

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA
COLEGIADO DO CURSO DE CIÊNCIA DA COMPUTAÇÃO

Jordana Sarmenghi Salamon

Portal de Medição de Software

Projeto de Graduação apresentado ao Departamento de Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Bacharel em Ciência da Computação.

Orientador: Monalessa Perini Barcellos

VITÓRIA
2016

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA
COLEGIADO DO CURSO DE CIÊNCIA DA COMPUTAÇÃO

Jordana Sarmenghi Salamon

Portal de Medição de Software

COMISSÃO EXAMINADORA

Prof. Monalessa Perini Barcellos, D. Sc.
Orientadora

Prof. Renata Silva Souza Guizzardi, D. Sc.

Prof. Vitor Estêvão Silva Souza, D. Sc.

Vitória, 02. de março de 2016

À minha família, meu pilar de sustentação

AGRADECIMENTOS

À Deus, por ter me dado a vida e por me acompanhar durante toda a minha caminhada.

Aos meus pais, Joaílso e Rosângela, por estarem presentes em todos os momentos e pelo amor incondicional dedicado por uma vida.

A minha avó Maria, por todos os ensinamentos e toda a paciência enquanto eu ainda estava descobrindo a vida. E a todos os familiares, tias, tios, primos e padrinhos que à sua maneira me ajudaram a crescer e a me tornar a pessoa que sou hoje. Sem vocês nada disso teria sido possível.

Ao meu amado Willian, pelo amor e companheirismo dispostos durante a realização deste sonho.

À minha orientadora Monalessa, pela oportunidade de realizar esse projeto e por todas as horas dedicadas a ele. Pela disponibilidade em responder minhas dúvidas e em revisar este texto, pelo vasto conhecimento transmitido e pela confiança depositada.

Aos companheiros de jornada, Higor, Mateus, Miguel e Thiago, pela amizade e por todos os obstáculos superados juntos. À professora Rosane, pelas conversas animadas e por todo o apoio durante essa caminhada. Aos demais amigos pela presença, incentivo, e pelos necessários e extremamente bem-vindos momentos de distração quando precisei.

Aos amigos e companheiros de NEMO, pela amizade e pelo ambiente descontraído e tão propício ao aprendizado. Obrigada por terem me acolhido e me deixado fazer parte do dia-a-dia de vocês.

Aos integrantes do projeto PADTEC e aos professores Maxuel e Anilton, pela grande oportunidade de aprendizado e crescimento profissional.

Aos professores Renata e Vitor, por terem se disponibilizado a ler e avaliar este projeto.

Enfim, agradeço a todos que de alguma forma contribuíram para a minha formação e para a realização deste trabalho.

RESUMO

Medição de software é uma prática importante para gerência de projetos de software e para melhoria dos processos de software. É considerada uma atividade fundamental entre as práticas de Engenharia de Software. Enquanto, no passado, muitas organizações de software não reconheciam a importância das atividades de medição, hoje ela é considerada uma prática básica da Engenharia de Software, estando presente nos principais padrões, normas e modelos que tratam melhoria de processos de software.

Dada sua relevância, medição de software tem sido tema de investigação há vários anos. Como resultado, na literatura existem vários trabalhos que abordam o tema, tais como: artigos, ferramentas, padrões, normas, *frameworks*, relatórios técnicos e outros. Esses trabalhos fornecem conhecimento sobre medição de software relevante para organizações, profissionais, estudantes e pesquisadores e, na maioria das vezes, estão registrados em diferentes fontes, o que usualmente requer tempo e esforço para encontrá-los.

Nos últimos anos, diversos trabalhos relacionados à medição de software têm sido desenvolvidos no NEMO (Núcleo de Estudos em Modelagem Conceitual e Ontologias) e, análogo ao que foi comentado anteriormente, esses trabalhos encontram-se dispersos em diferentes fontes e, algumas vezes, não estão disponíveis para acesso e uso. Além disso, durante o desenvolvimento desses trabalhos, notou-se que, muitas vezes, um mesmo material relevante para diversos trabalhos era buscado por pessoas diferentes, que despendiam tempo e esforço ao fazer isso. Se alguns materiais considerados fonte importante de conhecimento acerca de medição de software pudessem ser acessados facilmente a partir de um único ponto, o tempo e esforço necessários para encontrá-los poderia ser reduzido.

Buscando facilitar o acesso a trabalhos relacionados à medição de software desenvolvidos no NEMO, bem como a outros materiais considerados importantes fontes de conhecimento sobre esse domínio, este trabalho teve como objetivo desenvolver um Portal de Medição de Software para disponibilizar em um único ponto os diversos trabalhos desenvolvidos no NEMO, bem como outros trabalhos relevantes encontrados na literatura.

Palavras-chave: Medição de Software, Portal.

ABSTRACT

Software Measurement is an important practice for software project management and software processes improvement. It is considered a fundamental activity among the Software Engineering practices. While in the past many software organizations did not recognize the importance of measurement activities, today it is considered a basic practice in Software Engineering, being present in the main standards, norms and models that address software process improvement.

Given its relevance, software measurement has been a topic of investigation for many years. As a result, in the literature there are several works approaching this topic, such as: research papers, tools, standards, norms, frameworks, technical reports and others. These works provide relevant knowledge about software measurement for organizations, professionals, students and researchers. However, most of time, they are stored in different sources, usually demanding time and effort to access them.

In the last few years, several works related to software measurement have been developed at NEMO (Núcleo de Estudos em Modelagem Conceitual e Ontologias) and, similarly to the situation previously presented, this works are dispersed in different sources and, sometimes, are not available to access and use. Moreover, during the development of these works we have noticed that, many times, the same material is searched by different people in different works, spending time and effort. If some of the materials considered as important source of knowledge about software measurement could be easily accessed from a single point, the time and effort needed to find them could be reduced.

To facilitate accessing software measurement related works developed at NEMO, as well as other materials considered as important sources of knowledge about this domain, this work proposes a Software Measurement Web Portal to provide from a single point access to software measurement works developed at NEMO, as well as other relevant works found in literature.

Keywords: Software Measurement, Portal.

LISTA DE FIGURAS

- Figura 2.1 – Diagrama de Pacotes da Ontologia de Referência para Medição de Software
- Figura 2.2 - Modelo Comportamental Geral de OTMS.
- Figura 2.3 – Visão geral do IARM.
- Figura 2.4 – Visão Geral da Arquitetura de Referência para Medição de Software
- Figura 2.5 - Primeiras fases do processo de OBA-MSI.
- Figura 3.1 – Diagrama de Pacotes e os Subsistemas do Portal de Medição de Software.
- Figura 3.2 – Diagrama de Casos de Uso do Subsistema *gerenciaConteudo*
- Figura 3.3 – Diagrama de Casos de Uso do Subsistema *gerenciaUsuario*.
- Figura 3.4 – Diagrama de Classes do Subsistema *gerenciaConteudo*
- Figura 3.5 – Diagrama de Classes do Subsistema *gerenciaUsuario*
- Figura 3.6 – Arquitetura de Software Inicial
- Figura 3.7 – Arquitetura padrão para WIS baseada no padrão arquitetônico *Service Layer* (FOWLER, 2002).
- Figura 3.8 – Subdivisão dos subsistemas de acordo com a arquitetura do software
- Figura 3.9 – Diagrama de Classes do Componente de Domínio do Problema do subsistema *gerenciaConteudo*.
- Figura 3.10 – Modelo de aplicação genérico do *framework* nemo-utils
- Figura 3.11 – Diagrama do Componente de Gerência de Aplicação do subsistema *gerenciaConteudo*.
- Figura 3.12 - Diagrama do Componente de Controle do subsistema *gerenciaConteudo*
- Figura 3.13 - Modelo de navegação genérico do Componente de Visão do *framework* nemo-utils
- Figura 3.14 - Modelo de navegação do Componente de Visão da classe *Publication*
- Figura 3.15 – Protótipo da tela de cadastro de publicações
- Figura 3.16 - Diagrama de persistência do *framework* nemo-utils
- Figura 3.17 - Diagrama de persistência do subsistema *gerenciaConteudo*
- Figura 3.18 – Tela de *Login* do Portal de Medição de Software
- Figura 3.19 – Tela inicial do Administrador do Portal de Medição de Software
- Figura 3.20 – Tela inicial do usuário comum no Portal de Medição de Software
- Figura 3.21 – Tela de Publicações
- Figura 3.22 – Tela para cadastro de Publicações do Portal de Medição de Software
- Figura 3.23 – Tela de consulta de Publicações para o usuário do Portal de Medição de Software
- Figura 3.24 – Tela de formulário de envio de mensagem para o Administrador
- Figura 3.25 – Listagem de mensagens enviadas pelos usuários do Portal de Medição de Software
- Figura 3.26 – Tela de resposta a contato feito por usuário do Portal de Medição de Software

LISTA DE TABELAS

Tabela 2.1 – Grupos e Aspectos do CRMS

Tabela 4.1 – Objetivos específicos e sua situação na conclusão da monografia

SUMÁRIO

Capítulo 1 - Introdução	11
1.1 Introdução	11
1.2 Objetivos	13
1.3 Histórico do Desenvolvimento do Trabalho	13
1.4 Organização do Texto	14
Capítulo 2 – Medição de Software	15
2.1 Medição de Software	15
2.1.1 O Processo de Medição de Software	17
2.1.2 Medição de Software e Melhoria de Processos de Software	18
2.2 Trabalhos sobre Medição de Software Desenvolvidos no NEMO	19
2.2.1 Ontologia de Referência para Medição de Software (ORMS)	19
2.2.2 Ontologia de Tarefas para Medição de Software (OTMS)	22
2.2.3 Conjunto de Recomendações para Medição de Software (CRMS)	23
2.2.4 Instrumento para Avaliação de Repositório de Medição considerando sua Adequação ao CEP (IARM)	24
2.2.5 Arquitetura de Referência para Medição de Software (ARMS)	26
2.2.6 Abordagem Baseada em Ontologias para Integração de Ferramentas de Apoio à Medição de Software (OBA-MSI)	27
2.2.7 SoMeSPC – Ferramenta para Apoio à Medição de Software e Controle Estatístico de Processos	28
2.3 Portais Web	28
2.4 Considerações Finais do Capítulo	30
Capítulo 3 – Portal de Medição de Software	31
3.1 Requisitos e Modelagem Conceitual	31
3.1.1 Objetivo do Sistema	31
3.1.2 Descrição do Minimundo	31
3.1.3 Subsistemas do Portal de Medição de Software	33
3.1.4 Casos de Uso	33
3.1.5 Diagramas de Classes	35
3.2 Projeto do Sistema	36

3.2.1 Arquitetura de Software	37
3.2.2 Camada Lógica do Negócio	39
3.2.2.1 Componente Domínio do Problema	39
3.2.2.2 Componente de Gerência de Aplicação	40
3.2.3 Camada de Interface com o Usuário ou Lógica de Apresentação	41
3.2.3.1 Componente de Controle	42
3.2.3.2 Componente de Visão	42
3.2.4 Camada de Lógica de Acesso a Dados	44
3.3 Implementação	45
3.4 Considerações Finais do Capítulo	51
Capítulo 4 – Considerações Finais	53
4.1 Conclusões	53
4.2 Contribuições e Trabalhos Futuros	55
Referências Bibliográficas	56

Capítulo 1

Introdução

Este capítulo apresenta uma breve introdução ao tema do trabalho, seus objetivos, histórico do desenvolvimento e a organização deste documento.

1.1 Introdução

A crescente demanda do mercado por produtos e serviços de software de qualidade tem aumentado o interesse das organizações de software em melhorar seus processos, definindo estratégias e utilizando modelos, normas, padrões e *frameworks* de apoio à definição e melhoria de programas de medição.

Medição de software pode ser utilizada em vários contextos em uma organização. A medição auxilia o tomador de decisões a ter uma abordagem proativa às questões críticas inerentes aos projetos. As organizações podem, também, utilizar as informações providas pela medição para auxiliar o planejamento e avaliação dos projetos de software propostos, comparar o desempenho dos projetos correntes com seus planos, orientar os investimentos e decisões de melhoria de processos e auxiliar a prever se os projetos em andamento irão alcançar os objetivos inicialmente estabelecidos (BARCELLOS, 2008).

Medição de software também é fundamental para a melhoria de processos. Modelos de apoio à melhoria de processos de software, como, por exemplo, o CMMI (*Capability Maturity Model Integration*) (SEI, 2010) e o MR-MPS-SW (Modelo de Referência para a Melhoria do Processo de Software Brasileiro) (SOFTEX, 2012), propõem a melhoria dos processos em níveis. Nesses modelos, a medição de software é requerida desde os níveis iniciais. Nos níveis mais elevados, conhecidos como alta maturidade, o controle estatístico de processos (CEP) é requerido como uma evolução da medição de software iniciada nos níveis anteriores. À medida que o nível de maturidade dos processos aumenta, novas necessidades de informação são identificadas e a medição deve atendê-las. Dessa forma, o processo de medição e seus resultados orientam a evolução nos níveis de maturidade de melhoria de processos (DUMKE *et al.*, 2004).

Considerando a importância do processo de medição para as organizações de software, diversos trabalhos têm sido desenvolvidos. Em (BARCELLOS, 2009) foi proposta uma estratégia para auxiliar as organizações na preparação e implementação da medição de software

e controle estatístico de processos. Essa estratégia é composta por três componentes (BARCELLOS, 2009; BARCELLOS *et al.*, 2013): uma Ontologia de Referência para Medição de Software (BARCELLOS *et al.*, 2010a; BARCELLOS *et al.*, 2010b; c), que provê o vocabulário e a conceituação para o domínio medição de software, abrangendo aspectos tanto da medição tradicional (caracterizada nos níveis iniciais dos modelos de maturidade) quanto em alta maturidade; um Instrumento para Avaliação da Adequação de Bases de Medidas para o CEP (BARCELLOS *et al.*, 2010d; e), que apoia a avaliação e adequação de bases de medidas para o CEP; e um Conjunto de Recomendações para Medição de Software Adequada ao CEP (BARCELLOS *et al.*, 2013), que provê conhecimento para auxiliar as organizações a realizarem medição de software adequada ao CEP desde os níveis iniciais de maturidade.

A partir do trabalho de BARCELLOS (2009), alguns trabalhos relacionados à medição de software foram desenvolvidos no NEMO (Núcleo de Estudos em Modelagem Conceitual e Ontologias), grupo de pesquisa do Departamento de Informática da Universidade Federal do Espírito Santo. Embora esses trabalhos tenham o objetivo comum de apoiar aspectos relacionados à medição, eles foram desenvolvidos de maneira independente e não foram disponibilizadas de forma a facilitar seu acesso e uso. Além disso, embora trabalhos como monografias e dissertações estejam disponíveis no site do NEMO, vários artigos, relatórios e outros documentos têm sido produzidos e não possuem um ponto único de acesso. Alguns materiais encontram-se disponibilizados no site pessoal dos autores, outros em bibliotecas digitais e outros não estão disponibilizados para acesso público.

Além dos trabalhos desenvolvidos no NEMO, há na literatura diversos trabalhos que representam importantes fontes de conhecimento sobre medição de software. Esses trabalhos encontram-se dispersos em várias fontes, o que usualmente demanda esforço e tempo para encontrá-los.

Nos últimos anos, portais na web têm sido utilizados como meio para disponibilização de diversos tipos de informação. Eles oferecem um ponto único de acesso a uma vasta gama de informações, aplicações e serviços em um único ambiente (TATE *et al.*, 2007). O objetivo da criação de portais é prover, de forma eficiente, informações e serviços disponíveis em várias fontes para vários usuários. Nesse sentido, acredita-se que a criação de um portal para disponibilizar os trabalhos relacionados à medição de software desenvolvidos no NEMO, bem como trabalhos relevantes registrados na literatura sobre o tema, pode auxiliar o acesso a esse conteúdo por parte organizações, estudantes, pesquisadores e outros interessados em medição

de software. Cabe dizer que o conhecimento da autora deste trabalho sobre o tema medição de software, adquirido ao longo de um projeto de Iniciação Científica, reforçou a motivação para desenvolver um portal relacionado a medição de software.

