

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

Vinícius Gama Valory Frauches

MeMOS: Uma Ferramenta de Apoio à Medição e ao Monitoramento de Objetivos 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
2011

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

Vinícius Gama Valory Frauches

**MeMOS: Uma Ferramenta de Apoio à Medição e ao
Monitoramento de Objetivos de Software**

COMISSÃO EXAMINADORA

Prof. Monalessa Perini Barcellos, D. Sc.

Prof. Ricardo de Almeida Falbo, D. Sc.

Prof. Davidson Cury, D. Sc.

Vitória, 21 de Dezembro de 2011

Ao meu pai Ronaldo, inspirador e exemplo em minha vida
A minha mãe Fátima pela compreensão e carinho
À minha esposa Juliana, meu porto seguro
e ao meu filho Diogo de onde tirei forças para continuar

AGRADECIMENTOS

À Deus, pela força e pela fé que me proporcionou a conclusão deste trabalho.

Aos meus pais Ronaldo e Fátima pelo companheirismo, acolhimento e compreensão em todos esses anos. Por sempre apoiarem as minhas decisões, sempre tentarem passar o conhecimento individual de cada um. E por toda a experiência pessoal que adquiri até aqui.

À minha mais nova família, Juliana e Diogo, por me dar um objetivo a ser alcançado. Foram a razão de lutar continuamente nesta batalha. Em especial a minha esposa Juliana, que esteve sempre que eu precise de alguém para contar. Minha parceira que me ajudou e me afagou quando precisava.

Aos meus colegas do NEMO, que durante este trabalho foram a minha segunda família, onde passei maior parte do dia, nesse último semestre. Ao Julião, Renato, Júlio, Vítor, Ludimila e todos aqueles que auxiliaram neste trabalho, seja pelo conhecimento passado, seja pelo companheirismo que acolhe.

À minha orientadora Monalessa por ter sido mais do que uma orientadora, uma amiga e conselheira. Pelo conhecimento passado, tanto pessoal, quanto profissional. Sou grato pela oportunidade dada e pela confiança depositada em meu potencial.

À todos os professores da UFES que formaram, de alguma forma, a base do meu conhecimento profissional.

Enfim, agradeço a todos aqueles que colaboraram ou desejaram, de alguma forma, o meu crescimento e a realização profissional e pessoal.

RESUMO

Para atingir um grau satisfatório de competitividade, ou até mesmo a sobrevivência no mercado, organizações de software devem desenvolver projetos de qualidade e alinhados aos seus objetivos de negócio. Assim, faz-se necessário que as organizações monitorem seus projetos em relação ao alcance dos objetivos estabelecidos e executem ações a fim de corrigir possíveis desvios.

A monitoração dos objetivos pode ser realizada através da análise de informações que permitam conhecer o quão alinhado aos seus objetivos uma organização ou um projeto está. Para isso, é preciso que se tenha conhecimento acerca dos elementos da organização relevantes ao alcance dos objetivos. Esse conhecimento pode ser adquirido com a Medição de Software.

Medição de software é um importante processo da Engenharia de Software e é considerada uma prática básica. Ela é encontrada em boa parte das organizações de software bem sucedidas. Sendo um processo relativamente novo e tendo ganho mais atenção das organizações apenas há pouco tempo (pouco mais de uma década), o apoio computacional provido ainda é limitado.

Nesse contexto, este trabalho foi desenvolvido com o objetivo de definir um processo de medição e monitoramento de objetivos de software, com base nas propostas da literatura e desenvolver a versão inicial de uma ferramenta para apoiar o processo proposto. A ferramenta foi desenvolvida no contexto do ambiente ODE - *Ontology-based Development Environment* (FALBO et al., 2005). Seguindo a abordagem desse ambiente, o qual é baseado em ontologias, a ferramenta proposta utilizou como base a Ontologia de Medição de Software definida em (BARCELLOS, 2009).

Palavras-chave: medição de software, monitoramento de objetivos, ambiente ODE.

SUMÁRIO

CAPÍTULO 1 - INTRODUÇÃO	7
1.1 INTRODUÇÃO	7
1.2 OBJETIVOS	10
1.3 HISTÓRICO DE DESENVOLVIMENTO DO TRABALHO	10
1.4 ORGANIZAÇÃO DO TEXTO.....	11
CAPÍTULO 2 - MEDIÇÃO DE SOFTWARE.....	12
2.1 INTRODUÇÃO.....	12
2.2 MEDIÇÃO DE SOFTWARE.....	12
2.3 CONCEITOS DA MEDIÇÃO DE SOFTWARE	13
2.4 PROCESSO DE MEDIÇÃO DE SOFTWARE.....	15
2.5 DEFINIÇÃO E MONITORAMENTO DE OBJETIVOS	17
2.6 O AMBIENTE ODE.....	19
CAPÍTULO 3 - ESPECIFICAÇÃO DE REQUISITOS E ANÁLISE DA FERRAMENTA MEMOS	21
3.1 INTRODUÇÃO.....	21
3.2 PROCESSO DE MEDIÇÃO DE SOFTWARE E MONITORAMENTO DE OBJETIVOS.....	21
3.3 MEMOS: PROPÓSITO DO SISTEMA E DESCRIÇÃO DO MINIMUNDO	30
3.4 SUBSISTEMAS DA FERRAMENTA MEMOS	33
3.5 CASOS DE USO DA FERRAMENTA MEMOS.....	34
3.6 DIAGRAMAS DE CLASSES	40
CAPÍTULO 4 - PROJETO E IMPLEMENTAÇÃO DA FERRAMENTA MEMOS	50
4.1 INTRODUÇÃO.....	50
4.2 ARQUITETURA DE SOFTWARE	50
4.3 SUBSISTEMA <i>PLANEJAMENTO MEDICAO</i>	53
4.3.1 CAMADA LÓGICA DO NEGÓCIO	53
4.3.1.1 COMPONENTE DOMÍNIO DO PROBLEMA	53
4.3.1.2 COMPONENTE DE GERÊNCIA DE TAREFAS	55
4.3.2 CAMADA DE INTERFACE COM O USUÁRIO	55
4.3.3 CAMADA DE GERÊNCIA DE DADOS	56
4.4 DEMAIS SUBSISTEMAS DE MEMOS	59
4.5 A FERRAMENTA MEMOS	64
CAPÍTULO 5 - CONSIDERAÇÕES FINAIS.....	74
5.1 CONCLUSÕES	74
5.2 TRABALHOS FUTUROS.....	76
REFERÊNCIAS BIBLIOGRÁFICAS	77

Capítulo 1

Introdução

1.1 Introdução

Para permanecer no mercado com a atual competitividade, organizações de software devem desenvolver seus projetos com qualidade e de forma que seus objetivos de negócio sejam atendidos. Em outras palavras, os projetos realizados pelas organizações de software devem estar alinhados aos seus objetivos de negócio.

Cada projeto possui um conjunto de objetivos, estabelecidos com base nos objetivos da organização, que devem ser alcançados. Durante a execução dos projetos, é necessário verificar se os objetivos estão sendo alcançados e, caso não estejam, realizar ações corretivas para tratar os desvios.

O monitoramento dos objetivos dos projetos é realizado com o apoio da medição de software, que fornece dados coletados ao longo dos projetos e permite que a situação atual e progresso dos projetos sejam conhecidos.

De fato, a medição de software é uma das práticas mais importantes no contexto da Engenharia de Software. Segundo McGARRY *et al.* (2002), a maioria das organizações de software bem sucedidas implementa a medição como parte de suas atividades técnicas e gerenciais. A medição provê a informação necessária às tomadas de decisão que podem impactar no desempenho técnico e de negócio das organizações (BARCELLOS, 2008).

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 (SEI, 2010), através da área de processo Medição e Análise e do nível F do MR MPS (SOFTEX, 2011), através do processo Medição (BARCELLOS, 2009).

Medição de software pode ser utilizada em vários contextos em uma organização. 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. Ou seja, 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 de 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).

Apesar da importância da medição de software ser reconhecida, ainda há muitos problemas inerentes a esse tema. Isso ocorre, principalmente, pelo fato de a medição de software ainda ser um tema recente (BARCELLOS, 2008).

Um dos problemas relacionados à medição é a falta de um vocabulário consensual. Terminologias, conceitos, procedimentos e métodos de medição de software vêm sendo definidos na última década, porém, em particular, não há consenso para conceitos e terminologias, havendo duplicações e inconsistências nas propostas encontradas na literatura, inclusive nos termos mais comuns da área como medida, métrica e medição (GARCÍA *et al.*, 2006 *apud* BARCELLOS, 2009).

Outro problema é a falta de apoio computacional adequado para o processo de medição. No Brasil, algumas ferramentas foram propostas no contexto de ambientes de desenvolvimento de software como o TABA (ROCHA *et al.*, 1990) e o WebAPSEE (LIMA *et al.*, 2006). No TABA, a medição é apoiada pelas ferramentas Metrics e MedPlan (SCHNAIDER *et al.*, 2004) e no WebAPSEE a medição é apoiada por um conjunto de funcionalidades (não há um nome específico para o conjunto dessas funcionalidades) (RIBEIRO *et al.*, 2011). No entanto, essas ferramentas somente podem ser utilizadas no contexto dos seus ambientes, ou seja, para usar a ferramenta de medição, todo o ambiente deve ser utilizado.

No âmbito internacional, DUMKE e EBERT (2010) citam algumas ferramentas que apoiam o processo de medição. No entanto, a maioria das ferramentas diz respeito à automatização da medição de uma medida específica ou de um conjunto de medidas. Por exemplo, a ferramenta *Logiscop*, que realiza análise estática de sistemas. A única ferramenta citada pelos autores que, aparentemente, cobre todo o processo de medição é a ferramenta *e-measurement*. Contudo, essa ferramenta não foi encontrada para análise.

Considerando esses problemas, acredita-se que uma ferramenta que apoie a medição de software e monitoramento de objetivos seja útil às organizações que realizam ou desejam realizar as práticas relacionadas à medição. Cabe ressaltar que, no Brasil, com o crescimento do interesse por parte das organizações pelo MR MPS.BR – Modelo de Referência para Melhoria de Processo de Software Brasileiro (SOFTEX, 2011), a necessidade de se realizar medição vem se tornando mais evidente, uma vez que ela é um processo exigido no nível F do modelo, um dos níveis iniciais.

No NEMO, Núcleo de Estudos em Modelagem Conceitual e Ontologias do Departamento de Informática da UFES, há um projeto chamado ODE - *Ontology-based Development Environment* (FALBO *et al.*, 2005), o qual trata do desenvolvimento de um ambiente de Engenharia de Software fundamentado em ontologias. O ambiente foi originalmente desenvolvido para plataforma *desktop* e vem sendo migrado para plataforma *web*. Até este trabalho, ODE não possuía ferramenta de apoio à medição de software. Sendo assim, decidiu-se pelo desenvolvimento da ferramenta de apoio à medição no contexto do ODE. Para evitar a necessidade de se utilizar todo o ambiente ODE para que a ferramenta de medição possa ser utilizada, como ocorre no TABA e no WebAPSEE, decidiu-se pelo desenvolvimento de uma ferramenta que, embora faça parte do ODE, possa ser utilizada independentemente dele.

Uma vez que ODE é fundamentado em ontologias, a ferramenta de medição também a deve ser. Nesse sentido, foi escolhida a Ontologia de Medição de Software proposta em (BARCELLOS, 2009). Ela define um vocabulário comum ao domínio de medição de software e representa o conhecimento relevante sobre esse domínio, contribuindo para o compartilhamento e a reutilização desse conhecimento, bem como para a interoperabilidade semântica. Além disso, apesar de não ser foco deste trabalho, a Ontologia de Medição de Software foi desenvolvida considerando os aspectos relacionados à medição em alta maturidade, necessários nos níveis 4 e 5 do CMMI – *Capability Maturity Model Integration* (SEI, 2010) e nos níveis A e B do MR MPS.BR (SOFTEX, 2011).

Esse ponto pode ser considerado um diferencial deste trabalho, uma vez que, conforme discutido em (BARCELLOS, 2009), os maiores problemas para realizar a medição na alta maturidade estão relacionados ao início errado da medição. Ou seja, os problemas identificados nos níveis mais altos têm origem nos níveis iniciais e, muitas vezes, o que foi feito nos níveis iniciais precisa ser descartado nos níveis mais altos. Assim, uma vez que a ferramenta de medição proposta é baseada em uma ontologia que representa a conceituação do domínio de medição considerando tanto

aspectos da medição tradicional quanto da medição em alta maturidade, sua evolução para apoiar funcionalidades da alta maturidade é facilitada.

1.2 Objetivos

O objetivo principal deste trabalho é desenvolver, no ambiente ODE (*Ontology-based software Development Environment*) (FALBO *et al.*, 2005), a primeira versão de uma ferramenta de apoio ao processo para medição de software e monitoramento de objetivos em projetos de software definido durante a realização da Iniciação Científica que antecedeu este trabalho. Como objetivos específicos do projeto, têm-se:

- Identificar e documentar os requisitos da ferramenta;
- Realizar a modelagem comportamental e estrutural da ferramenta e documentar uma Especificação de Requisitos da mesma;
- Definir a arquitetura da ferramenta e detalhá-la em um Documento de Projeto;
- Implementar a versão preliminar da ferramenta.

1.3 Histórico de Desenvolvimento do Trabalho

Este trabalho foi realizado segundo as seguintes etapas:

- a) *Aquisição de conhecimento sobre o tema:* nesta etapa foi realizada uma revisão da literatura relacionada ao tema do trabalho, incluindo o estudo dos aspectos da Ontologia de Medição de Software (BARCELLOS, 2009) relevantes ao trabalho.
- b) *Definição do processo proposto:* nesta etapa o processo de medição de software e monitoramento de objetivos de projetos de software foi definido.
- c) *Estudo do Ambiente ODE:* nesta etapa, foram estudados aspectos da literatura relacionados a ferramentas CASE e ambientes de desenvolvimento de software, bem como sobre o ambiente ODE. Nesta etapa também foram realizados treinamentos e estudos sobre as tecnologias envolvidas no projeto, entre elas: linguagem Java, desenvolvimento Web, arquitetura em camadas, infraestrutura do ODE, framework Zkoss, framework Hibernate e a injeção de dependências do framework Spring.

- d) *Desenvolvimento da ferramenta:* esta etapa envolveu atividades de especificação de requisitos, análise, projeto, implementação e testes, segundo o paradigma da orientação a objetos, usando a linguagem de modelagem unificada (UML) e as tecnologias do ambiente ODE (serão descritas no capítulo 4).
- e) *Elaboração da Monografia:* nesta etapa foi realizada a escrita desta monografia.

Cabe ressaltar que este trabalho teve origem em dois projetos de Iniciação Científica (IC), um que tratava do processo de medição propriamente dito e outro, desenvolvido pelo autor, que tratava da definição e monitoramento de objetivos de projetos de software. Durante a IC, foram realizadas as atividades a) e b), foram iniciados os estudos individuais sobre as tecnologias envolvidas, foram identificados os requisitos da ferramenta e realizados os primeiros esboços dos casos de uso. Após o término do prazo dos projetos de IC, um dos alunos deixou o projeto e as propostas dos dois projetos foram unificadas neste trabalho. No contexto deste trabalho, foi feita a revisão dos requisitos e casos de uso identificados anteriormente, a elaboração da documentação de especificação de requisitos e de projeto e o desenvolvimento da ferramenta.

1.4 Organização do Texto

Este trabalho está organizado em 5 capítulos. Além desta Introdução, existem os capítulos a seguir:

Capítulo 2 - Revisão Bibliográfica: apresenta a fundamentação teórica do trabalho, que inclui medição de software, definição e monitoração de objetivos e o ambiente ODE,

Capítulo 3 – Especificação de Requisitos e Análise do Sistema: apresenta o processo de medição de software e monitoração de objetivos proposto, a descrição do minimundo contemplado pela ferramenta, seus casos de uso e modelos de classe.

Capítulo 4 – Projeto de Sistema, Implementação e Testes: apresenta a arquitetura definida para a ferramenta, detalha alguns dos componentes dessa arquitetura e apresenta algumas telas da ferramenta.

Capítulo 5 – Considerações Finais: apresenta as considerações finais do trabalho, destacando-se as contribuições e trabalhos futuros.

Capítulo 2

Medição de Software

2.1 Introdução

Este capítulo apresenta a fundamentação teórica deste trabalho. Está assim organizado: a Seção 2.2 apresenta a definição de medição e um breve histórico; na Seção 2.3 são apresentados aos conceitos centrais de medição de software; na Seção 2.4 são apresentados alguns padrões para o processo de medição; na Seção 2.5 discute-se a definição e o monitoramento de objetivos; e, na Seção 2.6 o ambiente ODE é apresentado.

2.2 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). Para essa avaliação quantitativa utilizam-se as medidas, as quais caracterizam, como foi dito, em termos quantitativos, alguma propriedade de uma entidade da Engenharia de Software (BASILI e ROMBACH, 1994 *apud* BARCELLOS, 2009).

Apesar de ser recente, medição de software não é um conceito novo. Mede-se software desde o primeiro compilador que contava o número de linhas de código de um programa. Em 1974, Donald Knuth utilizava dados obtidos em medições para demonstrar como o compilador FORTRAN poderia ser otimizado, baseado no uso específico de determinadas combinações da linguagem.

Até o início dos anos 80, a medição de software esteve em seu estágio primitivo, onde os objetivos de sua utilização em organizações de software não eram explícitos ou, se eram, não eram compreensíveis a seus membros. Nesse período, o escopo das medições estava limitado aos projetos, mais especificamente, às fases finais do desenvolvimento, ligadas mais à codificação propriamente dita que a qualquer outra atividade no ciclo de vida do software. As necessidades dos gerentes de projeto e demais envolvidos não eram entendidas e, assim, não podiam ser atendidas pela medição. Não havia uma metodologia de medição definida que fosse seguida e, como resultado, as métricas eram definidas baseando-se na intuição dos gerentes, não eram validadas e seus resultados não eram capazes de orientar ações de melhoria nos projetos. As atividades de medição que tinham

apoio automatizado eram limitadas a métricas referentes ao código, como, por exemplo, tempo de processamento de uma rotina e número de linhas de código (BARCELLOS, 2009).

Na primeira metade da década de 80 começaram a surgir propostas de modelos e métricas para compor uma metodologia de medição. O escopo das medições passava a atender os projetos, em todos os seus níveis, e os processos. Os objetivos da medição foram entendidos e o *feedback* positivo às medições realizadas começou a impulsionar as equipes de projetos de software a utilizarem os modelos e métricas propostos (BARCELLOS, 2009).

Em seguida, nos anos 90, foram desenvolvidos modelos para o processo de medição baseados na melhoria de processos e qualidade total, fornecendo as diretrizes e infraestrutura necessárias para definir, coletar, validar e analisar métricas.

Atualmente, como dito antes, a medição de software é um dos temas mais importantes na Engenharia de Software, sendo essa importância evidenciada por sua inclusão nos requisitos do nível 2 do CMMI (*Capability Maturity Model Integration*) (SEI, 2010) e do nível F do MR MPS (SOFTTEX, 2011) (BARCELLOS, 2009).

2.3 Conceitos da Medição de Software

Conforme discutido na Introdução deste trabalho, medição de software é considerada uma disciplina relativamente recente e, como tal, ainda não foram estabelecidos padrões consensuais para a medição de software (BARCELLOS, 2009). Em particular, não há consenso para conceitos e terminologias, havendo duplicações e inconsistências nas propostas encontradas na literatura, inclusive nos termos mais comuns da área (GARCÍA et al., 2006 *apud* BARCELLOS, 2009).

Devido a essa heterogeneidade do vocabulário relacionado à medição de software, BARCELLOS (2009) desenvolveu uma Ontologia de Medição de Software que define um vocabulário comum ao domínio de medição de software. Essa ontologia é utilizada como modelo conceitual de referência para este trabalho. A seguir são apresentados os conceitos dessa ontologia considerados centrais para este trabalho (BARCELLOS, 2011).

- **Medida:** quantificação de atributos de entidades. Pode ser medida base (ex.: *prazo estimado para o projeto, prazo real do projeto*) ou medida derivada (ex.: *aderência ao prazo do projeto, dada pela razão entre o prazo real do projeto e o prazo estimado para o projeto*).

- **Elemento Mensurável:** propriedade de uma entidade que pode ser quantificada. Ex.: a medida *prazo estimado para o projeto* quantifica o elemento mensurável *tempo*.
- **Entidade Mensurável:** entidade que pode ser caracterizada pela quantificação de seus atributos. Ex.: a medida *prazo estimado para o projeto* quantifica o atributo *tempo* da entidade *Projeto P*.
- **Unidade de Medida:** unidade por meio da qual uma medida pode ser expressa. Ex.: a medida *prazo estimado para o projeto* pode ser expressa em *horas*. A medida *aderência ao prazo do projeto* não possui unidade de medida.
- **Escala:** indica os valores que podem ser atribuídos a uma medida. Ex.: A medida *prazo estimado para o projeto* possui uma escala do tipo Absoluta que é composta pelos números reais positivos.
- **Procedimento de Medição:** procedimento que descreve como a coleta da medida deve ser realizada. Ex.: A medida *aderência ao prazo do projeto* poderia ter como procedimento de medição: Aplicar a fórmula de cálculo de medida que determina a razão entre o prazo real do projeto e o prazo estimado para o projeto.
- **Procedimento de Análise de Medição:** procedimento que descreve como os dados coletados para a medida devem ser representados e analisados. Ex.: A medida *aderência ao prazo do projeto* poderia ter como procedimento de análise: Representar em histograma as taxas de aderência ao prazo dos projetos. Valores superiores a 1 indicam que o projeto levou mais tempo que o previsto, valores menores que 1 indicam que o projeto levou menos o que o previsto e valores iguais a 1 indicam que o projeto levou exatamente o tempo que foi previsto. Analisar os valores medidos para os projetos e compará-los uns com os outros. Subagrupar os dados e representá-los em outros histogramas a fim de identificar e analisar: (i) as diferenças das taxas de acordo com as características dos projetos; (ii) as diferenças das taxas de acordo com a fase em que o projeto se encontra.
- **Definição Operacional de Medida:** detalhamento associado a uma medida que fornece informações sobre sua coleta e análise. Inclui procedimentos de medição e de análise, os papéis dos responsáveis pela medição e pela análise da medida, os momentos em que a medição e a análise devem ser realizadas e a periodicidade em que a medição e a análise devem ser realizadas.

