolhidas as ferramentas e ambientes a serem utilizados. Como linguagem de programação para escrever os códigos que seriam executados na aplicação, foi escolhido o PHP - Hypertext Preprocessor. O PHP é uma linguagem de script Open Source de uso geral, utilizado especialmente para o desenvolvimento web e que pode ser embutido no código HTML (Hypertext Markup Language) das páginas web.

O módulo de conhecimento, então, foi todo estruturado e implementado por meio dessa linguagem.

Como Sistema de Gerência de Banco de Dados (SGBD) foi utilizado o MySQL (MySQL, 2008), que utiliza a linguagem SQL (Structured Query Language - Linguagem de Consulta Estruturada) como interface. Sendo assim, foi realizado o mapeamento entre a OntoTest, mostrada na Seção 2.4 e o modelo relacional que foi desenvolvido para implementação do Portal, para que os conceitos definidos dentro da ontologia de teste fossem seguidos dentro do sistema.

O mapeamento da estrutura principal da OntoTest para o portal está representado na Figura 4.1, com os elementos principais da ontologia representados:

Nas Figuras 4.2, 4.3, 4.4, 4.5 e 4.6 são mostrados os modelos Entidade-Relacionamento do Portal referente a cada uma das sub-ontologias da OntoTest.

Figura 4.1: Modelo ER (Estrutura Principal)

Figura 4.2: Modelo ER Processos de Teste

Figura 4.3: Modelo ER Procedimentos de Teste

Figura 4.4: Modelo ER Passos de Teste

Figura 4.5: Modelo ER Artefatos de Teste

Além disso, para a implementação, foi utilizado um gerador de artefatos, classes, interfaces e documentação, que segue uma arquitetura em camadas e alguns padrões de projeto, agiliza e integra diversas fases do processo de desenvolvimento web com código PHP, denominado **ClassGenerator**. Esta ferramenta facilita a criação da estrutura em camadas dos objetos relacionados às classes da aplicação, conforme descrito a seguir:

A partir de um esquema relacional (base de dados), a ferramenta gera um conjunto de classes em PHP seguindo a estrutura apresentada na 4.7:

Figura 4.7: Arquitetura do ClassGenerator

O ClassGenerator realiza a geração de artefatos a partir de um modelo de mais alto-nível, para automatizar o desenvolvimento de Sistemas de Informação. Os artefatos são gerados com base em padrões reconhecidos pela comunidade de Engenheiros Web, e utilizam regras de mapeamento para a geração. O ClassGenerator foi utilizado com o objetivo de padronizar o código da aplicação e reduzir o tempo para implementação da estrutura do portal.

Por fim, como justificado em seções anteriores, tem-se o uso do EZPublish, que também é implementado em PHP e utiliza o MySQL como Sistema de Base de Dados, para a integração entre o módulo de conhecimento e os demais módulos do Portal de Conhecimento. Abaixo estão listadas as principais funcionalidades implementadas no EZPublish que foram utilizadas para a primeira versão do Portal.

- Modelo de conteúdo: habilita a definição de uma estrutura de conteúdo que pode ficar instantaneamente disponível para publicação;
- Interface Web: A edição e adição de conteúdo, bem como todo o acesso a estrutura do EZPublish é realizada por navegadores, como front-end dos editores disponíveis;
- Interface Administrativa: habilita a edição do conteúdo do site e de suas configurações;
- Editor Online: editor online WYSIWYG integrado ao site, com funções similares a de aplicações de processamento de texto;

- Características Web 2.0: interação com usuário e conteúdo gerado pelo usuário são feitas facilmente com os web sites com características da Web 2.0, com os blogs, fóruns, tagging e polls já embutidos como características do EZPublish.
- Suporte a diversos formatos: O conteúdo pode ser publicado em diversos formatos (PDF, RSS, email, etc);
- Otimização de Máquina de Busca: os conteúdos são indexados pelas máquinas de busca, para facilitar a localização de conteúdo. Além disso, mapas de site, urls amigáveis, e gerência de chaves XHTML são características embutidas no PHP;
- Gerência de Multi-linguagem: todo conteúdo pode ser traduzido para múltiplas linguagens;
- Gerenciamento de versões: o eZPublish mantém rastreamento automático de quais mudanças foram feitas pelos usuários nos conteúdos disponíveis e em que tempo, oferecendo a opção de reverter a mudança para uma anterior, facilitando o controle de versão de conteúdos;
- Publicação cruzada: significa colocar o mesmo conteúdo linkado em diferentes áreas dentro do ambiente. Sendo assim, o conteúdo pode ser reutilizado sem que haja duplicações e evitando inconsistências. Se o conteúdo aparecer em múltiplos locais, editando-se o objeto uma só vez, as alterações são refletidas em todos os locais;
- Auditoria: Notificações de mudança e logs de atividades detalhados oferecem as funções necessárias para verificar a integridade da informação;
- Segurança: controle de quais usuários e grupos de usuários podem ver, criar e atualizar seções específicas dentro do site.

4.4 Codificação

Com a utilização dos ambientes e ferramentas de desenvolvimento descritos na seção anterior, o portal foi implementado. O ClassGenerator, com o esquema do banco de dados, gerou, para cada tabela da base de dados uma classe do tipo Básica, sendo que para cada classe básica foram criadas também uma classe do tipo CadBD, uma classe do tipo Cad, bem como os métodos de acesso na classe Fachada. Além disso, foi criada a classe que realiza a conexão com a base de dados, denominada Conexão. Cada uma tem uma função diferente, conforme descrito abaixo:

- Classe Básica: Descreve a estrutura de dados de cada tabela mapeada. Fornece os atributos e um método set e get para cada um destes últimos, além de um setAll, que edita todos os atributos.;
- Classe Fachada: Interface única para o sistema. Todos os métodos que se comunicam com o Banco de Dados devem estar declarados nesta classe. Instancia um objeto da classe Cad e chama o método correspondente e retorna o resultado obtido pela classe Cad;
- Classe Cad: Faz o mapeamento entre o banco de dados e o sistema. Instancia a classe CadBD e chama o método correspondente, com o retorno dos resultados dessa classe;
- Classe CadBD: Conecta com o BD e executa os comando no mesmo. Instancia um objeto da classe conexão, que faz conexão com o banco, executa os comandos SQL, tratando o retorno. Centraliza os comandos SQL que pertencem a uma tabela.

A seguir, trechos de código que exemplificam a geração dessas classes para o Portal de Conhecimento.