1.2 Objetivos

O objetivo geral deste trabalho é *desenvolver um Portal de Medição de Software onde os diversos trabalhos relacionados à medição desenvolvidos no NEMO, bem como outros materiais que abordam medição de software (artigos, monografias, dissertações, teses, livros, relatórios técnicos, etc.), produzidos ou não no NEMO, sejam disponibilizados para consulta e uso por parte de pessoas e organizações interessadas em obter conhecimento sobre medição de software e apoio para sua implementação*. São objetivos específicos deste trabalho:

- i. Identificar trabalhos relevantes à medição de software para serem disponibilizados no Portal de Medição de Software.
- ii. Realizar a modelagem conceitual do Portal de Medição, incluindo aspectos comportamentais e estruturais.
- iii. Definir uma arquitetura para o Portal de Medição.
- iv. Implementar uma versão inicial para o Portal de Medição.

1.3 Histórico de Desenvolvimento do Trabalho

Para desenvolvimento deste trabalho foram realizadas as seguintes etapas:

(i) *Revisão Bibliográfica*: O trabalho teve início com uma revisão bibliográfica sobre medição de software e portais, na qual foram lidos materiais (livro, tese, dissertações e artigos científicos) relevantes ao trabalho.

(ii) *Estudo de Tecnologias*: Nesta etapa foi realizado o estudo de tecnologias relevantes para o desenvolvimento do Portal de Medição de Software, destacando-se: Linguagem de Programação JAVA; Linguagem de programação para web Javascript, Ambiente de Desenvolvimento *Eclipse EE IDE for Web Developers*; Servidor Web *Wildfly*; Banco de Dados *MySQL Workbench*; *Hibernate* (*framework* que realiza o mapeamento Objeto/Relacional); *CDI* (*framework* que provê uma fábrica de *beans* com injeção automática de dependências); *JSF* (*framework* responsável pela criação das interfaces gráficas); e *JPA* (*Java Persistence API*).

(iii) *Elaboração da Documentação do Portal de Medição de Software*: Nesta etapa foi criada a documentação do sistema. Inicialmente foi elaborado o Documento de Requisitos, apresentando uma descrição geral do minimundo do sistema e definição dos requisitos e regras de negócio. Em seguida, foram elaborados a Especificação de Requisitos e o Projeto do Sistema, contendo subsistemas, casos de uso, modelo estrutural, arquitetura e projeto detalhado de cada componente da arquitetura.

(iv) *Implementação e Testes da Ferramenta*: Nesta etapa o Portal de Medição de Software foi implementado e testado, tendo sido realizados testes ao longo do desenvolvimento, bem como após a conclusão de funcionalidades.

(v) *Elaboração da Monografia*: Nesta etapa foi realizada a escrita desta monografia.

1.4 Organização do Texto:

Neste capítulo foi apresentada uma introdução, contendo uma breve descrição do contexto em que o trabalho está inserido e a motivação que levou ao seu desenvolvimento. Além deste capítulo, a monografia possui outros três, a saber:

Capítulo 2 – Medição de Software: Apresenta uma breve fundamentação teórica sobre medição de software e portais web, incluindo informações sobre trabalhos relacionados à medição de software desenvolvidos no NEMO e disponibilizados no Portal de Medição de Software.

Capítulo 3 – Portal de Medição de Software: Apresenta os requisitos do sistema, a descrição do minimundo contemplado pelo sistema, seus casos de uso, modelos de classe, arquitetura definida, detalhes dos componentes e algumas telas do sistema.

Capítulo 4 – Considerações Finais: Apresenta as considerações finais do trabalho, incluindo algumas dificuldades encontradas, contribuições e experiência adquirida em seu desenvolvimento. Também são identificados alguns trabalhos futuros.

Capítulo 2

Medição de Software e Portais Web

Este capítulo apresenta os principais aspectos teóricos que fundamentam este trabalho. Ele está organizado em 3 seções, a saber: a Seção 2.1 aborda medição de software; a Seção 2.2 apresenta trabalhos relacionados a medição de software desenvolvidos no NEMO; a Seção 2.3 fala sobre portais web; e, por fim, na Seção 2.4 são apresentadas as considerações finais do capítulo.

2.1 Medição de Software

Medição de software é uma avaliação quantitativa de qualquer aspecto dos processos e produtos da Engenharia de Software, que permite seu melhor entendimento e, com isso, auxilia o planejamento, controle e melhoria do que se produz e de como é produzido (BASS et al., 1999). O objetivo mais importante de sua aplicação é prover informação quantitativa para apoiar a tomada de decisões (FENTON e NEIL, 2000).

A medição de software começou a ser praticada na década de 70, inicialmente apenas para medir o número de linhas de código dos programas produzidos. Na década de 80, outras medidas – relacionadas às fases finais do desenvolvimento – começaram a ser utilizadas, porém os objetivos da realização da medição nas organizações não eram explícitos ou, se eram, não eram compreensíveis aos seus membros, resultando em medições inúteis não alinhadas às necessidades das organizações ou dos projetos. Nos anos 90, impulsionados por algumas aplicações bem-sucedidas, foram desenvolvidos modelos para o processo de medição baseados na melhoria dos processos e nos princípios da qualidade total, fornecendo as diretrizes e a infraestrutura básica para definir, coletar, validar e analisar medidas (BASS *et al.*, 1999; FENTON e NEIL, 1999).

Atualmente, a medição de software é um dos temas mais importantes na Engenharia de Software. Enquanto, no passado, muitas organizações de software não reconheciam a importância das atividades de medição e as tratavam apenas como “mais uma coisa a ser feita”, hoje ela é considerada uma prática básica da Engenharia de Software, sendo evidenciada por sua inclusão nos requisitos do nível 2 do CMMI (CHRISIS *et al.*, 2006), através da área de processo Medição e Análise e do nível F do MR-MPS-SW (SOFTEX, 2012) (BARCELLOS, 2009).

Para que a medição de software seja eficientemente realizada em uma organização e dê o retorno necessário para ser percebida como prática fundamental à sobrevivência e ao crescimento organizacional, sua implementação deve ser orientada para apoiar a tomada de decisão nos âmbitos técnico e de negócios. Segundo McGARRY *et al.* (2002), esse é o principal diferencial entre as organizações que se beneficiam com os resultados de seu programa de medição e as organizações que simplesmente despendem tempo e esforço para acumular dados inúteis. Ainda segundo esses autores, a maioria das organizações de software bem-sucedidas implementa a medição como parte de suas atividades, englobando os níveis técnico, gerencial e estratégico. Nessas organizações a medição provê a informação necessária às tomadas de decisão que impactam no desempenho técnico e de negócio, sendo essa informação disponibilizada para os tomadores de decisão em todos os níveis da organização, de acordo com seus objetivos (FENTON e NEIL, 2000; PUTNAM e MYERS, 2003; BRIMSON, 2004 *apud* BARCELLOS, 2009).

Além de apoiar fortemente a tomada de decisão, a medição auxilia e acelera o aprendizado organizacional, uma vez que a análise dos dados coletados nos projetos provê a fundação necessária para o aprendizado através de cada projeto e, conseqüentemente, para o aprendizado organizacional. Também auxilia a organização a perceber e entender as diferenças entre o seu desempenho e o desempenho exigido pelo mercado, permitindo que ela otimize seus processos técnicos e de negócio quando necessário. Isso é possível, pois as medidas coletadas nos projetos da organização podem ser combinadas e, através da utilização de diferentes técnicas, analisadas para satisfazer as diferentes necessidades de informação organizacionais (GOPAL *et al.*, 2002; BRIMSON, 2004).

Um gerente de projetos pode, por exemplo, utilizar as informações providas pela medição para realizar uma comunicação mais eficiente, traçar os objetivos específicos dos projetos, identificar e corrigir problemas antecipadamente, tomar decisões chave e justificar tais decisões. Mas é importante ressaltar que, assim como qualquer outra ferramenta gerencial ou técnica, a medição não é capaz de garantir o alcance dos objetivos. Entretanto, ela é capaz de auxiliar os tomadores de decisão a terem uma abordagem proativa às questões críticas inerentes aos projetos, o que contribui para que os objetivos sejam alcançados (BARCELLOS, 2009).

As organizações podem, ainda, utilizar as informações providas pela medição para elaborar planos realistas para os projetos, comparar o desempenho dos projetos correntes com seus planos, orientar os investimentos e decisões de melhoria de processos e auxiliar a prever

se os projetos em andamento irão alcançar os objetivos inicialmente estabelecidos (NIESSINK e VLIET, 2001). Segundo McGARRY *et al.* (2002), essas são ações comuns em organizações de software maduras, onde há utilização da medição em todo o ciclo de vida do software e as informações obtidas são consideradas e utilizadas como recurso estratégico na condução desse ciclo.

Somente quando as informações obtidas na análise dos dados coletados são utilizadas para direcionar as ações necessárias às organizações e seus projetos é que o objetivo fundamental da medição é alcançado e percebido pelas organizações, fator que contribui para a real institucionalização de um programa de medição eficiente (BARCELLOS, 2009).

2.1.1 O Processo de Medição de Software

O processo de medição de software pode ser definido como um conjunto de passos que deve orientar a realização da medição em uma organização. Um processo de medição eficiente é fator crítico ao sucesso da medição na organização, pois é ele que direciona as atividades a serem realizadas para que com os resultados da análise dos dados coletados seja possível a identificação de tendências e antecipação aos problemas, a fim de prover melhor controle dos custos, redução dos riscos, melhoria da qualidade e, conseqüentemente, alcance dos objetivos técnicos e de negócio (WANG e LI, 2005).

Para realizar a medição de software, inicialmente, a organização deve planejá-la. Baseado em seus objetivos, a organização deve definir que entidades (processos, produtos e etc.) a considerar para a medição de software e quais de suas propriedades (tamanho, custo, tempo, etc.) devem ser medidas. A organização também deve definir que medidas serão usadas para quantificar esses elementos. Para cada medida, uma definição operacional deve ser especificada, indicando, entre outros, como a medida deve ser coletada e analisada. Uma vez planejada, a medição pode começar. A execução da medição envolve coletar dados para as pedidas definidas, de acordo com suas definições operacionais. Uma vez que os dados são coletados, eles devem ser analisados, também seguindo as orientações estabelecidas pelas definições operacionais correspondentes. Finalmente, o processo de medição e seus produtos devem ser avaliados para identificar potenciais melhorias (BARCELLOS *et al.*, 2010).

2.1.2 Medição de Software e Melhoria de Processos de Software

Melhoria de processos de software é uma abordagem para definição, organização e implementação de processos de software que sejam eficientes para uma organização (KILPI, 2001). Implantar melhoria de processos em uma organização baseia-se fundamentalmente na identificação e realização das mudanças que levam à melhoria dos processos. Inicialmente a medição é utilizada para identificar as necessidades de mudanças. Posteriormente, quando as mudanças são realizadas, a medição é utilizada para avaliar os resultados das alterações. Em outras palavras, a melhoria de processos é um ciclo contínuo e a medição é um de seus principais pilares (BARCELLOS, 2009).

Alguns modelos de apoio à melhoria de processo de software, como o CMMI (SEI, 2010) e o MR-MPS-SW (SOFTEX, 2012), orientam a melhoria de processos em níveis. Nos níveis iniciais (níveis 2 e 3 do CMMI e níveis G a C do MR-MPS-SW) é realizada a medição dita tradicional, que envolve o uso de técnicas estatísticas usuais para a análise de dados e comparação de valores reais com planejados. Na alta maturidade (níveis 4 e 5 do CMMI e níveis B e A do MR-MPS-SW) é realizado o controle estatístico de processos, a fim de entender o comportamento dos processos, prever seu desempenho nos projetos e prever se serão capazes de alcançar os objetivos estabelecidos.

Utilizar a medição em um contexto de alta maturidade na melhoria de processos de software requer que novos conceitos e práticas sejam adicionados aos programas de medição tradicionais, uma vez que a melhoria de processos em alta maturidade requer conhecimento consistente do comportamento dos processos em suas execuções nos projetos da organização (BARCELLOS, 2009).

Conhecer e controlar o comportamento dos processos é de suma importância para as organizações de software, pois os processos que elas utilizam para produzir seus produtos e serviços têm papel crítico na execução dos planos e estratégias que buscam alcançar os objetivos organizacionais. Organizações que são capazes de controlar seus processos são capazes de prever a qualidade de seus produtos e serviços, custos, cronogramas e melhorar a eficiência, eficácia e rentabilidade de suas operações técnicas e de negócios (BRIMSON, 2004).

O conhecimento e o controle do comportamento dos processos têm início na realização das atividades de medição. Considerando as medidas coletadas, a aplicação de métodos e técnicas apropriados é capaz de fornecer as informações necessárias para orientar o alcance dos objetivos técnicos e de negócio, uma vez que identifica problemas, tendências e dá

embasamento às tomadas de decisão, principalmente àquelas que dizem respeito à melhoria de desempenho dos processos (BARCELLOS, 2009).

Um processo sob controle é um processo estável e, como tal, tem comportamento que pode ser replicado. Assim, é possível prever seu desempenho em futuras execuções e, assim, preparar planos atingíveis e melhorar o processo continuamente. Por outro lado, um processo que varia além dos limites esperados é um processo instável e as causas dessas variações (ditas causas especiais) devem ser investigadas e tratadas por ações de melhoria a fim de estabilizar o processo. Uma vez que os processos são estáveis, seus níveis de variação podem ser estabelecidos e mantidos, sendo possível prever seus resultados. Assim, também é possível identificar os processos que são capazes de alcançar as metas estabelecidas e os processos que estão falhando no cumprimento das metas. Nesse caso, ações para mudar o processo de modo a torna-lo capaz devem ser realizadas (FLORAC *et al*, 1999).

Primeiro, é necessário entender os objetivos de negócio registrados no planejamento estratégico. Em seguida, os processos relacionados aos objetivos de negócio são identificados e as medidas utilizadas para fornecer informações quantitativas sobre o desempenho dos processos são definidas. Dados são coletados para as medidas, armazenados e utilizados na análise do comportamento dos processos usando técnicas estatísticas. Se um processo é instável, as causas especiais devem ser tratadas. Se ele é incapaz, deve ser mudado. Finalmente, se é capaz, ele pode ser melhorado continuamente (FLORAC *et al*, 1999).

2.2 Trabalhos sobre Medição de Software desenvolvidos no NEMO

Nesta seção são apresentadas informações sobre os principais trabalhos relacionados a medição de software desenvolvidos nos últimos anos no NEMO e que serão disponibilizados no Portal de Medição de Software.

2.2.1 Ontologia de Referência para Medição de Software (ORMS)

Uma ontologia de referência de domínio é um tipo especial de modelo conceitual, representando um modelo de consenso dentro de uma comunidade. É uma solução independente de especificação, com o objetivo de fazer uma descrição clara e precisa das entidades do domínio para propósitos de comunicação, aprendizado e resolução de problemas. Idealmente, uma ontologia de referência deve ser construída baseada em distinções ontológicas

por definidas em uma ontologia de fundamentação (GUARINO, 1998 *apud* GUIZZARDI, 2005).

A Ontologia de Referência para Medição de Software (ORMS) é uma ontologia de referência de domínio fundamentada na *Unified Foundational Ontology* (UFO) (GUIZZARDI, 2005). Ela foi desenvolvida com o propósito de prover um vocabulário comum e conhecimento relevante sobre o domínio de Medição de Software, incluindo aspectos da medição tradicional e da medição em alta maturidade (BARCELLOS *et al.*, 2013).

ORMS é baseada em diversos padrões (tais como CMMI (SEI, 2010), ISO/IEC 15939 (ISO/IEC 15939, 2007), Practical Software Measurement (PSM) (McGARRY *et al.*, 2002) e IEEE Std. 1061 (IEEE, 1998)), e em aspectos de medição de software de alta maturidade. Uma vez que o domínio de medição de software está fortemente relacionado aos domínios de processos de software e organizações, a ORMS reutiliza partes da conceituação descritas em uma ontologia de processos de software (BRIGUENTE *et al.*, 2011) e em uma ontologia de organização de software (VILLELA *et al.*, 2005) (FALBO *et al.*, 2014).

ORMS é subdividida em oito subontologias. A Figura 2.1 apresenta um diagrama de pacotes com as subontologias da ORMS, as ontologias integradas e as relações entre elas. Em seguida, é feita uma breve descrição das subontologias.

