

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

Allan Clipes Rosa

**Evolução de MePPLa - uma Linguagem de Padrões  
para Planejamento de Medição visando ao Controle  
Estatístico de Processos 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

Coorientador: Daisy Ferreira Brito

**Vitória, ES**

**2019**

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

Allan Clipes Rosa

**Evolução da MePPLa - uma Linguagem de  
Padrões para Planejamento de Medição  
visando ao Controle Estatístico de Processos  
de Software**

COMISSÃO EXAMINADORA

---

Prof<sup>ª</sup> Monalessa Perini Barcellos, D.Sc.

---

Prof. Vítor Estêvão Silva Souza, D. Sc.

---

Prof<sup>ª</sup>. Jordana Sarmenghi Salamon, M. Sc.

Vitória-ES11, 11 de Dezembro de 2019.

À minha família, que me compõe integralmente.

## **AGRADECIMENTOS**

Aos meus pais, Adolfo e Cássia, por todo o amor e suporte que sempre me foi dado, assim como me daram a base de quem sou hoje. A minha irmã, Melyna, pela amizade e compreensão que construímos.

Às minhas avós, Eva e Terezinha, e minha Tia, Renata, pela oportunidade de ter quatro mães nessa jornada e pelos exemplos de vida que me deram.

Aos meus amigos, pelo companheirismo e compreensão nos momentos de dificuldade.

Aos professores da universidade, pelo compartilhamento do conhecimento e pelas oportunidades de crescimento.

Aos servidores da universidade, por criarem um espaço que permitiu essa jornada.

À minha orientadora e coorientadora, Monalessa e Daisy, pela oportunidade de desenvolver este trabalho e por toda o esforço realizado na elaboração dele.

Aos professores Vítor e Jordana, por aceitarem o convite para compor a banca para a avaliação deste trabalho, e contribuírem na avaliação dele.

A todos aqueles que me ajudaram, em grau qualquer, a tornar essa experiência mais rica ou sutil.

E finalmente, a Deus, pela minha vida e por permitir que tudo ocorra da maneira que deve ocorrer.

## RESUMO

Medição de software é um processo indispensável para que organizações atinjam a maturidade no desenvolvimento de software. Na gerência de projetos, é uma prática fundamental para a melhoria dos processos, visto que ela apoia tomadas de decisão em níveis organizacional e de projetos, através de dados coletados. Dessa forma, a medição deve ser adequada aos objetivos de cada projeto e da organização. Modelos de maturidade como o CMMI e MR-MPS-SW tratam a melhoria de processos em níveis, onde a medição de software começa nos níveis iniciais e nos níveis de maturidade mais elevados incluem a prática do controle estatístico de processos.

Utilizar controle estatístico de processos requer certos cuidados, e a seleção de medidas apropriadas ao controle estatístico de processos tem sido indicada como uma dificuldade na implementação da prática. A partir da literatura e de experiências práticas, é possível identificar conjuntos de medidas recorrentes em contextos de objetivos semelhantes. Um padrão pode ser entendido como uma solução para um problema recorrente. Portanto, padrões para planejamento de medição apresentam soluções para problemas relacionados ao planejamento de medição, por exemplo, quais medidas selecionar para criar um plano de medição. Padrões podem ser organizados em linguagens de padrões, que buscam representar os padrões e suas relações e definir um processo que guie na seleção e utilização dos padrões. Utilizar linguagens de padrões favorece o reúso, contribuindo com a produtividade. Considerando esse contexto, em (TOMAZ, 2017) foi proposta MePPLa (*Measurement Planning Pattern Language*) uma linguagem de padrões para o planejamento de medição visando ao controle estatístico de processos, e MePPLa Tool, uma ferramenta de apoio ao uso de MePPLa.

Este trabalho teve como objetivo evoluir MePPLa, acrescentando a ela um processo e novos padrões. Além disso, o trabalho buscou melhorar *MePPLa Tool*, incluindo uma funcionalidade para facilitar incorporar à ferramenta futuras evoluções de MePPLa.