- **Medição:** ação de medir, ou seja, de atribuir um valor a uma medida, executando seu procedimento de medição. Ex.: medição do *prazo previsto para o projeto* obtendo-se o valor 500.
- **Análise de Medição:** ação de analisar os dados coletados para uma medida, executando seu procedimento de análise. Ex.: análise da *aderência ao prazo do projeto*, concluindo-se que os projetos que envolveram o uso de nova tecnologia apresentaram uma aderência 10% menor aos prazos previstos do que a média das aderências dos projetos que não utilizaram novas tecnologias.
- **Indicador:** medida utilizada para analisar o alcance de objetivos. Ex.: *aderência ao prazo do projeto* poderia ser um indicador para o objetivo *melhorar a aderência dos projetos aos planos*.

2.4 Processo de Medição de Software

O processo de medição é um conjunto de passos que deve orientar a realização da medição na organização, levando em consideração o modelo de informação definido.

Um processo de medição eficiente é fator crítico ao sucesso da medição na organização, pois é ele quem direciona as atividades a serem realizadas para que, ao final, 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 aos objetivos técnicos e de negócio (WANG e LI, 2005 *apud* BARCELLOS, 2009).

Existem várias propostas para o processo de medição que, apesar de possuírem pequenas diferenças entre si, são basicamente compostas por: definição e coleta das métricas, análise das medidas coletadas e utilização dos resultados da análise em ações. A seguir são brevemente descritas algumas das propostas para o processo de medição (BARCELLOS, 2008):

- BASILI e ROMBACH (1994b) propõem um processo para a utilização da medição de software nas organizações composto por seis fases, baseando-se no Paradigma de Melhoria da Qualidade (BASILI, 1993): (i) caracterizar os projetos correntes e seus ambientes utilizando métricas; (ii) identificar os objetivos e metas quantificáveis; (iii) definir o modelo de medição a ser utilizado; (iv) executar os processos, construir os produtos, coletar, validar e analisar as métricas, a fim de definir ações corretivas, se necessário; (v) analisar os dados da medição para avaliar as práticas utilizadas,

determinar problemas e realizar recomendações para os projetos futuros; e, (vi) armazenar o conhecimento obtido para uso em projetos futuros.

- ISO/IEC 15939 (2002) define um processo de medição que é descrito como um modelo que define as atividades do processo de medição que são requeridas para, adequadamente, especificar que informações de medição são necessárias, como as medidas serão realizadas, como seus resultados serão analisados e como avaliar se os resultados são válidos. O processo consiste de quatro atividades que são sequenciadas em um ciclo iterativo, permitindo um *feedback* e melhoria contínua do processo. Ele é uma adaptação do ciclo PDCA (*Plan-Do-Check-Act*), comumente utilizado como base para a melhoria da qualidade. Suas atividades são: (i) estabelecer e manter comprometimento com a medição; (ii) planejar o processo de medição; (iii) executar o processo de medição; e, (iv) avaliar a medição.
- IEEE Std 1061-1998 (1998) também propõe um processo de medição, porém, seu contexto é limitado aos objetivos de qualidade de software. O processo é composto por cinco passos: (i) estabelecer os requisitos de qualidade de software; (ii) identificar as métricas de qualidade de software; (iii) implementar as métricas; (iv) analisar os resultados das métricas; e, (v) validar as métricas.
- *Practical Software Measurement* – PSM (McGARRY *et al.*, 2002) define uma abordagem para medição de software orientada às necessidades de informação organizacionais aderente à ISO/IEC 15939 e, como ela, possui dois componentes: um modelo de informação de medição e um processo de medição. Considerando o modelo de informação definido, o PSM propõe um processo de medição composto por quatro fases: (i) planejamento; (ii) execução; (iii) avaliação; e (iv) comprometimento.

Observando-se as primeiras etapas dos processos de medição propostos nas abordagens apresentadas, nota-se que elas são responsáveis pela identificação e definição das medidas, bem como pela associação destas aos objetivos organizacionais. Essas etapas são de grande importância para a medição, pois são elas que definem que informações serão fornecidas para apoiar as tomadas de decisão (BARCELLOS, 2009).

Há algumas abordagens que apoiam a identificação e seleção de medidas adequadas à avaliação do alcance dos objetivos organizacionais. Uma das abordagens mais conhecidas é o *Goal Question Metric* - GQM (BASILI *et al.*, 1994) que considera que, para cada objetivo estabelecido, é possível

determinar questões cujas respostas estão associadas a medidas. A Figura 2.1 apresenta um exemplo para a relação de um objetivo de negócio, suas questões e medidas associadas.

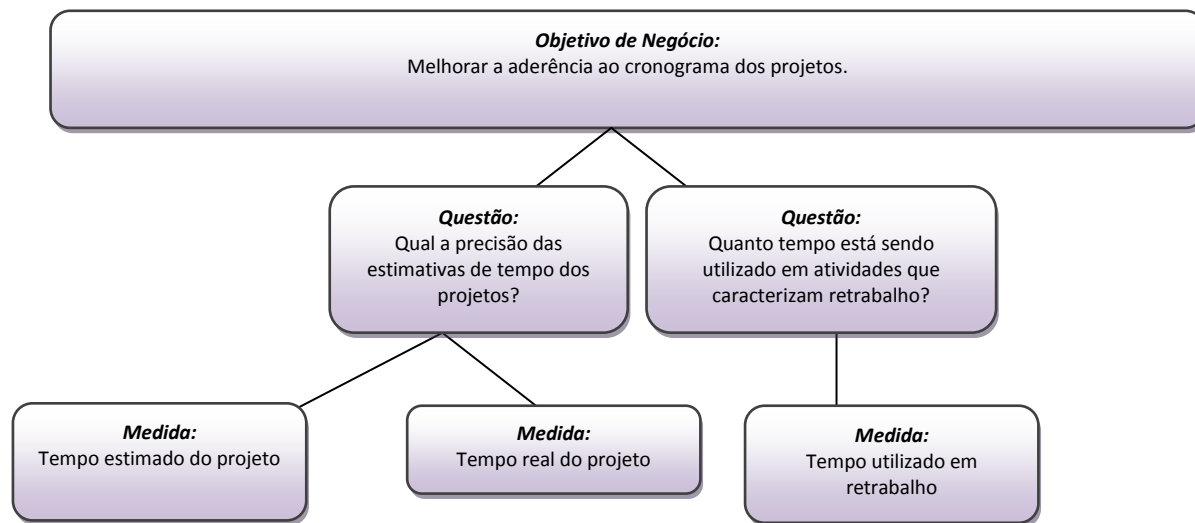


Figura 2.1 – Exemplo de aplicação do GQM (BARCELLOS, 2009).

2.5 Definição e Monitoramento de Objetivos

Organizações de software desenvolvem projetos que são realizados para atender objetivos específicos. Esses objetivos devem ser estabelecidos considerando o planejamento estratégico da organização, onde constam seus objetivos de negócio, definidos com foco na competitividade organizacional (PORTER, 2004).

A operacionalização do planejamento estratégico pode ser alcançada através da execução de projetos, pois a correta execução dos projetos tende a atender as metas e, conseqüentemente, os objetivos, missão e visão da empresa (PMBOK, 2008; SOUZA, 2008). Um projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo (PMBOK, 2008). Particularmente em organizações que desenvolvem produtos de software, o nível operacional normalmente está organizado em projetos e, assim, seu planejamento operacional passa a ser o planejamento de cada projeto a ser executado (BARRETO e ROCHA, 2009).

Uma das primeiras etapas do planejamento dos projetos é definir seus objetivos, isto é, objetivos específicos para o projeto que representem as restrições do projeto e necessidades do cliente, mantendo um alinhamento aos objetivos de negócio da organização. Os objetivos do projeto incluem os critérios mensuráveis do sucesso do projeto. Os projetos podem possuir uma ampla

variedade de objetivos técnicos, de negócio, custo, cronograma e qualidade (ISO/IEC, 2008; PMBOK, 2008). Como objetivos operacionais, os objetivos dos projetos proveem a base para medição do progresso em direção ao alcance dos objetivos estratégicos (MARKOVIC e KOWALKIEWICZ, 2008).

Existem diversas abordagens que tratam da definição de objetivos, tais como: GQM - *Goal Question Metrics* (BASILI *et al.*, 1994), *Goal-Question-(Indicator)-Measure* - GQ(I)M (PARK *et al.*, 1996), COBIT - *Control Objectives for Information Technology* (IT GOVERNANCE INSTITUTE, 2007) e BSC - *Balanced Score Card* (KAPLAN e NORTON, 1996). As duas primeiras abordagens citadas são específicas para organizações de software. As demais podem ser aplicadas em qualquer tipo de organização.

O *Goal Question Metric* – GQM (BASILI *et al.*, 1994), citado anteriormente, é uma abordagem específica para organizações de software. Seguindo essa abordagem, os objetivos são definidos de modo que sejam operacionalmente tratáveis, por meio do refinamento para um conjunto de questões quantificáveis que são utilizadas para extrair as informações apropriadas. Medidas são derivadas com base nas questões, formalizando o processo e levando à escolha e/ou definição de medidas relevantes. Por fim, os dados coletados são interpretados no contexto dos objetivos e questões definidos (SOLINGEN e BERGHOUT, 1999).

O *Goal-Question-(Indicator)-Measure* - GQ(I)M (PARK *et al.*, 1996) é uma variação do GQM que inclui a definição de indicadores para o monitoramento dos objetivos.

O COBIT, proposto pelo Instituto de Governança de TI, descreve um conjunto de objetivos de negócio genéricos e um conjunto de objetivos de TI, também genéricos, relacionados aos objetivos de negócio. Em relação à medição, o COBIT identifica alguns indicadores e medidas que podem ser usados para monitorar as atividades, os processos e os objetivos.

Por fim, o BSC (KAPLAN e NORTON, 1996) é um *framework* para descrever, implementar e gerenciar a estratégia de uma organização em todos os seus níveis, relacionando objetivos, iniciativas e medidas para esta estratégia. Ele traduz a visão e a estratégia da organização em um conjunto abrangente de indicadores de desempenho que provê a estrutura para o sistema de medição e gerenciamento estratégico. O conjunto de indicadores inclui medidas financeiras, que refletem os resultados de ações tomadas no passado, e, de forma complementar, inclui as medidas operacionais, tais como medidas para avaliar a satisfação do cliente, processos internos, inovação e melhoria organizacional que guiarão o desempenho financeiro futuro da organização.

Apesar de abordarem a definição e monitoramento dos objetivos de forma distinta, há algo comum entre as abordagens citadas: a utilização de medidas e indicadores para monitorar o alcance dos objetivos. Dessa forma, percebe-se que a medição é essencial para o monitoramento do alcance dos objetivos.

2.6 O Ambiente ODE

O ambiente ODE (*Ontology-based software Development Environment*) (FALBO *et al.*, 2003) é um Ambiente de Desenvolvimento de Software (ADS) centrado em processo e baseado em ontologias. Esse ambiente é desenvolvido no NEMO - Núcleo de Estudos em Modelagem Conceitual e Ontologias, no Departamento de Informática da UFES.

A equipe do Projeto ODE é composta por professores, alunos de doutorado, mestrado e graduação da UFES. Recentemente, professores do IFES (Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo), que foram membros do Projeto ODE quando alunos da UFES, retornaram à equipe com alguns de seus alunos de graduação.

Algumas características de ODE que merecem destaque são: a uniformidade de conceitos provida pelas ontologias, que facilita a integração, deixa o ambiente mais homogêneo e torna mais efetiva a comunicação entre pessoas e entre ferramentas; a forte base em conhecimento, que permite que o ambiente ofereça um apoio especializado ao usuário na realização de suas tarefas e possibilita que as informações geradas mantenham-se interligadas e consistentes ao longo de todo o processo; e o foco em ferramentas gerenciais, uma vez que a gerência é uma área de grande importância e ainda carente em termos de ferramentas.

O ODE constitui um conjunto de ferramentas de apoio a áreas da engenharia de software como, apoio à definição de processos (BERTOLLO, SEGRINI *et al.*, 2006; SEGRINI, 2009), de acompanhamento de projetos (MORO *et al.*, 2005), de gerência de requisitos (MARTINS *et al.*, 2006), de gerência de riscos (FALBO *et al.*, 2004) e de gerência de conhecimento (NATALI *et al.*, 2003).

ODE foi originalmente desenvolvido em plataforma *desktop*, porém com o crescimento do ambiente, as mudanças tecnológicas e as necessidades de organizações clientes, em 2009 teve início a migração do ODE para a plataforma *web*.

A migração vem sendo realizada gradativamente. Algumas ferramentas consideradas centrais, como a de apoio à definição de processos e a de apoio à caracterização de projetos foram

selecionadas para serem as primeiras a serem migradas. Além das ferramentas que serão migradas, novas ferramentas estão sendo desenvolvidas para o ODE, dentre elas, a ferramenta MeMOS, proposta neste trabalho.

Cabe ressaltar que uma grande diferença da versão ODE para web em relação à versão para *desktop* é que espera-se que na versão atual as ferramentas possam ser utilizadas tanto integradas ao ODE quanto isoladas.

Capítulo 3

Especificação de Requisitos e Análise da Ferramenta

MeMOS

3.1 Introdução

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).

Este capítulo aborda os resultados da Engenharia de Requisitos da ferramenta MeMOS e apresenta o processo que a ferramenta apoia. Na Seção 3.2 é apresentado o processo de medição e monitoramento de objetivos definido neste trabalho; na Seção 3.3 é apresentado o objetivo da ferramenta e a descrição do minimundo; na Seção 3.4 são apresentados os subsistemas de MeMOS; na Seção 3.5 são apresentados diagramas de casos de uso e na Seção 3.6 são apresentados os modelos de classes.

3.2 Processo de Medição de Software e Monitoramento de Objetivos

A seguir é apresentado o processo de medição de software e monitoramento de objetivos que deve ser apoiado por MeMOS. O processo baseou-se nas propostas da literatura citadas no Capítulo 2 e no conhecimento provido pela Ontologia de Medição de Software (BARCELLOS, 2009). Ele é composto por 5 atividades, que podem ser divididas em subatividades.

Atividade 1: Planejar Medição

Esta atividade consiste no planejamento da medição alinhada aos objetivos estratégicos da organização. Durante sua execução é elaborado o Plano de Medição.

Subatividade 1.1: Selecionar Objetivos Estratégicos relevantes à Medição

Descrição: Nesta subatividade, os objetivos estratégicos registrados no Planejamento Estratégico da organização são analisados e aqueles que são relevantes para a medição de software são selecionados. Um objetivo estratégico expressa a intenção pela qual ações estratégicas são planejadas e realizadas em uma organização. Por exemplo, “aumentar o número de clientes em 10%”.

Pré-atividade: Não há.

Entradas: Planejamento Estratégico da Organização.

Saídas: Objetivos estratégicos relevantes à medição selecionados.

Subatividade 1.2: Selecionar/Definir Objetivos de Software

Descrição: Com base nos objetivos estratégicos podem ser definidos objetivos de software, que são objetivos relacionados à área de software da organização, que contribuem para o alcance dos objetivos estratégicos aos quais estão relacionados. O objetivo “obter avaliação MPS.BR nível C” é um exemplo de objetivo de software. Objetivos de software podem ter sido definidos no Planejamento Estratégico ou no Planejamento Tático da organização. Nesse caso, nesta subatividade, os objetivos de software relacionados aos objetivos estratégicos relevantes à medição são selecionados. Caso a organização não tenha estabelecido os objetivos de software, nesta subatividade eles são definidos.

Pré-atividade: Registrar Objetivos Estratégicos relevantes à Medição.

Entradas: Objetivos estratégicos relevantes à medição.

Saídas: Objetivos de software definidos/selecionados.

Subatividade 1.3: Definir Objetivos de Medição

Descrição: Nesta subatividade, objetivos de medição são definidos com base nos objetivos de software ou nos objetivos estratégicos. No momento da definição de um objetivo de medição, deve ser informado seu tipo, podendo ser: *Objetivo de Monitoramento e Controle de Projetos* (por exemplo, “melhorar a aderência dos projetos aos planos”), *Objetivo de Medição de Qualidade* (por exemplo, “reduzir o número de defeitos dos produtos entregues”) ou *Objetivo de Medição de Desempenho* (por exemplo, “conhecer e melhorar o desempenho dos processos críticos”) (BARCELLOS, 2009).

Objetivos de Monitoramento e Controle de Projetos são, geralmente, definidos desde o início de um programa de medição, onde o monitoramento e controle dos projetos baseiam-se essencialmente na realização de estimativas para os projetos, medição de valores ao longo de seu desenvolvimento e comparação dos valores coletados com os valores planejados. *Objetivos de Medição de Qualidade* também estão presentes desde o início de um programa de medição, sendo definidas medidas que quantificam aspectos da qualidade dos produtos e processos. As medições são realizadas ao longo dos projetos e os valores coletados,

geralmente, são comparados entre si e com valores coletados anteriormente. Por fim, *Objetivos de Análise de Desempenho de Processos*, tipicamente, são identificados por uma organização quando ela já passou pelos níveis iniciais de maturidade (caracterizados pelos níveis 2 e 3 do CMMI e pelos níveis G a C do MR MPS.BR) e está realizando as práticas que caracterizam os níveis mais elevados (níveis 4 e 5 do CMMI e A e B do R MPS.BR). Nesses níveis é realizada a análise do desempenho dos processos nos projetos e comparação deste com o desempenho esperado para o processo no âmbito organizacional. Para isso, são utilizadas técnicas do controle estatístico de processos.

Pré-atividade: Definir Objetivos de Software.

Entradas: Objetivos de software.

Saídas: Objetivos de medição definidos/selecionados.

Subatividade 1.4: Identificar Necessidades de Informação

Descrição: Nesta subatividade, a partir dos objetivos de medição, são identificadas as informações necessárias à análise do alcance dos objetivos de medição. Por exemplo, a necessidade de informação “conhecer a estabilidade dos requisitos dos projetos após homologação junto ao cliente” pode ser identificada a partir do objetivo de medição “melhorar a aderência dos projetos aos planos”.

Pré-atividade: Definir Objetivos de Medição.

Entradas: Objetivos de medição.

Saídas: Necessidades de informação identificadas.

Subatividade 1.5: Identificar Processos Relacionados

Descrição: Nesta subatividade, os processos que estão relacionados aos objetivos de medição definidos são identificados. Um processo está relacionado a um objetivo de medição se ele é fonte de dados para as necessidades de informação identificadas para o objetivo de medição. Por exemplo, o processo de Gerência de Requisitos está relacionado ao objetivo de medição “melhorar a aderência dos projetos aos planos”, pois é fonte de dados para a necessidade de informação “conhecer a estabilidade dos requisitos dos projetos após homologação junto ao cliente”.

Pré-atividade: Identificar Necessidades de Informação.

Entradas: Objetivos de medição e necessidades de informação.

Saídas: Processos relacionados aos objetivos de medição identificados.

Subatividade 1.6: Definir/Selecionar Medidas

Descrição: Esta subatividade consiste na definição das medidas para atender às necessidades de informação identificadas. Para cada medida deve ser estabelecida uma definição operacional. Caso existam medidas previamente definidas, elas podem ser selecionadas. A definição operacional de uma medida deve incluir (BARCELLOS, 2009):

- i. Nome:* nome da medida.
- ii. Definição:* descrição sucinta da medida.
- iii. Mnemônico:* sigla utilizada para identificar a medida.
- iv. Tipo de Medida:* classificação da medida quanto à sua dependência funcional, podendo uma medida ser uma medida base ou uma medida derivada.
- v. Entidade Medida:* entidade que a medida mede. Exemplos: organização, projeto, processo, atividade, recurso humano, recurso de hardware, recurso de software e artefato, dentre outros.
- vi. Propriedade Medida:* propriedade da entidade medida quantificada pela medida. Exemplos: tamanho, custos, defeitos, esforço etc.
- vii. Unidade de Medida:* unidade de medida em relação à qual a medida é medida. Exemplos: pessoa/mês, pontos de função, reais etc.
- viii. Tipo de Escala:* natureza dos valores que podem ser atribuídos à medida. Exemplos: escala nominal, escala intervalar, escala ordinal, escala absoluta e escala taxa.
- ix. Valores da Escala:* valores que podem ser atribuídos à medida. Exemplos: números reais positivos, {alto, médio, baixo} etc. Para medidas com escala do tipo absoluta ou taxa, ao determinar os valores da escala, é preciso identificar a precisão a ser considerada (0, 1 ou 2 casas decimais).
- x. Intervalo esperado dos dados:* limites de valores da escala definida de acordo com dados históricos ou com metas estabelecidas. Exemplo: [0, 10].
- xi. Procedimento de Medição:* descrição do procedimento que deve ser realizado para coletar uma medida. A descrição do procedimento de medição deve ser clara, objetiva e não ambígua.
- xii. Fórmula de Cálculo de Medida:* fórmula utilizada no procedimento de medição de medidas derivadas, para calcular o valor atribuído à medida considerando-se sua

relação com outras medidas ou com outros valores. Exemplo: aderência ao cronograma = tempo real / tempo estimado.