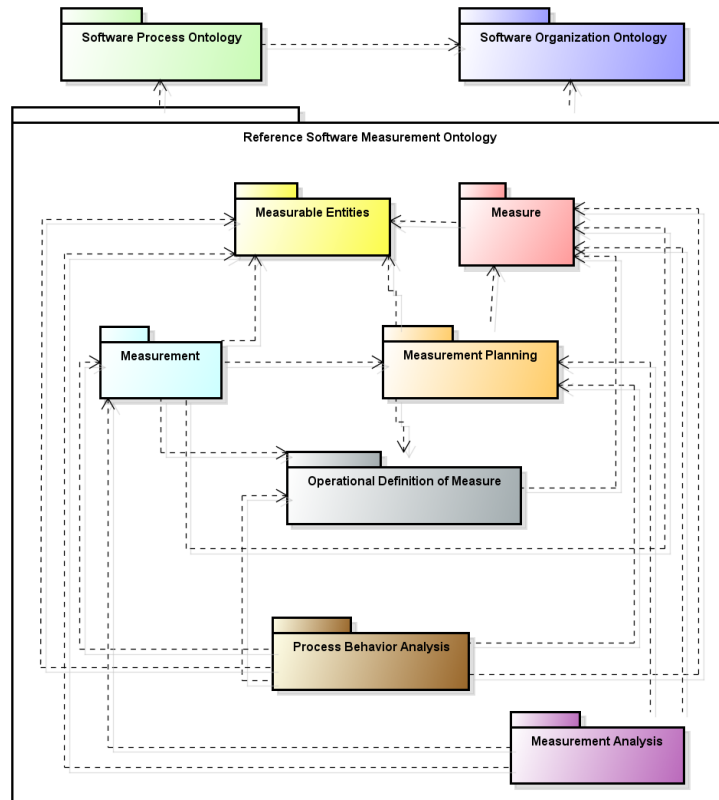


Figura 2.1 – Diagrama de Pacotes da Ontologia de Referência para Medição de Software

- *Subontologia de Entidades Mensuráveis (**Measurable Entities**)*: trata das entidades que podem ser submetidas à medição e das suas propriedades que podem ser medidas.
- *Subontologia de Medida (**Measure**)*: trata de medidas de software e do relacionamento das medidas com as entidades e elementos mensuráveis.
- *Subontologia de Definição Operacional de Medidas (**Operational Definition of Measure**)*: trata do detalhamento de aspectos relacionados à coleta e análise de medidas.
- *Subontologia de Planejamento de Medição (**Measurement Planning**)*: trata de aspectos relacionados ao planejamento da medição de software.
- *Subontologia de Medição (**Measurement**)*: trata da medição propriamente dita, ou seja, a coleta e armazenamento dos dados para as medidas.
- *Subontologia de Análise de Medição (**Measurement Analysis**)*: trata da análise dos dados coletados para as medidas para obtenção das informações de apoio às decisões.

- *Subontologia de Análise de Comportamento de Processos (Process Behavior Analysis)*: trata da aplicação da análise dos dados coletados na análise de comportamento de processos utilizando controle estatístico de processos.

2.2.2 Ontologia de Tarefa de Medição de Software (OTMS)

Apesar do crescente uso de ontologias de domínio, o mesmo não ocorre com ontologias de tarefas (MARTINS; FALBO, 2008). Isso é um problema, já que em termos de funcionalidade e integração de processos de aplicações de software, ontologias de tarefas podem ser usadas para atribuir significado a serviços, funcionalidades, atividades e suas informações relacionadas. No caso de integração de diferentes padrões, novamente, ontologias de tarefas podem ser usadas para atribuir significado a atividades e boas práticas (BARCELLOS; FALBO, 2013).

Ontologias de tarefa devem capturar dois tipos principais de conhecimento (MARTINS; FALBO, 2008): (i) o conhecimento comportamental, representando a decomposição das tarefas e fluxos de controle, e (ii) o conhecimento estrutural, que trata dos papéis desempenhados pelas entidades do domínio no cumprimento da tarefa. Dessa forma, modelos diferentes são necessários para representar tais conhecimentos.

Com o objetivo de estabelecer uma conceituação comum a respeito do processo de medição de software, foi desenvolvida a Ontologia de Tarefas de Medição de Software (OTMS), que é fundamentada na *Unified Foundational Ontology* (UFO) (GUIZZARDI, 2005). OTMS pode ser utilizada, entre outros, para tratar problemas de interoperabilidade semântica decorrentes da utilização conjunta de diferentes padrões relacionados à medição, bem como apoiar a integração semântica de aplicações de software que suportam medição (BARCELLOS; FALBO, 2013).

Seguindo a abordagem proposta em (MARTINS; FALBO, 2008), a OTMS é representada por diagramas de atividades da UML, que tratam o aspecto comportamental da tarefa, e por diagramas de classes, que modelam os conceitos envolvidos na tarefa e seus relacionamentos. A OTMS foi desenvolvida buscando um alinhamento com a ORMS. Dessa forma, o modelo estrutural da OTMS é equivalente aos modelos conceituais da ORMS. Vale destacar que, embora seja usado o termo ontologia de tarefa (consagrado na área de ontologias), na verdade, a OTMS é uma ontologia de processo, uma vez que ela descreve o processo de medição como um todo e não apenas tarefas com baixo nível de granularidade (BARCELLOS; FALBO, 2013).

O modelo comportamental geral de OTMS é apresentado na Figura 2.2.

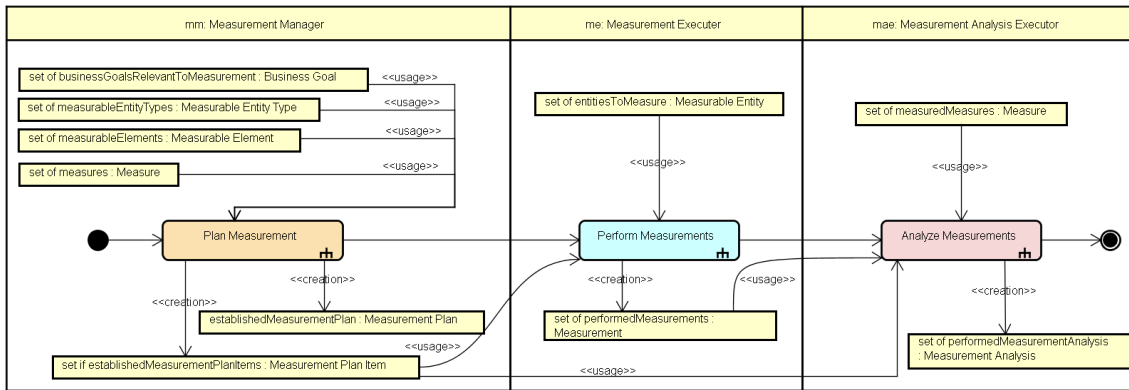


Figura 2.2 - Modelo Comportamental Geral de OTMS.

2.2.3 Conjunto de Recomendações para Medição de Software (CRMS)

O crescente interesse por parte das organizações de software em alcançar a alta maturidade tem motivado a utilização do controle estatístico de processos de software (CEP). O CEP foi originalmente proposto para apoiar programas de melhoria de processos na área da manufatura. Apesar de não ser novidade para a indústria em geral, seu uso em organizações de software é considerado recente e há, ainda, dúvidas sobre sua implementação (CARD *et al.*, 2008).

O CRMS foi desenvolvido com o objetivo de reunir informações que auxiliem as organizações a realizarem medição de software de forma adequada ao CEP. Apesar de o CRMS visar à medição em alta maturidade, há várias recomendações que lidam com a medição realizada nos níveis iniciais. Isso ocorre, pois grande parte dos problemas relacionados ao uso do CEP tem origem nos níveis iniciais. Frequentemente, esses problemas têm menor impacto nos resultados da medição tradicional e, por isso, apenas são percebidos quando as práticas do CEP têm início (BARCELLOS *et al.*, 2014).

O CRMS apresenta recomendações para 19 aspectos divididos em 5 grupos. A Tabela 2.1 apresenta os grupos e aspectos tratados no CRMS.

Tabela 2.1 – Grupos e Aspectos do CRMS

Grupos	Aspectos
Preparação para Medição de Software	Criação do Repositório de Medidas Caracterização dos Projetos Identificação de Projetos Similares Identificação das Versões dos Processos
Alinhamento da Medição aos Objetivos da Organização	Identificação de Objetivos de Medição Identificação de Necessidades de Informação de acordo com Objetivos de Medição Identificação de Medidas para atender as Necessidades de Informação
Definição de Medidas de Software	Definição Operacional de Medidas Nível de Granularidade de Medidas Normalização de Medidas Medidas Correlatas
Realização de Medições de Software	Execução de Medições Consistentes Validação de Dados Coletados Registro de Contexto de Medição Volume de Dados Coletados
Análise de Medições de Software	Periodicidade de Análise de Medição Agrupamento de Dados para Análise Identificação de <i>Baseline</i> de Desempenho de Processos Determinação da Capacidade de Processos

O CRMS foi originalmente definido no formato de guia e, em seguida, foi desenvolvida uma ferramenta de apoio à utilização e avaliação das suas recomendações (LEÃO, 2012).

2.2.4 Instrumento para Avaliação de Repositório de Medição considerando sua Adequação ao CEP (IARM)

O IARM visa avaliar o grau de adequação de repositórios de medição ao CEP e prover ações de adequação quando é possível adequar um repositório para ser aplicado nesse contexto. O instrumento foi definido no formato de um guia, composto por um conjunto de *checklists*, orientações para a avaliação de cada requisito presente nos *checklists* e ações de adequação necessárias caso um determinado requisito não seja atendido. A Figura 2.3 apresenta uma visão geral do IARM.

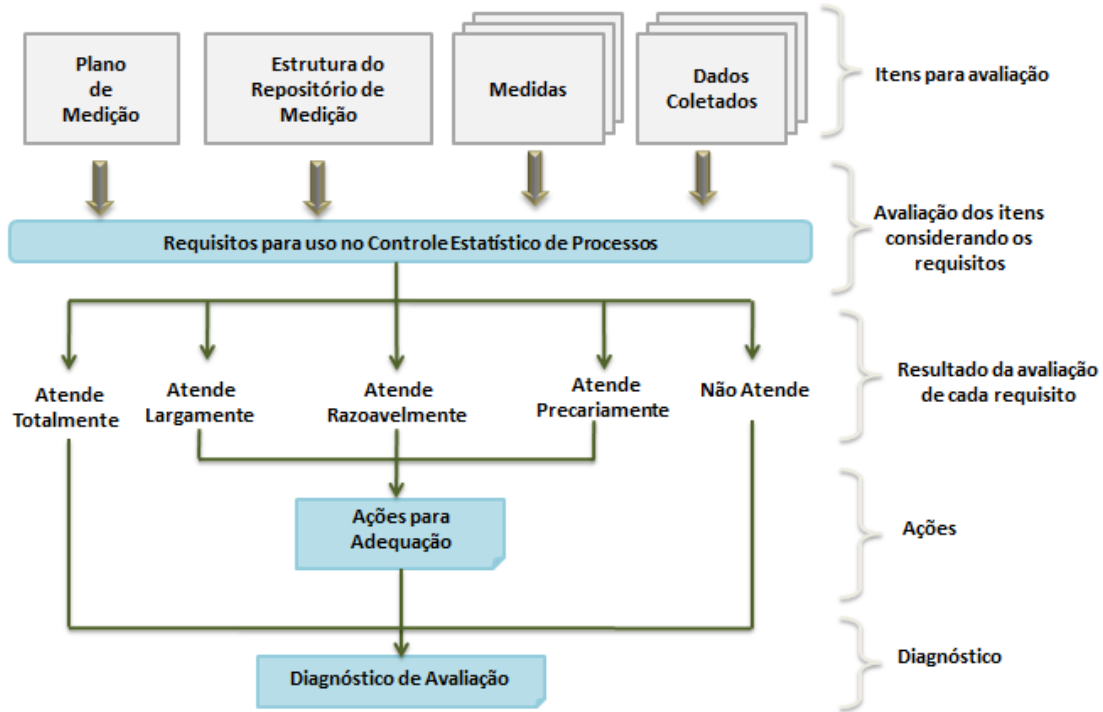


Figura 2.3 – Visão geral do IARM.

O IARM considera a avaliação de quatro itens: o **Plano de Medição**, a **Estrutura do Repositório de Medição**, as **Medidas** propriamente ditas e os **Dados Coletados** para essas medidas. Cada item é submetido à avaliação considerando-se um conjunto de requisitos. A avaliação de cada item segundo cada requisito pode produzir um dos seguintes resultados:

- (i) *Atende Totalmente*: o item avaliado satisfaz totalmente o requisito e nenhuma ação de alteração é necessária em relação ao requisito considerado.
- (ii) *Atende Largamente, Atende Razoavelmente ou Atende Precariamente*: o item avaliado não satisfaz o requisito, mas é possível realizar ações que irão adequá-lo a fim de satisfazer o requisito em questão. O grau de atendimento do item ao requisito (Largamente, Razoavelmente ou Precariamente) está diretamente relacionado com o esforço necessário para realizar as ações que levarão o item a atender o requisito em questão. Quanto mais esforço, menor grau de atendimento.
- (iii) *Não Atende*: o item avaliado não satisfaz o requisito e não há ações possíveis para adequá-lo.

Quando o resultado da avaliação de um requisito é *Atende Largamente*, *Atende Razoavelmente* ou *Atende Precariamente*, são sugeridas *Ações para Adequação*. Essas ações

são orientações que visam auxiliar na realização de correções que permitam a utilização do item avaliado no controle estatístico de processos.

Os resultados da avaliação são registrados em um Diagnóstico de Avaliação, que inclui, a avaliação detalhada de cada item, ações de adequação sugeridas e o grau de adequação de cada item e do repositório de medição como um todo ao controle estatístico de processos, dado em percentual.

2.2.5 Arquitetura de Referência para Medição de Software (ARMS)

Em (MARETTO, 2013) foi proposta uma Arquitetura de Referência para Medição de Software, a qual foi desenvolvida com base na Ontologia de Referência para Medição de Software apresentada anteriormente. A arquitetura é independente de tecnologia e pode ser utilizada para que organizações definam suas próprias arquiteturas, considerando os aspectos tecnológicos a elas inerentes.

A Figura 2.4 apresenta a visão geral da Arquitetura de Referência para Medição de Software. Dados provenientes da organização, dos projetos e de seus artefatos são capturados por meio de utilitários de entrada e são armazenados em um repositório de medição. Esses dados são recuperados, analisados e os resultados são apresentados aos interessados por meio dos utilitários de saída (MARETTO, 2013), ou seja, ferramentas computacionais que apoiam a apresentação de dados.

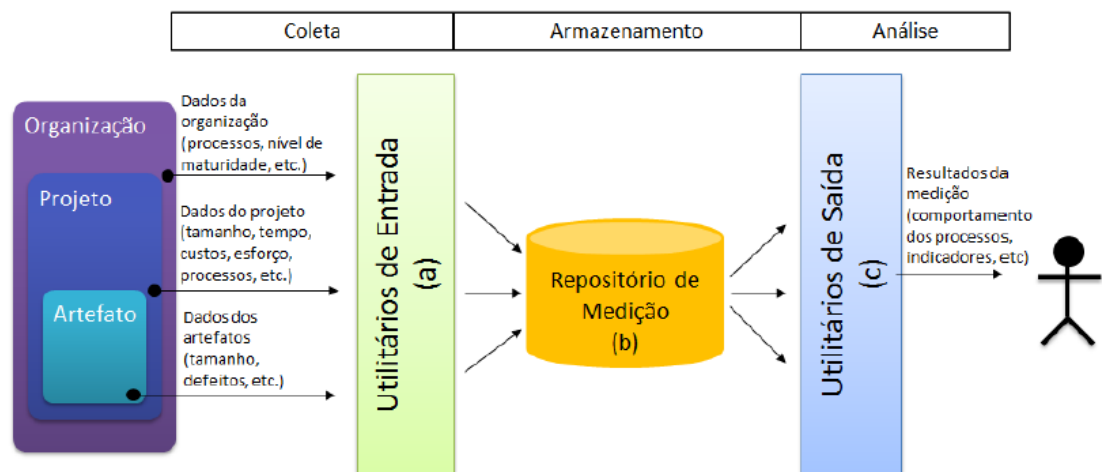


Figura 2.4 – Visão Geral da Arquitetura de Referência para Medição de Software

Conforme mostra a Figura 2.2, a arquitetura de referência para medição é composta por três componentes:

- (a) *Utilitários de Entrada*: componente responsável pela captura dos dados relevantes à medição de software, a saber: dados referentes à organização como um todo, aos projetos e aos artefatos produzidos nos projetos. Em uma solução computacional para medição de software, os utilitários de entrada devem possuir um conjunto de funcionalidades que permita a captura dos dados necessários à medição de software. Na arquitetura de referência, esse componente é descrito por meio de requisitos funcionais.
- (b) *Repositório de Medição*: componente responsável pelo armazenamento dos dados de medição. Tem como base principal a Ontologia de Referência para Medição de Software e é descrito por meio de modelos de classes em UML (*Unified Modeling Language*), descrições detalhadas desses modelos e restrições de integridade.
- (c) *Utilitários de Saída*: componente responsável pela análise dos dados e apresentação dos resultados para apoio à tomada de decisão. É descrito por meio de requisitos funcionais.