**Palavras-chave:** Medição de Software, Controle Estatístico de Processos, Linguagem de Padrões, MePPLa.

# SUMÁRIO

<b>Capítulo 1 Introdução .....</b>	<b>8</b>
1.1 Introdução .....	8
1.2 Objetivos .....	10
1.3 Histórico do Desenvolvimento do Trabalho.....	11
1.4 Organização do Texto.....	12
<b>Capítulo 2 Fundamentação Teórica .....</b>	<b>13</b>
2.1 Medição de Software e Controle Estatístico de Processos.....	13
2.2 Linguagem de Padrões .....	17
2.3 Measurement Planning Pattern Language (MePPLa) .....	18
2.4 Systematic Approach for creating Measurement Planning Pattern Languages (SAMPPLa) .....	22
2.4.1 Desenvolver Fonte para Extração dos Padrões.....	22
2.4.2 Desenvolver Linguagem de Padrões .....	24
<b>Capítulo 3 Evolução de MePPLa .....</b>	<b>26</b>
3.1 Desenvolver Fonte Para Extração de Padrões.....	26
3.1.1 Definir Propósito e Contexto de Aplicação da Linguagem de Padrões .....	26
3.1.2 Selecionar Processos.....	26
3.1.3 Identificar Conjunto de Objetivos de Medição e Medidas para Extração dos Padrões .....	26
3.1.4 Selecionar Objetivos de Medição para o CEP .....	27
3.1.5 Selecionar Medidas Adequadas para o CEP.....	27
3.1.6 Rever e Refinar Processos, Objetivos de Medição e Medidas para o CEP .....	27
3.2 Desenvolver Linguagem de Padrões.....	27
3.2.1 Identificar Padrões de Planejamento de Medição .....	28
3.2.2 Construir a Linguagem de Padrões .....	29
<b>Capítulo 4 Evolução de MePPLa Tool.....</b>	<b>32</b>
4.1 Introdução .....	32
4.2 Requisitos e Modelagem Conceitual .....	33
4.2.1 Casos de Uso.....	34
4.2.3 Diagramas de Classes.....	35
4.3 Projeto de Sistema .....	37
4.3.1 Arquitetura de Software.....	38
4.3.2 Camada Lógica do Negócio.....	38
4.3.3 Camada de Interface com o Usuário .....	40
4.3.4 Camada de Gerência de Dados.....	42
4.4 MePPLa Tool.....	43
<b>Capítulo 5 Considerações Finais .....</b>	<b>51</b>
5.1 Conclusões.....	51
5.2 Trabalhos Futuros .....	54

**Apêndice A..... 59**

# Capítulo 1

## Introdução

---

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

### 1.1 Introdução

Medição de software é um processo indispensável para as organizações atingirem a maturidade no desenvolvimento de software (SEI, 2010). Entretanto, medir software custa caro e envolve um esforço considerável em termos de recursos humanos. Assim, quando uma organização realiza a medição simplesmente por medir ou para atender a requisitos de modelos de maturidade, como o CMMI - *Capability Maturity Model Integration* (SEI, 2010) e o MR-MPS-SW - Modelo de Referência para Melhoria de Processo de Software Brasileiro (SOFTEX, 2016), não se atingem os objetivos propostos pela medição de software (ROCHA *et al.*, 2012).

De acordo com o nível de maturidade da organização, a medição de software é realizada de diferentes formas. Na alta maturidade, além das medições tradicionais, que fundamentalmente buscam comparar valores executados com valores planejados, é necessário realizar Controle Estatístico de Processos (CEP). O CEP possibilita conhecer o comportamento dos processos, determinar seu desempenho em execuções anteriores para, então, prever o desempenho dos projetos presentes e futuros e analisar se os objetivos esperados podem ser alcançados (FLORAC; CARLETON, 1999).

A aplicação do CEP requer que os processos de software possuam medidas adequadas, dados válidos e úteis, armazenados em uma base organizacional de medidas e um conjunto de processos padrão definidos e implementados (FLORAC; CARLETON, 1999). Entretanto, as organizações encontram dificuldades em selecionar medidas adequadas a esse contexto (BARCELLOS, 2009a). Tomaz (2017) ressalta que, na literatura, existem vários trabalhos que apresentam medidas que podem ser aplicadas ao CEP, mas que, no entanto, as informações encontram-se dispersas, tornando seu acesso às vezes ineficiente.

A partir de um conjunto de medidas utilizadas no CEP, é possível identificar alguns *padrões* formados por medidas que podem ser usadas no monitoramento de objetivos e para realizar o controle estatístico de certos processos (TOMAZ, 2017).