- xiii. *Responsável pela Medição*: papel desempenhado pelo recurso humano responsável pela coleta da medida. É importante que o responsável pela medição seja fonte direta das informações a serem fornecidas na medição. Exemplos: analista de sistemas, programador, gerente do projeto etc.
- xiv. *Momento da Medição*: momento em que deve ser realizada a coleta e registro de dados para a medida. O momento da coleta deve ser uma atividade do processo definido para o projeto ou de um processo organizacional. Exemplos: na atividade Homologar Especificação de Requisitos, na atividade Realizar Testes de Unidade etc.
- xv. *Periodicidade de Medição*: frequência de coleta da medida. Exemplos: diária, mensal, uma vez por fase, uma vez por projeto, uma vez em cada ocorrência da atividade designada como momento da medição etc. É indispensável que haja coerência entre a periodicidade de medição e o momento de medição.
- xvi. *Procedimento de Análise*: descrição do procedimento que deve ser realizado para representar e analisar os dados coletados para uma medida, incluindo, além do procedimento propriamente dito, as ferramentas analíticas que devem ser utilizadas (por exemplo: histograma, gráfico de controle XmR etc.). A descrição do procedimento de análise deve ser clara, objetiva e não ambígua. Um procedimento de análise de medição pode ser baseado em critérios de decisão (por exemplo, utilizando-se uma meta como referência) e, nesse caso, os critérios de decisão considerados (incluindo suas premissas e conclusões) devem ser claramente estabelecidos. Medidas que não são analisadas isoladamente não precisam ter procedimento de análise definido. Por exemplo: se a medida número de requisitos alterados só for submetida à análise quando utilizada na composição da medida taxa de alteração de requisitos, não há necessidade de definir seu procedimento de análise.
- xvii. *Momento da Análise de Medição*: momento em que deve ser realizada a análise de dados coletados para a medida. O momento da análise deve ser uma atividade do processo definido para o projeto ou de um processo organizacional como, por exemplo, em atividades de monitoramento de projeto.

- xviii. *Periodicidade da Análise:* frequência de análise de dados da medida. Exemplos: diária, mensal, uma vez por fase, uma vez por projeto, uma vez em cada ocorrência da atividade designada como momento da análise etc. É indispensável que haja coerência entre a periodicidade de análise de medição e o momento da análise de medição.
- xix. *Responsável pela Análise:* papel desempenhado pelo recurso humano responsável pela análise da medida. É importante que o responsável pela análise de medição seja apto a aplicar o procedimento de análise e tenha conhecimento organizacional que propicie a correta interpretação dos dados e fornecimento de informações que apoiem as tomadas de decisão. Exemplos: gerente do projeto, gerente de qualidade etc.

Pré-atividade: Identificar Processos Relacionados.

Entradas: Necessidades de informação.

Saídas: Medidas definidas (ou selecionadas).

Subatividade 1.7: Registrar Plano de Medição

Descrição: Nesta subatividade as saídas de todas as subatividades anteriores são organizadas e registradas na forma de um Plano de Medição

Pré-atividade: Definir/Selecionar medidas.

Entradas: saídas das subatividades 1.1 a 1.6

Saídas: Plano de Medição

Atividade 2: Estabelecer Valores de Referência para os Indicadores

Descrição: Esta atividade consiste no planejamento de valores que serão utilizados como referência para a análise do alcance dos objetivos. Quando uma medida é utilizada para indicar o alcance de um determinado objetivo, ela assume o papel de indicador. Para cada indicador devem ser estabelecidos valores ou intervalos de valores que indiquem o alcance ou ameaça ao alcance dos objetivos. Por exemplo, pode ser definido um intervalo de valores considerados bons, que indica o alcance do objetivo, um intervalo de valores considerados regulares, que indica uma provável ameaça ao alcance do objetivo, e um intervalo de valores considerados ruins, que indica uma real ameaça ao alcance do objetivo.

Pré-atividade: Registrar Plano de Medição.

Entradas: Plano de Medição.

Saídas: Valores de referência dos indicadores estabelecidos.

Atividade 3: Executar Medições

Esta atividade consiste na coleta de dados para as medidas, validação e armazenamento desses dados.

Subatividade 3.1: Coletar Dados

Descrição: Nesta subatividade são coletados dados para as medidas definidas. A coleta dos dados de uma medida deve ser realizada de acordo com o procedimento de coleta estabelecido em sua definição operacional.

Pré-atividade: Estabelecer Valores de Referência para Indicadores.

Entradas: Plano de Medição (definição operacional das medidas).

Saídas: Dados coletados.

Subatividade 3.2: Validar Dados

Descrição: Nesta subatividade, os dados coletados para as medidas devem ser validados. A validação dos dados antes de seu armazenamento é fundamental para evitar que dados incorretos sejam armazenados. A validação dos dados pode ser feita comparando-se o dado coletado com as informações contidas na definição operacional da medida, como por exemplo, seu tipo e valores esperados.

Pré-atividade: Coletar dados.

Entradas: Dados coletados, Plano de Medição (definição operacional das medidas)

Saídas: Dados validados.

Subatividade 3.3: Armazenar Dados

Descrição: Esta subatividade consiste no armazenamento dos dados em um repositório organizacional de medidas. Ao armazenar os dados de uma medição devem ser incluídos: o valor medido, a data da medição, o executor da medição, o momento real da medição (atividade na qual a medição foi realizada) e o contexto da medição, que descreve as condições sob as quais a medição foi realizada (por exemplo, ao medir o número de requisitos alterados durante um determinado período em um projeto, poderia ser registrado o seguinte contexto: “medição realizada após alteração na legislação que rege o domínio tratado pelo sistema, o que contribuiu para o elevado número de alterações registradas”).

Pré-atividade: Validar dados.

Entradas: Dados validados.

Saídas: Dados armazenados.

Atividade 4: Analisar Medições

Nesta atividade os dados coletados para as medidas são analisados e os resultados das análises são registrados.

Subatividade 4.1: Selecionar Dados para Análise

Descrição: Nesta subatividade é realizada a seleção dos dados que devem ser analisados. A seleção dos dados pode considerar critérios como o período de coleta dos dados e os tipos de projetos nos quais os dados foram coletados, dentre outros.

Pré-atividade: Armazenar dados.

Entradas: Dados armazenados, critérios para seleção dos dados para análise.

Saídas: Dados selecionados.

Subatividade 4.2: Representar Dados

Descrição: Esta subatividade consiste em representar graficamente os dados selecionados, a fim de facilitar sua análise.

Pré-atividade: Selecionar Dados para Análise.

Entradas: Dados selecionados, Plano de Medição (definição operacional das medidas).

Saídas: Representação gráfica dos dados.

Subatividade 4.3: Analisar Dados

Descrição: Nesta subatividade os dados são analisados e os resultados da análise são registrados para que sejam disponibilizados às partes interessadas.

Pré-atividade: Representar Dados.

Entradas: Representação gráfica dos dados, Plano de Medição (definição operacional das medidas).

Saídas: Resultados da análise registrados.

Atividade 5: Monitorar Objetivos de Medição

Esta atividade consiste na monitoração dos objetivos de medição, utilizando-se os valores de referência estabelecidos para os indicadores e os dados coletados.

Subatividade 5.1: Verificar Alcance dos Objetivos de Medição

Descrição: Nesta subatividade é realizada a verificação da aderência dos valores reais dos indicadores em relação aos valores de referência para eles estabelecidos.

Pré-atividade: Analisar Dados.

Entradas: Valores reais dos indicadores, valores de referência dos indicadores.

Saídas: Registros de monitoramento.

Subatividade 5.2: Identificar Ações Corretivas

Descrição: Caso os valores reais dos indicadores encontrem-se nas faixas de valores que indicam ameaça ao alcance dos objetivos, devem ser sugeridas ações corretivas para tratar os desvios identificados.

Pré-atividade: Verificar Alcance dos Objetivos de Medição

Entradas: Registros de monitoramento.

Saídas: Ações corretivas identificadas.

A seguir, na Tabela 3.1, é apresentado um exemplo hipotético contendo possíveis resultados da execução do processo proposto:

Atividade	Subatividade	Resultado
<i>Planejar Medição</i>	Registrar Objetivos Estratégicos relevantes à Medição	Reduzir em 10% os custos operacionais.
	Definir Objetivos de Software	Reduzir em 15% os custos com retrabalho nos projetos.
	Definir Objetivos de Medição	Diminuir o número de alterações de requisitos dos projetos.
	Identificar Necessidades de Informação	Qual é a estabilidade dos requisitos?
	Identificar Processos Relacionados	Processo de Gerência de Requisitos
	Definir /Selecionar Medidas	Número de Requisitos Homologados (NRH) Número de Requisitos Alterados (NRA) Taxa de Alteração dos Requisitos (TAR) * a definição operacional da medida Taxa de Alteração dos Requisitos (TAR) encontra-se após esta tabela.
	Registrar Plano de Medição	Plano de Medição registrado.
<i>Estabelecer Metas para Indicadores</i>	Estabelecer Metas para Indicadores	Indicador: Taxa de Alteração dos Requisitos (TAR) Bom = [0, 0.2] Regular=]0, 0.4] Ruim = > 0.4
<i>Executar Medições</i>	Coletar Dados	Dados coletados no projeto X: TAR = 0.2, 0.2, 0.3 Dados coletados no projeto Y: TAR = 0.2, 0.3, 0.4

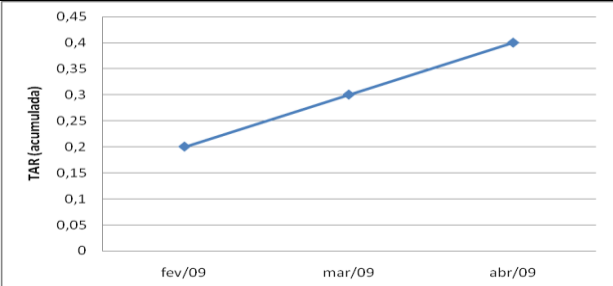
Atividade	Subatividade	Resultado								
<i>Executar Medições</i>	Validar Dados	Dados válidos.								
	Armazenar Dados	Dados armazenados na base de dados.								
<i>Analisar Medições</i>	Selecionar Dados para Análise	Dados coletados no projeto Y: 0.2, 0.3, 0.4								
	Representar Dados	 <table border="1"> <caption>Dados do Gráfico de TAR (acumulada)</caption> <thead> <tr> <th>Mês</th> <th>TAR (acumulada)</th> </tr> </thead> <tbody> <tr> <td>fev/09</td> <td>0,2</td> </tr> <tr> <td>mar/09</td> <td>0,3</td> </tr> <tr> <td>abr/09</td> <td>0,4</td> </tr> </tbody> </table>	Mês	TAR (acumulada)	fev/09	0,2	mar/09	0,3	abr/09	0,4
	Mês	TAR (acumulada)								
fev/09	0,2									
mar/09	0,3									
abr/09	0,4									
Analisar Dados	TAR ascendente, mostrando que à medida que se avança nas etapas de desenvolvimento do projeto aumenta o número de modificações nos requisitos.									
<i>Monitorar Objetivos de Medição</i>	Verificar Alcance dos Objetivos de Medição	TAR atual do projeto Y (0.4) encontra-se na faixa Regular.								
	Identificar Ações Corretivas	Realizar uma revisão detalhada dos requisitos junto ao cliente e redefinir a <i>baseline</i> dos requisitos do projeto.								

Tabela 3.1 – Exemplo de resultados da execução do processo definido.

3.3 MeMOS: Propósito do Sistema e Descrição do Minimundo

A ferramenta MeMOS tem como propósito auxiliar na realização do processo de medição de software de forma alinhada aos objetivos organizacionais e de projetos, apoiando, também, a análise do alcance desses objetivos.

O minimundo considerado por MeMOS é descrito a seguir. Vale ressaltar que a descrição aqui apresentada é uma versão resumida da descrição completa do minimundo. Essa descrição completa do minimundo, bem como toda a documentação de MeMOS pode ser encontrada no site do Projeto ODE (<https://sites.google.com/site/projetoode>).

Organizações de software realizam medições para que seja possível avaliar e melhorar, dentre outros, seus produtos, processos e projetos.

A medição tem início em seu planejamento, onde as medidas que deverão ser coletadas e analisadas são definidas. É de suma importância que a medição seja realizada de maneira alinhada aos objetivos de negócio da organização. Assim, as medidas definidas devem ser capazes de fornecer informações que apoiem a análise do alcance dos objetivos da organização.

O planejamento da medição começa com a identificação dos objetivos estratégicos da organização, registrados em seu Planejamento Estratégico, que são relevantes para a medição de

software. Com base nos objetivos estratégicos, podem ser definidos objetivos de software, que são objetivos relacionados à área de software da organização, os quais contribuem para o alcance dos objetivos estratégicos aos quais estão relacionados. Vale destacar que é possível que os objetivos de software tenham sido definidos no Planejamento Estratégico ou no Planejamento Tático da organização. Nesse caso, os objetivos de software relacionados aos objetivos estratégicos relevantes à medição devem ser identificados.

Os objetivos de software são utilizados como base para a definição de objetivos de medição. Um objetivo de medição pode ser de três tipos: *Objetivo de Monitoramento e Controle de Projetos*, *Objetivo de Medição de Qualidade* ou *Objetivo de Medição de Desempenho*.

A partir dos objetivos de medição, são identificadas suas necessidades de informação. Após a identificação das necessidades de informação, os processos que estão relacionados aos objetivos de medição definidos devem ser identificados. Um processo está relacionado a um objetivo de medição se ele é fonte de dados para as necessidades de informação identificadas para o objetivo de medição. O próximo passo do planejamento da medição consiste na definição das medidas para atender às necessidades de informação identificadas. Para cada medida deve ser estabelecida uma definição operacional. Caso existam medidas previamente definidas, elas podem ser selecionadas.

O resultado do planejamento da medição é um documento chamado Plano de Medição, onde são registrados os objetivos, as necessidades de informação, os processos e as medidas definidos, bem como a relação entre esses elementos.

Para que as medidas definidas no Plano de Medição possam ser utilizadas no monitoramento do alcance dos objetivos, devem ser estabelecidos os valores que serão utilizados como referência na análise do alcance dos objetivos. Quando uma medida é utilizada para indicar o alcance de um determinado objetivo, ela assume o papel de indicador. Para cada indicador devem ser estabelecidos valores ou intervalos de valores que indiquem o alcance ou ameaça ao alcance dos objetivos. Após o planejamento da medição ser concluído e os valores dos indicadores serem estabelecidos, a medição passa a ser executada. Para isso, dados são coletados para as medidas de acordo com procedimento de medição estabelecido em suas definições operacionais. Uma vez coletados, os dados devem ser avaliados para que não sejam armazenados incorretamente. A validação dos dados pode ser feita comparando-se o dado coletado com as informações contidas na definição operacional da medida. Uma vez validados, os dados são armazenados em um repositório organizacional de medidas. Ao armazenar os dados de uma medição, devem ser incluídos: o valor medido, a data da medição, o

executor da medição, o momento real da medição (atividade na qual a medição foi realizada), o projeto onde a medição foi realizada e o contexto da medição, que descreve as condições sob as quais a medição foi realizada (por exemplo, ao medir o número de requisitos alterados durante um determinado período em um projeto, poderia ser registrado o seguinte contexto: “medição realizada após alteração na legislação que rege o domínio tratado pelo sistema, o que contribuiu para o elevado número de alterações registradas”).

Uma vez que os dados são armazenados, é possível realizar sua análise. Para isso é necessário selecionar os dados a serem analisados, podendo ser utilizados critérios, como: período de coleta dos dados e os tipos de projetos nos quais os dados foram coletados, dentre outros. Os dados selecionados devem ser representados por meio de gráficos sugeridos no procedimento de análise de medição da medida. Esse procedimento também será utilizado para orientar a análise dos dados. Os resultados da análise devem ser registrados no repositório de medidas e disponibilizados às partes interessadas.

Os dados coletados para as medidas podem ser analisados sob diversas perspectivas. Por exemplo, dados podem ser utilizados para comparar projetos (os dados coletados em cada projeto para uma certa medida são comparados entre si). Também podem ser utilizados para monitorar um projeto. Por exemplo, dados coletados para o esforço realizado em um projeto podem ser comparados com os valores de esforço estimados para aquele projeto.

Para o monitoramento dos objetivos de medição, os valores coletados para as medidas são comparados com os valores de referência estabelecidos para os indicadores e os dados coletados. Assim, o alcance dos objetivos de medição é analisado realizando a verificação da aderência dos valores reais dos indicadores em relação aos valores de referência para eles estabelecidos. Caso os valores reais dos indicadores encontrem-se nas faixas de valores que indicam ameaça ao alcance dos objetivos, devem ser sugeridas ações corretivas para tratar os desvios identificados.

Com base na descrição do minimundo foram identificados os requisitos funcionais, não funcionais e regras de negócio de MeMOS. Tabelas contendo essas informações encontram-se no Documento de Requisitos de MeMOS, disponível site do Projeto ODE (<https://sites.google.com/site/projetoode>).

3.4 Subsistemas da Ferramenta MeMOS

Por tratar de um domínio relativamente complexo, MeMOS foi dividida em subsistemas, como mostra a Figura 3.1, que apresenta o diagrama de pacotes da ferramenta MeMOS.

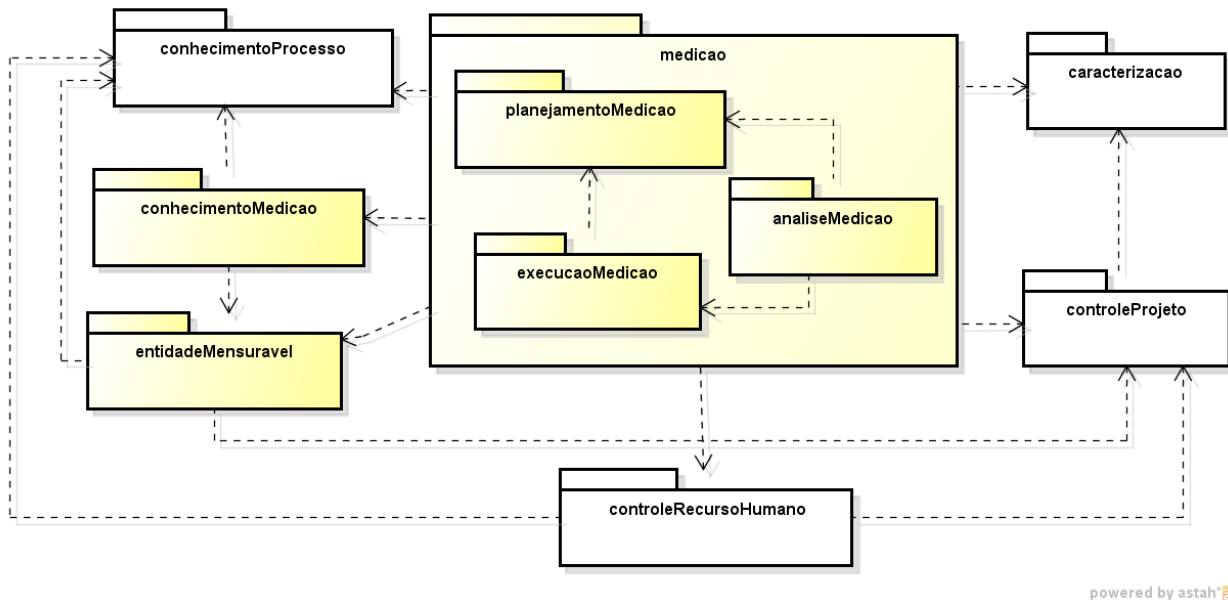


Figura 3.1 - Diagrama de Pacotes da Ferramenta MeMOS.

Os pacotes na cor branca representam pacotes oriundos do ODE e que são utilizados pela ferramenta MeMOS. São eles:

- *conhecimentoProcesso*: contém as classes que representam conhecimento acerca de processos de software. Baseia-se em conceitos da Ontologia de Processos de Software, que é a principal ontologia de ODE.
- *controleProjeto*: contém as classes e funcionalidades necessárias ao registro de projetos.
- *caracterizacao*: envolve a infraestrutura genérica para caracterização de itens de software e uso dessa caracterização para recuperar itens similares.
- *controleRecursoHumano*: contém as classes que armazenam os recursos humanos de uma organização.

Os pacotes da cor amarela foram desenvolvidos durante este trabalho exclusivamente para a ferramenta MeMOS. São eles:

- *conhecimentoMedicao*: contém as classes que representam conhecimento acerca de medição de software. Baseia-se em conceitos da Ontologia de Medição de Software (BARCELLOS, 2009).
- *entidadeMensuravel*: contém as classes que representam as entidades mensuráveis, segundo a Ontologia de Medição de Software (BARCELLOS, 2009) e a estrutura de ODE.
- *medicao* : trata da realização da medição propriamente dita, ou seja, da definição do plano de medição, coleta de dados para as medidas definidas e análise dos dados coletados. Para melhor organização, este pacote é subdividido em : *planejamentoMedicao* (trata da elaboração do plano de medição e estabelecimento de valores de referência para análise), *execucaoMedicao* (trata da coleta e armazenamento dos dados) e *analiseMedicao* (trata da análise dos dados coletados).