A classe Conexão contém os métodos para estabelecer conexão e manipular as transações sobre o BD. Na Figura 4.8 é apresentado um trecho de código desta classe.

```

1  <?
2  include_once dirname(__FILE__)."/adodb/adodb.inc.php";
3
4  class Conexao {
5      var $conexao;
6      var $usuario;
7      var $senha;
8      var $bdPath;
9      var $bd;
10     var $tipoBd;
11     var $charSet;
12     var $buffers;
13     var $dialect;
14     var $autoCommit;
15
16     function Conexao() {
17         $this->usuario = "root";
18         $this->senha = "*****";
19         $this->bdPath = "localhost";
20         $this->bd = "knowledgeportal";
21         $this->tipoBd = "mysql";
22         $this->charSet = "ASCII";
23         $this->buffers = 100;
24         $this->dialect = 3;
25         $this->autoCommit = true;
26         // Cria uma instancia do objeto
27         // Configura uma conexão com a biblioteca
28         $this->conexao = NewADOConnection($this->tipoBd);
29         $this->conexao->charSet = $this->charSet;
30         $this->conexao->buffers = $this->buffers;
31         $this->conexao->dialect = $this->dialect;
32         // Abre a conexão com o banco
33         if (!$this->conexao->Connect($this->bdPath,$this->usuario,$this->senha,$this->bd)) {

```

Figura 4.8: Classe Conexão

As classes BD contêm os métodos para acesso à base de dados. Elas servem de ponte de comunicação entre a classe Fachada e o banco de dados. A classe BD recebe as requisições provenientes da classe Fachada e realiza o acesso ao banco através da linguagem

SQL (Structured Query Language). Para as operações de inserção, alteração e exclusão, é realizada a chamada ao método setFamilia, retornando “verdadeiro” se a operação for executada com sucesso e “falso” caso contrário. Para as consultas executadas sobre o banco de dados, são feitos os métodos get, com o resultado sendo transformado em objeto e retornado à classe Fachada. Já nas classes básicas, existem os métodos get e set para operar com os atributos dos objetos. Os métodos set são responsáveis por recuperar os dados dos atributos e os set por atribuir novos valores aos atributos dos objetos. Nas Figuras 4.9 e 4.10 são apresentados trechos de código das classes BD e Básicas respectivamente.

```

5 include_once dirname(__FILE__).'/class.CadBD.php';
6 class CadTestRequisicaoBD extends CadBD {
7
8     function CadTestRequisicaoBD($conexao = "") {
9         $this->CadBD($conexao);
10     }
11
12     function inserirTestRequisicao($idTestRequisicao, $idRequisicaoType, $nome) {
13         $sql = "INSERT INTO testRequisicao
14             VALUES ($idTestRequisicao, '$idRequisicaoType', '$nome')";
15         return $this->executarTransacao($sql);
16     }
17
18     function alterarTestRequisicao($idTestRequisicao, $idRequisicaoType, $nome) {
19         $sql = "UPDATE testRequisicao
20             SET
21                 idRequisicaoType = '$idRequisicaoType',
22                 nome = '$nome'
23             WHERE
24                 idTestRequisicao = '$idTestRequisicao'";
25         return $this->executarTransacao($sql);
26     }
27
28     function excluirTestRequisicao($idTestRequisicao) {
29         $sql = "DELETE FROM testRequisicao
30             WHERE
31                 idTestRequisicao = '$idTestRequisicao'";
32         return $this->executarTransacao($sql);
33     }
34
35     function getTestRequisicao($idTestRequisicao) {
36         $sql = "SELECT * FROM testRequisicao
37             WHERE

```

Figura 4.9: Classe BD

```

5 class SupportModuleType {
6
7     function SupportModuleType($idSupportModuleType, $descricao) {
8         $this->idSupportModuleType = $idSupportModuleType;
9         $this->descricao = $descricao;
10     }
11
12     function setAll($descricao = false) {
13         if ($descricao != false) {
14             $this->setDescricao($descricao);
15         }
16     }
17
18     function getIdSupportModuleType() { return $this->idSupportModuleType; }
19
20     function getDescricao() { return $this->descricao; }
21
22     function setIdSupportModuleType($x) { $this->idSupportModuleType = $x; }
23
24     function setDescricao($x) { $this->descricao = $x; }
25
26 }

```

Figura 4.10: Classe Básica

A classe fachada é a única a interagir com o usuário e possui o acesso a todos os métodos das classes BD. Ela efetua as operações solicitadas pelos usuários a partir dos formulários das páginas da aplicação e os envia a classe BD correspondente para, assim, retornar os objetos resultantes, como mostrado na Figura 4.11.

```

74 include_once dirname(__FILE__).'/class.CadTestrequirementset.php';
75 include_once dirname(__FILE__).'/class.Conexao.php';
76 class Fachada {
77     var $conexao;
78     function Fachada() {
79         $this->conexao = NULL;
80     }
81
82     //Métodos de Conexao
83     function getConexao() {
84         return $this->conexao;
85     }
86
87     function iniciarTransacao() {
88         $this->conexao = new Conexao();
89         $this->conexao->iniciarTrans();
90     }
91
92     function commitTransacao() {
93         $this->conexao->commitTrans();
94         $this->conexao = NULL;
95     }
96
97     function rollBackTransacao() {
98         $this->conexao->rollBackTrans();
99         $this->conexao = NULL;
100     }
101
102     //Métodos da Classe Activity
103     function inserirActivity($activity) {
104         $cadActivity = new CadActivity($this->getConexao());
105         return $cadActivity->inserirActivity($activity);
106     }
107
108     function alterarActivity($activity) {
109         $cadActivity = new CadActivity($this->getConexao());
110         return $cadActivity->alterarActivity($activity);
111     }
112
113     function excluirActivity($activity) {
114         $cadActivity = new CadActivity($this->getConexao());
115         return $cadActivity->excluirActivity($activity);
116     }
117 }

```

Figura 4.11: Classe Fachada

Pode-se perceber que, através das classes, é realizado o mapeamento objeto-relacional entre os dados da base de dados, que é relacional, e o programa, desenvolvido com a utilização de modelo de objetos. O mapeamento ocorre quando o resultado da consulta em SQL é transformado em um dos objetos do modelo.

A partir das classes principais da ontologia, são montadas as páginas de interface para o acesso dos usuários, em que são inseridos os códigos PHP de manipulação das informações junto com os códigos HTML de visualização de informações via navegador. Nas páginas, são realizadas as chamadas para as funções das classes citadas acima e a estruturação dos relacionamentos entre as diversas classes.