2.2.6 Abordagem Baseada em Ontologias para Integração de Ferramentas de Apoio à Medição de Software (OBA-MSI)

Em (FONSECA, 2015) é proposta OBA-MSI (*Ontology-Based Approach for Measurement Systems Integration*), uma especialização de OBA-SI (*Ontology-Based Approach for Semantic Integration*) (CALHAU, 2011) que visa auxiliar organizações a estabelecer um processo de medição adequado e definir apoio ferramental adequado para ele por meio da integração de ferramentas, a qual é conduzida com base na Ontologia de Referência de Medição de Software (ORMS) e na Ontologia de Tarefa de Medição de Software (OTMS). Na Figura 2.5 é apresentada uma visão geral das primeiras fases de OBA-MSI, que são as fases que diferem de OBA-SI. As fases seguintes (*Projetar Integração, Implementar Integração, Testar Integração e Implantar Conjunto Integrado*) não apresentadas na figura, são idênticas às fases de OBA-SI.

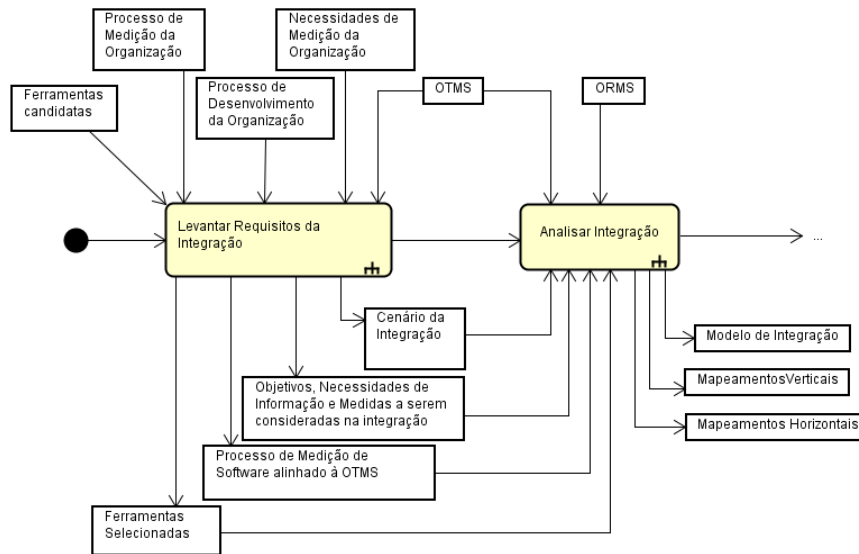


Figura 2.5 - Primeiras fases do processo de OBA-MSI.

2.2.7 SoMeSPC – Ferramenta para Apoio à Medição de Software e Controle Estatístico de Processos

Em (MARETTO, 2013) foi desenvolvida uma ferramenta a partir da Arquitetura de Referência para Medição de Software (ARMS) apresentada anteriormente. SoMeSPC é uma ferramenta capaz de apoiar o processo de medição considerando aspectos tanto da medição tradicional quanto da alta maturidade.

Em (FONSECA, 2015) e (CASTRO, 2016) a ferramenta foi evoluída para permitir a disponibilização de serviços e facilitar sua integração com outras ferramentas, visando potencializar a realização de medições automáticas. Dessa forma, a ferramenta pode ser utilizada isoladamente ou integrada a outras.

2.3 Portais Web

Repositórios digitais e bibliotecas digitais estão hoje entre as ferramentas mais comuns para gerenciamento e disseminação de coleções digitais de objetos de âmbito cultural, educacional e outros tipos de conteúdo na Web (KOUTSOMITROPOULOS *et al.*, 2010).

Um portal Web é um site que agrega informações de várias fontes na web e organiza o material de uma maneira fácil e amigável (XIAO *et al.*, 2005). Ao longo da última década, o número de organizações que fornecem portais da Web tem crescido dramaticamente. Essas organizações fornecem portais que complementam, substituem ou estendem os serviços existentes para sua base de clientes (YANG *et al.*, 2004).

Numerosos usuários em todo o mundo usam portais da Web para obter informações para seu trabalho e para ajudar com a tomada de decisão. Esses usuários, ou consumidores de dados, precisam garantir que os dados obtidos são adequados para suas necessidades. Da mesma forma, as organizações que fornecem portais da Web precisam oferecer dados que atendam às necessidades dos usuários, ajudando-os a atingir seus objetivos (CALERO *et al.*, 2008).

De acordo com (WELLS *et al.*, 2000), um portal ideal é baseado em oito áreas de funcionalidades: Pesquisa e Navegação, Integração de Informação (Gestão de Conteúdo), Personalização, Notificação (Tecnologia *Push*), Gerenciamento de Tarefas e Fluxo de Trabalho, Colaboração e *Groupware*, Integração de aplicações e *Business Intelligence*, e Funcionalidades de Infraestrutura.

Considerando o contexto de utilização de um portal, ele pode ser classificado como público ou corporativo (DIAS, 2001). O **portal público**, também denominado portal Internet, portal *web* ou portal de consumidores, provê ao consumidor uma única interface à imensa rede de servidores que compõem a Internet. Sua função é atrair, para o seu *site*, o público em geral que navega na Internet. Quanto maior o número de visitantes, maior a probabilidade do estabelecimento de comunidades virtuais que potencialmente comprarão o que os anunciantes daquele *site* têm para vender. Assim como a televisão, o rádio e a mídia impressa, o portal público estabelece um relacionamento unidirecional com seus visitantes e constitui-se em uma mídia adicional para o *marketing* de produtos (DIAS, 2001).

O **portal corporativo**, no mundo institucional, tem o propósito de expor e fornecer informações específicas de negócio, dentro de determinado contexto, auxiliando os usuários de sistemas informatizados corporativos a encontrar as informações de que precisam para fazer frente aos concorrentes (REYNOLDS e KOULOPOULOS, 1999). O portal corporativo é considerado por (REYNOLDS e KOULOPOULOS, 1999) como uma evolução do uso das Intranets, incorporando a essa tecnologia novas tecnologias que possibilitam identificação, captura, armazenamento, recuperação e distribuição de grandes quantidades de informações de múltiplas fontes, internas e externas, para os indivíduos e equipes de uma instituição (DIAS, 2001).

2.4 Considerações Finais do Capítulo

Para tratar de temas fundamentais ao entendimento deste trabalho, este capítulo apresentou o conteúdo relacionado a medição de software, trabalhos desenvolvidos no NEMO e portais web. Foram apresentados a Ontologia de Referência para Medição de Software, a Ontologia de Tarefas para Medição de Software, o Conjunto de Recomendações para Medição de Software, o Instrumento para Avaliação de Repositório de Medição considerando sua Adequação ao CEP, a Abordagem Baseada em Ontologias para Integração de Ferramentas de Apoio à Medição de Software e a Ferramenta para Apoio à Medição de Software e Controle Estatístico de Processos. Também foram apresentadas algumas informações sobre portais web.

No próximo capítulo são apresentados os principais resultados das fases de desenvolvimento do Portal de Medição de Software, como os aspectos de análise, projeto e implementação da ferramenta, a fim de se fornecer uma visão do principal produto resultante deste trabalho.

Capítulo 3

Portal de Medição de Software

Este capítulo apresenta os principais resultados produzidos ao longo do desenvolvimento do Portal de Medição de Software. Na Seção 3.1 são apresentados os resultados produzidos durante a especificação e análise de requisitos do sistema. Na Seção 3.2 são apresentados resultados produzidos durante a fase de projeto do sistema. Na Seção 3.3 são apresentadas algumas telas do Portal de Medição de Software. Na Seção 3.4 são apresentadas as considerações finais do capítulo.

3.1 Requisitos e Modelagem Conceitual

A Engenharia de Requisitos é o processo pelo qual os requisitos de um produto de software são coletados, analisados, documentados e gerenciados ao longo de todo o ciclo de vida do software (AURUM e WOHLIN, 2005). Nesta seção são apresentados o objetivo do sistema, seu minimundo, subsistemas, diagrama de casos de uso e diagrama de classes.

3.1.1 Objetivo do Sistema

O *Portal de Medição de Software* tem como propósito disponibilizar diversos trabalhos relacionados à medição de software, visando facilitar o acesso de organizações e indivíduos a conteúdo e ferramentas de apoio ao domínio de medição de software.

3.1.2 Descrição do Minimundo

No Núcleo de Estudos em Modelagem Conceitual e Ontologias (NEMO) foram desenvolvidos diversos trabalhos relacionados à medição de software. Atualmente, alguns desses trabalhos são disponibilizados no site do NEMO (<http://nemo.inf.ufes.br>), outros nos sites dos autores e outros não estão disponíveis para acesso via Internet. Buscando-se unificar e facilitar o acesso a esses trabalhos, decidiu-se por desenvolver um Portal de Medição de Software, no qual os trabalhos relacionados a esse tema desenvolvidos no NEMO e outros materiais relevantes a esse domínio podem ser disponibilizados.

A versão inicial o Portal deve permitir acesso a cinco categorias de itens: Publicações, Dicionário de Termos, Ferramentas, Ontologias e Outros Trabalhos.

O Portal deve ter uma área para que os usuários enviem mensagens ao administrador. Para isso, deve haver uma caixa de mensagens para o administrador no Portal. Para enviar uma mensagem, o usuário deverá informar seu nome, e-mail para contato, título e texto da mensagem. As mensagens enviadas devem ser armazenadas na caixa de mensagens do

administrador, que receberá uma notificação por e-mail comunicando a existência de novas mensagens para ele no Portal. Deve ser possível ao administrador visualizar as mensagens enviadas pelos usuários e respondê-las. A resposta a uma mensagem deve ser enviada para o e-mail do usuário.

O conteúdo do Portal deve poder ser atualizado sempre que necessário. Para isso, deve ser possível a um usuário com perfil de administrador usar funcionalidades presentes no próprio Portal para inserir novos itens a serem disponibilizados e alterar ou excluir itens nele já existentes. Para a primeira versão do Portal, apenas itens relacionados às categorias previamente definidas podem ser inseridos, ou seja, não deve ser permitido criar uma nova categoria de itens no Portal. Para inserir um item na categoria *Publicações*, é necessário informar seu tipo (artigo, relatório técnico, monografia, dissertação de mestrado, tese de doutorado, livro, capítulo de livro, padrão ou outros), título, ano, autores, referência completa, resumo, arquivo da publicação e se a publicação é do NEMO ou não. Para inserir um termo no *Dicionário de Termos* é necessário informar nome e descrição. Para inserir uma *Ontologia* é necessário informar nome, descrição e o arquivo ou *link* da ontologia. Para inserir uma *Ferramenta*, devem ser informados nome, descrição e o arquivo ou *link* da ferramenta. Por fim, para inserir um item em *Outros Trabalhos*, deve-se nome, descrição e o arquivo ou *link* para o item.

Para acesso aos itens das categorias *Publicações* e *Outros Trabalhos* devem ser disponibilizadas funcionalidades que permitam a realização de buscas.

3.1.3 Subsistemas do Portal de Medição de Software

A Figura 3.1 mostra os subsistemas identificados no contexto do presente projeto. As descrições são apresentadas a seguir.

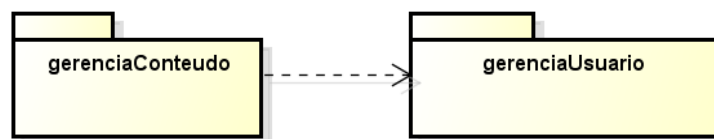


Figura 3.1 – Diagrama de Pacotes e os Subsistemas do Portal de Medição de Software.

- **gerenciaConteudo:** Contém as classes necessárias para o gerenciamento das publicações, termos do dicionário, ferramentas, ontologias e outros trabalhos registrados no Portal.

- **gerenciaUsuario:** Contém as classes relacionadas ao gerenciamento dos usuários do Portal, que inclui procedimentos de autenticação e comunicação entre usuário comum e administrador do Portal.

3.1.4 Casos de Uso

O modelo de casos de uso visa apresentar as funcionalidades que um sistema deve oferecer a seus usuários, assim como seus fluxos de execução. Os atores do Portal de Medição são:

- **Administrador:** Pessoa responsável pelo registro, atualização e disponibilização das publicações, termos do dicionário, ontologias, ferramentas e outros trabalhos. Também é responsável por responder mensagens dos usuários.
- **Usuário:** Pessoa que utiliza o Portal de Medição de Software.

A seguir são apresentados os casos de uso presentes nos subsistemas identificados. Para cada caso de uso são apresentadas descrições breves. A descrição completa dos casos de uso foi registrada no Documento de Especificação de Requisitos.

A Figura 3.2 apresenta o diagrama de casos de uso do subsistema *gerenciaConteudo*.

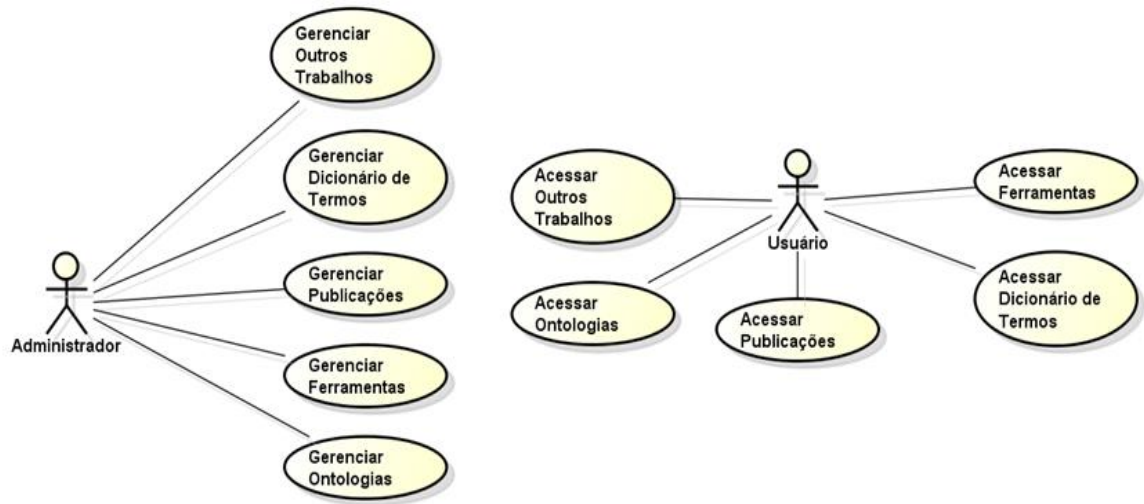


Figura 3.2 – Diagrama de Casos de Uso do Subsistema gerenciaConteudo

Um usuário do Portal com credenciais de administrador possui a autonomia para realizar as operações de gerenciamento do sistema, tais como: gerenciar publicações, gerenciar termos no dicionário, gerenciar ferramentas, gerenciar ontologias e gerenciar outros trabalhos.

Esses casos de uso são considerados operações CRUD (*Create, Read, Update, Delete*). O administrador do sistema é o único tipo de usuário com controle sobre criação, edição e deleção das informações disponíveis no Portal. O usuário do sistema utiliza funcionalidades para acessar o conteúdo disponível no Portal.

A Figura 3.3 apresenta o diagrama de casos de uso do subsistema *gerenciaUsuario*.

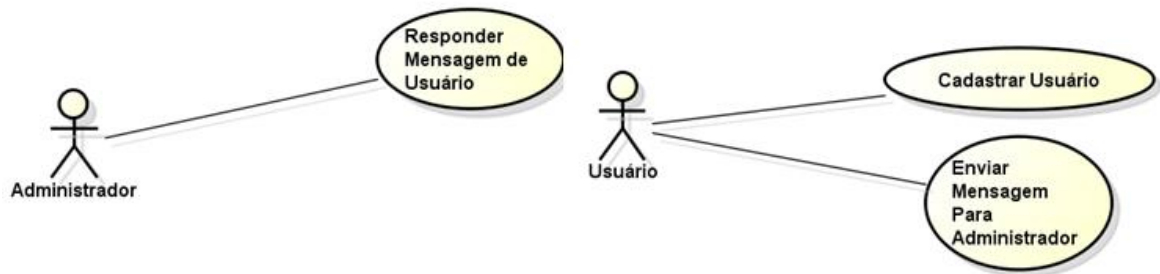


Figura 3.3 – Diagrama de Casos de Uso do Subsistema gerenciaUsuario.