3.5 Casos de Uso da Ferramenta MeMOS

Em um levantamento de requisitos, é interessante capturar uma porção discreta da funcionalidade e representá-la pela descrição e diagramação de casos de uso. Nesse contexto, os atores são aqueles que interagem com o sistema e executam as funcionalidades descritas pelo diagrama. Os atores identificados no contexto deste trabalho são: o *Engenheiro de Software* que é responsável por realizar os cadastros básicos necessários para a realização da medição, o *Gerente de Qualidade* que fica responsável pela medição como uma forma de qualificar os processos da organização de uma forma geral, o *Gerente de Projeto* que ficará responsável pela medição na parte que tange projetos executados pela organização. Um outro ator denominado *Membro da Equipe do Projeto* será utilizado para representar o membro da equipe que estiver responsável pela coleta de um determinado dado. A seguir serão apresentados os casos de uso da ferramenta MeMOS, organizados por subsistema. Para cada caso de uso são apresentadas breves descrições. A descrição completa dos casos de uso encontra-se no Documento de Especificação de Requisitos de MeMOS, disponível site do Projeto ODE (<https://sites.google.com/site/projetoode>).

3.5.1 Subsistema *conhecimentoMedicao*

A Figura 3.2 apresenta o diagrama de caso de uso do subsistema *conhecimentoMedicao*. Após a figura são apresentadas as descrições.



Figura 3.2 - Diagrama de Casos de Uso do subsistema conhecimentoMedicao.

- **Cadastrar Elemento Mensurável:** Este caso de uso permite ao engenheiro de software incluir, excluir, atualizar e consultar elementos mensuráveis, ou seja, características que podem ser medidas em um tipo de entidade mensurável.
- **Cadastrar Unidade de Medida:** Este caso de uso permite ao engenheiro de software incluir, excluir, atualizar e consultar unidades de medida.
- **Cadastrar Escala:** Este caso de uso permite ao engenheiro de software incluir, excluir, atualizar e consultar escalas.
- **Cadastrar Valor Escala:** Este caso de uso permite ao engenheiro de software incluir, excluir, atualizar e consultar valores que fazem parte de uma escala.
- **Cadastrar Medida:** Este caso de uso permite ao engenheiro de software incluir, excluir, atualizar e consultar medidas.
- **Cadastrar Periodicidade:** Este caso de uso permite ao engenheiro de software incluir, excluir, atualizar e consultar frequências em que medições e análise de medições devem ser realizadas.

- **Cadastrar Procedimento de Medição:** Este caso de uso permite ao engenheiro de software incluir, excluir, atualizar e consultar os procedimentos de coleta de dados para medidas.
- **Cadastrar Procedimento de Análise de Medição:** Este caso de uso permite ao engenheiro de software incluir, excluir, atualizar e consultar os procedimentos de análise de medição que podem ser usados para representar e analisar as medidas.
- **Cadastrar Método Analítico:** Este caso de uso permite ao engenheiro de software incluir, excluir, atualizar e consultar os métodos analíticos que podem ser utilizados para representar e analisar valores medidos.

3.5.2 Subsistema *entidadeMensuravel*

A figura 3.3 apresenta o diagrama de caso de uso do subsistema *entidadeMensuravel*.



Figura 3.3 - Diagrama de Casos de Uso do subsistema *entidadeMensuravel*.

O caso de uso **Identificar Entidade Mensurável** permite ao engenheiro de software criar, consultar, editar e excluir entidades mensuráveis.

3.5.3 Subsistema *medicao*

A seguir são apresentados os casos de uso do pacote *medicao*, organizados de acordo com a subdivisão desse pacote em outros três: *planejamentoMedicao*, *execucaoMedicao* e *analiseMedicao*.

3.5.3.1 Subsistema *planejamentoMedicao*

A Figura 3.4 apresenta o diagrama de caso de uso do subsistema *planejamentoMedicao*. Em seguida os casos de uso são descritos.

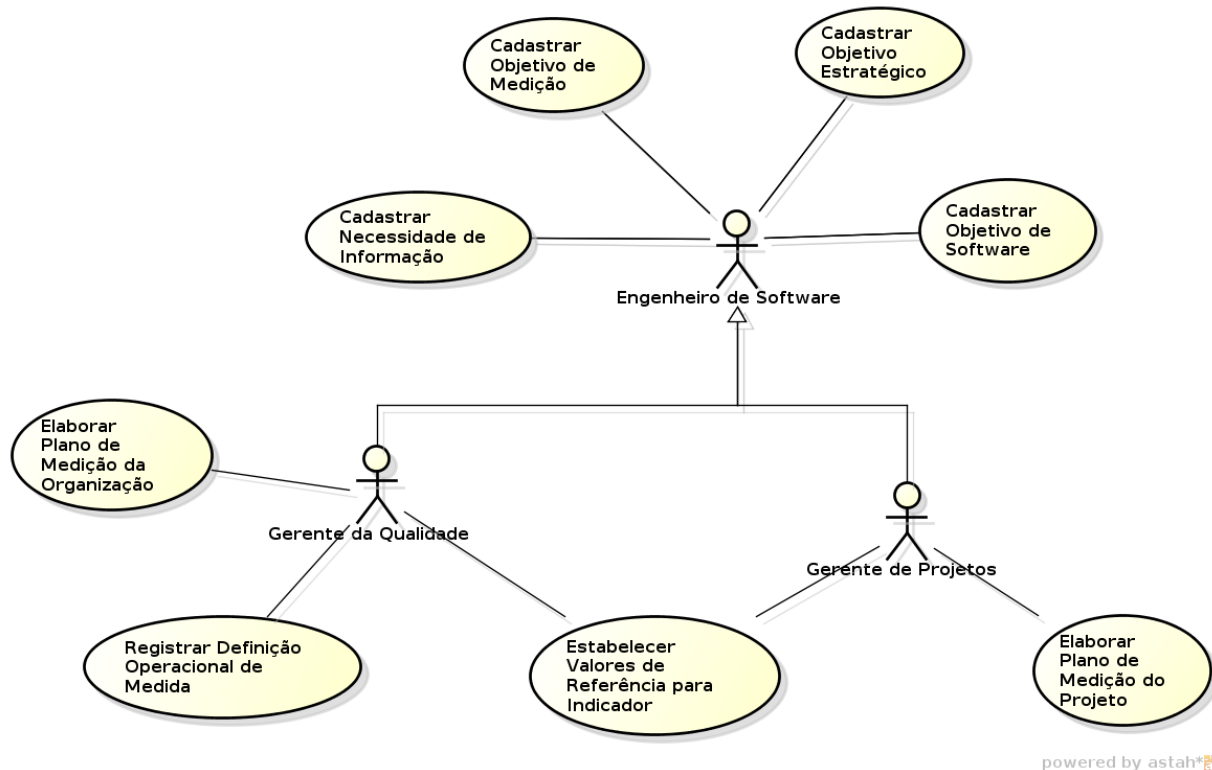


Figura 3.4 - Diagrama de Casos de Uso do subsistema *planejamentoMedicao*.

- **Cadastrar Objetivo Estratégico:** Este caso de uso permite ao engenheiro de software incluir, excluir, atualizar e consultar os objetivos estratégicos da organização.
- **Cadastrar Objetivo de Software:** Este caso de uso permite ao engenheiro de software incluir, excluir, atualizar e consultar objetivos da área de software da organização a partir dos objetivos estratégicos.
- **Cadastrar Objetivo de Medição:** Este caso de uso permite ao engenheiro de software incluir, excluir, atualizar e consultar objetivos de medição, baseados em objetivos de software e/ou objetivos estratégicos.
- **Cadastrar Necessidade de Informação:** Este caso de uso permite ao engenheiro de software incluir, excluir, atualizar e consultar necessidades de informação a partir dos objetivos de medição.
- **Elaborar Plano de Medição da Organização:** Este caso de uso permite ao gerente de qualidade elaborar planos de medição para uma organização.

- **Elaborar Plano de Medição do Projeto:** Este caso de uso permite o gerente de projetos elaborar planos para medição para projetos, a partir de planos de medição da organização.
- **Registrar Definição Operacional de Medida:** Este caso de uso permite aos gerente de qualidade incluir, excluir, atualizar e consultar definições operacionais de medidas.
- **Estabelecer Valores de Referência para Indicadores:** Este caso de uso permite o gerente de qualidade e de projeto estabelecer valores de referência que serão utilizados para analisar o alcance dos objetivos.

3.5.3.2 Subsistema *execucaoMedicao*

A Figura 3.5 apresenta o diagrama de caso de uso do subsistema *execucaoMedicao*.

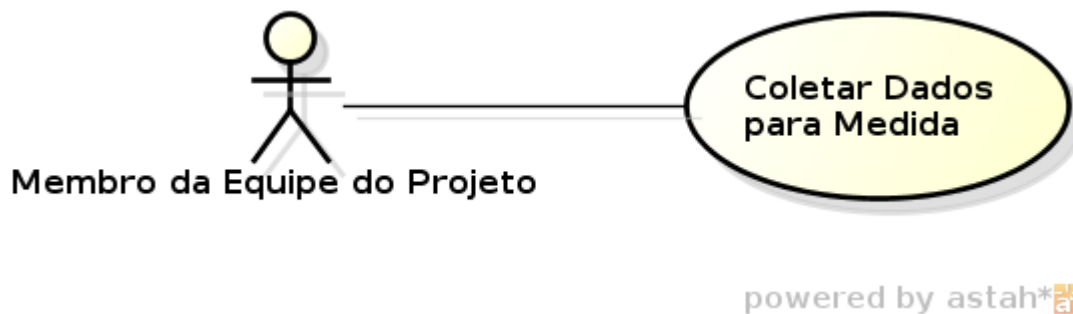
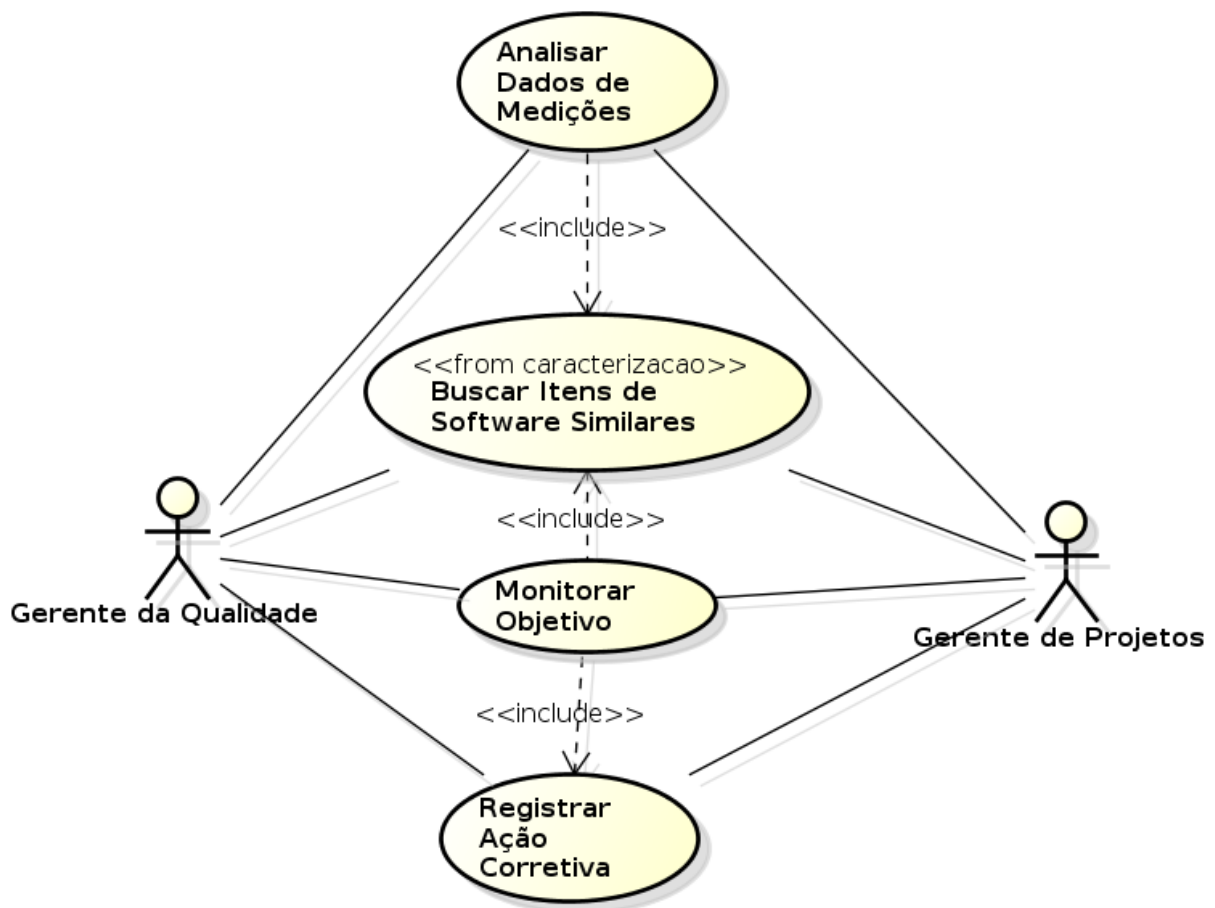


Figura 3.5 - Diagrama de Classe do subsistema *execucaoMedicao*.

O caso de uso **Coletar Dados para Medida** permite o Membro da Equipe registrar a coleta dos dados coletados para as medidas.

3.5.3.3 Subsistema *analiseMedicao*

A Figura 3.6 apresenta o diagrama de caso de uso do subsistema *analiseMedicao*. Em seguida os casos de uso são descritos.



powered by astah*

Figura 3.6 - Diagrama de caso de uso do subsistema *analiseMedicao*.

- **Analisar Dados de Medições:** Este caso de uso permite tanto ao gerente de qualidade quanto o de projetos selecionar dados para análise, representá-los e registrar a análise dos dados.
- **Monitorar Objetivo:** Este caso de uso permite tanto ao gerente de qualidade quanto o de projetos monitorar o alcance dos objetivos, utilizando os valores de referência como base.
- **Registrar Ação Corretiva:** Este caso de uso permite tanto ao gerente de qualidade quanto o de projetos registrar as ações corretivas que devam ser realizadas para corrigir desvios encontrados durante a análise dos dados medidos e do monitoramento dos objetivos.

3.6 Diagramas de Classes

O modelo conceitual estrutural visa capturar e descrever as informações (classes, associações e atributos) que o sistema deve representar para prover as funcionalidades descritas na seção anterior. A seguir, são apresentados os diagramas de classes de cada um dos subsistemas identificados no contexto deste projeto. Após os diagramas são apresentadas breves descrições das classes. A documentação completa das classes, incluindo a identificação das restrições de integridade, está registrada no Documento Especificação de Requisitos de MeMOS, disponível site do Projeto ODE (<https://sites.google.com/site/projetoode>).

3.6.1 Subsistema *conhecimentoMedicao*

Ao longo do subsistema *conhecimentoMedicao* é representado o conhecimento a respeito de medição. Como o conhecimento de uma organização pode ser particular da mesma, esse subsistema apresenta casos de uso, já apresentados, para cadastramento de entidades além daquelas comum às organizações. A Figura 3.7 representa o diagrama de classes do subsistema *conhecimentoMedicao*.

- **TipoEntidadeMensuravel:** classe que armazena os tipos de entidades que podem ser medidas.
- **KElementoMensuravel:** classe que armazena as características que podem ser medidas em um tipo de entidade mensurável.
- **KMedida:** classe que armazena as medidas definidas em uma organização, com o objetivo de atender as necessidades de informação.
- **KMedidaBase:** classe que armazena medidas funcionalmente independentes das outras.
- **KMedidaDerivada:** classe que armazena medidas funcionalmente dependentes de outras medidas.
- **KEscala:** classe que armazena as escalas das medidas.
- **KValorEscala:** classe que armazena os valores que fazem parte de cada escala.
- **KPeriodicidade:** Classe que armazena a frequência em que medições e análise de medições podem ser realizadas.
- **KProcedimentoMedicao:** classe que armazena os procedimentos de coleta de dados para medidas .
- **KProcedimentoAnaliseMedicao:** classe que armazena os procedimentos de análise de medição que podem ser usados para analisar as medidas.
- **KMetodosAnalitico:** classe que armazena métodos analíticos que podem ser utilizados para representar e analisar valores.
- **KUnidadeMedida:** classe que armazena unidades de medidas nas quais medidas podem ser expressas.

3.6.2 Subsistema *entidadeMensuravel*

No pacote *entidadeMensuravel*, é representado as entidades mensuráveis e seus tipos. Assim, cada entidade mensurável (i.e. *ProjetoMensuravel*, *AtividadeMensuravel*), é instância de uma entidade a qual ela representa (*Projeto* e *KAtividade*, respectivamente). A Figura 3.8 ilustra o diagrama de classes do subsistema *entidadeMensuravel*.

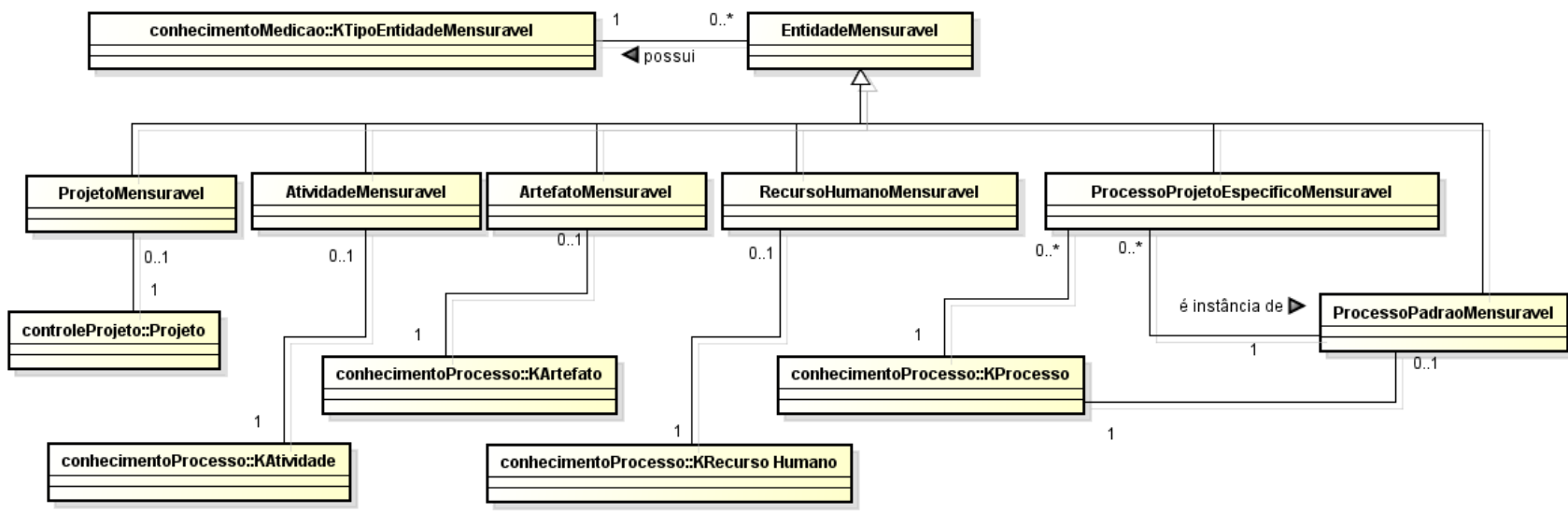


Figura 3.8 - Diagrama de Classes do subsistema *entidadeMensuravel*

- **EntidadeMensuravel:** classe que armazena as entidade que podem ser medidas.
- **ProjetoMensuravel:** classe que armazena os projetos registrados em ODE como entidades mensuráveis.
- **AtividadeMensuravel:** classe que armazena as atividades registradas em ODE como entidades mensuráveis.
- **ArtefatoMensuravel:** classe que armazena registrados em ODE como entidades mensuráveis.
- **RecursoHumanoMensuravel:** classe que armazena os recursos humanos registrados em ODE como entidades mensuráveis.
- **ProcessoProjetoEspecificoMensuravel:** classe que armazena os processos específicos de projetos registrados em ODE como entidades mensuráveis.
- **ProcessoPadraoMensuravel:** classe que armazena os processos padrão registrados em ODE como entidades mensuráveis.

3.6.3 Subsistema *medicao*

Como foi apresentado anteriormente, o pacote *medicao* é dividido em três subpacotes, *planejamentoMedicao*, *execucaoMedicao* e *analiseMedicao*, cujos diagramas de classe são apresentados a seguir.

3.6.3.1 Subsistema *planejamentoMedicao*

Durante o planejamento de uma medição produz-se um plano de medição, onde se encontram informações de como será realizada a execução e a análise da medição. Dessa forma o elemento principal do subsistema *planejamentoMedicao* é a classe *PlanoMedicao*, seguido por *DefinicaoOperacionalMedida* que representa a forma como a execução e análise de uma medida serão executadas. A Figura 3.9 representa o diagrama de classes do subsistema *planejamentoMedicao*.