## Capítulo 1 - Introdução

Deutsch *et al.* (2004) afirmam que *padrões* são veículos para o encapsulamento de conhecimento e permitem capturar o que deve ser feito para resolver um dado problema. Um *padrão* pode ser entendido como uma solução bem-sucedida para um problema recorrente. Assim, padrões que apoiem a seleção de medidas a serem incluídas em um Plano de Medição podem apresentar soluções para problemas relacionados ao Planejamento de Medição (TOMAZ, 2017).

*Padrões* normalmente são utilizados em conjunto, podendo ser organizados em uma *Linguagem de Padrões (LP)*. Uma LP tem como objetivo prover um processo para resolução sistemática dos problemas através da representação dos padrões e suas relações de forma a guiar o usuário na seleção e utilização dos padrões (DEUTSCH, 2004; BUSCHMANN, 2007).

Para realizar a medição de software, primeiramente a organização precisa planejá-la, considerando seus objetivos. Como resultado dessa atividade, tem-se um Plano de Medição que deve conter os objetivos da medição, as necessidades de informação e as medidas que serão utilizadas para atender a essas necessidades, incluindo orientações para coleta e análise dessas medidas (BASILI *et al.*, 1994; BASILI, 1992; SOLINGEN & BERGHOUT, 1999).

Existem algumas abordagens propostas na literatura que auxiliam o planejamento da medição, como, por exemplo, o *Goal-Question-Metric (GQM)* (BASILI *et al.*, 1994 e BASILI, 1992; SOLINGEN & BERGHOUT, 1999). Trata-se de um método corriqueiramente utilizado devido a sua simplicidade de uso, bem como a estrutura *top down*, que, partindo-se de questões ligadas aos objetivos de medição, permite a identificação de necessidades de informação, as quais podem ser atendidas por meio de medidas (ROCHA, SOUZA e BARCELLOS, 2012).

Considerando a dificuldade de elaboração de planos de medição adequados ao CEP de software, Tomaz (2017) propôs uma linguagem de padrões para apoiar o planejamento de medição visando ao CEP, denominada MePPLa – *Measurement Planning Pattern Language*. Em MePPLa, cada padrão é uma solução composta de um objetivo de medição, um processo a ser submetido ao CEP e medidas que permitem analisar o comportamento do processo e monitorar o objetivo referido. Os padrões presentes em MePPLa foram identificados com base em resultados de um mapeamento sistemático da literatura (BRITO, BARCELLOS, 2016; TOMAZ, 2017) e um survey com profissionais da indústria (TOMAZ, 2017) que investigaram medidas utilizadas no CEP. MePPLa foi criada seguindo a abordagem SAMPPLa (*Systematic Approach for creating Measurement Planning Pattern Languages*), uma abordagem para guiar a criação de LPs para planejamento de medição visando ao CEP, também proposta por Tomaz (2017). Essa abordagem define os passos a serem seguidos

## Capítulo 1 - Introdução

para a criação ou evolução de uma linguagem de padrões para planejamento de medição de software, incluindo desde o desenvolvimento das fontes para extração de padrões, até o desenvolvimento da linguagem de padrões propriamente dita.

Para potencializar o uso de MePPLa, foi desenvolvida uma ferramenta, *MePPLa Tool*, que permite que o usuário navegue pelo modelo comportamental de MePPLa e selecione os padrões que deseja usar. Após a seleção dos padrões, a ferramenta produz um arquivo contendo a especificação dos padrões selecionados. Esse arquivo pode ser exportado no formato do MSWord e pode ser editado, podendo ser adaptado (por exemplo, inserindo-se objetivos estratégicos) para se transformar em um plano de medição (TOMAZ, 2017).

Atualmente, MePPLa inclui padrões relacionados aos processos Gerência de Projetos, Codificação e Testes. No total, em (TOMAZ, 2017) foram definidos 28 padrões, sendo 12 relacionados ao processo Gerência de Projetos, 6 relacionados ao processo Codificação e 10 relacionados ao processo Testes.

Linguagens de padrões podem estar sempre em evolução, podendo, por exemplo, ter novos padrões adicionados e novas relações identificadas (DEUTSCH *et al*, 2004; BUSCHMANN, 2007). Dessa forma, busca-se cada vez mais identificar processos da engenharia de software que possam ser submetidos ao CEP, ampliando as possibilidades de utilização de MePPLa para apoiar o planejamento de medição.