Na Figura 4.12 é exibida a página que implementa o cadastro do processo, ela recebe as requisições de cadastro e informações preenchidas pelo usuário a partir do navegador, e em seguida, realiza a chamada as funções referentes às classes do processo para fazer o armazenamento dos dados preenchidos.

```

include_once dirname(__FILE__).'/classes/class.Fachada.php';
session_start();
$fachada = new Fachada();
if($_POST['btnSalvar']){
    $processtype = new Processtype($_POST['txtIdProcessType'], $_POST['txtDescription']);
    $res = $fachada->inserirProcesstype($processtype);
    $msg = ($res) ? 1:2;
    header('location:admProcesstype.php?msg='.$urlencode($msg));
}
?>
<HTML>
<HEAD>
<?php include_once dirname(__FILE__).'/template/templateHeaders.php';?>
</HEAD>
<BODY>
<?php include_once dirname(__FILE__).'/template/templateComeco.php';?>
<div id='divConteudo'>
<FORM name='frmProcesstype' method='POST' action='cadProcesstype.php' onSubmit='return formulario.validaForm();'>
<TABLE>
<caption>Register Processtype</caption>
<tr>
<td class='itemForm'>Description:</td>
<td><INPUT type='text' name='txtDescription' size='60' value='' maxlength='255' class='textForm'>
</td>
</tr>
<tr>
<td colspan='2'><INPUT type='submit' name='btnSalvar' value='Salvar' class='buttonForm' onClick = ''>
</td>
</tr>
</TABLE>
</FORM>
</div>
<SCRIPT language='javascript'>
formulario = new FormContext(document.frmProcesstype);
</SCRIPT><?php include_once dirname(__FILE__).'/template/templateFim.php';?>
</BODY>
</HTML>

```

Figura 4.12: Código de Interface de Cadastro de Processo de Teste

Na Figura 4.13 é exibida uma página de consulta, em que é feita a chamada a um método que irá acessar informações da base de dados para recuperação e posterior visualização via navegador.

Com a utilização do ClassGenerator foram geradas três classes para cada tabela, que contém um total de 73 tabelas. As classes *básicas* possuem em média 48 linhas de código (LOC), as classes *cadBD* foram geradas com média de 58 LOC e as classes *Cad*, com média de 30 LOC. Quantificando essa média com o número de tabelas da base, há um total de 3780 LOC. Além disso, na classe fachada, que realiza a interface entre as classes da

```

include_once dirname(__FILE__).'/classes/class.FrontController.php';
session_start();
$frontController = new FrontController();
$vetTestingphase = $frontController->iniciarAdm($_GET['pag'],'testingphase','admTestingphase.php');
??
<HTML>
<HEAD>
<?php include_once dirname(__FILE__).'/template/templateHeaders.php';??>
</HEAD>
<BODY>
<?php include_once dirname(__FILE__).'/template/templateComeco.php';??>
<?Util::getMessage($_GET['msg'])??>
<div id='divConteudo'>
<div id='tituloPagina'>
<div id='divImagemMarcador'>
</div>
Ger&ecirc;ncia de Testingphase</div>
<FORM name='formAction' method='POST' action=''>
<TABLE>
<tr>
<th scope='col'>&nbsp;</th>
<th scope='col'>IdPhaseType</th>
<th scope='col'>IdTestingStrategyProcedure</th>
<th scope='col'>Name</th>
</tr>
<tr>
<td><input type='radio' name='rdOpcao' value='<?Util::Encode($testingphase->getIdTestingPhase())??>' class=''>
</td>
<td><?=$testingphase->getIdPhaseType();??></td>
<td><?=$testingphase->getIdTestingStrategyProcedure();??></td>
<td><?=$testingphase->getName();??></td>
</tr>
??

```

Figura 4.13: Código de Interface de Consulta

camada de negócio com as paginas de apresentacao, foram geradas 2086 linhas de código. Sendo assim, para o módulo de conhecimento, no qual foi implementada a OntoTest, foram geradas uma quantidade estimada em 5866 LOC.

Nas páginas de apresentação, existem os código HTML intercalados às chamadas em códigos PHP para a classe Fachada. Por apresentarem mais código HTML que códigos PHP, elas não foram consideradas para estimar o tamanho do código implementado.

O esforço de desenvolvimento foi minizado com a utilização do gerador, uma vez que, a partir da base, eram obtidas toda a lógica de negócios e acesso aos dados. A parte a ser alterada ficava por conta da apresentação e relacionamento entre as páginas de apresentação, para que fossem realizados os cadastros, consultas e visualizações das informações da base de conhecimento do portal. Além disso, a utilização da OntoTest facilitou a implementação da base de dados, uma vez que as regras eram determinadas pela ontologia e precisavam ser mapeadas para a base.

O EZPublish, sendo uma ferramenta de gestão de conteúdo já pronta, precisou ser instanciada para estar de acordo com as necessidades do portal. Sendo assim, seu modelo de conteúdo foi formado com base nos conceitos e definições da OntoTest, sendo possível armazenar os conteúdos referentes às entidades da ontologia dentro de cada modelo definido no EZPublish, e realizando-se uma associação com a entidade da base de dados da ontologia. As consultas são realizadas dentro da própria ferramenta, sendo assim, é possível fazer uma variedade de consultas para os conteúdos existentes na base do EZPublish.

Além disso, foi utilizado o fórum como originalmente existente no EZPublish, podendo os usuários acessar as diversas áreas do fórum, para criar tópicos, participar das discussões

e inserir comentários sobre o assunto de teste de aplicações web. Na próxima seção são mostradas algumas ilustrações das páginas de interface do portal de e do EzPublish.

4.5 Cenários de Uso

Após a implementação do portal e projeto de integração com uma ferramenta customizável, é possível realizar a apresentação de como será realizada a interação entre os usuários e o sistema.

Para cada usuário, têm-se as restrições de acesso, que estão divididos nos módulos do portal. Nesta seção são mostradas as interfaces de uso do portal, dentro de cada módulo, com algumas das telas implementadas do portal, que permitem que o usuário insira as informações no portal e as visualize, e além disso, algumas telas do EzPublish, que mostram os meios que os usuários têm para colaborar com a disseminação do conhecimento sobre o que está armazenado no portal.

1. Participação no módulo de conhecimento

- (a) Cadastro de Processos de Teste (Testing Process): é o primeiro nível de cadastro de uma informação seguindo a OntoTest: o usuário seleciona no menu a opção para cadastro de processo de teste. Em seguida, insere os dados do processo de teste, e se necessário, inclui um novo tipo de processo de teste (Processo Type). Na Figura 4.14 é apresentada a tela para o cadastro de processos de teste;
- (b) Cadastro de Passos de Teste (Testing Step): Com a existência de um processo de teste previamente cadastrado, o usuário solicita pelo menu o cadastro de um passo de teste associado a esse cadastro. Assim, será apresentada ao usuário uma tela que contém os processos já cadastrados, que ele deve selecionar para cadastrar o respectivo passo. O usuário pode cadastrar um ou mais passos para cada processo. Nas figuras 4.15 e 4.16 é mostrada a sequência de seleção de processo e cadastro de passos de teste associado ao processo selecionado, respectivamente;
- (c) Participação no módulo de colaboração
 - i. EzPublish: ferramenta de gestão de conteúdo customizável a ser integrada com o portal, na Figura 4.17 é apresentado o módulo de administração do EzPublish, em que são inseridos os conceitos de categorias, é realizada a administração dos usuários e as demais funcionalidades oferecidas pela ferramenta. Na Figura 4.18 é mostrada a interface a ser apresentada como tela inicial para o usuário, que deverá ser integrada à estrutura do portal;

- ii. Fórum: Na Figura 4.19 é apresentado o modelo para o fórum do portal, com o projeto de integração do EzPublish ao portal:

É importante deixar claro que os processos que serão cadastrados, bem como as informações relacionadas a esse processo, deverão ser mapeadas corretamente para obedecer à OntoTest. Como existe o uso de terminologias por vezes diferenciadas na área de teste de software, alguns termos que descrevem os processos e suas relações podem não estar de acordo com as definições da ontologia. Sendo assim, é necessário fazer uma equivalência entre o processo a ser inserido com os conceitos da base de dados.