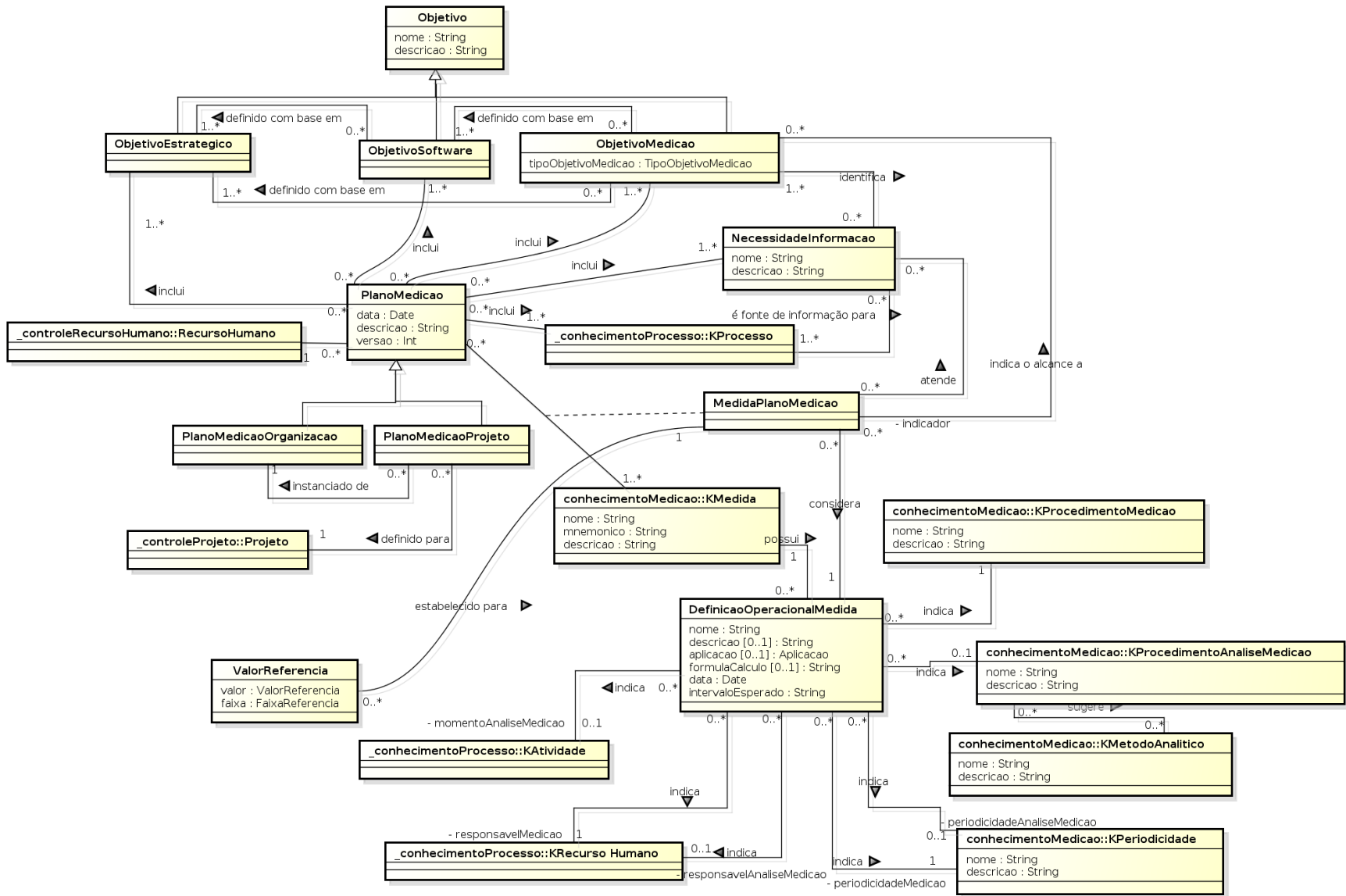


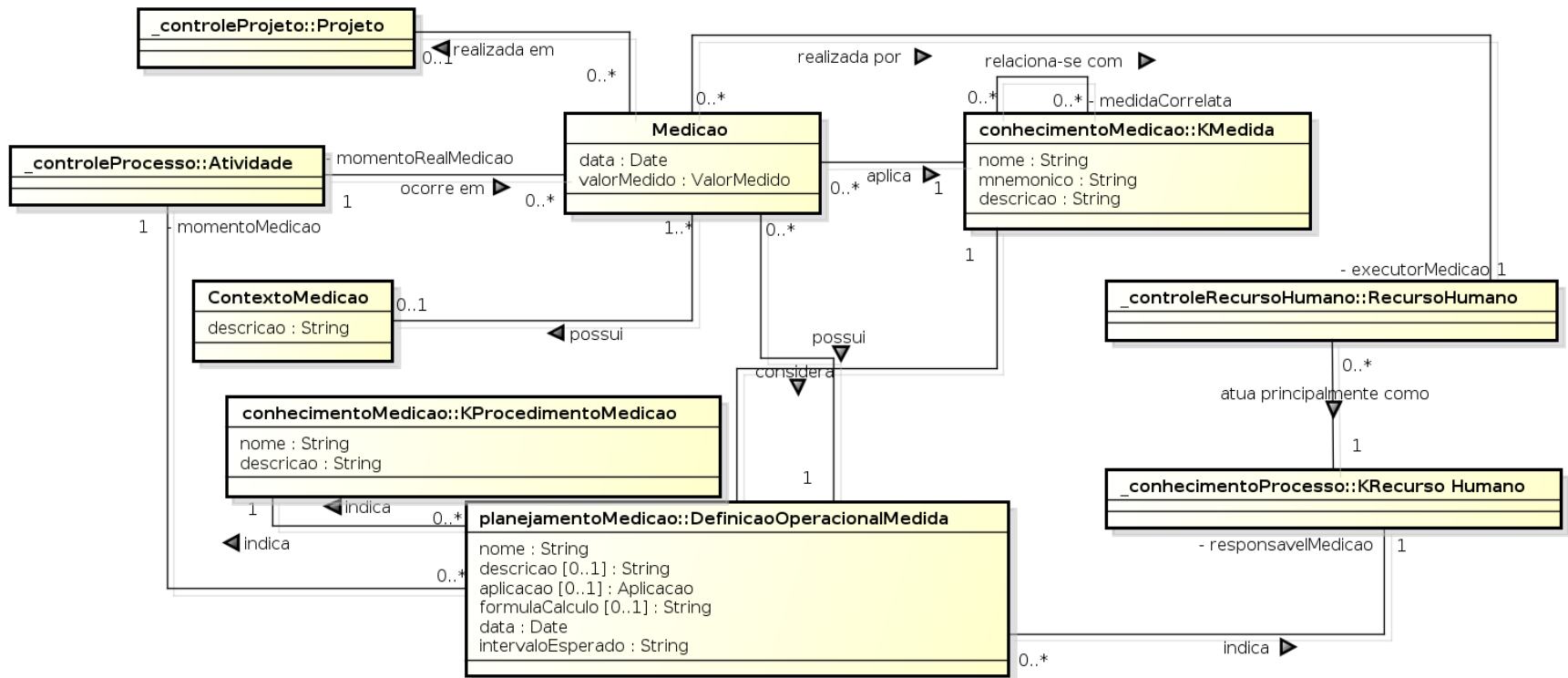
Figura 3.9 - Diagrama de Classe do subsistema *planejamentoMedicao*.

- **Objetivo:** classe que armazena os objetivos de uma organização.
- **ObjetivoEstrategico:** classe que armazena os objetivos estratégicos da organização.
- **ObjetivoSoftware:** classe que armazena objetivos da área de software da organização, identificados a partir dos objetivos estratégicos.
- **ObjetivoMedicao:** classe que armazena objetivos de medição, definidos com base em objetivos de software e/ou objetivos estratégicos.
- **NecessidadeInformacao:** classe que armazena as necessidades de informação identificadas a partir de objetivos de medição.
- **PlanoMedicao:** classe que armazena os planos de medição da organização e dos projetos.
- **PlanoMedicaoOrganizacao:** classe que armazena os planos de medição relativos à organização.
- **PlanoMedicaoProjeto:** classe que armazena os planos de medição dos projetos.
- **DefinicaoOperacionalMedida:** classe que armazena definições operacionais das medidas.
- **MedidaPlanoMedicao:** classe que relaciona um plano de medição com as medidas e definições operacionais utilizadas.
- **ValorReferencia:** classe que armazena valores de referência que são utilizados para analisar o alcance dos objetivos.

3.6.3.2 Subsistema *execucaoMedicao*

Como é indicado na seção 3.2, após o desenvolvimento de um plano de medição, é iniciada a coleta de dados.. Para armazenar o contexto, os responsáveis, o momento e a definição operacional desta fase, bem como os dados obtidos, é utilizado a classe *Medicao* e *ContextoMedicao*. A Figura 3.10 representa o diagrama de classes do subsistema *execucaoMedicao*. As classes pertencentes a esse subsistema são:

- **Medicao:** classe que representa as medições executadas em uma organização.
- **ContextoMedicao:** classe que armazena o contexto em que uma medição foi executada.

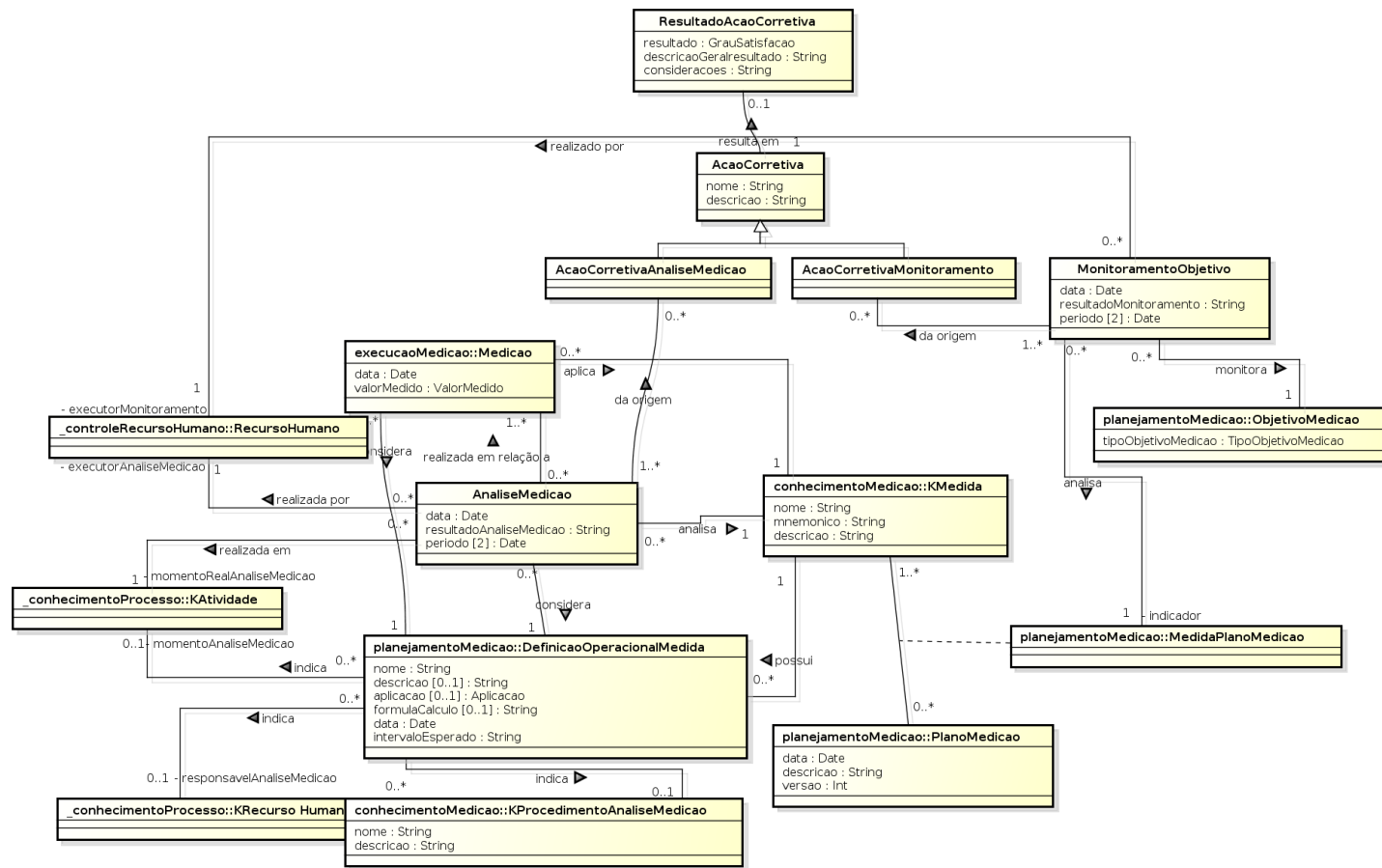


powered by astah*

Figura 3.10 - Diagrama de Classe do subsistema *execucaoMedicao*.

3.6.3.3 Subsistema *analiseMedicao*

No Subsistema *analiseMedicao* é possível verificar que pode-se armazenar análises de medição e monitoramentos de objetivo. A partir desses é possível gerar ações corretivas e acompanhá-las, registrando seus resultados. A Figura 3.11 representa o diagrama de classes do subsistema *analiseMedicao*.



powered by astah

Figura 3.11 - Diagrama de Classes do subsistema *analiseMedicao*.

- **AnaliseMedicao:** classe que representa as análises de medição executadas em uma organização.
- **MonitoramentoObjetivo:** classe que armazena dados relativos ao monitoramento do alcance de um objetivo.
- **AcaoCorretiva:** classe que armazena informações sobre ações executadas com o intuito de corrigir desvios de objetivos em uma análise ou monitoramento.
- **AcaoCorretivaAnaliseMedicao:** classe que armazena informações de ações corretivas para corrigir desvios encontrados durante a análise de medição.
- **AcaoCorretivaMonitoramento:** classe que armazena informações de ações corretivas para corrigir desvios encontrados durante o monitoramento.
- **ResultadoAcaoCorretiva:** classe que armazena informações referentes aos resultados das ações corretivas realizadas.

Capítulo 4

Projeto e Implementação da Ferramenta MeMOS

4.1 Introdução

Na fase de Projeto, o sistema é modelado de forma que os aspectos tecnológicos que serão utilizados para a implementação do sistema sejam levados em conta. Esses aspectos envolvem: linguagem de programação e *frameworks* utilizados, características de interface com o usuário, arquitetura de software e de hardware e forma de persistência de dados.

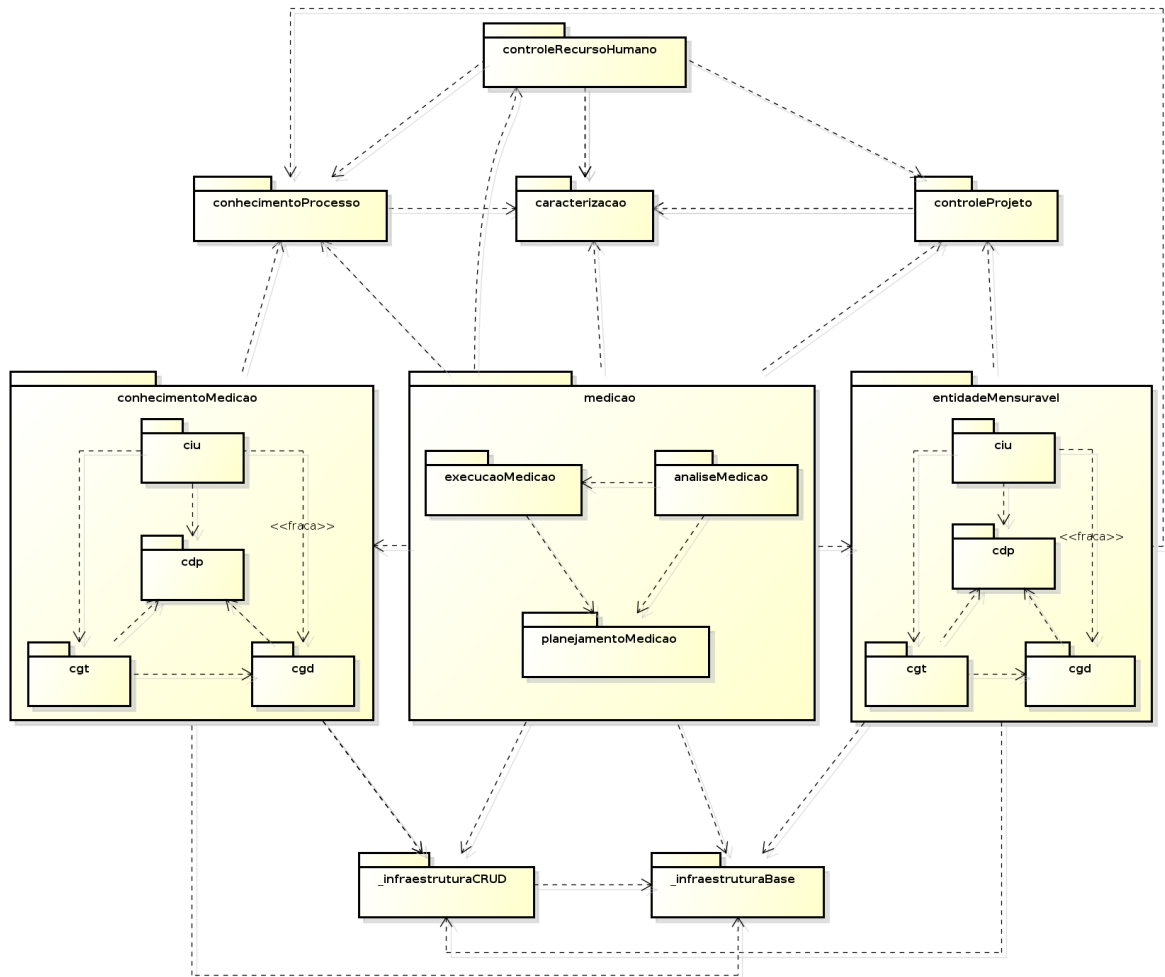
Neste capítulo são apresentados os principais aspectos do Projeto de Sistema da ferramenta MeMOS. Algumas de suas telas também são apresentadas. Na Seção 4.2 a arquitetura de software de MeMOS é descrita. Na Seção 4.3 os componentes da arquitetura são detalhados e na Seção 4.4 são apresentadas algumas telas da ferramenta. Vale ressaltar que na Seção 4.3 são detalhados todos os componentes de apenas um subsistema da ferramenta. O detalhamento de projeto dos demais subsistemas encontra-se no Documento de Projeto de Sistema de MeMOS, disponível no site do Projeto ODE (<https://sites.google.com/site/projetoode>). No Documento de Projeto de Sistema de MeMOS, além do detalhamento dos componentes da arquitetura de todos os subsistemas, também há informações sobre as tecnologias utilizadas e sobre as táticas de projeto realizadas para atender os requisitos não funcionais identificados no Documento de Requisitos de MeMOS (também disponível no site do Projeto ODE).

4.2 Arquitetura de Software

A arquitetura de software da ferramenta MeMOS baseia-se na combinação de camadas e partições. Inicialmente, para cada subsistema identificado na fase de análise foi definida uma partição. Cada uma dessas partições, por sua vez, está organizada em três camadas, a saber: camadas de Interface com o Usuário (ciu), que trata de aspectos relacionados às interfaces gráficas com os usuários; Lógica de Negócio (cln), onde é implementada a lógica de negócio; e Gerência de Dados (cgd), responsável pela persistência de objetos. A camada de Lógica de Negócio, por sua vez, é subdividida em dois componentes: Componente de Domínio do Problema (cdp) e Componente de Gerência de Tarefas (cgt).

Além das partições definidas a partir dos subsistemas identificados na fase de análise, foram reutilizados os componentes `_infraestruturaBase` e `_infraestruturaCRUD`, presentes na arquitetura do

ODE e próprios para o reúso. A Figura 4.1 mostra o projeto da arquitetura de software da ferramenta MeMOS. Uma vez que o pacote *medicao* é subdividido em três outros subsistemas, a fim de não poluir visualmente a figura, o detalhamento das camadas do pacote *medicao* é apresentado na Figura 4.2.



powered by astah®

Figura 4.1 – Arquitetura de Software de MeMOS.

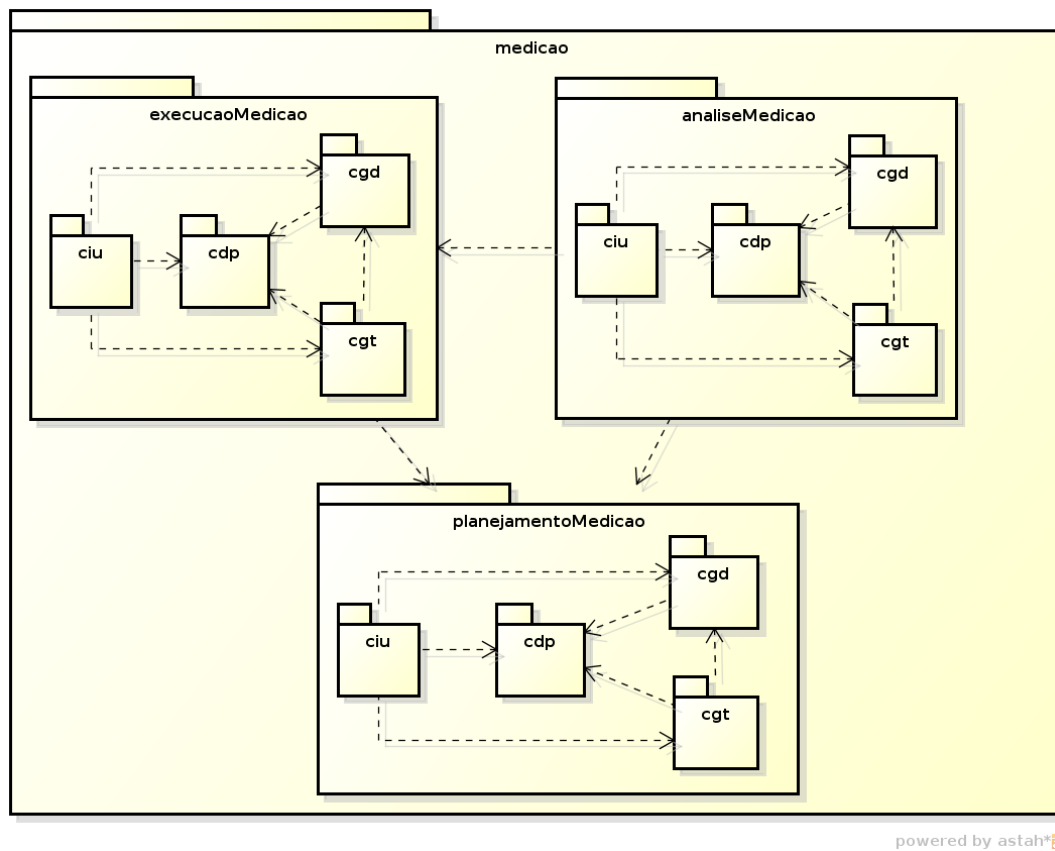


Figura 4.2 – Detalhes da Arquitetura de Software do subsistema *medicao*.