O usuário pode realizar seu cadastro no Portal de Medição de Software, informando nome, e-mail, login e senha para acesso. Ele também pode mensagens ao administrador, que serão armazenadas pelo Portal. O administrador, por sua vez, pode responder às mensagens enviadas pelos usuários.

3.1.5 Diagramas de Classes

A orientação a objetos oferece conceitos bastante apropriados para a modelagem de sistemas, como objetos, relações de composição e hierarquias. Modelos baseados em uma perspectiva humana da realidade são úteis para a compreensão de problemas, para a comunicação com os *stakeholders* e para a realização de atividades ao longo do processo de desenvolvimento de software. Assim, o modelo de análise tem como objetivo obter e descrever as informações relevantes para que as funcionalidades do sistema sejam construídas (FALBO, 2012). A seguir são apresentados os diagramas de classes de cada um dos subsistemas do Portal de Medição de Software. Outras informações, tais como glossário do projeto, que não são apresentadas neste capítulo, foram descritas no Documento de Especificação de Requisitos.

A Figura 3.4 apresenta o diagrama de classes do subsistema *gerenciaConteudo*.

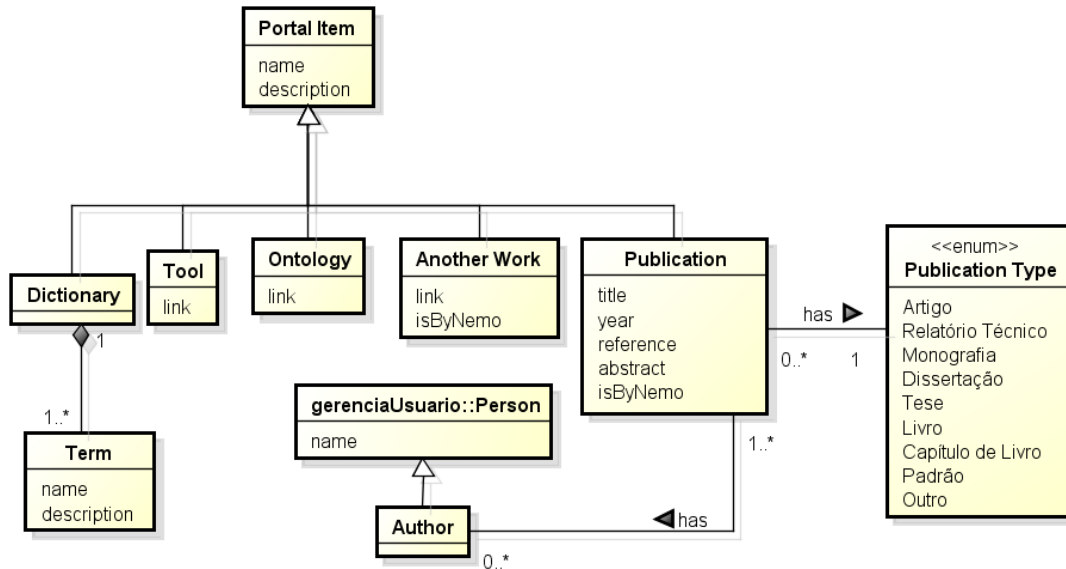


Figura 3.4 – Diagrama de Classes do Subsistema gerenciaConteudo

No Portal de Medição de Software são registrados itens (*Portal Item*), que podem ser: Dicionário (*Dictionary*), Ferramenta (*Tool*), Ontologia (*Ontology*), Publicação (*Publication*) ou Outro Trabalho (*Another Work*). Um Dicionário é composto por Termos (*Term*). Publicações possuem Autores (*Author*) e Tipo de Publicação (*Publication Type*). Publicações e Outros Trabalhos podem ter sido desenvolvidos no NEMO (*isByNEMO = true*) ou não (*isByNEMO = false*).

A Figura 3.5 apresenta o diagrama de classes do subsistema *gerenciaUsuario*.

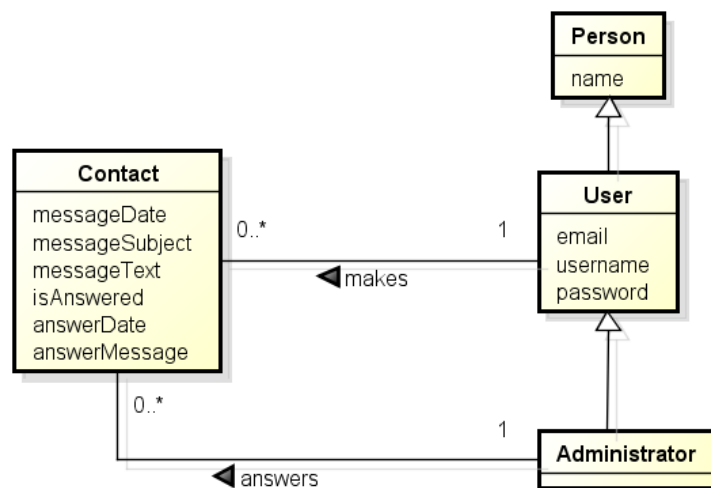


Figura 3.5 – Diagrama de Classes do Subsistema gerenciaUsuario

Usuários do Portal de Medição de Software (*User*) realizam Contato (*Contact*), enviando uma mensagem (*messageText*), em uma determinada data (*messageDate*) e com um certo assunto (*messageSubject*). Contatos são respondidos pelo Administrador do Portal (*Administrator*), que envia uma resposta (*answerMessage*), em uma determinada data (*answerDate*). Quando um contato é respondido, *isAnswered = true*.,

3.2 Projeto de Sistema

Uma vez que os artefatos da etapa de análise são capturados, inicia-se a etapa de projeto, onde os modelos de análise são estendidos para comportarem decisões relacionadas a implementação. Nesta fase deve-se levar em consideração aspectos tecnológicos, como: linguagem de programação, *frameworks* a serem utilizados, características de interface com o usuário, arquitetura de software e persistência de dados (FALBO, 2011).

As tecnologias utilizadas no desenvolvimento do Portal de Medição são:

- *Linguagem de Programação Java*: linguagem orientada a objetos e independente de plataforma.
- *Java Persistence API (JPA)*: API para persistência de dados por meio de mapeamento objeto-relacional.
- *Java Server Faces (JSF)*: Especificação Java para a construção de interfaces de usuário baseadas em componentes para aplicações web e que implementa o padrão MVC.
- *Contexts and Dependency Injection for Java EE (CDI)*: *Framework* gratuito para injeção de dependências do sistema, que garante o baixo acoplamento entre os diferentes módulos do sistema.
- *Apache Maven*: Ferramenta de gerência de projeto baseada em *project object model* (POM) para simplificar o gerenciamento das dependências de bibliotecas do software.
- *Hibernate. Framework* de mapeamento objeto-relacional gratuito que implementa a JPA.
- *Framework NEMO Utils*: Repositório de *frameworks* úteis para projetos baseados em padrões Java para Web.
- *Wildfly*: Servidor Web.
- *MySQL Workbench*: Sistema gerenciador de banco de dados relacional gratuito.

Neste trabalho, foi adotado o método FrameWeb, proposto em (SOUZA, 2007), que é um método que possui o intuito de auxiliar na fase de Projeto de sistemas de informação para a web que utilizem *frameworks*, devido ao fato de serem propostos modelos de projeto que retratam mais fielmente a implementação do sistema. Este método foi utilizado no projeto do Portal de Medição de Software para auxiliar na modelagem de partes do sistema, como o modelo do Componente de Visão.

3.2.1 Arquitetura de Software

A arquitetura de software do Portal de Medição de Software baseia-se na combinação de camadas e partições. Inicialmente, para cada subsistema identificado na fase de análise foi definida uma partição, como mostra a Figura 3.6.

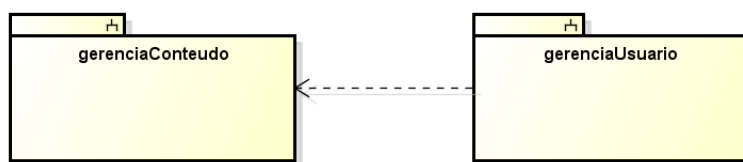


Figura 3.6 – Arquitetura de Software Inicial

Cada uma dessas partições, por sua vez, está organizada em três camadas, a saber: camada de Lógica de Apresentação, que trata de aspectos relacionados às interfaces gráficas que os usuários manipulam; camada de Lógica de Negócio, onde é implementada a lógica do sistema; camada de Lógica de Acesso a Dados, onde é implementada a persistência dos objetos.

Utilizando-se o padrão arquitetônico de camada de serviço, a camada de Lógica de Negócio é subdividida em dois componentes: Componente de Domínio do Problema e Componente de Gerência de Aplicação. Utilizando o padrão Modelo-Visão-Controlador (MVC), a camada de Lógica de Apresentação é subdividida em dois componentes: Componente de Visão e Componente de Controle. A Figura 3.7 apresenta a visão geral das camadas e seus relacionamentos. Na figura as camadas definidas pelo padrão MVC não são ilustradas.

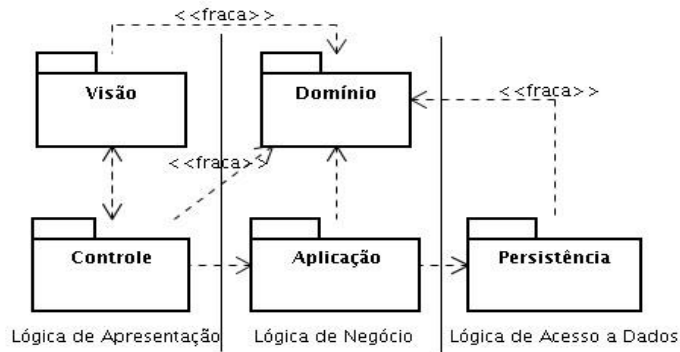


Figura 3.7 – Arquitetura padrão para WIS baseada no padrão arquitetônico *Service Layer* (FOWLER, 2002).

Além das partições definidas a partir dos subsistemas identificados na fase de análise, foram reutilizados componentes presentes na arquitetura do *framework nemo-utils* e próprios para reúso. *Nemo-utils* é um *framework* desenvolvido no NEMO e tem como objetivo auxiliar na implementação de aplicações web que utilizem os *frameworks* JSF, JPA e CDI.

A Figura 3.8 apresenta a subdivisão de cada subsistema nas camadas descritas anteriormente, a saber: Camada de Lógica de Apresentação, Camada de Lógica de Negócio e Camada de Lógica de Acesso a Dados.

As seções subsequentes apresentam um fragmento do projeto detalhado dos componentes da arquitetura. O fragmento selecionado para ser apresentado diz respeito ao subsistema *gerenciaConteudo*.

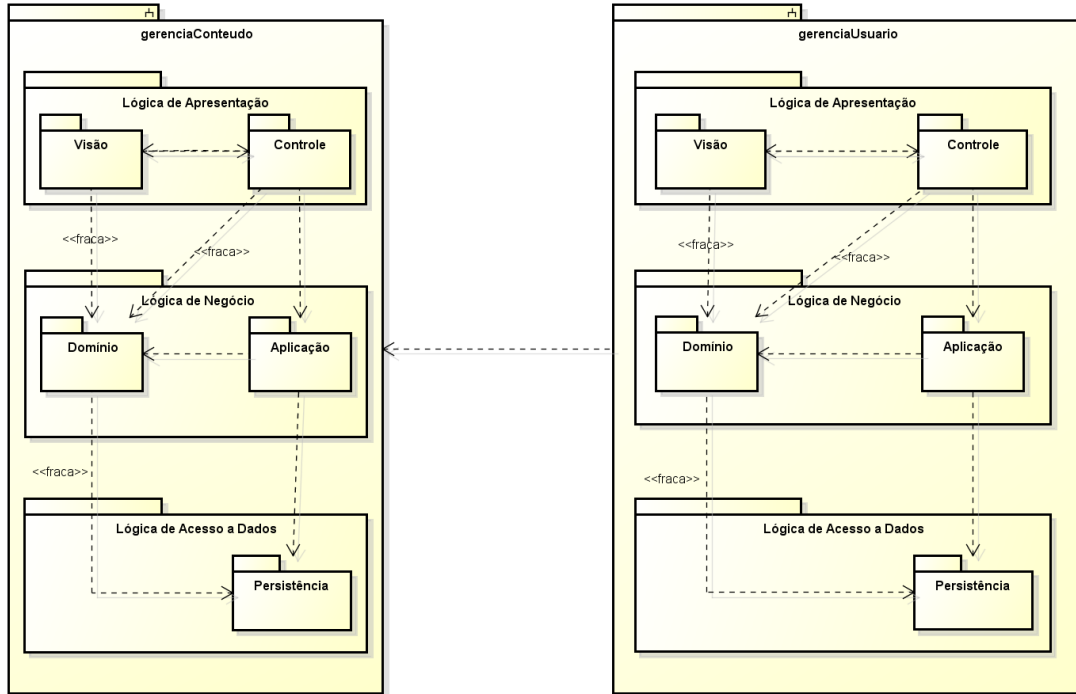


Figura 3.8 – Subdivisão dos subsistemas de acordo com a arquitetura do software

3.2.2 Camada Lógica do Negócio

Como mencionado anteriormente, para organizar a camada de lógica de negócio foi escolhido o padrão camada de serviço. Assim, a camada é dividida em dois componentes: Componente de Domínio do Problema e Componente de Aplicação. O Componente de Aplicação trata a lógica da aplicação, que recebe e trata as requisições da interface, e Componente de Domínio do Problema trata dos conceitos do domínio do problema, advindos do modelo estrutural derivado da etapa de análise. A seguir, o projeto desses componentes é apresentado.

3.2.2.1 Componente Domínio do Problema

A Figura 3.9 apresenta o diagrama de classes do Componente de Domínio do Problema do subsistema *gerenciaConteudo*.

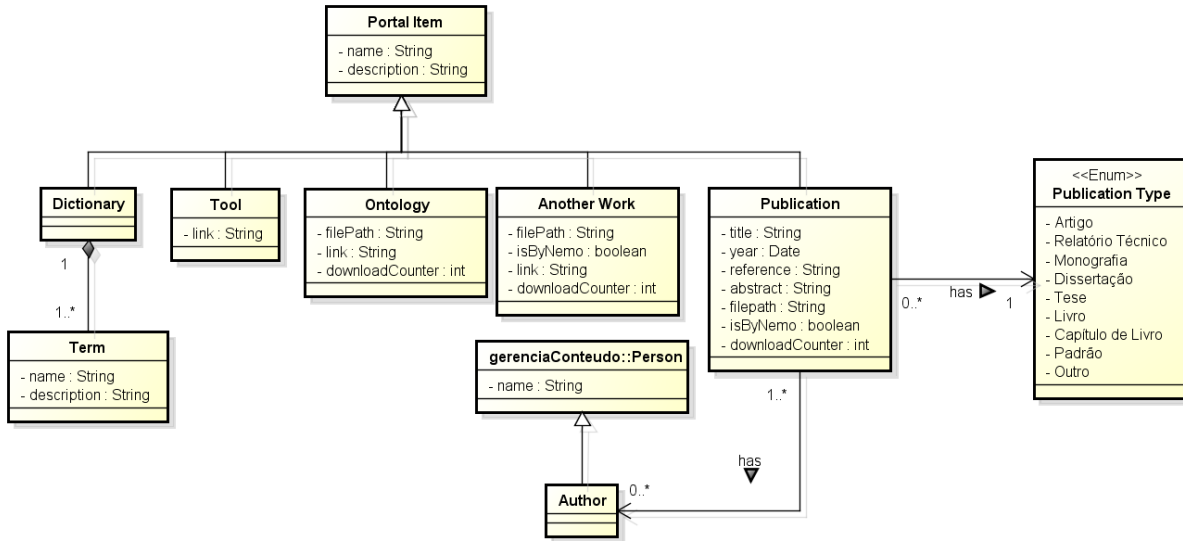


Figura 3.9 – Diagrama de Classes do Componente de Domínio do Problema do subsistema *gerenciaConteudo*.

Todas as classes do diagrama de classes estendem de *PersistenceObjectSupport*, uma classe fornecida pelo *framework nemo-utils*. Dessa forma, a parte cadastral do sistema se comporta de forma uniforme seguindo a arquitetura proposta pelo *framework* que também fornece para as classes um atributo para identificação dos seus objetos.