Os especialistas, responsáveis pela inserção das informações, deverão colher as informações que serão cadastradas e observar se os termos e relações descritas no processo são comuns à ontologia implementada para o portal. Em seguida, farão a inserção dos dados. Com isso, serão armazenados os dados do processo segundo os relacionamentos existentes no mapeamento da ontologia para o portal.

Figura 4.14: Cadastro de Processo

Management of Testing process				
	IdUser	IdProcessType	Name	Description
☐	0	4	Teste 1	Descrição do Processo 1
☐	0	3	Teste 2	Descrição do Teste 2
<input type="button" value="Excluir"/> <input type="button" value="Editar"/> <input type="button" value="Detalhes"/>				

Figura 4.15: Seleção de Processo

Home Testing Process ▶ Testing Step ▶ Activity ▶ Testing Resource ▶ Testing Artifact ▶ Testing Strategy Procedure ▶	<div style="text-align: right;">Register Testingstep</div> <table> <tr> <td>Description:</td> <td>Descrição do Passo de Teste</td> </tr> <tr> <td>IdtestingProcess:</td> <td>1</td> </tr> <tr> <td>IdtestingLifeCycle:</td> <td>1</td> </tr> <tr> <td colspan="2"> <input type="button" value="Salvar"/> </td> </tr> </table>	Description:	Descrição do Passo de Teste	IdtestingProcess:	1	IdtestingLifeCycle:	1	<input type="button" value="Salvar"/>	
Description:	Descrição do Passo de Teste								
IdtestingProcess:	1								
IdtestingLifeCycle:	1								
<input type="button" value="Salvar"/>									

Figura 4.16: Cadastro de Passo

The screenshot displays the EzPublish administration interface. At the top, there's a navigation bar with 'eZ publish version 3.9.1' and a search bar. Below it, a menu includes 'Estrutura do conteúdo', 'Biblioteca de mídia', 'Contas de usuário', 'Loja Web', 'Design', 'Configuração', and 'Minha conta'. The main content area is titled 'Home [Frontpage]' and shows the 'Detalhes' tab. It lists the page's name as 'Home', its billboard as 'knowledgePortalBanner', and its layout with columns for 'Press Releases', 'Latest Information From This Company', and 'Boxes'. On the right, a sidebar shows the current user 'Marcella Costa' with options to 'Mudar informações', 'Mudar a senha', and 'Desconectar'. Below the main content, the 'Traduções [1]' section shows the 'Português' language set as the principal one for the 'por-BR' localization. A checkbox at the bottom indicates the use of the principal language if no priority translation exists.

Figura 4.17: Módulo de Administração do EzPublish

Figura 4.18: Página Inicial do EzPublish

The screenshot shows the 'Forums' page within the 'Test Knowledge Portal'. The page has a navigation bar with 'Home / Community / Forums'. A left sidebar lists 'Community' links: Multimedia, Downloads, Pictures, Forums, Technology, General, and Setup & design. The main content area is titled 'Forums' and contains a table of forum topics.

Forums Table:

Forum	Topics	Posts	Last reply
Technology Tech discussion.	0	0	
General An example of the general forum.	3	4	Where can I find more information 25/06/2007 17:20
Setup & design Etiam erat urna, tincidunt id.	0	0	

At the bottom of the page, there is a copyright notice: 'Copyright © 2006 eZ systems AS. All rights reserved. Powered by eZ Publish™ Content Management System.'

Figura 4.19: Projeto de Fórum

4.6 Considerações Finais

Neste capítulo, foram apresentadas as principais funcionalidades implementadas do Portal de Conhecimento sobre Ferramentas, Estratégias e Critérios de Aplicações Web, bem como as interfaces do portal de conhecimento, que serão apresentadas aos usuários.

Para o desenvolvimento do portal, foram seguidas as características, modelos e arquitetura como as definidas na Seção 2.2. Além disso, foi seguido um processo bem definido, considerando os estudos realizados para aplicações web e portais detalhado na Seção 3.3. Com o estudo desses processos, foram definidas as fases que deveriam ser aplicadas, com base no processo WAE (Web Application Extension) definido em Conallen (2002). Adicionalmente, esse processo foi estendido, com a determinação da utilização de uma ontologia que definisse a estruturação do conteúdo do portal, a OntoTest (Barbosa et al., 2006), bem como a adição de uma fase para a seleção do ambiente customizável pronto, o EzPublish (eZSystems, 2007).

A inclusão da OntoTest ofereceu maior rigor na hierarquização dos conceitos e relacionamentos entre esses conceitos dentro do domínio dos testes de software, facilitando a compreensão do domínio, bem como para apoiar a interoperabilidade entre sistemas que queiram se relacionar ao portal. Já a utilização do EzPublish como ferramenta de gestão de conteúdo facilitou o desenvolvimento, por evitar a necessidade de desenvolver as funcionalidades de colaboração, oferecendo-as já implementadas para serem integradas ao portal de acordo com sua estrutura.

O desenvolvimento e implementação do Portal estão em seu estágio não completo. Como portal, ele pode ainda possuir mais funcionalidades além das implementadas, sendo assim a versão implementada não é a versão final, embora já tenha várias funcionalidades que demonstrem sua aplicabilidade dentro do cenário de testes de software para aquisição, compartilhamento e evolução de conhecimento nessa área.

Atualmente, não existem dados inseridos no Portal, a OntoTest foi implementada na base de dados do sistema e os relacionamentos corretos foram mapeados nas páginas de navegação do portal. Pretende-se então, para validar o uso do portal, inserir as informações mostradas na Seção 2.3 para que haja uma base inicial de informações.

Em seguida, o portal será disponibilizado para que os usuários possam realizar as pesquisas e, com o uso do sistema, armazenar informações da área de teste de aplicações web, que estarão, então, acessíveis para visualização.

No próximo capítulo, são apresentadas as considerações finais do trabalho, com uma visão geral da pesquisa realizada para o desenvolvimento do portal, e a enumeração das principais contribuições e os trabalhos futuros que podem ser realizados a partir deste.

Conclusão

5.1 Considerações Gerais

Neste trabalho, foi mostrado o desenvolvimento do Portal do conhecimento de Estratégias, Critérios, Técnicas e Ferramentas de Teste de Aplicações Web, como forma de fomentar o conhecimento existente na área, para estudantes e pesquisadores, e a indústria.