A dependência entre o Componente de Interface com o Usuário (ciu) e o Componente de Gerência de Dados (cgd) indica que a arquitetura utiliza uma implementação em camadas abertas. Essa dependência é fraca, pois indica apenas leitura de dados, em especial, para montagem de interfaces.

A seguir é apresentado o projeto detalhado dos componentes da arquitetura dos subsistemas de MeMOS. Como dito na introdução deste capítulo, apenas um subsistema tem todos os componentes descritos no texto desta monografia. Assim, na próxima seção são descritos os componentes da arquitetura do *subsistema planejamentoMedicao*. Para os demais subsistemas, são apresentados os modelos de classes de seus componentes de domínio do problema.

4.3 Subsistema *planejamentoMedicao*

4.3.1 Camada Lógica do Negócio

Para organizar a camada de lógica de negócio deste subsistema e dos demais subsistemas de MeMOS, foi escolhido o padrão Camada de Serviço. Sendo assim, essa camada é dividida em dois componentes: Componente de Domínio do Problema (cdp) e Componente de Gerência de Tarefas (cgt), como apresentado anteriormente nas figuras 4.1 e 4.2. Esse padrão utiliza um componente para tratar a lógica de aplicação (o cgt), o qual recebe as requisições da interface, e um componente para tratar os conceitos do domínio do problema, advindos do modelo conceitual estrutural elaborado na fase de análise (o cdp). A seguir, o projeto desses dois componentes é apresentado.

4.3.1.1 Componente Domínio do Problema

A Figura 4.3 apresenta o diagrama de classes do cdp do subsistema *planejamentoMedicao*. Pode-se observar que os tipos específicos de domínio identificados na etapa de análise passaram a ser representados como classes no modelo de classes de projeto, as classes associativas tornaram-se classes regulares e as navegabilidades entre as classes foram identificadas.

4.3.1.2 Componente de Gerência de Tarefas

No projeto do CGT do subsistema *planejamentoMedicao*, optou-se por estabelecer uma classe de aplicação para cada caso de uso proposto. A Tabela 4.1 sumariza as relações existentes entre as classes do CGT e os casos de uso por elas tratados.

Tabela 4.1 – Classes do CGT e Casos de Uso do subsistema *planejamentoMedicao*.

Classe	Casos de Uso
AplCadastrarObjetivoEstrategico	Cadastrar Objetivo Estratégico
AplCadastrarObjetivoSoftware	Cadastrar Objetivo de Software
AplCadastrarObjetivoMedicao	Cadastrar Objetivo de Medição
AplCadastrarNecessidadeInformacao	Cadastrar Necessidade de Informação
AplCadastrarDefinicaoOperacionalMedida	Registrar Definição Operacional de Medida
AplElaborarPlanoMedicaoOrganizacao	Elaborar Plano de Medição da Organização
AplElaborarPlanoMedicaoProjeto	Elaborar Plano de Medição do Projeto
AplEstabelecerValorReferencia	Estabelecer Valores de Referência para Indicadores

Uma vez que o projeto do CGT está fortemente relacionado ao projeto da Interface com o Usuário, um único diagrama foi elaborado, o qual é mostrado na Figura 4.4.

4.3.2 Camada de Interface com o Usuário

A Figura 4.4 apresenta o diagrama de classes (parcial) do CIU do subsistema *planejamentoMedicao*. O diagrama referente à elaboração de plano de medição.

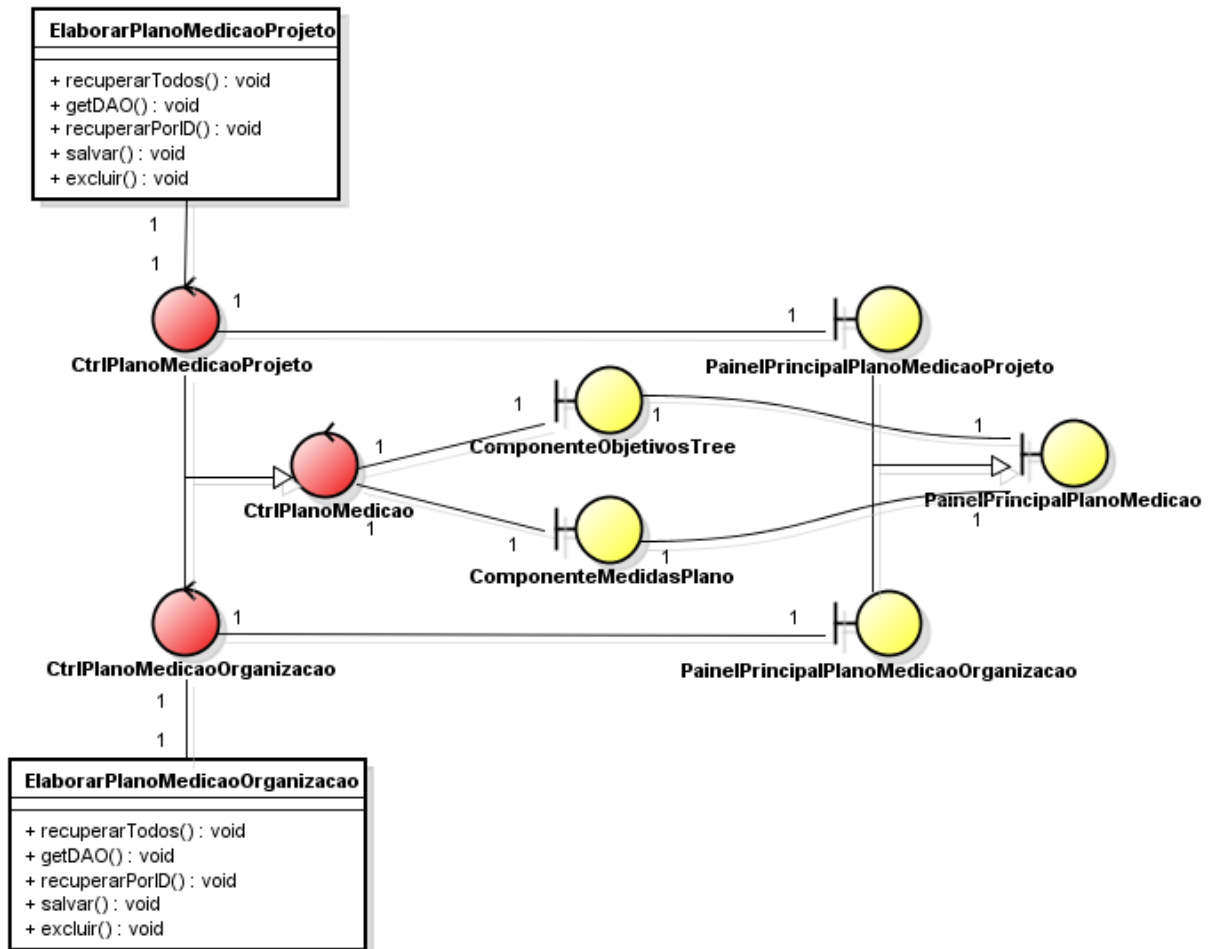


Figura 4.4 - Diagrama de Classes (parcial) do CIU do Subistema planejamentoMedicao.

Uma vez que as funcionalidades dos casos de uso Elaborar Plano de Medição da Organização e Elaborar Plano de Medição do Projeto são praticamente as mesmas, foi criada uma classe controladora geral para a elaboração de plano de medição e, a partir dela, os controladores específicos da elaboração do plano de medição da organização e do projeto foram especializadas.

4.3.3 Camada de Gerência de Dados

A persistência dos objetos em MeMOS é realizada em um banco de dados relacional, utilizando a infraestrutura de persistência desenvolvida no contexto do Projeto ODE. Essa infraestrutura utiliza JPA com o *framework* de persistência Hibernate, juntamente com o Sistema de Gerência de Banco de Dados PostgreSQL v8.4 e adota o Padrão DAO. A Figura 4.5 apresenta as principais classes do utilitário de Persistência.

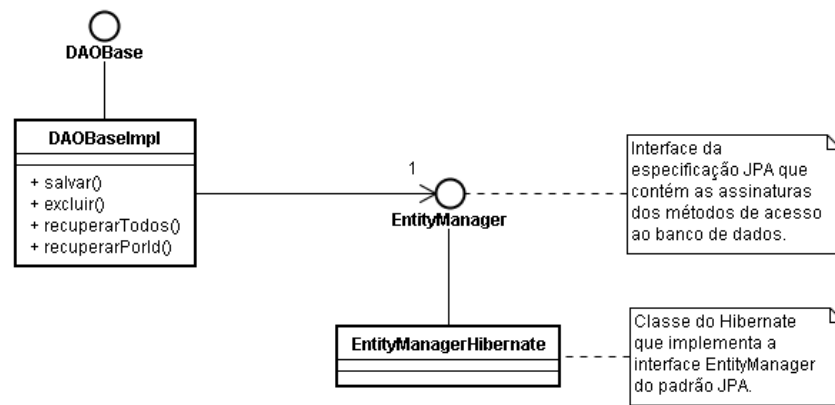
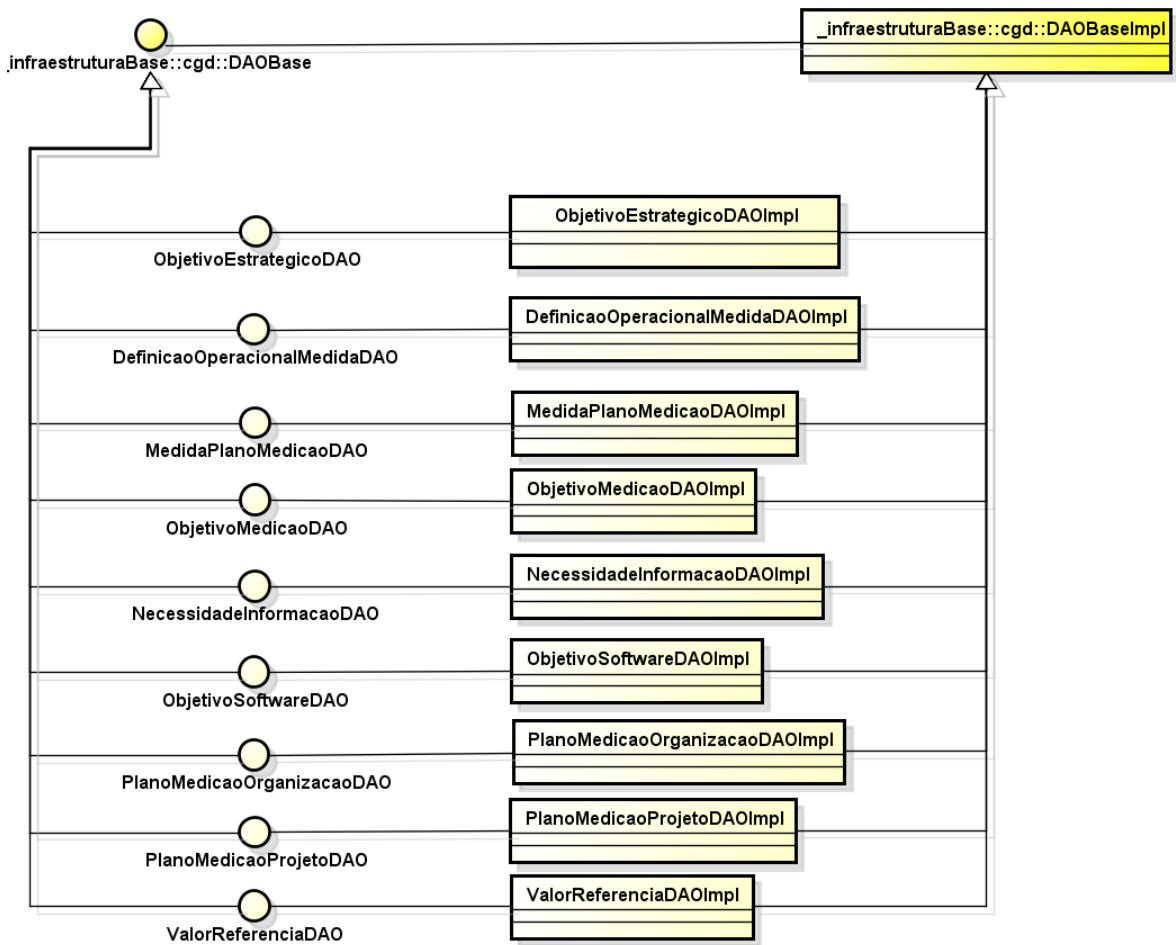


Figura 4.5 - Infraestrutura de persistência do Projeto ODE.

Dessa forma, para cada classe de domínio do problema a ser persistida, foram criadas uma classe (*)DAOImpl e uma interface (*)DAO correspondente. A primeira herda de DAOBaseImpl, uma classe genérica que possui as funcionalidades básicas de acesso ao mecanismo de persistência e deve implementar a interface DAO associada. Já a interface DAO da classe a ser persistida deve herdar da interface genérica DAOBase. Seguindo essa abordagem, cada classe a ser persistida tem uma correspondente classe de persistência, responsável pela interação com o banco de dados relacional, e implementa uma interface correspondente, como mostra a Figura 4.6.

Todas as classes de domínio do problema de todos os subsistemas de MeMOS seguem esse mesmo padrão para persistência.

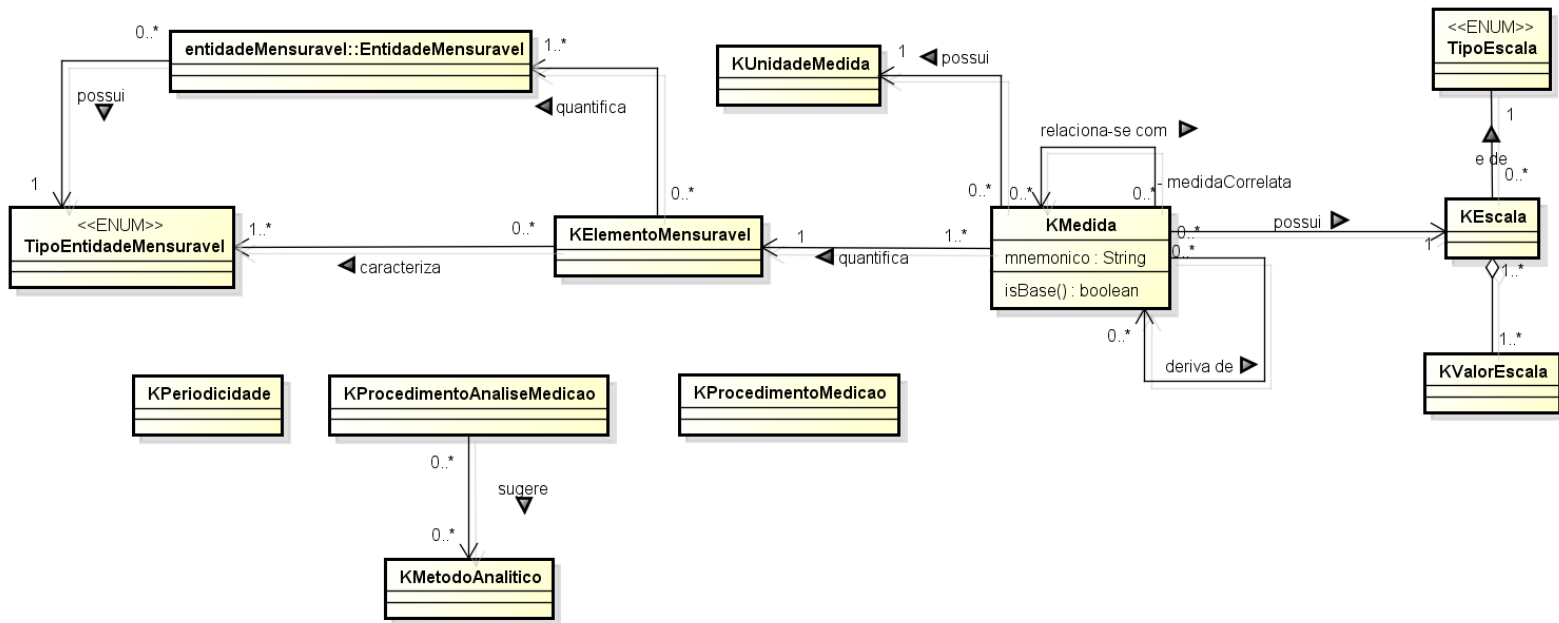


powered by astah™

Figura 4.6 – CGD do Subsistema *planejamentoMedicao*.

4.4 Demais Subsistemas de MeMOS

A seguir, nas figuras 4.7, 4.8, 4.9 e 4.10 são apresentados os modelos de classe do componente de domínio do problema dos demais subsistemas de MeMOS, respectivamente: *conhecimentoMedicao*, *entidadeMensuravel*, *execucaoMedicao* e *analiseMedicao*.



powered by astah

Figura 4.7 - Diagrama de Classes do cdp do subsistema *conhecimentoMedicao*.

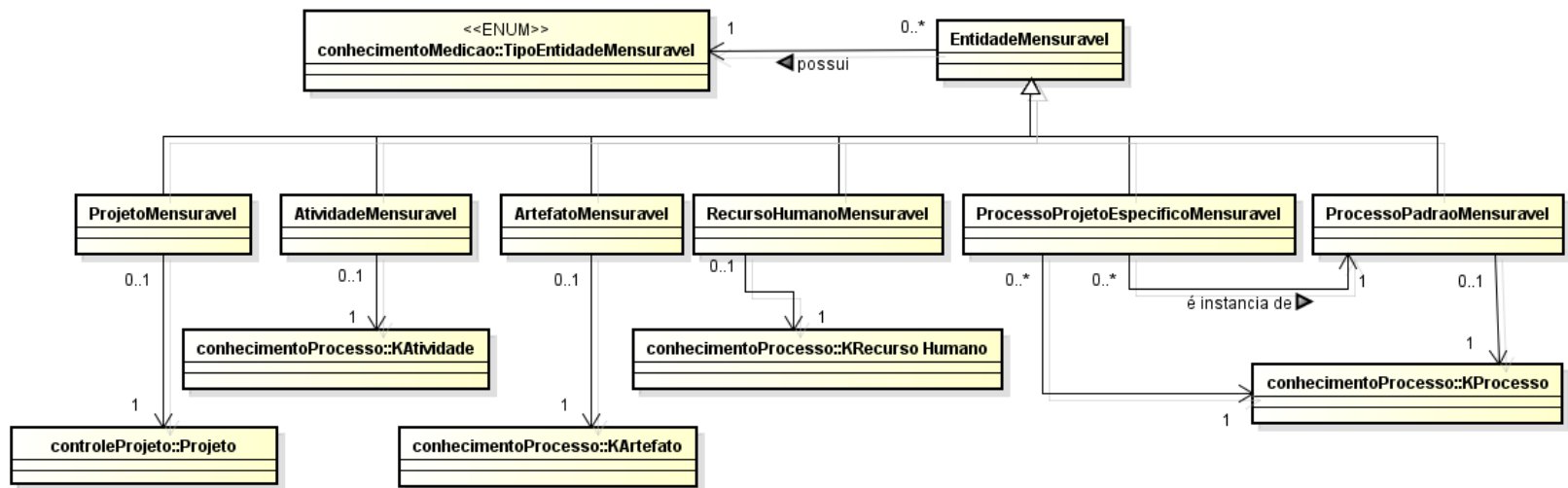


Figura 4.8 - Diagrama de Classes do cdp do subsistema *entidadeMensuravel*.