Neste trabalho, em suma, MePPLa foi evoluída utilizando-se a abordagem SAMPPLa, fornecendo um acervo maior de processos para serem submetidos ao CEP. Acrescentou-se um novo processo da engenharia de software à Linguagem de Padrões. Também foram realizadas melhorias na ferramenta *MePPLa Tool* com o objetivo de facilitar o processo de incorporação de evoluções da linguagem MePPLa, como a que foi realizada ao se incluir um novo processo<sup>1</sup>.

### 1.2 Objetivos

O **objetivo geral** deste trabalho é *evoluir a Linguagem de Padrões MePPLa e sua ferramenta de apoio*. Esse objetivo geral pode detalhado nos seguintes **objetivos específicos**:

- i. Selecionar um novo processo para ser tratado em MePPLa e definir os padrões a ele associados;
- ii. Evoluir MePPLa, adicionando a ela o novo processo e padrões relacionados;

---

<sup>1</sup> Neste trabalho, quando se usa o termo MePPLa, se está referindo à linguagem de padrões em si. O termo MePPLa Tool é usado para se referir à ferramenta desenvolvida para apoiar o uso de MePPLa.

## Capítulo 1 - Introdução

- iii. Evoluir a ferramenta de apoio à MePPLa (*MePPLa Tool*) para incorporar as evoluções feitas na linguagem;
- iv. Evoluir a ferramenta de apoio à MePPLa (*MePPLa Tool*) para facilitar o processo de incorporação de evoluções da linguagem MePPLa à ferramenta.

### 1.3 Histórico do Desenvolvimento do Trabalho

Para se atingir os objetivos propostos, no desenvolvimento deste trabalho foram realizadas as seguintes atividades:

- i. *Revisão Bibliográfica:* O trabalho iniciou-se com uma revisão bibliográfica sobre Medição de Software, Controle Estatístico de Processos e Linguagem de Padrões, incluindo a Linguagem de Padrões para Planejamento de Medição (MePPLa) e a abordagem para criação de Linguagem de padrões para Planejamento de Medição (SAMPPLa). Analisou-se artigos científicos e trabalhos acadêmicos relevantes a este trabalho.
- ii. *Seleção de processo e identificação dos padrões a serem acrescentados à MePPLa:* Nesta atividade foi realizada a evolução de MePPLa. Foi selecionado o processo de Revisão como novo processo para ser tratado por MePPLa. Nove padrões relacionados a esse processo foram identificados e organizados em MePPLa. A especificação de MePPLa foi atualizada incluindo o processo e os padrões criados.
- iii. *Estudo da ferramenta MePPLa Tool:* Nesta atividade foi feito um estudo da ferramenta proposta por Tomaz (2017), a *MePPLa Tool*, para entendimento do código e das tecnologias utilizadas para implementação das melhorias propostas.
- iv. *Elaboração da Documentação da MePPLa Tool:* Nesta etapa foi definida a evolução a ser feita em *MePPLa Tool* e foi elaborada a documentação da ferramenta, tratando aspectos relacionados à análise e especificação de requisitos e ao projeto do sistema.
- v. *Implementação e Testes:* Nesta atividade a evolução da ferramenta foi implementada e testada.
- vi. *Elaboração da Monografia:* Nesta etapa, foi realizada a escrita desta monografia.

## Capítulo 1 - Introdução

### 1.4 Organização do Texto

Neste capítulo foi apresentada uma introdução com as ideias principais deste trabalho, contendo uma contextualização, breve descrição do domínio do problema, os objetivos a serem alcançados e o histórico de desenvolvimento. Além deste capítulo, a monografia possui outros quatro, a saber:

*Capítulo 2 – Fundamentação Teórica:* Apresenta os principais aspectos teóricos relacionados a medição de software, controle estatístico de processos e linguagem de padrões. A Linguagem de Padrões para Planejamento de Medição de Software (MePPLa) e abordagem SAMPPLa, usada para apoiar a criação de MePPLa, também são introduzidas neste capítulo.

*Capítulo 3 – Evolução de MePPLa:* apresenta os resultados da evolução de MePPLa, que inclui a seleção de um novo processo, identificação dos padrões e organização desses em MePPLa.