3.2.2.2 Componente de Gerência de Aplicação

Como os casos de uso do subsistema *gerenciaConteudo* são do tipo CRUD (criar, atualizar, consultar e deletar) e o *framework nemo-utils* proporciona classes genéricas que cobrem esse tipo de caso de uso, todas as classes do componente estendem *CrudServiceBean* e implementam uma interface que estende *CrudService*, ambas pertencentes ao *framework*. A classe *CrudServiceBean* implementa os principais métodos dos casos de uso de tipo CRUD independentemente de domínio, tornando a implementação das classes do sistema voltadas para as questões específicas relacionadas ao domínio da aplicação. A Figura 3.10 apresenta o modelo de aplicação genérico do *framework nemo-utils*.

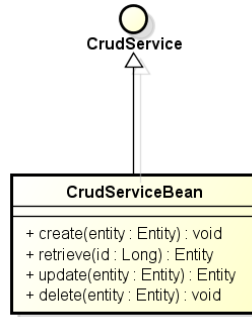


Figura 3.10 – Modelo de aplicação genérico do *framework nemo-utils*

A Figura 3.11 mostra o diagrama do componente de gerência de aplicação de todas as classes do subsistema *gerenciaConteudo*.

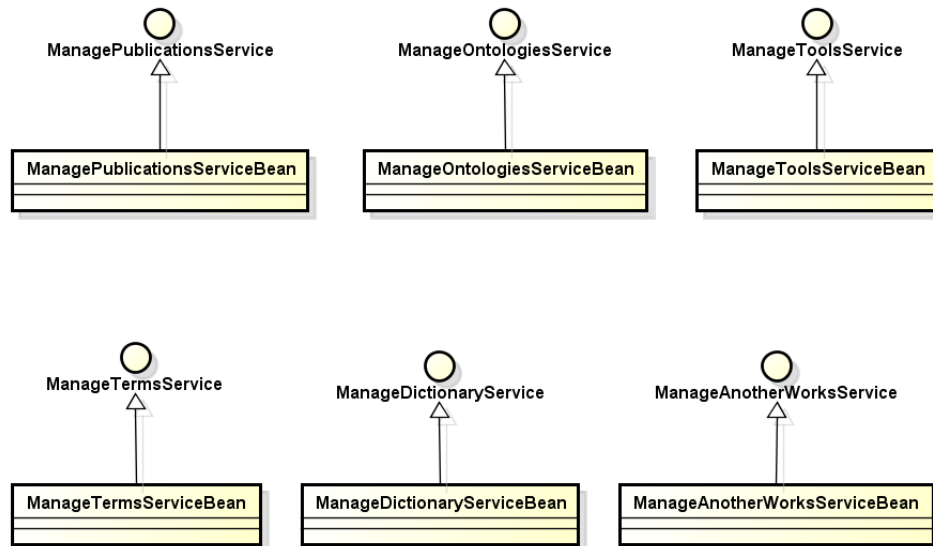


Figura 3.11 – Diagrama do Componente de Gerência de Aplicação do subsistema *gerenciaConteudo*.

3.2.3 Camada Lógica de Apresentação

Para organizar a camada de Lógica de Apresentação foi adotado o padrão Modelo-Visão-Controlador (MVC). Sendo assim, essa camada foi dividida em dois componentes: Componente de Visão, representado pelas páginas Web, e o Componente de Controle, que media as interações entre usuários e sistema. Esse padrão utiliza uma parte *model* para tratar os dados, as regras de negócio e a lógica da aplicação, uma parte *view* que trata da representação dos dados na tela, e uma parte *controller* que trata da mediação entre a entrada fornecida pelo

usuário e os comandos necessários para a realização das ações requeridas por ele no *model* ou *view*. O *framework* que implementa o padrão MVC escolhido para auxiliar a implementação do Portal foi o *framework JSF*. A seguir, o projeto desses componentes é apresentado.

3.2.3.1 Componente de Controle

Todas as classes do componente de Controle estendem a classe *CrudController* fornecida pelo *framework nemo-utils*. A Figura 3.12 descreve as classes desse componente referentes ao subsistema *gerenciaConteudo*.

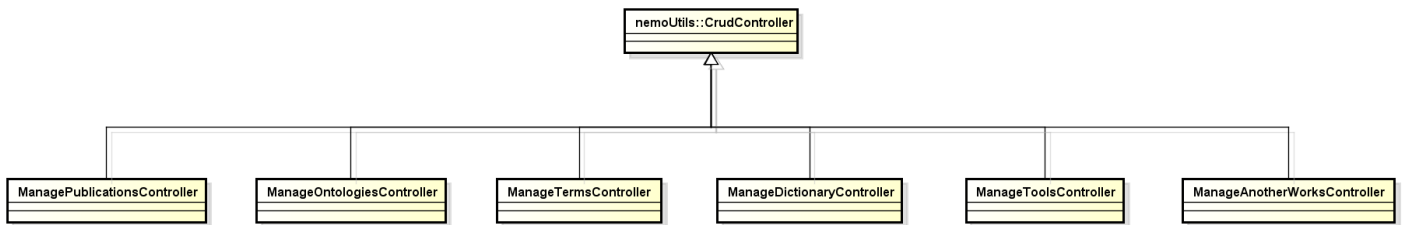


Figura 3.12 - Diagrama do Componente de Controle do subsistema *gerenciaConteudo*

3.2.3.2 Componente de Visão

As funcionalidades do tipo CRUD são similares em seu fluxo de execução e interação com o usuário. Assim, sua implementação é facilitada pela utilização do *framework nemo-utils*, que fornece um modelo de navegação simplificado e eficiente. A Figura 3.13 mostra o modelo de navegação genérico de uma funcionalidade CRUD implementado no *framework nemo-utils*. Como todos os casos de uso do subsistema *gerenciaConteudo* são do tipo CRUD, seus modelos de navegação e de páginas são extensões do modelo genérico do *framework nemo-utils*, sendo alteradas apenas as entidades utilizadas.

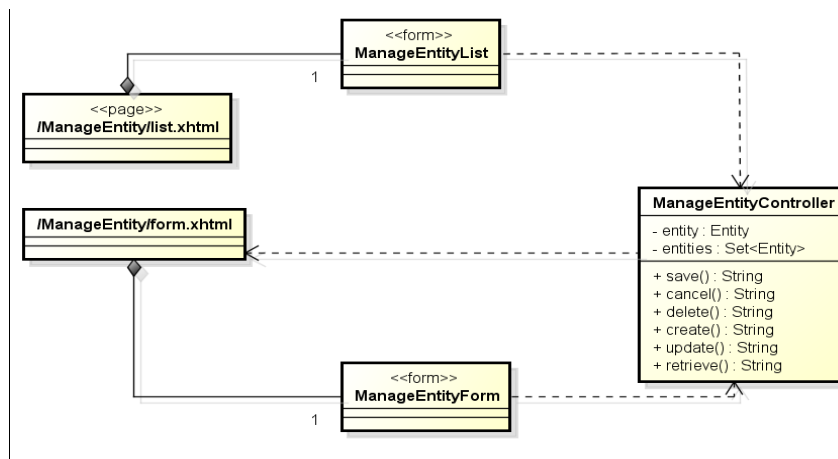


Figura 3.13 - Modelo de navegação genérico do Componente de Visão do *framework nemo-utils*

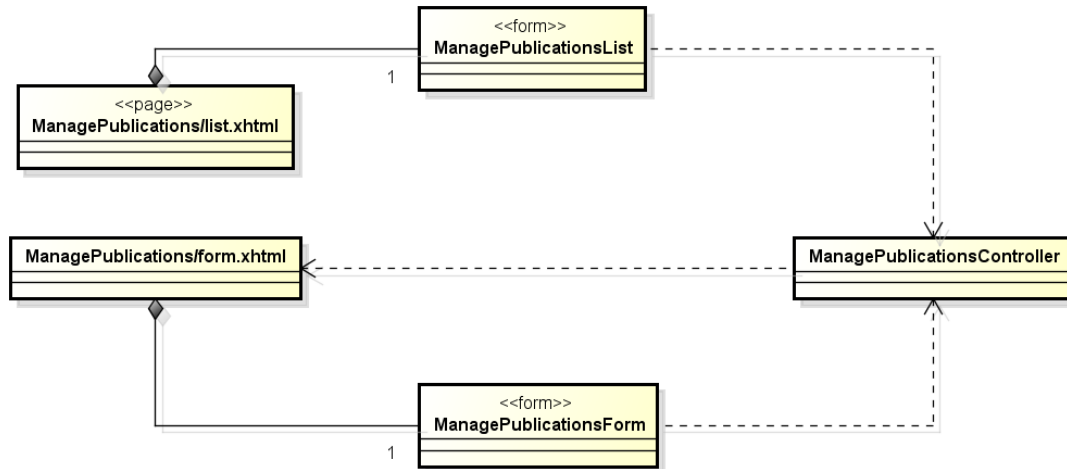


Figura 3.14 - Modelo de navegação do Componente de Visão da classe *Publication*

Na construção do projeto das interfaces, utilizou-se prototipação. Um protótipo tem por objetivo simular a aparência e funcionalidade do software a ser construído, permitindo realçar características de interface relevantes ao desenvolvimento da aplicação. A ferramenta utilizada para o desenvolvimento do protótipo da interface foi *Balsamiq Mockups 3*. Na Figura 3.15 é apresentada como exemplo uma tela definida durante a prototipação da interface.

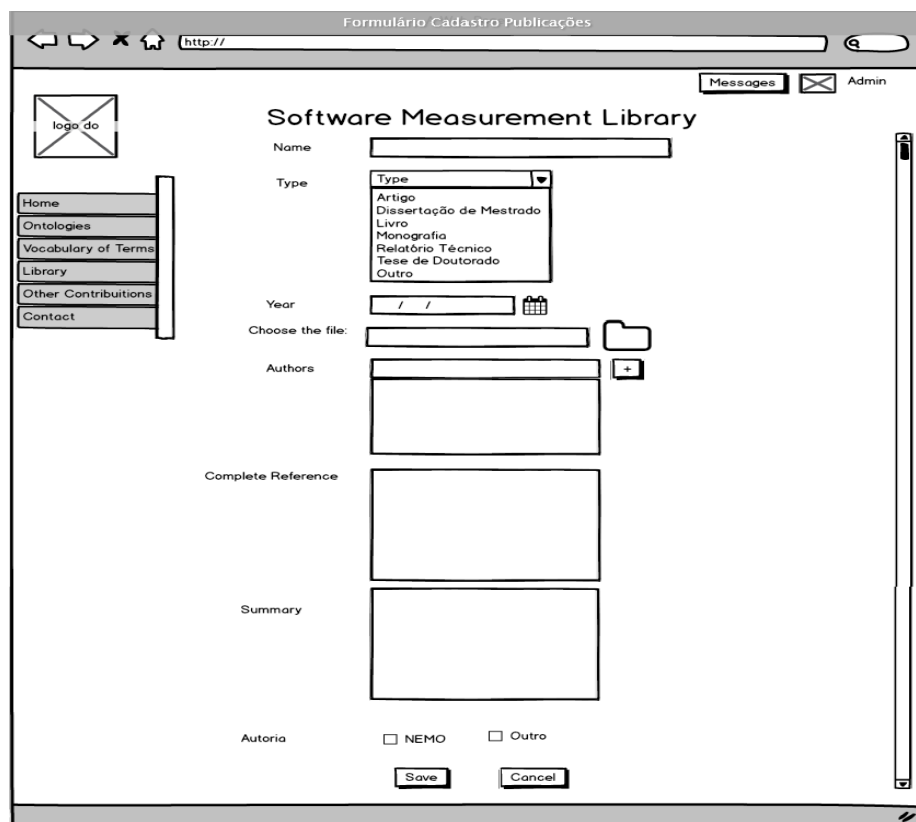


Figura 3.15 – Protótipo da tela de cadastro de publicações

3.2.4 Camada Lógica de Acesso a Dados

A persistência dos objetos do Portal de Medição de Software é realizada em um banco de dados relacional, utilizando a infraestrutura de persistência apresentada pelo *framework nemo-utils*. Essa infraestrutura utiliza JPA com o *framework* de persistência *Hibernate* e adota o Padrão DAO. Na Figura 3.16 está representado o diagrama de persistência genérico provido pelo *framework nemo-utils*.

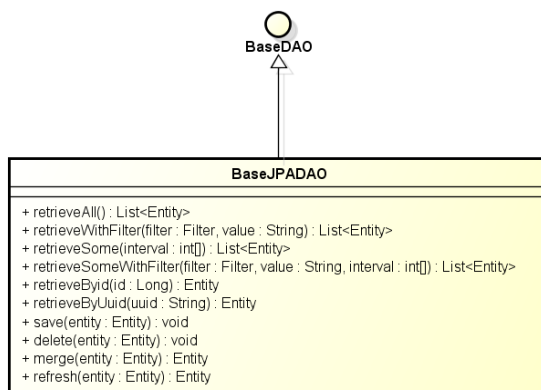


Figura 3.16 - Diagrama de persistência do *framework nemo-utils*

Para cada classe de domínio a ser persistida devem ser criadas uma interface (*)DAO e uma classe (*)JPADAO correspondente. A primeira deve herdar da interface genérica BaseDAO. Já a segunda deve herdar de BaseJPADAO, que possui as funcionalidades básicas de acesso ao mecanismo de persistência, além de implementar a interface DAO correspondente. Assim, cada classe a ser persistida terá uma classe de persistência correspondente, responsável pela interação com o banco de dados relacional, e implementará uma interface correspondente, como mostra a Figura 3.17. Para simplificação, na figura as relações de herança entre as interfaces (*)DAO específicas e a interface BaseDAO e entre as classes (*)JPADAO e a classe BaseJPADAO não são representada.

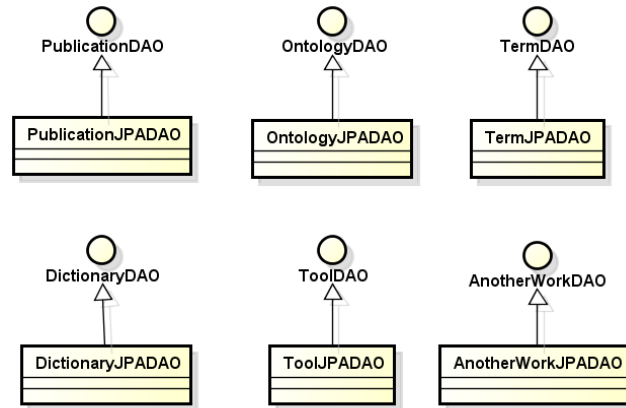


Figura 3.17 - Diagrama de persistência do subsistema *gerenciaConteudo*

Vale ressaltar que, como comentado na descrição do Componente Domínio do Problema, as classes de domínio a serem persistidas herdam da classe *PersistentObjectSupport* do *framework nemo-utils*, que provê identificadores únicos para os objetos (ids) a serem usados para mapear os objetos em memória com as correspondentes linhas das tabelas do banco de dados relacional.

3.3 Implementação

A engenharia de software inclui todas as atividades envolvidas no desenvolvimento de software, desde os requisitos iniciais do sistema até a manutenção e o gerenciamento do sistema implantado. O estágio mais crítico desse processo é, naturalmente, a implementação do sistema, estágio em que é criada uma versão executável do software. A implementação pode envolver o desenvolvimento de programas em alto ou baixo nível de linguagens de programação, bem como customização e adaptação de sistemas genéricos de prateleira, para atender aos requisitos específicos de uma organização (SOMMERVILLE, 2011).

Nesta seção são apresentadas algumas telas do Portal de Medição de Software. De forma geral, o Portal de Medição de Software é composto por um conjunto de funcionalidades cadastrais, que permitem o armazenamento e consulta a conteúdo como, por exemplo, cadastro de publicações, e por um conjunto de funcionalidades que apoiam o contato entre o usuário comum e o administrador do sistema. Todo portal deve possuir um link para o mundo externo. Portanto, no Portal de Medição esse link é provido pelos trabalhos não desenvolvidos no NEMO que são inclusos no Portal.