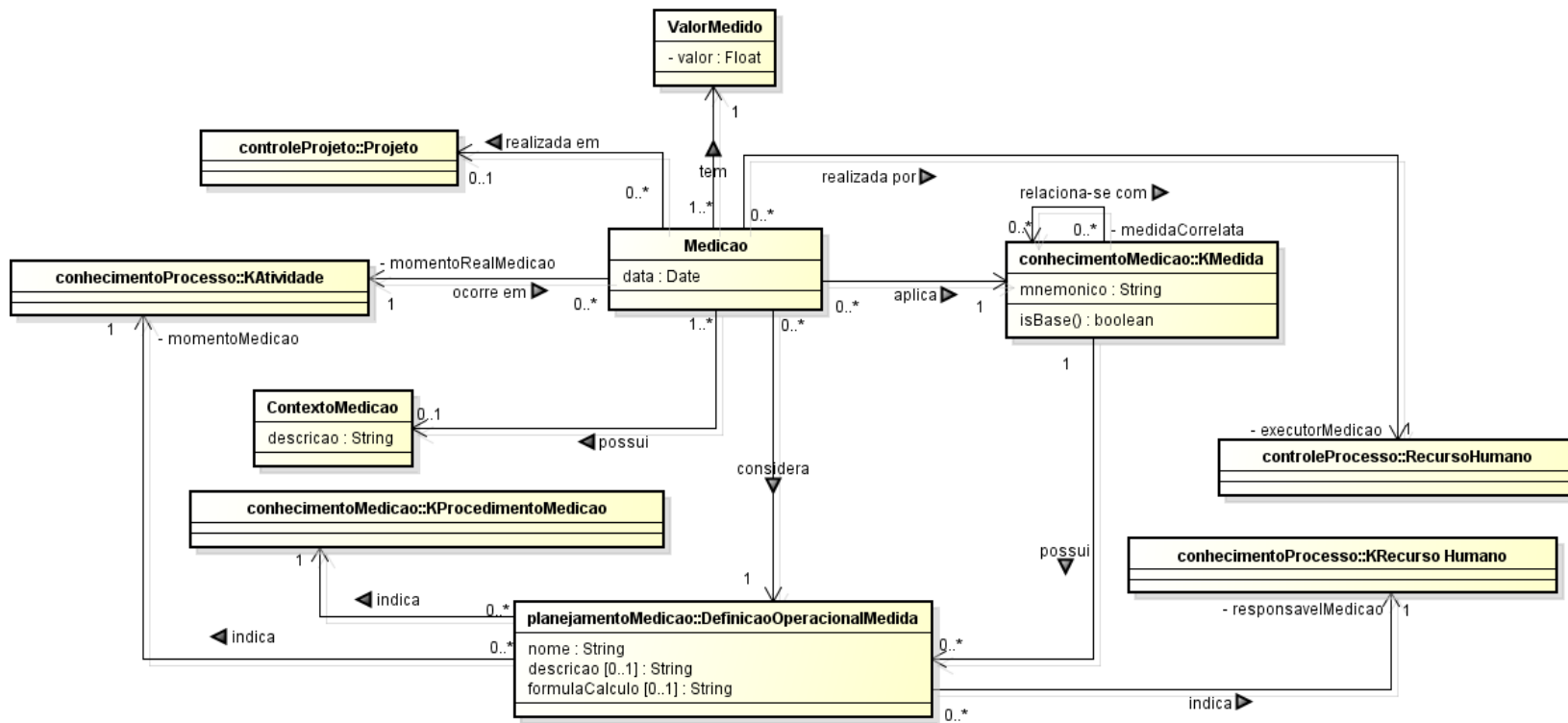
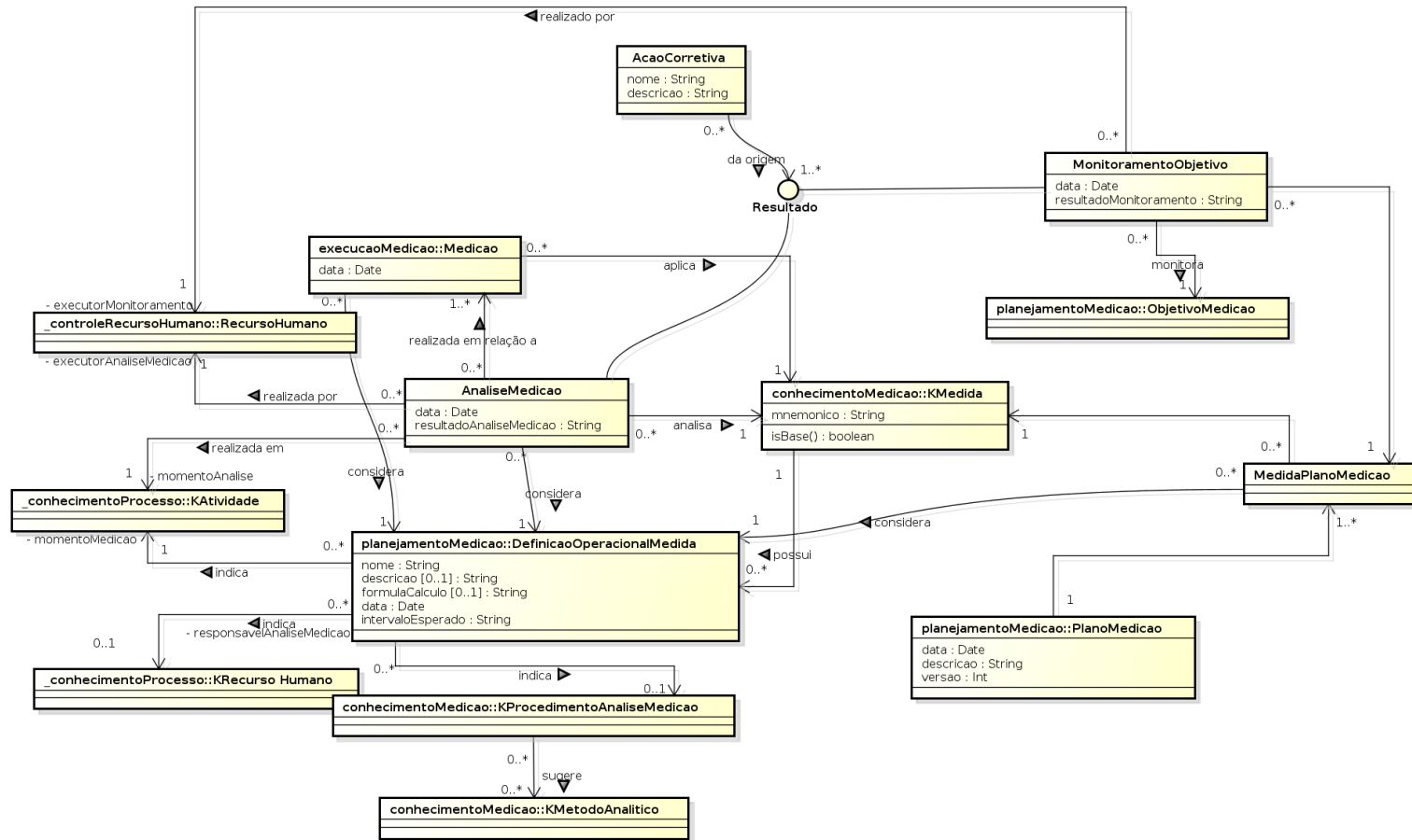


Figura 4.9 - Diagrama de Classes do cdp do subsistema *execucaoMedicao*.



powered by astah

Figura 4.10 - Diagrama de Classes do cdp do subsistema *analisarMedicao*.

4.5 A Ferramenta MeMOS

Nesta seção são apresentadas algumas telas de MeMOS. Basicamente, MeMOS é composta por um conjunto de funcionalidades que permitem o armazenamento de dados iniciais necessários à execução do processo de medição e monitoramento e por um outro conjunto de funcionalidades que apoia a realização do processo propriamente dito.

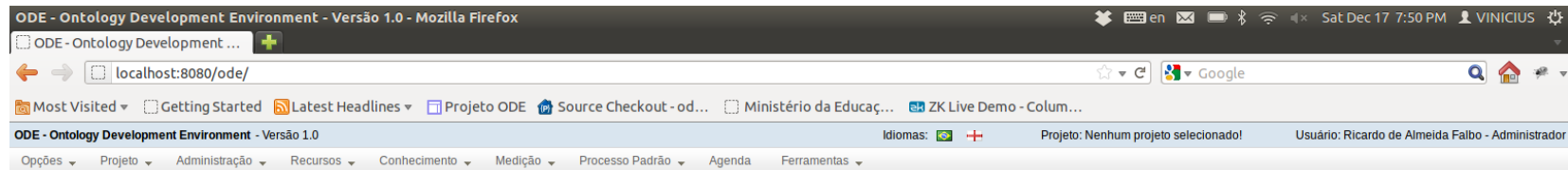
A porção de MeMOS que contém as funcionalidades que permitem o armazenamento de dados de apoio à execução do processo de medição e monitoramento está presente nos subsistemas *conhecimentoMedicao* e *entidadeMensuravel* descritos anteriormente, e representam o conhecimento da organização acerca de medição de software e as entidades mensuráveis presentes na mesma. Em termos mais simples, são cadastros úteis para a execução do processo de medição e monitoramento em si. Por se tratar de funcionalidades, basicamente, de cadastros, essa porção da ferramenta não será detalhada nesta seção.

Vale ressaltar que conhecimentos considerados comuns às organizações (por exemplo: entidades mensuráveis, escalas, periodicidades e unidades de medidas) são previamente armazenados na ferramenta.

A porção de MeMOS diretamente relacionada à execução da medição e monitoramento de objetivos inclui funcionalidades referentes ao planejamento da medição, estabelecimento de valores de referência para os indicadores, à execução da medição, à análise dos dados coletados e ao monitoramento dos objetivos.

Planejamento da Medição

Como definido no processo descrito na seção 3.2, os resultados do planejamento da medição são registrados em um Plano de Medição. Assim, o processo de medição tem início com o registro de um Plano de Medição da Organização. Para isso, são informadas a versão do plano, a data de criação e uma breve descrição. O registro de um Plano de Medição da Organização é realizado na tela apresentada na Figura 4.11.



Plano de Medição

Plano de Medição ▾ Objetivos ▾

Controle | Objetivos e Necessidades de Informação | Medidas

Versão*: 1.2

Responsável*: Analista ▾

Data*: 18/09/2011 📅

Descrição:

Salvar

javascript:

Figura 4.11 – Tela de registro de Plano de Medição da Organização.

Para elaborar o Plano de Medição da Organização o engenheiro de software identifica os objetivos estratégicos, de software e de medição que constarão no plano. A definição dos objetivos em MeMOS é realizada na tela apresentada na Figura 4.12, onde o engenheiro de software seleciona em uma estrutura de árvore os objetivos que farão parte do Plano de Medição da Organização. Embora não seja apresentado na figura, novos objetivos podem ser inseridos pelo usuário.

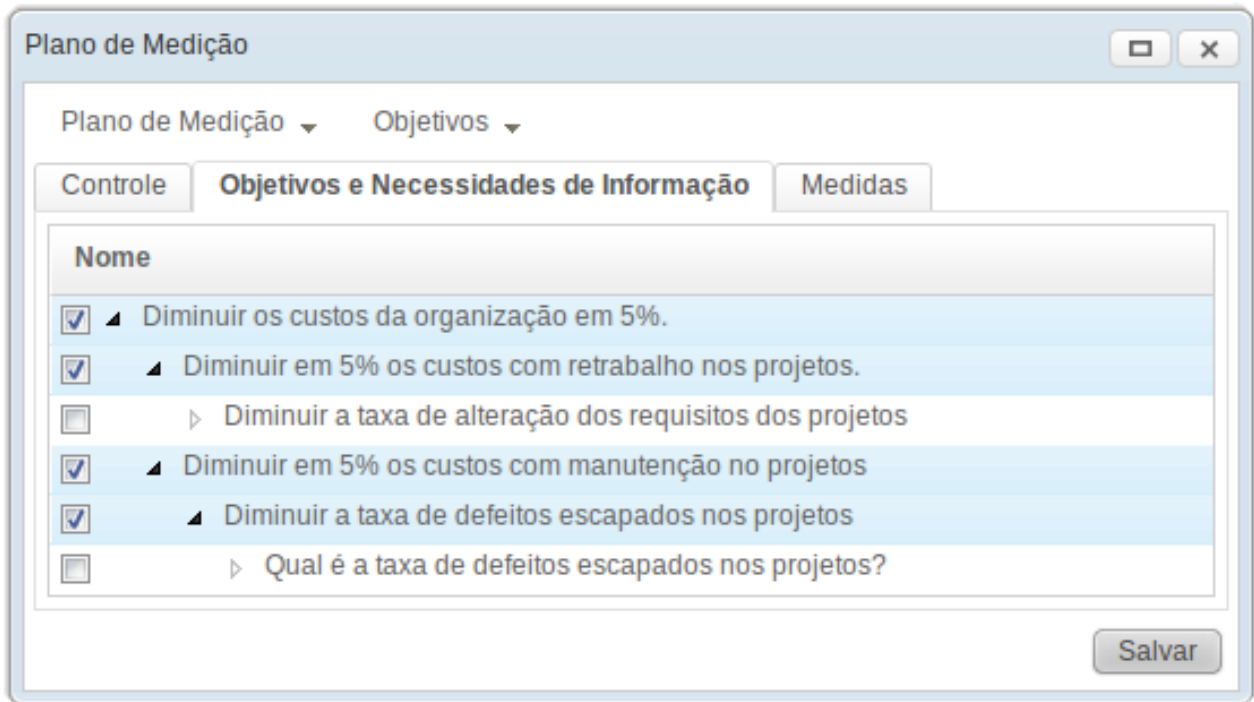


Figura 4.12 – Tela de identificação dos objetivos para o Plano de Medição da Organização.

Após identificar os objetivos, o engenheiro de software identifica as necessidades de informação associadas aos objetivos de medição, como ilustra a tela da Figura 4.13.

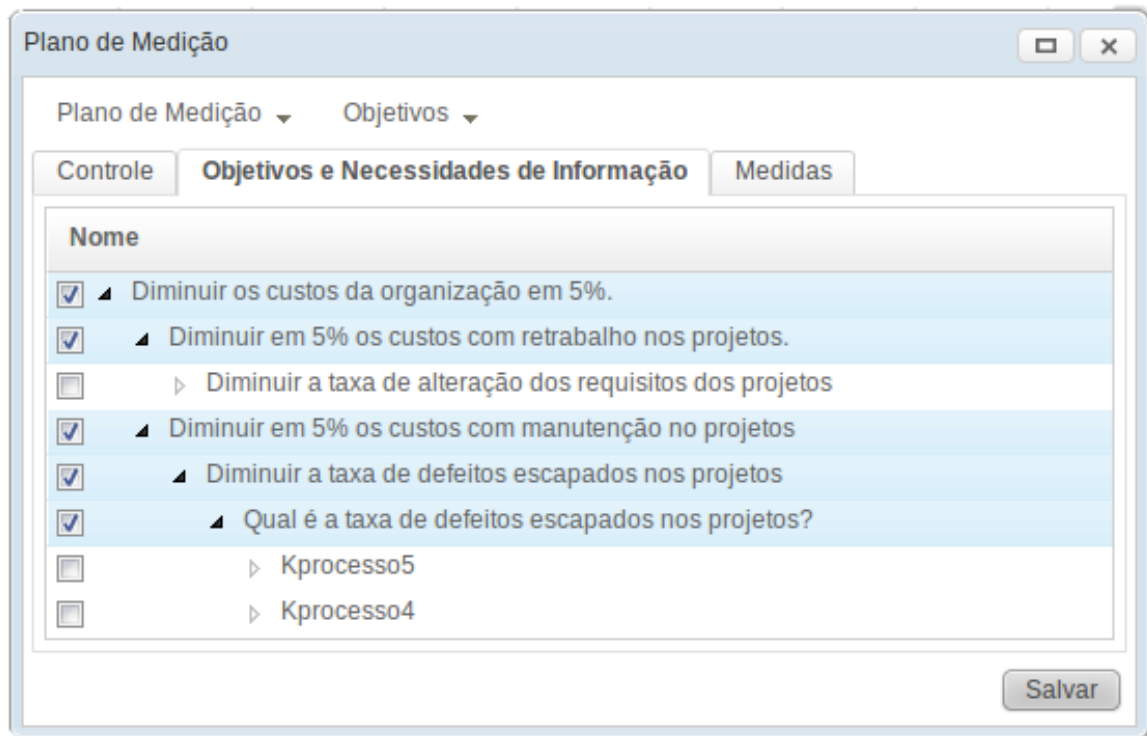


Figura 4.13 – Tela de identificação das necessidades de informação para o Plano de Medição da Organização.

Em seguida, o engenheiro de software identifica as medidas que atendem às necessidades de informação selecionadas (Figura 4.14).

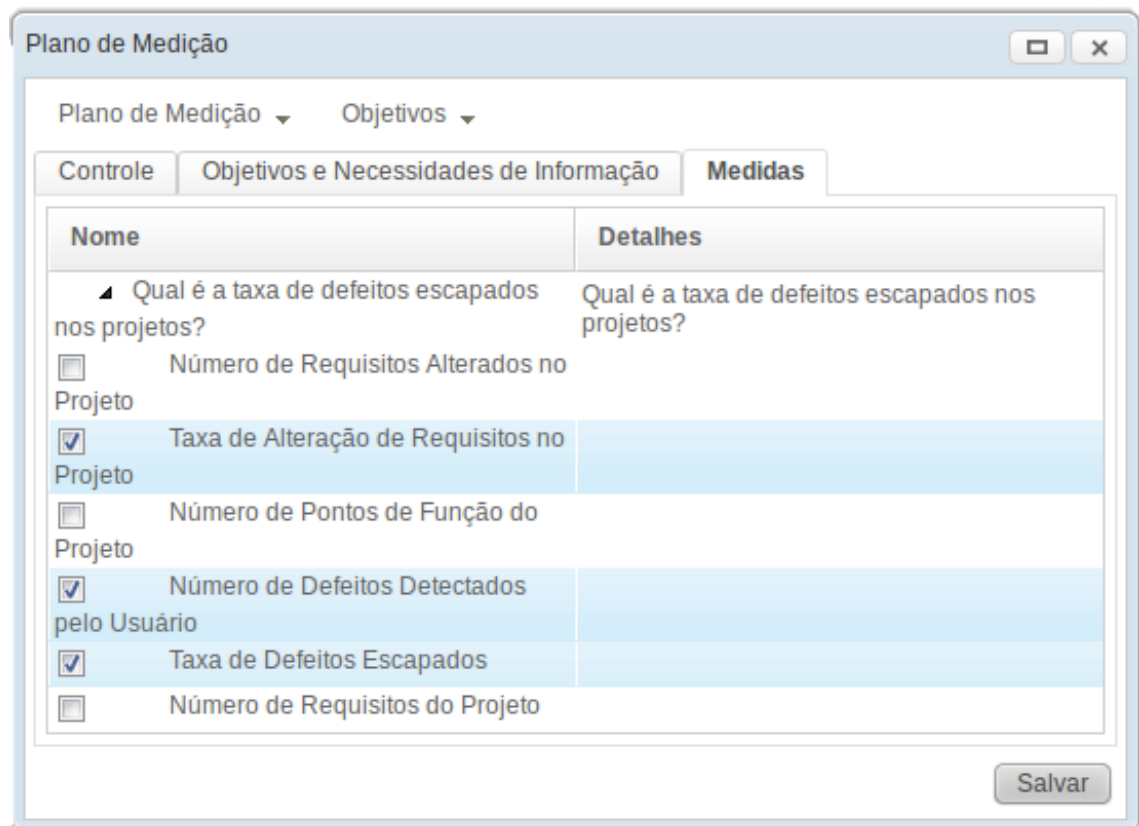


Figura 4.14 – Tela de identificação das medidas do Plano de Medição da Organização.

Para cada medida, o engenheiro de software deve informar a definição operacional que será considerada no Plano de Medição da Organização. Isso se faz necessário, pois uma mesma medida pode possuir diferentes definições operacionais, dependendo de sua aplicação. Uma mesma medida pode ser usada para monitoramento tradicional de projetos e para análise de desempenho de processos. No entanto, suas definições operacionais poderiam ser diferentes para atender às exigências de cada aplicação. Por exemplo, se a medida taxa de aderência ao cronograma é usada na medição tradicional, a definição operacional utilizada incluiria um procedimento de análise e método analítico da estatística clássica e uma periodicidade de coleta e análise mensal. Por outro lado, se essa medida fosse utilizada na análise de desempenho de processos, a definição operacional utilizada incluiria um procedimento de análise e método analítico do controle estatístico de processos e uma periodicidade de coleta e análise semanal. Na Figura 4.15 a tela utilizada para registrar uma definição operacional é apresentada.

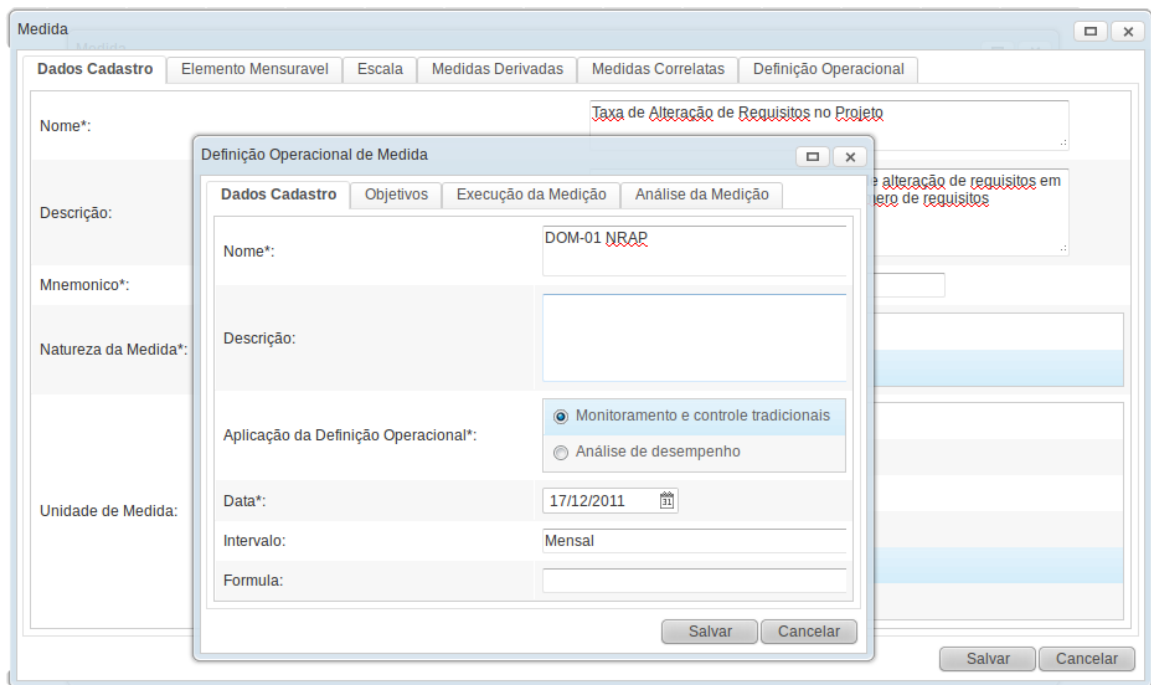


Figura 4.15 – Tela de registro de definição operacional para medidas.