A importância de formas de gerir o conhecimento é apresentada como desafio, pois o conhecimento costuma estar distribuído entre diversas fontes, inclusive humanas. Sendo assim, torna-se fundamental haver uma maneira eficaz de distribuir esse conhecimento, desde a sua aquisição e armazenamento, até o seu compartilhamento e evolução.

Os portais, por suas características, realizam a integração entre pessoas, aplicações e serviços para o gerenciamento do conhecimento existente acerca de determinado assunto de interesse, permitindo, então, que as informações sejam disseminadas de maneira integrada e que seja possível compartilhar o conhecimento existente em torno de uma área de interesse.

O desenvolvimento do portal voltado para a área de testes de aplicações web advém da necessidade de reunir conhecimento sobre as atividades de garantia de qualidade das aplicações web, considerando dois pontos:

- A necessidade da realização de atividades de teste, como um dos elementos críticos em qualquer projeto de software, para a garantia da qualidade na revisão final da especificação, projeto e geração de código;

- A crescente demanda de desenvolvimento desse tipo de aplicação, que possui características diferenciadas das aplicações tradicionais, sendo este um tema relevante dentro da área.

Seguindo as exigências da área de Engenharia de Software, que prediz a utilização de metodologias adequadas para o desenvolvimento de sistemas, houve a necessidade de um estudo que envolvesse processos de desenvolvimento de sistemas sobre a plataforma web, na qual o portal foi implementado. Assim, tendo como objetivo definir o processo utilizado, aplicando-se regras para o estudo, como a revisão sistemática e o estudo dirigido, obteve-se um esquema que foi considerado adequado ao escopo do trabalho.

Além disso, para facilitar a estruturação e organização do conteúdo a ser armazenado, foi utilizado um “modelo” de informação, com a utilização da OntoTest, definida em Barbosa et al. (2006). Com a OntoTest, foi possível realizar a implementação do portal, com a base de dados obedecendo à ontologia, que classifica e organiza os conceitos da área de teste e permite que o conteúdo seja armazenado a partir de uma série de regras e relacionamentos que promove uma melhor administração da informação, tanto para as opções de inclusão, como as operações de atualizações e consultas na base.

A utilização da OntoTest possibilitou organizar melhor a informação distribuída no portal, facilitando a compreensão do domínio dos conceitos de teste de software e o uso desses conceitos para a aquisição, compartilhamento e evolução do conhecimento dentro da aplicação

Assim, o Portal de Conhecimento sobre Critérios, Técnicas, Estratégias e Ferramentas de Teste de aplicações web foi desenvolvido para facilitar o ciclo de aquisição e compartilhamento de conhecimento, disseminando o conteúdo da área. O portal será disponibilizado para o público em geral, sendo assim, para as pessoas terem acesso às informações armazenadas, bastará realizar o cadastro e utilizar as funcionalidades dos módulos de consulta e colaboração do portal.

5.2 Contribuições

Este trabalho apresentou como principal contribuição o estudo sobre os portais de conhecimento, considerando os principais conceitos a respeito dessa tecnologia, suas classificações, as principais características, e além disso, o processo utilizado para realizar o seu desenvolvimento.

Têm-se como principais contribuições do trabalho:

- Caracterização da tecnologia dos portais, com o estudo de classificação, modelos de portais, principais funcionalidades existentes em um portal e arquitetura de portais;

- Realização de um estudo sobre principais processos de desenvolvimento que podem ser aplicados aos portais, divididos em:
 - Estudo dirigido de processos de desenvolvimento de aplicações web, que apresentou as características de quatro processos encontrados na literatura, fazendo-se a comparação entre as fases e indentificando-se similaridades e diferenças de cada um, que poderiam ser utilizados no contexto de desenvolvimento de portais web;
 - Revisão sistemática de processos de desenvolvimento de portais web, que procurou identificar de maneira sistemática os processos específicos para portais, traçando a comparação entre tais processos, tal como realizado no estudo dirigido.
- A partir do estudo sobre os processos de desenvolvimento, extensão do processo de desenvolvimento de aplicações web (WAE - Conallen (2002)) com a adição de uma fase encontrada em processos específicos para o desenvolvimento de portais (fase de integração de ferramenta) e com a utilização de uma ontologia como base para estruturação do conteúdo do portal;
- Projeto de integração da estrutura do portal a uma ferramenta customizável, para ampliar o nível de funcionalidades do portal.

5.3 Trabalhos Futuros

De forma a ampliar as contribuições oferecidas por esse trabalho, e dando continuidade às atividades realizadas, têm-se como trabalhos futuros:

- Realizar a implementação das funcionalidades descritas na seção de requisitos do portal, com a implementação de um mecanismo de busca mais otimizado para a consulta às informações da base de conhecimento;
- Realizar a integração proposta do EzPublish com as páginas do portal, para relacionar os conteúdos de material de apoio ao módulo de conhecimento;
- Realizar novos estudos para a viabilização de novas funcionalidades, como a integração de módulos educacionais que permitam a utilização do conhecimento armazenado para o ensino, presencial e à distância;
- Inserir dados reais no portal, seguindo a estrutura da Ontotest;
- Ampliar o escopo do portal para o teste de software de quaisquer aplicações, o que se torna viável pois foi utilizada uma ontologia de teste de qualquer aplicação;

- Integrar a estrutura do portal ao ambiente definido em Domingues (2005), que identifica o estado da arte e da prática para os testes de software orientados a objeto, ampliando o escopo de conhecimento de teste de aplicações tanto web quanto orientadas a objeto.

Referências

- Alavi, M.; Leidner, D. Knowledge management systems: Issues, challenges, and benefits. *Communications of the AIS*, v. 1, n. 7, 1999.
- Aneja, A.; Rowan, C.; Brooksby, B. Corporate portal framework for transforming content chaos on intranets. *Intel Technology Journal Q1*, 2000.
- April, A.; Deshanais, J.-M.; Dumke, R. A formalism of ontology to support a software maintenance knowledge-based system. In: *18th International Conference on Software Engineering and Knowledge Engineering (SEKE'06)*, Califórnia, EUA, 2006, p. 331—336.
- Balushi, T. H. A.; Sampaio, P. R. F.; Dabhi, D.; Loucopoulos, P. Performing requirements elicitation activities supported by quality ontologies. In: *18th International Conference on Software Engineering and Knowledge Engineering (SEKE'06)*, Califórnia, EUA, 2006, p. 343—348.
- Barbosa, E. F.; Maldonado, J. C.; Vincenzi, A. M. R.; Delamaro, M. E.; Souza, S. R. S.; Jino, M. Introdução ao teste de software. Minicurso apresentado no XIV Simpósio Brasileiro de Engenharia de Software, 2000.
- Barbosa, E. F.; Nakagawa, E. Y.; Maldonado, J. C. Ontotest: An ontology of software testing. In: *18th International Conference on Software Engineering and Knowledge Engineering (SEKE'06)*, Sao Francisco, Estados Unidos, 2006.
- Baresi, L.; Garzotto, G.; Paolini, P. Extending uml for modelling web applications. In: *Procs. 34th Hawaii International Conference on System Sciences*, 2001.
- Basili, V. R.; Shull, F.; Lanubile, F. Building knowledge through families of experiments. *IEEE Transactions on Software Engineering*, v. 25, n. 4, p. 456—473, 1999.