Para a versão inicial do Portal foram disponibilizados:

- *Publicações:* artigos, monografias e dissertações relacionadas à medição produzidas por alunos do NEMO e a tese de doutorado (BARCELLOS, 2009)
- *Dicionário de Termos:* termos definidos a partir da Ontologia de Referência de Medição de Software, considerando a versão atual da ontologia originalmente proposta em (BARCELLOS, 2009).
- *Ferramentas:* SoMeSPC, ferramenta de apoio à medição de software e controle estatístico de processos desenvolvida em (MARETTO, 2013) e evoluída em (CASTRO, 2016).
- *Ontologias:* Ontologia de Referência de Medição de Software (versão atual da ontologia proposta em (BARCELLOS, 2009)) e Ontologia de Tarefa de Medição de Software (versão atual da ontologia parcialmente descrita em BARCELLOS e FALBO, 2013).
- *Outros Trabalhos:* IARM – Instrumento para Avaliação de Repositório de Medição para CEP, CRMS – Conjunto de Recomendações para Medição de Software Adequada ao CEP, Arquitetura de Referência para Medição de Software (MARETTO, 2013), OBA-MSI – *Ontology-Based Approach for Measurement Systems Integration* (FONSECA, 2015).

A porta de entrada do sistema se dá através da tela de *Login*, que permite o registro e a autenticação dos usuários, como mostra a Figura 3.18. Essa funcionalidade é importante para diferenciar usuários que podem inserir conteúdo no portal daqueles que só podem consultá-lo.

Portal de Medição de Software

Login

Nome de usuário*:

Senha*:

Figura 3.18 – Tela de *Login* do Portal de Medição de Software

Enquanto não houver a autenticação de um usuário, a única ação possível é o registro de usuário, uma vez que as funções do menu são selecionadas e exibidas de acordo com o nível de controle de acesso aplicado ao utilizador do sistema.

No caso do Administrador, uma vez autenticado ele possui permissão para gerenciar as funcionalidades cadastrais do sistema, bem como gerenciar a funcionalidade de resposta às mensagens de contato enviada pelos usuários, como mostra a Figura 3.19.

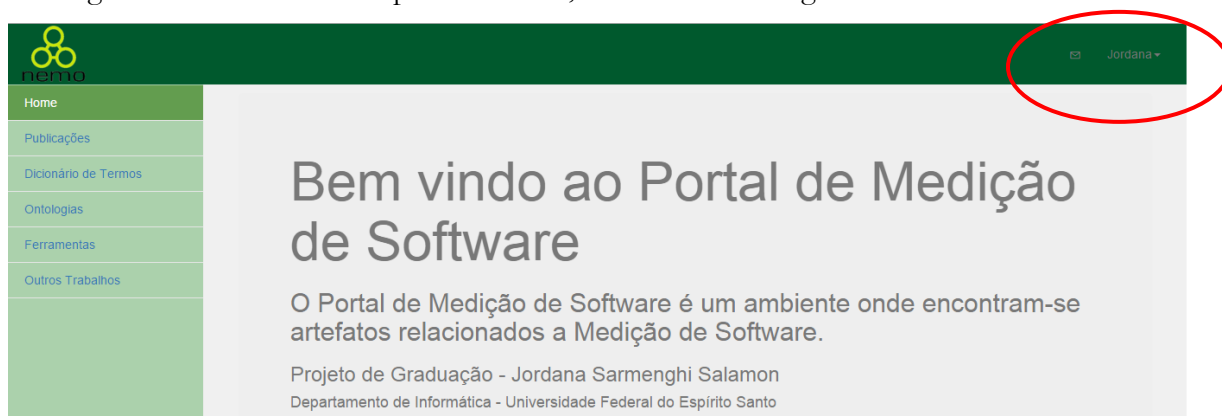


Figura 3.19 – Tela inicial do Administrador do Portal de Medição de Software

No caso do usuário comum, uma vez autenticado ele possui permissão para visualizar os itens cadastrados no sistema, bem como enviar mensagens de Contato ao Administrador do sistema, como mostra a Figura 3.20.



Figura 3.20 – Tela inicial do usuário comum no Portal de Medição de Software

As diferenças entre as telas apresentadas nas figuras 3.19 e 3.20 ocorrem no menu à esquerda e na barra superior. Quanto ao menu à esquerda, na tela do administrador ele comporta somente *links* para as categorias de itens a serem cadastrados, enquanto que na tela

do usuário comum ele inclui também um *link* para o formulário de contato. Quanto à barra superior, na tela do administrador há um ícone de envelope onde o administrador pode acessar as mensagens enviadas pelo usuário. Esse ícone não é mostrado na tela de usuário comum.

Ao clicar em qualquer opção do menu cadastral à esquerda, o Administrador é encaminhado para uma tela que apresenta a listagem de itens armazenados para a opção selecionada. Como exemplo, na Figura 3.21 é apresentada a tela para cadastro de publicações. As ações possíveis de serem realizadas são exibidas em botões na parte inferior da tela.

Título	Ano	Tipo	NEMO?
Uma Arquitetura de Referência para Medição de Software	2013	Dissertacao	true

Figura 3.21 – Tela de Publicações

Ao solicitar a inclusão de um novo item, clicando-se no botão “Novo”, a tela para cadastro é apresentada, como mostra a Figura 3.22.

The image shows a web interface for creating a new publication. On the left is a green sidebar menu with items: Home, Publicações, Dicionário de Termos, Ontologias, Ferramentas, and Outros Trabalhos. The main content area is titled 'Criar nova Publicação'. It contains the following fields and controls:

- Título*:** A text input field.
- Tipo*:** A dropdown menu with 'Artigo' selected.
- Ano:** A text input field with a calendar icon on the left.
- Autores*:** A long text input field.
- Seleção de arquivo:** A section with a button 'Escolher arquivo', the text 'Nenhum arquivo selecionado', and an 'Upload' button.
- Resumo:** A large text area with a small icon in the bottom right corner.
- Referência Completa:** A text area with a small icon in the bottom right corner.
- Autoria:** A dropdown menu with 'NEMO' selected.
- Buttons:** 'Cancelar' and 'Salvar' buttons at the bottom.

Figura 3.22 – Tela para cadastro de Publicações do Portal de Medição de Software

As funcionalidades providas para as outras opções do menu referentes a casos de usos cadastrais são similares às apresentadas até o momento. Para os casos de uso que não se referem a cadastros, são apresentadas telas diferentes. Para o usuário comum, por exemplo, não são apresentadas as telas que permitem o registro, exclusão ou alteração de itens. Por outro lado, são apresentadas telas que permitem consulta a itens e retornam os itens selecionados pelos critérios fornecidos pelo usuário, como mostra a Figura 3.23.

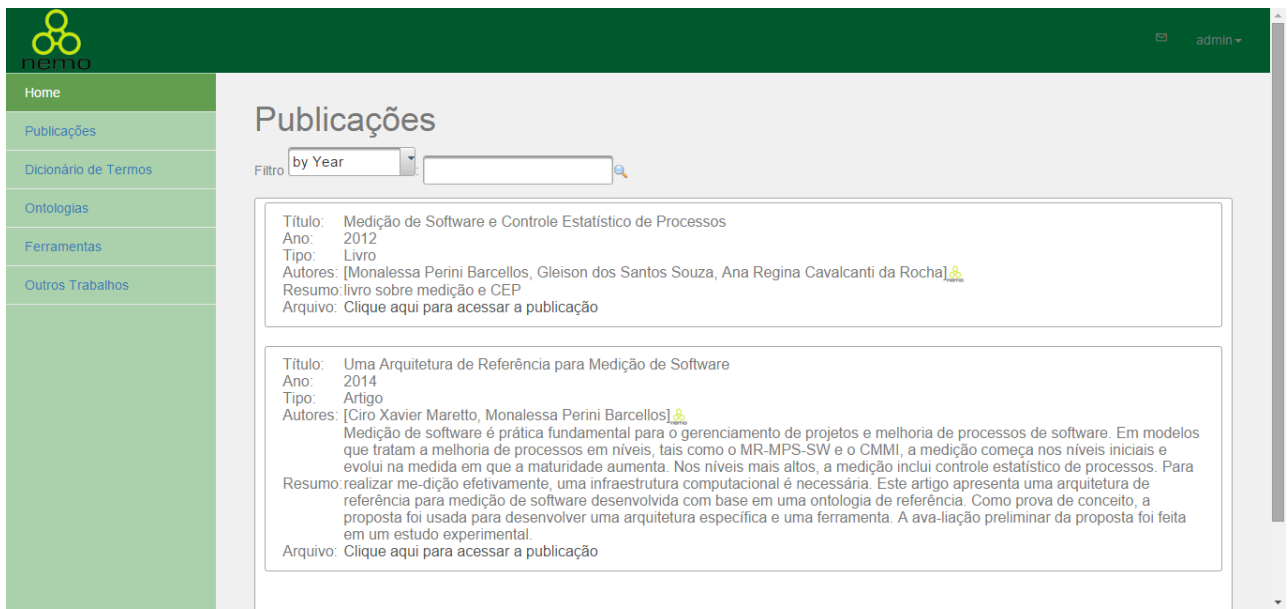


Figura 3.23 – Tela de consulta a Publicações para o usuário do Portal de Medição de Software

Para os casos de uso que lidam com o envio de mensagens e respostas entre os usuários e o administrador, são providas as telas apresentadas a seguir. Para enviar uma mensagem, o usuário comum deve clicar na opção Contato no menu exibido a esquerda, enquanto que o Administrador deve clicar no ícone de envelope exibido na barra superior.

Para o usuário comum, é exibido um formulário a ser preenchido com assunto e mensagem, como mostra a Figura 3.24. Assim que a mensagem é enviada, um e-mail é enviado para o Administrador, informando-o que ele possui uma mensagem não lida no Portal.



Figura 3.24 – Tela de formulário de envio de mensagem para o Administrador

Para o Administrador, é exibida uma listagem das mensagens enviadas, como mostra a Figura 3.25.

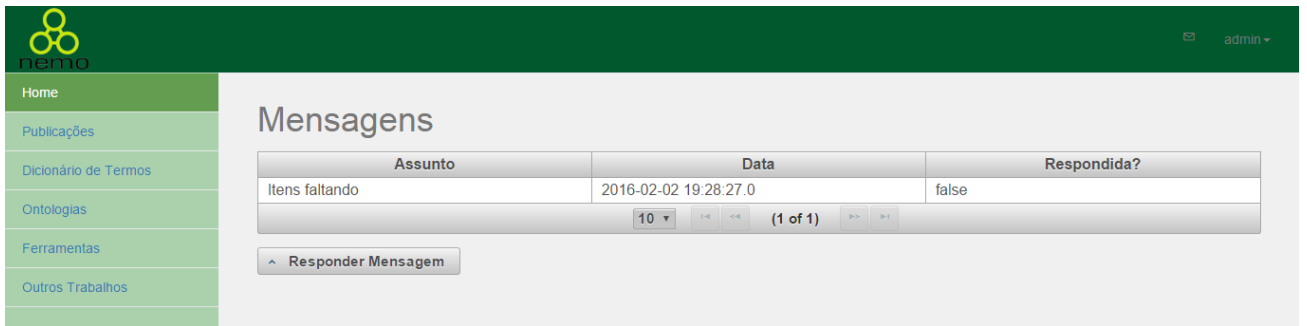


Figura 3.25 – Listagem de mensagens enviadas pelos usuários do Portal de Medição de Software

Ao selecionar uma mensagem a ser respondida, é exibido o formulário com as informações sobre a mensagem enviada e com os campos para o preenchimento da resposta, como mostra a Figura 3.26. A resposta é enviada em um e-mail para o usuário que enviou a mensagem original.

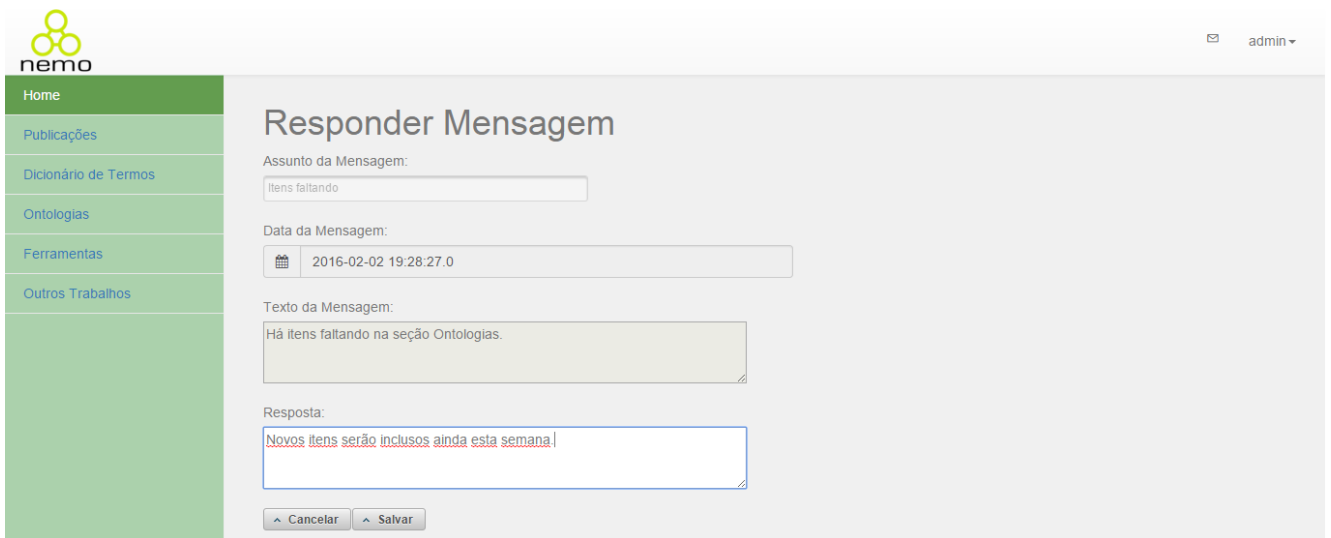


Figura 3.26 – Tela de resposta a contato feito por usuário do Portal de Medição de Software

3.4 Considerações Finais do Capítulo

Este capítulo apresentou o desenvolvimento da ferramenta Portal de Medição de Software, tendo sido apresentados os principais artefatos produzidos nas fases de análise, projeto e implementação.

Segundo a classificação de portais apresentada por (DIAS, 2001), o Portal de Medição de Software possui, em sua versão inicial, características majoritariamente atribuídas a portais públicos. Porém, há a possibilidade de desenvolvimento de funcionalidades voltadas para um

público específico, como os usuários do laboratório NEMO, por exemplo. Assim, embora (DIAS, 2001) não tenha definido explicitamente um portal de tipo misto, o Portal de Medição de Software se encaixa nessa categoria.

No próximo capítulo são apresentadas as considerações finais do trabalho.

Capítulo 4

Conclusão

Neste capítulo são realizadas as considerações finais deste trabalho, sendo apresentadas suas principais contribuições e perspectivas de trabalhos futuros.

4.1 Considerações Finais

A crescente demanda por qualidade nos produtos e produtividade no desenvolvimento de software e o entendimento de que qualidade e produtividade dependem da qualidade dos processos têm aumentado o interesse das organizações pela melhoria de seus processos de software (BARCELLOS *et al.*, 2012). Nesse contexto, medição de software é fundamental.

Além disso, medição de software é um dos principais processos de apoio à gerência de projetos e é crucial para a avaliação da qualidade de produtos de software e do desempenho e capacidade dos processos de software organizacionais (ISO/IEC, 2007). No contexto da gerência de projetos, a medição auxilia no desenvolvimento de planos realísticos, no monitoramento do progresso do projeto, na identificação de problemas e no embasamento para a tomada de decisão (MCGARRY *et al.*, 2002). No contexto organizacional, a medição apoia a análise do desempenho dos processos e a identificação de ações para melhorar a competitividade das organizações (TARHAN; DEMIRÖRS, 2006).

Diversos trabalhos sobre medição de software têm sido desenvolvidos e registrados na literatura nas últimas décadas. No entanto, esses trabalhos encontram-se dispersos e o que requer esforço e tempo para encontrá-los. Nessa mesma linha, no NEMO têm sido desenvolvidos diversos trabalhos sobre medição de software que, também, não possuíam um ponto único de acesso.

Neste trabalho foi desenvolvida a primeira versão de um Portal de Medição de Software cujo propósito é disponibilizar trabalhos relacionados à medição, especialmente aqueles desenvolvidos no NEMO. Embora tenha sido disponibilizado um conjunto limitado de itens no Portal de Medição, ele tem capacidade de ser escalado para atender a um conjunto maior de categorias de itens a serem disponibilizados. Além disso, este trabalho apresenta a versão inicial da seleção de itens a serem disponibilizados, a qual é limitada e deverá ser ampliada em breve e, assim, o portal será disponibilizado para uso.