*Capítulo 4 – Evolução de MePPLa Tool:* Este capítulo apresenta os principais resultados produzidos ao longo da evolução de *MePPLa Tool*, incluindo os casos de uso e modelos de classe da ferramenta, bem como a arquitetura e algumas telas resultantes da evolução de *MePPLa Tool*.

*Capítulo 5 – Considerações Finais:* São realizadas as considerações finais deste trabalho, sendo apresentadas suas principais contribuições e perspectivas de trabalhos futuros.

## Capítulo 2

# Fundamentação Teórica

---

*Este capítulo apresenta os principais aspectos teóricos que fundamentam este trabalho. Ele está organizado em cinco seções, a saber: A Seção 2.1, aborda Medição de Software e Controle Estatístico de Processos (CEP). A Seção 2.2 trata aspectos relacionados a Linguagem de Padrões. Na Seção 2.3, MePPLa é apresentada. Por fim, na Seção 2.4 é apresentada a abordagem SAMPPLa.*

### 2.1 Medição de Software e Controle Estatístico de Processos

Medição de Software é uma avaliação quantitativa de qualquer aspecto dos processos e produtos da Engenharia de Software, que permite seu melhor entendimento, dessa forma auxiliando o planejamento, controle e melhoria do que se produz e de como é produzido. São utilizadas medidas, para realizar essa avaliação quantitativa (BASS *et al*, 1999).

De acordo com Solingen & Berghout (1999), Medição de Software é um processo contínuo de definir, coletar e analisar dados relacionados aos processos e produtos de software, a fim de entendê-los, controlá-los e fornecer informações relevantes para melhorá-los.

O elemento básico da medição, que propicia a análise quantitativa, são as medidas. Elas caracterizam, em termos quantitativos, alguma propriedade de um objeto da Engenharia de Software (BASILI; ROMBACH, 1994). A norma ISO/IEC 15939:2001 (ISO/IEC, 2001) define que *medida* é a variável à qual um valor é atribuído com o resultado de uma medição.

Fenton e Neil (1999) e Bass *et al* (1999) (*apud* Barcellos, 2009a) registram que a prática da *medição de software* se iniciou na década de 1970, embora, por um bom tempo, seus objetivos não fossem claros e seus resultados sem expressão de utilidade. Somente nos anos 1990 que, impulsionados por algumas aplicações bem-sucedidas, desenvolveu-se modelos para o procedimento de medição baseados na melhoria de processos e nos princípios da qualidade total, fornecendo diretrizes e infraestrutura básica para definir, coletar, validar e analisar medidas.

No passado, diversas organizações de software não reconheciam a importância da medição. Atualmente, ela é considerada uma atividade fundamental entre as práticas de Engenharia de Software, evidenciada por sua inclusão nos requisitos dos níveis iniciais dos modelos que tratam da melhoria de processos em níveis (BARCELLOS, 2010).

O principal objetivo da medição de software é prover informação quantitativa para apoiar a tomada de decisões, ou seja, as organizações de software definem medidas e coletam dados que, após analisados e selecionados, fornecem informações que sejam úteis à tomada

## Capítulo 2 - Fundamentação Teórica

de decisões, auxiliando o estudo e análise dos objetivos fixados e à identificação das ações corretivas e de melhoria (BARCELLOS, 2010).

Para realizar medição de software, inicialmente, uma organização deve planejá-la. Baseada em seus objetivos, a organização deve definir quais entidades (processos, produtos, etc.) devem ser consideradas para a medição de software e quais propriedades (tamanho, custo, tempo, etc.) devem ser mensuradas. A organização deve definir também quais medidas devem ser usadas para quantificar essas propriedades. Para cada medida, uma definição operacional de medida deve ser especificada, indicando como a medida deve ser coletada e analisada. Uma vez planejada, pode-se executar a medição. A execução da medição envolve coletar dados para as medidas definidas, armazená-los e analisá-los. A análise de dados fornece informação para a tomada de decisão, apoiando a identificação de ações corretivas. Finalmente, o processo de medição e seus produtos devem ser avaliados visando identificar potenciais melhorias (BARCELLOS; FALBO; ROCHA, 2010).