- Beizer, B. *Software testing techniques*. 2 ed. Van Nostrand Reinhold Company, 1990.
- Benbya, H.; Passiante, G.; Belbaly, N. A. Corporate portals: a tool for knowledge management synchronization. *International Journal of Information Management*, v. 24, p. 201–220, 2004.
- Bianchini, S. L. *Avaliação de Metodologias de Desenvolvimento de Sistemas Web*. Dissertação de Mestrado, ICMC/USP, São Carlos/SP - Brasil, em andamento, 2005.
- Bianchini, S. L. *Avaliação de métodos de desenvolvimento de aplicações Web*. Dissertação de Mestrado, ICMC-USP, São Carlos – SP, em andamento, 2006.
- Briand, L.; Labiche, Y. Empirical studies of software testing techniques: challenges, practical strategies, and future research. *SIGSOFT Softw. Eng. Notes*, v. 29, n. 5, p. 1–3, 2004.
- Carvalho, M. R. *HMBS/M – um método orientado a objetos para o projeto e o desenvolvimento de aplicações hipermídia*. Dissertação de Mestrado, ICMC-USP, São Carlos, SP, 1998.
- Carvalho, M. R.; Oliveira, M. C. F.; Masiero, P. C. HMBS/M – an object oriented method for hypermedia design. In: *V Simpósio Brasileiro de Multimídia e Sistemas Hipermídia (SBMIDIA '99)*, Goiânia, GO, 1999, p. 43–62.
- Ceri, S.; Fraternali, P.; Bongio, A. Web modeling language (webml): a modeling language for designing web sites. In: *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications netowrking*, Amsterdam, The Netherlands, The Netherlands: North-Holland Publishing Co., 2000, p. 137–157.
- Chan, A. *Um Ambiente para de Apoio ao Uso de Padrões de Software e Requisitos de Teste no Desenvolvimento de Aplicações Web*. Dissertação de Mestrado, ICMC/USP, São Carlos/SP - Brasil, em andamento, 2005.
- Compuware Compuware BoundsChecker. Disponível em <http://www.compuware.com/>. Acesso em 27/02/2006, 2006.
- Conallen, J. Modeling web application architectures with uml. *Commun. ACM*, v. 42, n. 10, p. 63–70, 1999.
- Conallen, J. *Buildind Web applications with UML*. 2nd. ed. Addison-Wesley, 2002.
- Corcho, O.; Cima, A. L.; Gomes-Perez, A. A platform for the development of semantic web portals. In: *ICWE'06: Proceedings of the Sixth International Conference on Web Engineering*, New York, NY, USA: ACM Press, 2006, p. 145–152.

- Costa, M. L. S.; Maldonado, J. C. *Processos de desenvolvimento de portais web: Uma revisão sistemática*. Relatório Técnico, ICMC-USP/SC, São Carlos/SP, Brasil, em andamento, 2008.
- DeepSoftware Deep Log Analyzer. Disponível em <http://www.deep-software.com>. Acesso em 27/02/2006, 2006.
- Devedzic, V. Research community knowledge portals. *International Journal of Knowledge and Learning*, v. 1, p. 96–112, 2005.
- Dias, C. Corporate portals: a literature review of a new concept in information management. *International Journal of Information Management*, v. 21, p. 269–287, 2001.
- Domingues, A. L. S. *Aplicações Web: definição e análise de recursos de teste e validação*. Tese de Doutorado, ICMC/USP, São Carlos/SP - Brasil, em andamento, 2005.
- Doria, E. S. *Replicação de Estudos Empíricos em Engenharia de Software*. Dissertação de Mestrado, ICMC/USP, São Carlos/SP - Brasil, 2001.
- Duarte, K. C.; Falbo, R. A. Uma ontologia de qualidade de software. In: *Anais do WQS'2000, Workshop de Qualidade de Software - XIV Simpósio Brasileiro de Engenharia de Software*, João Pessoa-PB, Brasil, 2000.
- Eckerson, W. 15 rules for enterprise portals. Disponível em <http://my.theebsgroup.com/pls/portal/>. Acesso em 14/01/2006, 1999a.
- Eckerson, W. New version fulfills enterprise portal requirements. Disponível em http://206.130.104.53/paperfree/017_eckerson_plumtree.pdf. Acesso em 14/01/2006, 1999b.
- Elbaum, S.; Karre, S.; Rothermel, G. Improving web application testing with user session data. In: *ICSE '03: Proceedings of the 25th International Conference on Software Engineering*, Washington, DC, USA: IEEE Computer Society, 2003, p. 49–59.
- Envive Envive Prophecy. Disponível em <http://www.envive.com>. Acesso em 27/02/2006, 2006.
- eZSystems eZ Publish Enterprise Content Management System. Disponível em <http://ez.no/>. Acesso em 27/04/2007, 2007.
- Falbo, R. A.; Menezes, C. S.; Rocha, A. R. Using ontologies to improve knowledge integration in software engineering environments. In: *World Multiconference on Systemic, Cybernetics and Informatics / 4th International Conference on Information Systems Analysis and Synthesis (SCI98/ISAS98)*, Orlando, EUA, 1997, p. 1–8.

- Fensel, D. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. 2 ed. Springer, 2003.
- Fernandes, K. J.; Raja, V.; Austin, S. Portals as a knowledge repository and transfer tool - vizcon case study. *Technovation*, v. 25, n. 11, p. 1281–1289, 2005.
- Finjan Software Surfincheck Firewall. Disponível em <http://www.finjan.com/>. Acesso em 27/02/2006, 2006.
- Fons, J.; Pelechano, V.; Pastor, O.; Albert, M.; Valderas, P. Extending an oo method de develop web applications. In: *The Twelfth International World Wide Web Conference*, 2003.
- Frappaolo, C. What's your knowledge iq. Disponível em http://www.delphigroup.com/consulting/case_studies/knowledge_IQ.htm. Acesso em 27/02/2006, 2002.
- Fraternali, P.; Paolini, P. Model-driven development of web applications: the autoweb system. In: *ACM Transactions on Office Information Systems*, 2000.
- Friedman, M. H. Creating a knowledge management vocabulary. Disponível em <http://www.intelligentkm.com/feature/06/SideBar2.shtml>, 2002.
- Galligan, M. P. T.; Hopkins, M. J. L.; Kehoe, P. D. F. An approach to deliver an early return on investment during the development of a corporate web portal. In: *SCC '05: Proceedings of the 2005 IEEE International Conference on Services Computing*, Washington, DC, USA: IEEE Computer Society, 2005, p. 175–184.
- Garavelli, A. C.; Gorgoglione, M.; Scozzi, B. Managing knowledge transfer by knowledge technologies. *Technovation*, v. 22, p. 269–279, 2002.
- Garzotto, F.; Schwabe, D.; Paolini, P. HDM – A Model Based Approach to Hypertext Application Design. *ACM Transactions on Information Systems*, v. 11, n. 1, p. 1–26, 1993.
- Ginige, A.; Murugesan, S. Guest editors' introduction: The essence of web engineering - managing the diversity and complexity of web application development. *IEEE MultiMedia*, v. 8, n. 2, p. 22–25, 2001a.
- Ginige, A.; Murugesan, S. Guest editors' introduction: Web engineering - an introduction. *IEEE MultiMedia*, v. 8, n. 1, p. 14–18, 2001b.
- Gómez, J.; Cachero, C.; Pastor, O. Conceptual modeling of device-independent web applications. *IEEE MultiMedia*, v. 8, n. 2, p. 26–39, 2001.