Para o desenvolvimento, foram realizadas atividades de engenharia de software que levaram à elaboração de diversos artefatos, destacando-se: Documento de Requisitos, Documento de Especificação de Requisitos, Documento de Projeto do Sistema e o sistema propriamente dito, o qual foi desenvolvido a partir da documentação gerada nas etapas anteriores à implementação.

No Capítulo 1 foram identificados os objetivos a ser alcançados neste trabalho. Na Tabela 4.1 é apresentado para cada objetivo específico o produto que serve como evidência de seu alcance.

Tabela 4.1 – Objetivos específicos e sua situação na conclusão da monografia.

Objetivo	Status	Resultado
Identificar trabalhos relevantes à medição de software para serem disponibilizados no Portal de Medição de Software.	Parcialmente atendido	Trabalhos do NEMO identificados (vide Seção 2.2) e trabalhos disponibilizados no Portal de Medição de Software. Embora alguns itens tenham sido selecionados, ainda é necessário ampliar a seleção e incluir novos itens no Portal.
Realizar a modelagem conceitual do Portal de Medição, incluindo aspectos comportamentais e estruturais.	Atendido	Elaboração do Documento de Requisitos do Portal de Medição de Software e do Documento de Especificação de Requisitos do Portal de Medição de Software (vide Seção 3.1).
Definir uma arquitetura para o Portal de Medição.	Atendido	Elaboração do Documento de Projeto do Portal de Medição de Software (vide Seção 3.2).
Implementar o Portal de Medição.	Atendido	Portal de Medição de Software implementado (vide Seção 3.3)

Em relação à experiência adquirida durante a realização deste trabalho destaca-se a integração de diversas disciplinas e conceitos vistos ao longo do curso de graduação. Foram aplicados aprendizados das disciplinas de Engenharia de Software, Engenharia de Requisitos, Projeto de Software, Programação III, Desenvolvimento Web e Web Semântica, cujos ensinamentos foram imprescindíveis para a realização deste projeto. Além disso, o estudo e uso do *framework* FrameWeb auxiliou na complementação da modelagem do sistema.

Para a execução deste trabalho, passou-se por todas as fases de desenvolvimento de software – Levantamento e Análise de Requisitos, Projeto, Implementação, Testes e Implantação. Além disso, foi muito importante a consolidação do conhecimento acerca da utilização das várias tecnologias e *frameworks* mencionados no Capítulo 3. Por fim, mas não menos importante, ressalta-se o aprendizado acerca de medição de software, iniciado durante um projeto de Iniciação Científica e estendido ao presente trabalho.

Considerando as dificuldades encontradas para desenvolver este trabalho destaca-se o projeto e implementação de interfaces, que demandaram esforço e tempo no estudo do *framework* JSF e das implementações do *framework* a serem utilizadas. Além disso, destacam-se também dificuldades encontradas na modelagem conceitual do sistema e nas tomadas de decisão em relação ao projeto e implementação.

4.2 Contribuições e Trabalhos Futuros

A principal contribuição deste trabalho é o Portal de Medição de Software. Além desta, também pode-se destacar como contribuição o conjunto de itens selecionados para serem disponibilizados no portal.

Em relação a esses itens, é necessário realizar uma busca mais ampla e selecionar novos itens a serem inclusos no Portal de Medição. Devido à limitação de tempo e às dificuldades encontradas na fase de implementação, a seleção de itens foi limitada.

Ao final do processo de desenvolvimento de um software, novas necessidades e funcionalidades podem ser identificadas. Além disso, a manutenção do software é um trabalho constante, que extrapola o tempo de implantação do sistema e se estende por toda a vida útil do software. Assim, no âmbito do Portal de Medição foram identificadas algumas melhorias, a saber:

- Permitir a submissão de material por outras pessoas para ser avaliado pelo administrador que, se aprovar o material, poderá disponibilizá-lo;
- Implementar mecanismos de buscas para retornar itens considerando todos os trabalhos armazenados (atualmente, as buscas por itens devem ser feitas em cada categoria de publicação);
- Buscar materiais candidatos a serem inseridos no portal automaticamente a partir dos dados registrados no site do NEMO (ou até mesmo em outros sites);
- Incluir contador de *downloads* dos materiais;
- Incluir funcionalidade de *chat*, onde os usuários podem se cadastrar e se comunicar no próprio portal;
- Identificar as instituições das publicações e incluir funcionalidade que permita identificar aquelas que têm realizado pesquisa na área de medição de software, segundo os dados disponibilizados no portal.

Referências Bibliográficas

- AURUM, A., WOHLIN, C., 2005, **“Engineering and Managing Software Requirements”**, Springer-Verlag
- BARCELLOS, M. P., FALBO, R. A., 2013, **A Software Measurement Task Ontology**. ACM SAC 2013 28th Symposium On Applied Computing. Coimbra - Portugal.
- BARCELLOS, M. P., 2008, **“Uma Abordagem para Controle Estatístico de Processos em Organizações de Alta Maturidade”**, Exame de Qualificação para o Doutorado, COPPE/UFRJ - Rio de Janeiro – Brasil.
- BARCELLOS, M.P., 2009, **"Uma Estratégia para Medição de Software e Avaliação de Bases de Medidas para Controle Estatístico de Processos em Organizações de Alta Maturidade "**, Tese de Doutorado, COPPE/UFRJ - Universidade Federal do Rio de Janeiro, Rio de Janeiro - RJ, Brasil.
- BARCELLOS, M.P., FALBO, R.A. and Dalmoro, R., 2010a, **"A Well-Founded Software Measurement Ontology"**, In: Proceedings of the 6th International Conference on Formal Ontology in Information Systems (FOIS 2010), Toronto - Canadá.
- BARCELLOS, M.P., FALBO, R.A. and ROCHA, A.R., 2010b, **"Establishing a Well-founded Conceptualization about Software Measurement in High Maturity Levels"**, In: Proceedings of the 7th International Conference on the Quality of Information and Communications Technology (QUATIC 2010), Oporto - Portugal, 467-472.
- BARCELLOS, M.P., FALBO, R.A. and ROCHA, A.R., 2010c, **"A Well-founded Software Process Behavior Ontology to Support Business Goals Monitoring in High Maturity Software Organizations"**, In: Proceedings of the IEEE International EDOC Enterprise Computing Conference Workshops, Vitória - ES, 253-262.
- BARCELLOS, M.P., ROCHA, A.R. and FALBO, R.A., 2010d, **"Evaluating the Suitability of a Measurement Repository for Statistical Process Control"**, In: Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. New York: ACM, 1-10.

- BARCELLOS, M.P., ROCHA, A.R. and FALBO, R.A., 2010e, "**IABM: Instrumento para Avaliação de Bases de Medidas para Controle Estatístico de Processos de Software**", Anais do IX Simpósio Brasileiro de Qualidade de Software (SBQS 2010), Belém - PA, 119-133.
- BARCELLOS, M. P., ROCHA, A. R., SANTOS, G., "**Medição de Software e Controle Estatístico de Processos**" (ISSN 1679-1878). 1. ed. Brasília - DF: Ministério da Ciência, Tecnologia e Inovação - SEPIN - PBQP Software, 2012. 232p.
- BARCELLOS, M.P. and FALBO, R.A., 2013, "**A Software Measurement Task Ontology**", In: Proceedings of the 28th Symposium on Applied Computing, Coimbra-Portugal.
- BARCELLOS, M.P., FALBO, R.A. and ROCHA, A.R., 2013, "**A Strategy for Preparing Software Organizations for Statistical Process Control**", Journal of the Brazilian Computer Society, In press.
- BASS, L., BELADY, L., BROWN, A., FREEMAN, P., ISENSEE, S., KAZMAN, R., KRASNER, H., MUSA, J., PFLEEGER, S., VREDENBURG, K., WASSERMAN, T., 1999, **Constructing Superior Software**, Software Quality Institute Series, Macmillan Technical Publishing
- BRIMSON, J. A., 2004, "**Stop Cane Dancing and Integrate Statistical Process Control (SPC) into your Process Based Management System**", Measurement Business Excellence, v. 8, n. 2, pp. 15-22.
- CALERO, C., CARO, A., PIATTINI, M., 2008, "**An Applicable Data Quality Model for Web Portal Data Consumers**", World Wide Web Journal, Volume 11, Issue 4, pp 465-484
- CALHAU, R. F., 2011, "**Uma Abordagem Baseada em Ontologias para Integração Semântica de Sistemas**". (Dissertação de Mestrado). Universidade Federal do Espírito Santo.
- CARD, D.N., DOMZALSKI, K., DAVIES, G., 2008, "**Making Statistics Part of Decision Making in an Engineering Organization**", IEEE Software 25, 37-47
- CASTRO, H. Evolução de uma Ferramenta de Medição para Apoiar a Medição de Software no Contexto de Métodos Ágeis, 2016. (trabalho em andamento). Universidade Federal do Espírito Santo.
- CHRISSIS, M. B., KONRAD, M., SHRUM, S., 2006, CMMI (Second Edition): **Guidelines for Process Integration and Product Improvement**, Addison-Wesley.

- DIAS, C.A., 2001, **“Portal Corporativo: Conceitos e Características”**, Ci. Inf., Brasília, v. 30, n. 1, p. 50-60, jan./abr. 2001
- DUMKE, R., CÔTÉ, I. and ANDRUSCHAK, O. T., 2004, **“Statistical Process Control (SPC) - A Metric-based Point of View of Software Processes Achieving the CMMI Level Four, Technical Report”**, Dept. of Computer Science, University of Magdeburg, Germany.
- FALBO, R. A. **Engenharia de Requisitos: Notas de aula. 2012**. Disponível em: <http://www.inf.ufes.br/~falbo/files/Notas_Aula_Engenharia_Requisitos.pdf>. Acesso em: 3 nov. 2015.
- FALBO, R. A. **Projeto de Sistemas: Notas de aula. 2011**. Disponível em: <http://www.inf.ufes.br/~falbo/files/Notas_Aula_Projeto_Sistemas_2.pdf>. Acesso em: 3 nov. 2015.
- FALBO, R. A., RUY, F., GUIZZARDI, G., BARCELLOS, M. P., & ALMEIDA, J. P. (2014). Towards an Enterprise Ontology Pattern Language. Proceedings of the 29th ACM Symposium On Applied Computing (SAC 2014), Gyeongju, Korea.
- FENTON, N. E., NEIL, M., 1999, **“Software Metrics: Success, Failures and New Directions”**, Journal of Systems and Software, v. 47, pp. 149-157.
- FENTON, N. E., NEIL, M., 2000, **“Software Metrics: a Roadmap”**, In: Proceedings of the 22nd International Conference on Software Engineering - ICSE 2000, San Francisco Bay, USA, pp. 359- 370.
- FLORAC, W.A., CARLETON, A.D., **“Measuring the Software Process: Statistical Process Control for Software Process Improvement”**, Addison-Wesley, Boston USA, 1999.
- FOWLER, M., SCOTT, K. **UML Distilled: Applying the Standard Object Modeling Language**. Addison-Wesley Object Technology Series, 1997.
- FOWLER, M. **Patterns of Enterprise Application Architecture**. Addison-Wesley, ISBN 0321127420, novembro 2002.
- FRAUCHES, V., 2011, **“MeMOS: Uma Ferramenta de Apoio à Medição e ao Monitoramento de Objetivos de Software”**, Trabalho de Conclusão de Curso - Ciência da Computação, Departamento de Informática, UFES, Vitória – ES.

- GOPAL, A., KRISHNAN, M. S., MUKHOPADHYAY, T., GOLDENSON, D. R., 2002, **"Measurement Programs in Software Development: Determinants of Success"**, IEEE Transactions on Software Engineering, v. 28, n. 9, pp. 863-875.
- GUARINO, N., 1998, **"Formal ontology and information systems"**. In: Proceedings of the international conference in formal ontology and information systems—FOIS'98, Trento, pp 3–15
- GUIZZARDI, G., 2005, **"Ontological foundations for structural conceptual models"**, Universal Press, The Netherlands
- ISO/IEC, 2007, ISO/IEC 15939 (E) Software Engineering – **"Software Measurement Process, International Organization for Standardization and the International Electrotechnical Commission"**, Geneva, Switzerland.
- IEEE, 1998, **Std 1061—IEEE standard for a software quality metrics methodology**
- KILPI, T., 2001, **"Implementing a Software Metrics Program at Nokia"**, IEEE Software 18 (nov./dez. 2001), p. 72-76
- KOUTSOMITROPOULOS, D. A., SOLOMOU, G. D., ALEXOPOULOS, A. D., PAPTAEODOROU, T. S., 2010, **"Semantic Web Enabled Digital Repositories"**, International Journal on Digital Libraries, vol. 10, Issue 4, pp. 179-199
- LEÃO, A.F.C., 2012, **"RecMed: Uma Ferramenta de Apoio ao Uso de um Conjunto de Recomendações para Medição de Software Adequada ao Controle Estatístico de Processos"**, Projeto Final de Graduação do Curso de Ciência da Computação, Departamento de Informática, Universidade Federal do Espírito, Vitória - ES, Dezembro.
- MARETTO, C. X., 2013, **"Uma Arquitetura de Referência para Medição de Software"**, Tese de Mestrado do Programa de Pós-Graduação do Departamento de Informática, Universidade Federal do Espírito Santo, Vitória - ES, Dezembro.
- MARTINS, A.F., FALBO, R.A., 2008, **"Models for representing task ontologies"**. In Proceedings of the 3rd Workshop on Ontologies and Their Applications, Brazil.
- McGARRY, J., CARD, D., JONES, C., LAYMAN, B., CLARK, E., DEAN, J., HALL, F., 2002, **"Practical Software Measurement: Objective Information for Decision Makers"**, Addison Wesley, Boston, USA.

- NIESSINK, F., VLIET, H., 2001, **"Measurement Program Success Factors Revisited"**, Information and Software Technology, v. 43, n. 10, pp. 617-628.
- PUTNAM, L., MYERS, W., 2003, **"Five Core Metrics: The Intelligence Behind Successful Software Management"**, Dorset House Publishing.
- REYNOLDS, H., KOULOPOULOS, T., 1999, **"Enterprise Knowledge Has a Face"**, Intelligent Enterprise, v. 2, n. 5, p. 29-34, Mar. 1999. [online], abril 2000. [<http://www.intelligententerprise.com/993003/feat1.shtml>]
- SCHNEIDEWIND, N. F., 2002, **"Body of Knowledge for Software Quality Measurement"**, IEEE Computer, v. 35, n. 2, pp. 77-83.
- SEI (2010). **Capability Maturity Model Integration - CMMI-DEV 1.3**. Pittsburgh, Pennsylvania, USA, Carnegie Mellon Software Engineering Institute.
- SOFTEX: **MPS.BR: Melhoria de Processo do Software Brasileiro - Guia Geral MPS de Software:2012**. Disponível em: <http://www.softex.br/mpsbr> (2012)
- SOMMERVILLE, I., 2011, **Engenharia de Software**, 9ª Edição. São Paulo: Pearson Prentice Hall
- SOUZA, 2007, **"FrameWeb: um Método baseado em Frameworks para o Projeto de Sistemas de Informação Web"**, Dissertação de Mestrado, Departamento de Informática, Universidade Federal do Espírito Santo.
- TARHAN, A.; DEMIRÖRS, O. 2006, **"Investigating Suitability of Software Process and Metrics for Statistical Process Control"**. p.88–99
- TATE, M.E., J. Hope, B. Barnes, S. , 2007, **"Perceived Service Quality in a University Web Portal: Revising the E-Qual Instrument"**.
- VILLELA, K., ROCHA, A. R., TRAVASSOS, G. H., 2005, **"The use of an enterprise ontology to support knowledge management in software development environments"**. J Braz Comput Soc 11(2):45–59
- WANG, Q., LI, M., 2005, **"Measuring and Improving Software Process in China"**, In: Proceedings of International Symposium on Empirical Software Engineering - ISESE 2005, Hoosa Head, Australia, pp. 183-192.

WELLS, D., SHERMA, M., HARRIS-JONES, C., 2000, **“Enterprise Portals: New Strategies For Information Delivery”**, Ovum, Retrieved from www.ovum.com

XIAO, L., DASGUPTA, S., 2005, **“User Satisfaction with Web Portals: An empirical Study”**. In: Gao, Y. (eds.) *Web Systems Design and Online Consumer Behavior*, pp. 193–205. Idea Group, Hershey, PA (2005)

YANG, Z., CAI, S., ZHOU, Z., ZHOU, N., 2004, **“Development and validation of an instrument to measure user perceived service quality of information presenting Web portals”**. *Inf. Manage.* 42, 575–589 (2004), Elsevier Science