A elaboração do Plano de Medição para um projeto é realizada seguindo-se, basicamente, o mesmo procedimento adotado para elaborar o Plano de Medição. No entanto, ao dar início ao Plano de Medição do Projeto, o engenheiro de software deve informar o Plano de Medição da Organização que servirá de base. O Plano de Medição da Organização é, então, instanciado (copiado) para o projeto e o engenheiro de software poderá realizar as alterações necessárias. A Figura 4.16 apresenta a tela de criação de um plano de medição de projeto.

Figura 4.16 – Tela de registro de um Plano de Medição do Projeto.

Estabelecimento dos Valores de Referência para os Indicadores

Com o plano de medição definido, é possível identificar valores de referência para os indicadores que serão utilizados para analisar o alcance dos objetivos de medição. Para isso, o usuário seleciona o indicador e informa os valores de referência associados aos termos *bom*, *regular* e *ruim*. A Figura 4.17 apresenta a tela onde é feito o registro dos valores de referência em MeMOS.

Vale ressaltar que o registro dos valores de referência pode ser feito no âmbito organizacional ou no âmbito de um projeto específico. Os valores de referência organizacionais são utilizados durante a análise dos dados de vários projetos. Os valores de referência de um projeto, por sua vez, são utilizados apenas no contexto daquele projeto. A possibilidade de haver valores diferentes para a organização e para os projetos baseia-se na possibilidade de um projeto possuir objetivos específicos, o que pode, por exemplo, levar a valores de referência mais rígidos que os da organização como um todo.

Valores de Referência

Data do Estabelecimento: 02/09/2011

Plano de Medição: a Organização - Versão 1.3

Objetivo Estratégico: Diminuir os custos da organização em 5%

Objetivo de Software: Diminuir em 5% os custos com retrabalho

Objetivo de Medição: Diminuir a taxa de alteração dos requisito

<input type="checkbox"/> Indicador	Bom	Regular	Ruim
<input type="checkbox"/> Número de Requisitos Alterados no Projeto			
<input checked="" type="checkbox"/> Taxa de Alteração de Requisitos no Projeto	0.1	0.3	0.5
<input type="checkbox"/> Número de Pontos de Função do Projeto			
<input checked="" type="checkbox"/> Número de Defeitos Detectados pelo Usuário	3	6	10
<input checked="" type="checkbox"/> Taxa de Defeitos Escapados	0.8	0.6	0.3
<input type="checkbox"/> Número de Requisitos do Projeto			

Salvar

Figura 4.17 – Tela de registro dos valores de referência para indicadores.

Execução da Medição

A execução da medição consiste na coleta e armazenamento dos dados para as medidas. A coleta dos dados é realizada em MeMOS por meio da tela apresentada na Figura 4.18. Para registrar uma medição, o usuário informa a medida para a qual o dado foi coletado, a definição operacional considerada, a data da medição, o valor medido, o projeto em que a medição foi realizada e as informações de contexto da medição

Medida:	axa de Alteração de Requisitos no Projeto
Definição Operacional:	DOM-01 NRAP
Valor Medido:	0.3
Data da Medição:	04/10/2011
Projeto:	Projeto A
Executor da Medição:	Ricardo de Almeida Falbo
Momento da Medição:	Levantar Requisitos
Contexto da Medição	--

Salvar

Figura 4.18 – Tela de registro de medições.

Análise de Medições e Monitoramento de Objetivos

Para realizar a análise dos dados, o engenheiro de software deve, inicialmente, selecionar os dados para análise. Para isso, o engenheiro de software informa a medida a ser analisada, a definição operacional a ser utilizada na análise, o período a ser considerado e os projetos a serem utilizados. O usuário pode escolher um projeto específico. A Figura 4.19 apresenta a tela que é utilizada para a seleção dos dados para análise.

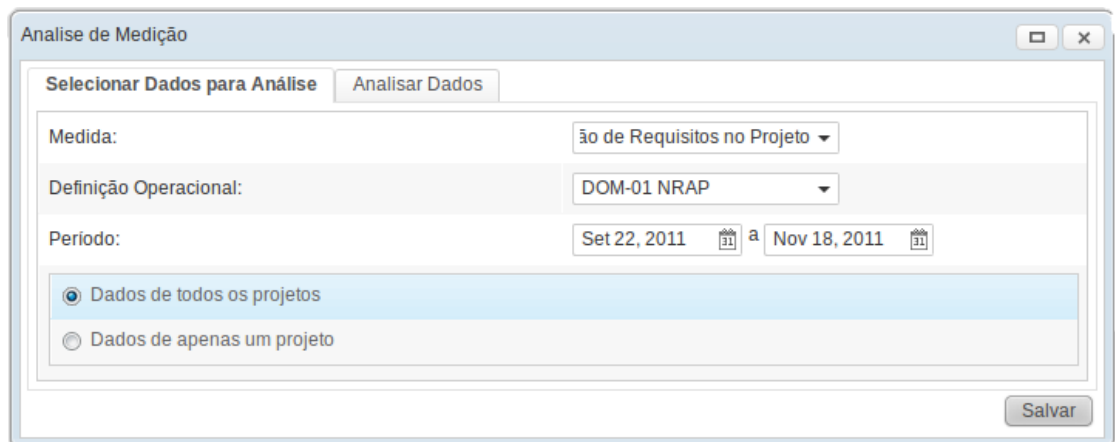


Figura 4.19 – Tela de seleção de dados para análise.

Selecionados os dados, a análise pode ser realizada. Um gráfico representando os dados selecionados é exibido em uma tela de MeMOS (Figura 4.19) e o engenheiro de software pode, então, analisar os dados e registrar o resultado da análise. Para facilitar a análise dos dados para monitoramento dos objetivos, os valores de referência são indicados em linhas coloridas no gráfico. Uma linha vermelha indica o valor de referência considerado *ruim*, uma linha amarela indica o valor *regular* e uma linha verde indica o valor *bom*.

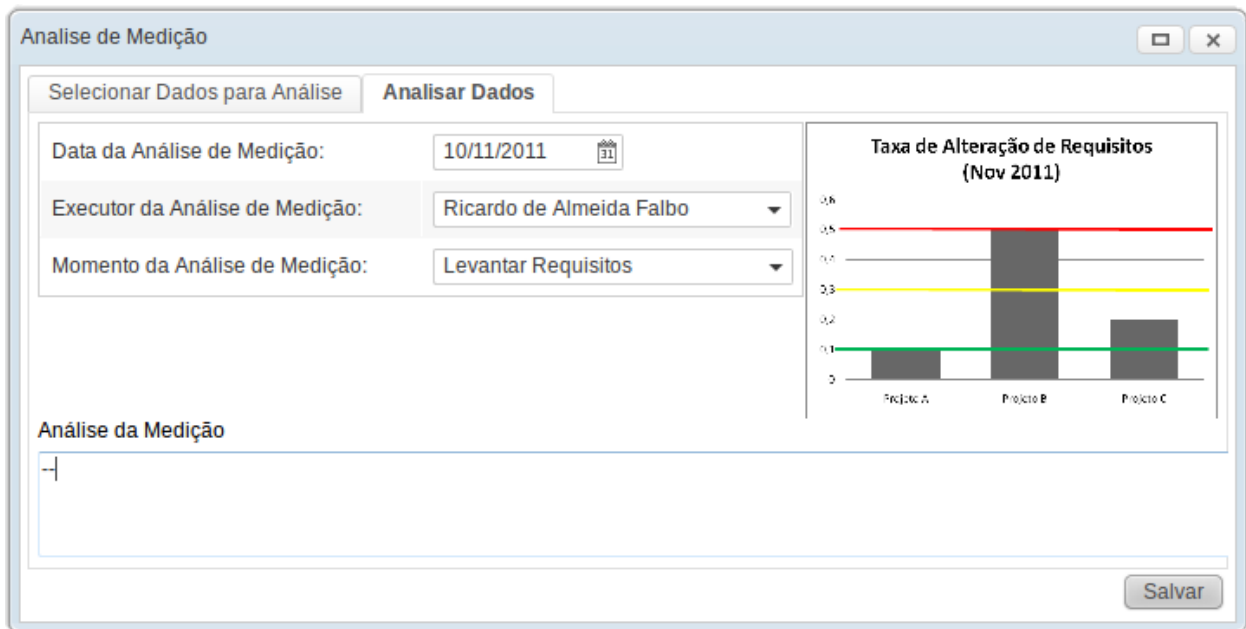


Figura 4.20 – Tela de registro da análise das medições.

Capítulo 5

Considerações Finais

5.1 Conclusões

Organizações de software precisam definir para seus projetos objetivos de software viáveis e alinhados aos seus objetivos de negócio. Além disso, devem acompanhá-los continuamente de modo a detectar possíveis ameaças ao seu alcance e, quando necessário, realizar ações corretivas. O monitoramento dos objetivos de projetos e da organização baseia-se na análise de dados coletados ao longo dos projetos, o que ocorre quando se realiza a medição de software.

A medição de software é considerada uma das atividades mais importantes para a gerência e melhoria de processos e produtos de software, uma vez que fornece subsídios para a elaboração de planos realistas para os projetos e possibilita o monitoramento da aderência da execução dos projetos em relação a seus planos. Isso é possível, pois as medidas, ao serem coletadas e armazenadas, podem ser analisadas através de métodos e fornecerem informações importantes para a tomada de decisão, envolvendo a identificação e realização de ações corretivas e preventivas que orientem os projetos e processos a alcançarem os objetivos para eles estabelecidos (BARCELLOS, 2008).

Nesse contexto, este projeto teve como objetivo desenvolver uma ferramenta para apoiar o processo de medição e monitoramento de objetivos de software definido durante a Iniciação Científica que antecedeu este trabalho. A ferramenta foi desenvolvida no contexto do Projeto ODE (FALBO *et al.*, 2005), um projeto que conta com a participação de professores, alunos de iniciação científica, graduação, mestrado e doutorado no Núcleo de Estudos em Modelagem Conceitual e Ontologias (NEMO) do Departamento de Informática.

Algumas ferramentas de apoio à medição já foram desenvolvidas, porém ainda há necessidades não atendidas, pois, como discutido no capítulo 2, algumas dessas ferramentas são limitadas a medidas específicas e outras precisam ser utilizadas juntamente com ambientes de Engenharia de Software completos.

Uma preocupação presente durante o desenvolvimento da ferramenta MeMOS foi a necessidade de que a interface favorecesse a execução das atividades a serem realizadas pelo usuário. Nesse sentido, buscou-se definir uma interface alinhada ao processo. Infelizmente, devido a limitações de tempo, algumas decisões relacionadas à interface não puderam ser completamente implementadas no contexto deste trabalho.

Em relação à experiência adquirida durante este trabalho, destaca-se a aquisição de conhecimento relacionado à medição de software e monitoramento de objetivos, que é conhecimento importante para um profissional de Engenharia de Software. Particularmente, o estudo da abordagem GQM (BASIL *et al*, 1994) possibilitou o entendimento da abordagem baseada em objetivos e necessidades de informação (questões) para realizar medição em uma organização de software.

Além disso, este trabalho possibilitou o aprendizado adicional ao adquirido via atividades regulares da graduação, já que permitiu o estudo de novas tecnologias, como os *framework* *Zkoss*, para interface Web, *Hibernate*, para persistência de dados, e *Spring*, para injeção de dependência, o banco de dados relacional *Postgres* e a linguagem de programação Java.

Além desses aprendizados, muitos temas estudados ao longo do curso de Ciência da Computação foram colocados em prática durante o desenvolvimento deste trabalho como, por exemplo, especificação de requisitos, análise, projeto de sistemas, implementação e testes. Colocar em prática a teoria aprendida para desenvolver um software do início ao fim de uma versão operacional, permitiu integrar o conhecimento para executar as etapas do ciclo de vida de um software.

Considerando as dificuldades encontradas para desenvolver este trabalho, destaca-se a rotatividade de alunos que trabalham no Projeto ODE, o que pode resultar na queda da produtividade de novos alunos, uma vez que há fuga do conhecimento adquirido por alunos experientes no ambiente. Apesar da equipe atual do projeto estar contando com a colaboração de membros experientes, a maior parte da equipe corrente ingressou no projeto recentemente. Havendo menos pessoas experientes no projeto, algumas mudanças ocorreram em ritmo lento o que acabou levando à necessidade de reimplementar alguns itens.

Em relação às contribuições deste trabalho, acredita-se que a documentação elaborada de forma detalhada e a ferramenta propriamente dita sejam as principais contribuições.

É importante ressaltar que, uma vez que o processo a ser apoiado pela ferramenta é um processo relativamente grande, a versão de MeMOS produzida neste trabalho possui algumas limitações. A principal delas é a não cobertura de todo o processo, uma vez que a parte de monitoramento dos objetivos propriamente dita não foi implementada. Além disso, alguns itens ficaram pendentes, como a inclusão de gráficos para a análise. A interface também apresenta aspectos que precisam ser melhorados.

5.2 Trabalhos Futuros

No final do desenvolvimento de um software, tipicamente novas necessidades são identificadas. A manutenção e evolução de software deve ser um trabalho constante, de forma que o ciclo de vida não finalize na homologação, mas permaneça ao longo de toda a vida do software.

Sendo a versão de MeMOS produzida neste trabalho uma versão inicial, algumas necessidades identificadas podem ser realizadas em trabalhos futuros.

Uma dessas necessidades é a definição e implementação de um mecanismo de validação dos dados coletados para as medidas. Atualmente, não é feita validação dos dados antes do armazenamento.

Outra necessidade é o desenvolvimento da versão de MeMOS integrada ao ODE. O ambiente ODE, como dito anteriormente, contém um conjunto de ferramentas integradas. Como tal, há a necessidade de que uma ferramenta comunique-se e conheça o comportamento de outra, porém o desenvolvimento de cada ferramenta é individual. Como dito na Introdução desta monografia, embora faça parte do ODE, MeMOS pode ser utilizada independentemente dele. A versão produzida neste trabalho permite o uso de MeMOS de forma independente de ODE. No entanto, organizações podem desejar usar a ferramenta integrada ao ODE, usufruindo, também, das outras funcionalidades do ambiente. Para isso, uma nova versão deve ser criada ou uma estratégia de integração como, por exemplo, um sistema de eventos, deve ser implementada.

Outro trabalho futuro é a evolução de MeMOS para apoiar as práticas da medição em alta maturidade. A versão atual de MeMOS contempla a medição tradicional, ou seja, a medição realizada nos níveis iniciais dos modelos de maturidade como MR MPS.BR e CMMI. Para apoiar as práticas da alta maturidade, que envolve o controle estatístico de processos, a ferramenta deve ser evoluída. O uso da Ontologia de Medição de Software (BARCELLOS, 2009) como base para MeMOS favorece essa evolução, pois a ontologia contempla aspectos da medição tradicional e em alta maturidade e, para evoluir MeMOS, basta considerar a parte da ontologia que não foi considerada neste trabalho e estender a ferramenta para atendê-la.

Referências Bibliográficas

- AURUM, A., WOHLIN, C., 2005, *Engineering and Managing Software Requirements*, Springer-Verlag
- 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, *Estratégia para Medição de Software e Avaliação de Bases de Medidas para Controle Estatístico de Processos de Software em Organizações de Alta Maturidade*, Tese de D. Sc. PESC, COPPE/UFRJ, Rio de Janeiro, Brasil.
- BARCELLOS, M. P., 2011, Medição de Software, Material de Minicurso, *VII Workshop Anual do MPS.BR*, Campinas, São Paulo, Outubro.
- BARRETO, A. O. S., ROCHA, A. R., 2009, Definição e Monitoração de Objetivos de Software Alinhados ao Planejamento Estratégico, *Anais do VIII Simpósio Brasileiro de Qualidade de Software*, Ouro Preto, Brasil, Junho, p. 114-128.
- BASILI, V., CALDIERA, G., ROMBACH, H., 1994, “Goal Question Metric Paradigm”. In: *Encyclopedia of Software Engineering*, v. 1, John Wiley & Sons, pp. 528-532.
- 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.
- BERTOLLO, G., SEGRINI, B., FALBO, R. A. Definição de Processos de Software em um Ambiente de Desenvolvimento de Software Baseado em Ontologias. In: *V Simpósio Brasileiro de Qualidade de Software*, 2006, Vila Velha, Brasil. Anais...2006. p. 72-86.
- DUMKE, R., & EBERT, C., 2010,. *Software Measurement: Establish - Extract - Evaluate - Execute*: Springer-Verlag Berlin Heidelberg.
- FALBO, R. A., 2004, "*Experiences in Using a Method for Building Domain Ontologies*", In: Proceedings of the Fourth International Conference on Quality Software - QSIC'2004, IEEE Computer Society, Braunschweig, Germany, pp. 162-169.
- FALBO, R. A., A. C. C. NATALI, et al. 2003. "ODE: *Ontology-based software Development Environment*." IX *Congreso Argentino de Ciencias de la Computación*. La Plata, Argentina, Outubro de 2003. : 1124-1135.
- FALBO, R. A., RUIZ, F., & DALMORO, R. 2005. Using Ontologies to Add Semantics to a Software Engineering Environment. In: *Proceedings of the 17th International Conference on Software Engineering and Knowledge Engineering*, Taipei, China , 151-156.
- GARCÍA, F., BERTOIA, M. F., CALERO, C., VALLECILLO, A., RUIZ, F., PIATTINI, M., GENERO, M., 2006, "Towards a Consistent Terminology for Software Measurement Information and Software Technology", *Information and Software Technology*, v. 48, n. 8, pp. 631+644.
- IEEE, 1998, Std 1061 – *IEEE Standard for a Software Quality Metrics Methodology*.
- ISO/IEC, 2002, ISO/IEC 15939 – 2002 (E) *Software Engineering – Software Measurement Process*, International Organization for Standardization and the International Electrotechnical Commission, Geneva, Switzerland.

- ISO/IEC-12207, 2008, Systems and software engineering – Software life cycle processes, The International Organization for Standardization and the International Electrotechnical Commission.
- IT GOVERNANCE INSTITUTE, 2007. “*Control Objectives for Information and Related Technology*”, 4.1th edition, <http://www.itgi.org>.
- KAPLAN, R.S., NORTON, D.P., 1996, *The Balanced Scorecard Translating Strategy Into Action*, Boston, Massachusetts, Havard Business School Press.
- LIMA, A. et al., 2006, WebAPSEE: Um Ambiente Livre e Flexível Para Gerência de Processos de Software, 7º Forum do Software Livre, WSL 2006 - VII Workshop de Software Livre . Porto Alegre, Abril, 2006.
- MARKOVIC, I., KOWALKIEWICZ, M., 2008, “Linking Business Goals to Process Models in Semantic Business Process Modeling”, In: *12th International IEEE Enterprise Distributed Object Computing Conference*, pp. 332-338, Munich, Germany, September.
- MARTINS, A. F., J. C. NARDI, et al. 2006. ReqODE: Uma Ferramenta de Apoio à Engenharia de Requisitos Integrada ao Ambiente ODE. *Sessão de Ferramentas do XX Simpósio Brasileiro de Engenharia de Software - SBES'2006*, Florianópolis.
- 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.
- MORO, R. D., J. C. NARDI, et al. 2005. ControlPro: Uma Ferramenta de Acompanhamento de Projetos Integrada a um Ambiente de Desenvolvimento de Software. *XII Sessão de Ferramentas do Simpósio Brasileiro de Engenharia de Software*, SBES'2005 Uberlândia, Brasil.
- NATALI, A. C. C. and R. A. Falbo 2003. Gerência de Conhecimento em ODE. *Proceedings of the XVII Brazilian Symposium on Software Engineering - SBES'2003*. Manaus - Amazonas, Brazil: 270-285.
- PARK, R.E., GOETHERT, W.B., FLORAC, W.A., 1996, *Goal-Driven Software Measurement — A Guidebook*, Carnegie Mellon University, Pittsburgh, PA, CMU/SEI-96-HB-002.
- PMBOK, 2008, *PMBOK - A Guide to the Project Management Body of Knowledge*, Project Management Institute.
- PORTER, M., 2004, *Techniques for Analyzing Industries and Competitors*, Free Press.
- RIBEIRO, T.; NASCIMENTO, L.; LIMA, L.; LIMA REIS, C. A.; REIS, R. Q. Apoio à Medição em um ADS Centrado em Processos. In: *Sessão de Ferramentas do VII Workshop Anual do MPS, 2011, Campinas. VII Workshop Anual do MPS, 2011*.
- ROCHA, A. R., SOUZA, J. M., AGUIAR, T. C., 1990, "TABA: A Heuristic Workstation for Software Development", In: *Proceedings of COMPEURO 90*, Tel Aviv, Israel, pp. 126-129.
- SCHNAIDER, L., SANTOS, G., MONTONI, M., ROCHA, A.R., 2004, Uma Abordagem para Medição e Análise em Projetos de Desenvolvimento de Software, *III Simpósio Brasileiro de Qualidade de Software – SBQS 2004*, Brasília – DF.
- SEGRINI. B.M. *Definição de Processos Baseada em Componentes*. Dissertação de Mestrado. PPGI/UFES. Vitória, Brasil 2009.

- SEI, SOFTWARE ENGINEERING INSTITUTE. *CMMI for Development version v. 1.2* Pittsburg, PA. Software Engineering Institute, Carnegie Mellon University. USA, 2006. Disponível em: www.sei.cmu.edu Acesso em: 15 Jun 2010.
- SOFTEX, 2011. MPS,BR. *Melhoria de Processo do Software Brasileiro – Guia Geral 2011*. Disponível em: <http://www.softex.br/mpsbr>.
- SOLINGEN, R., BERGHOUT, E., 1999, *The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development*, McGrawHill.
- SOUZA, A.D., 2008a, *Uma Abordagem para Gerência Estratégica de Portfólio: Foco na Seleção de Projetos*, Dissertação de M. Sc. PESC, COPPE/UFRJ, Rio de Janeiro, Brasil.
- 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.