- Gruber, T. R. Toward principles for the design of ontologies used for knowledge sharing. In: *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Califórnia, Estados Unidos: Kluwer Academic Publishers, 1993.
- Gruhn, V.; Mocker, M.; Schöpe, L. Development of an electronic commerce portal system using a specific software development process. In: *CRPIT '02: Proceedings of the Fortieth International Conference on Tools Pacific*, Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2002, p. 93–101.
- Gu, A.; Henderson-Sellers, B.; Lowe, D. Web modelling languages: the gap between requirements and current exemplars. In: *AusWeb 2002 - The Eighth Australian World Wide Web Conference*, 2002.
- Guarino, N. Understanding, building, and using ontologies: A commentary to “using explicit ontologies in kbs development”, by van heijst, schreiber, and wielinga. In: *International Journal of Human and Computer Studies*, Padova, Itália, 1997, p. 293–310.
- Gwebu, K. Rethinking university portal design: The role of knowledge rooms in facilitating knowledge sharing. Disponível em <http://computing.breinstorm.net>. Acesso em 14/11/2005, 2000.
- Hazra, T. K. Building enterprise portals: principles to practice. In: *ICSE '02: Proceedings of the 24th International Conference on Software Engineering*, New York, NY, USA: ACM Press, 2002, p. 623–633.
- Höhn, E. N. *Técnicas de leitura e especificação de requisitos de software: estudos empíricos e gerência de conhecimento em ambientes acadêmico e industrial*. Dissertação de Mestrado, ICMC/USP, São Carlos/SP - Brasil, 2005.
- Howden, W. E. *Software Engineering and Technology: Functional Program Testing and Analysis*. New York: McGrall-Hill Book Co, 1987.
- JStudio JStudio SiteWalker. Disponível em <http://www.jstudio.de/English/LogOnENG.htm>. Acesso em 27/02/2006, 2006.
- JWebunit JWebunit. Disponível em <http://jwebunit.sourceforge.net/>. Acesso em 27/02/2006, 2006.
- Kappel, G.; Michlmayr, E.; Pröll, B.; Reich, S.; Retschitzegger, W. Web engineering - old wine in new bottles? In: *Web Engineering: 4th International Conference, ICWE 2004*, Berlin, Heidelberg: Springer-Verlag, 2004, p. 6–12.
- Kitchenham, B. *Procedures for performing systematic reviews*. Relatório Técnico 0401, Department of Computer Science - Keele University, 2004.

- Koch, N. *Software engineering for adaptive hypermedia systems: Reference model, modeling techniques and development process*. Tese de Doutorado, Ludwig-Maximilians University, Munich - Germany, 2000.
- Koch, N.; Mandel, L. *Using uml to design hypermedia applications*. Relatório Técnico, Ludwig-Maximilians-University, 1999.
- Lee, S. C.; Offutt, J. G. Generating test cases for xml-based web applications. In: *12th International Symposium on Software Reliability Engineering*, Hong Kong, 2001, p. 200–209.
- Li, M.; Wang, J.; Wong, Y. S.; Lee, K. S. A collaborative application portal for the mould industry. *International Journal of Production Economics*, v. 96, n. 2, p. 233–247, 2005.
- Liu, C. H.; Kung, D. C.; Hsia, P. Object-based data ow testing of web applications. In: *The First Asia-Pacific Conference on Quality Software (APAQs'00)*, Hong Kong-China, 2000a, p. 7–16.
- Liu, C. H.; Kung, D. C.; Hsia, P. Structural testing of web applications. In: *11th International Symposium on Software Reliability Engineering (ISSRE00)*, San Jose-California, 2000b, p. 84–96.
- Liu, C.-H.; Kung, D. C.; Hsia, P.; Hsu, C.-T. Object-based data flow testing of web applications. In: *APAQs '00: Proceedings of the The First Asia-Pacific Conference on Quality Software (APAQs'00)*, Washington, DC, USA: IEEE Computer Society, 2000c, p. 7.
- Liu, C.-H.; Kung, D. C.; Hsia, P.; Hsu, C.-T. Structural testing of web applications. In: *ISSRE '00: Proceedings of the 11th International Symposium on Software Reliability Engineering (ISSRE'00)*, Washington, DC, USA: IEEE Computer Society, 2000d, p. 84.
- Lucca, G. A. D.; Fasolino, A. R.; Faralli, F.; de Carlini, U. Testing web applications. In: *18th International Conference on Software Maintenance (ICSM 2002)*, Quebec-Canada, 2002, p. 310–319.
- Lucca, G. A. D.; Penta, M. D. Considering browser interaction in web application testing. In: *5th International Workshop on Web Site Evolution (WSE'03)*, Amsterdam.), The Netherlands: IEEE Computer Society, 2003, p. 74.
- Maldonado, J. *Critérios potenciais usos: Uma contribuição ao teste estrutural de software*. Tese de Doutorado, DCA/FEE/UNICAMP, 1991.
- Maldonado, J. C.; Barbosa, E. F.; Vincenzi, A. M. R.; Delamaro, M. E.; Souza, S. R. S.; Jino, M. Introdução ao teste de software. Notas Didaticas do ICMC-USP/SP, 2004.

- Mercury Interactive Mercury Interactive LoadRunner. Disponível em <http://www.mercury.com/>. Acesso em 27/02/2006, 2006.
- MicroQuill MicroQuill HeapAgent. Disponível em <http://www.microquill.com/>. Acesso em 27/02/2006, 2006.
- Murray, G. The portal is the desktop. 1999.
- Myers, G. J. *The Art of Software*. 2 ed. John-Wiley & Sons, Inc, 2004.
- MySQL MySQL Database Software Products. Disponível em <http://www.mysql.com/>. Acesso em 05/01/2008, 2008.
- Nakagawa, E. Y. *Uma contribuição ao projeto arquitetural de ambientes de engenharia de software*. Tese de Doutorado, ICMC-USP, São Carlos – SP, 2005.
- Nguyen, H. Q. *Testing applications on the web*. John Willey and Sons, Inc., 2001.
- Northwest Performance Software, Inc. Netscan Tools. Disponível em <http://www.nwpsw.com/>. Acesso em 27/02/2006, 2006.
- ParaSoft ParaSoft Insure++. Disponível em <http://www.parasoft.com/>, 2006.
- ParaSoftbreak ParaSoft SiteRuler. Disponível em <http://www.parasoft.com/jsp/home.jsp>. Acesso em 27/02/2006, 2006.
- Pastor, O.; Pelechano, V.; Insfránz, E.; Gómez, J. From object oriented conceptual modeling to automated programming in java. In: *ER '98: Proceedings of the 17th International Conference on Conceptual Modeling*, London, UK: Springer-Verlag, 1998, p. 183–196.
- Perry, D. E.; Kaiser, G. E. Object-Oriented programs and Testing. *Journal of Object-Oriented Programming*, v. 2, n. 5, p. 13—19, 1990.
- Pettersson, U.; Jarzabek, S. Industrial experience with building a web portal product line using a lightweight, reactive approach. In: *ESEC/FSE-13: Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering*, New York, NY, USA: ACM Press, 2005, p. 326–335.
- Pressman, R. S. *Software Engineering - A Practitioner's Approach*. 6 ed. McGraw-Hill, 2005.
- RadView RadView WebLoad,. Disponível em <http://www.radview.com/default.asp>. Acesso em 27/02/2006, 2006.

- Raol, J. M.; Koong, K. S.; Liu, L. C.; Yu, C. S. An identification and classification of enterprise portal functions and features. *Industrial Management & Data Systems*, v. 102, n. 7, p. 390–399, 2002.
- Rapps, S.; Weyuker, E. J. Data flow analysis techniques for program test data selection. In: *6th International Conference on Software Engineering*, Tokio, Japan, 1982, p. 272—278.
- Rapps, S.; Weyuker, E. J. Selecting software test data using data flow information. *IEEE Transactions on Software Engineering*, v. 11, n. 4, p. 364–375, 1985.
- Rational Rational Performance Studio. Disponível em <http://www-306.ibm.com/software/rational/>. Acesso em 27/02/2006, 2006a.
- Rational Rational Purify. Disponível em <http://www.rational.com/>. Acesso em 27/02/2006, 2006b.
- Rational Rational VisualTest. Disponível em <http://www.rational.com/>. Acesso em 27/02/2006, 2006c.
- Reis, T. P. C. *REQE - Uma Metodologia para Medição de Qualidade de Aplicações Web na Fase de Requisitos*. Dissertação de Mestrado, CIN/UFPE, Recife/PE - Brasil, 2004.
- Ricca, F.; Tonella, P. Analysis and testing of web applications. In: *ICSE '01: Proceedings of the 23rd International Conference on Software Engineering*, Washington, DC, USA: IEEE Computer Society, 2001, p. 25–34.
- Ricca, F.; Tonella, P. Testing processes of web applications. *Ann. Softw. Eng.*, v. 14, n. 1-4, p. 93–114, 2002.
- Rossi, G. *Um método orientado a objetos para o projeto de aplicação hipermídia*. Tese de Doutorado, PUC-RJ, Rio de Janeiro, RJ, 1996.
- RSW Software RSW eTester. Disponível em <http://www.rswsoftware.com/>. Acesso em 27/02/2006, 2006.
- Schwabe, D.; Pontes, R. A. *OOHDM-Web: Rapid prototyping of hypermedia applications in the WWW*. Relatório Técnico MCC 08, PUC-Rio, Rio de Janeiro, RJ, 1998.
- Segue Segue SilkTest. Disponível em <http://www.segue.com/>. Acesso em 27/02/2006, 2006.
- Shilakes, C. C.; Tylman, J. Enterprise information portals. Disponível em <http://www.sagemaker.com/home.asp?id=500&file=Company/WhitePapers/lynch.htm>, 1999.

- Smith, M. A. Portals: toward an application framework for interoperability. *Commun. ACM*, v. 47, n. 10, p. 93–97, 2004.
- Sneed, H. M. Testing a web application. In: *WSE '04: Proceedings of the Web Site Evolution, Sixth IEEE International Workshop on (WSE'04)*, Washington, DC, USA: IEEE Computer Society, 2004, p. 3–10.
- Swartout, B.; Patil, R.; Knight, K.; Russ, T. Toward distributed use of large-scale ontologies. In: *Proceedings Of Aaai'97 Spring Symposium Series Workshop On Ontological Engineering*, Califórnia, Estados Unidos: AAAI Press, 1997, p. 138–148.
- Tahboub, R.; Lazarescu, D.; Lazarescu, V. Software engineering and database web management in daleelcom design. In: *ITCC '05: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II*, Washington, DC, USA: IEEE Computer Society, 2005, p. 823–824.
- Terra, J. C. C.; Gordon, C. *Portais corporativos : a revolução na gestão do conhecimento*. 1 ed. São Paulo: Negócio, 2002.
- The Delphi Group Business portals: Applications & architecture. Disponível em <http://www.delphiweb.com/knowledgebase/>. Acesso em: 07/02/2006, 2001.
- Toledo, A. M. *Portais Corporativos: Uma ferramenta estratégica de apoio à Gestão de Conhecimento*. Dissertação de Mestrado, UFRJ, Rio de Janeiro/RJ - Brasil, 2002.
- UPPSALA University HTTPD Log Analyzers. Disponível em <http://www.uu.se/Software/Analyzers/>. Acesso em 27/02/2006, 2006.
- Villela, K.; Santos, G.; Gallota, C.; anda R.Negrão, R. M.; Farias, L.; Zlot, F.; Barcelos, M.; anda M. Oliveira, B. S.; Travassos, G. H.; Rocha, A. R. Toward distributed use of large-scale ontologies. In: *stendendo a Estação TABA para criação de Ambientes de Desenvolvimento de Software Orientados a Organização*, Rio de Janeiro-RJ, Brasil: AAAI Press, 2001, p. 332–337.
- Vincenzi, A. M. R. *Orientação a objeto: Definição e análise de recursos de teste e validação*. Tese de Doutorado, ICMC/USP, 2004.
- Watchfire Watchfire. Disponível em <http://www.watchfire.com/>. Acesso em 27/02/2006, 2006.
- Watson, J.; Fenner, J. understanding portals. *The Information Management Journal*, v. 34, n. 3, p. 18, 2000.

- White, C. Enterprise information portal requirements. Disponível em <http://www.decisionprocessing.com/papers/eip2.doc>, 1999.
- Xu, L.; Xu, B.; Jiang, J. Testing web applications focusing on their specialties. *SIG-SOFT Softw. Eng. Notes*, v. 30, n. 1, p. 10, 2005.
- Zelkowitz, M. V.; Wallace, D. R. Experimental models for validating technology. *Computer*, v. 31, n. 5, p. 23–31, 1998.