Para apoiar o planejamento de medição, a literatura oferece algumas propostas de abordagens, sendo o GQM (*Goal-Question-Metric*) uma das mais conhecidas. O GQM é uma abordagem sistemática para a adaptação e integração de objetivos com processos, produtos e perspectivas de interesse de qualidade, baseado nas necessidades específicas dos projetos e da organização (BASILI; ROMBACH, CALDIERA, 1994). Simplificando, o GQM afirma que a partir de objetivos é possível identificar necessidades de informação que podem ser atendidas por medidas. Desta forma, a partir de seus objetivos, as organizações podem derivar necessidades de informação e definir medidas capazes de atendê-las.

A abordagem GQM define medidas a partir de uma perspectiva *top-down*, analisa e interpreta os dados de medição numa perspectiva *bottom-up*. Tendo uma definição das medidas com um propósito definido, a informação alimentada pelas referidas medidas será interpretada e analisada, considerando esse propósito, para concluir se esse objetivo foi ou não alcançado (SOLINGEN; BERGOUT, 1999). A Figura 2.1 ilustra a estrutura hierárquica do GQM.

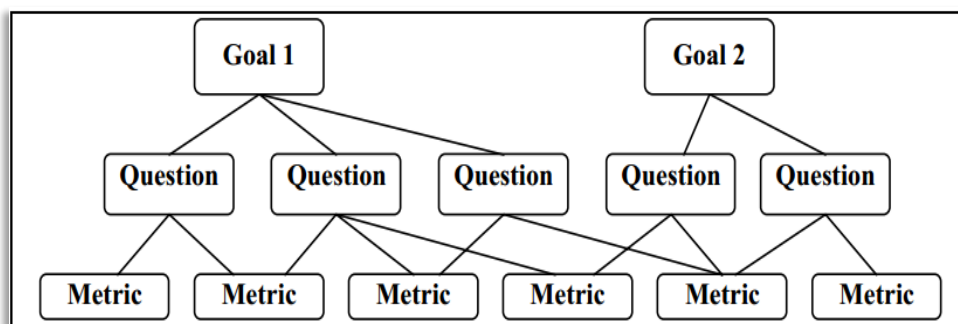


Figura 2.1: Estrutura hierárquica do modelo GQM (BASILI; CALDIERA; ROMBACH, 1994).

Dependendo do nível de maturidade da organização, a medição é realizada de diferentes formas. Nos níveis iniciais dos modelos de maturidade como CMMI e MR-MPS-SW, ela consiste, basicamente, na coleta de dados dos projetos e comparação desses dados com os valores planejados correspondentes. Na alta maturidade (níveis 4 e 5 do CMMI e B e A do MR-MPS-BR), a medição tradicional não é suficiente, sendo necessário realizar o controle estatístico de processos para conhecer seu comportamento, determinar seu desempenho em execuções anteriores e prever seu desempenho em execuções futuras, verificando se são capazes de alcançarem os objetivos estabelecidos (FLORAC; CARLETON, 1999).

Visando implementar um processo de melhoria contínua em linha de produção na área de manufatura, foi desenvolvido o Controle Estatístico de Processos (CEP), que envolve o uso de ferramentas estatísticas e técnicas de resolução de problemas com o objetivo de detectar padrões de variação no processo de produção, para garantir que os padrões de qualidade, estabelecidos para os produtos, fossem alcançados (WHEELER; CHAMBERS, 1992 e WHEELER, 1995). O CEP é utilizado para determinar se um processo está sob controle, sob o ponto de vista estatístico (LANTZY, 1992; BARCELLOS, 2009b).

O CEP também é um método de gestão de processos, através de análise estatística, que inclui definir, medir, controlar e melhorar os processos (FLORAC; CARLETON, 1999), por meio do qual é possível entender variações de processos e melhorar o seu desempenho (WHEELER, 1995). É conhecido por ser uma poderosa ferramenta para melhoria de processos, aumentando, desse modo, a qualidade e a produtividade dos processos (FLORAC; CARLETON, 1999).

Dois conceitos são importantes, no contexto do controle estatístico de processos: *estabilidade* e *capacidade*. Pode-se considerar que um processo é *estável* se o seu comportamento

## Capítulo 2 - Fundamentação Teórica

for previsível e repetível. Pode ser considerado *capaz* se, além de ser estável, o processo conseguir alcançar os objetivos propostos (BARCELLOS, 2009b).

No que toca a *estabilidade*, importa ressaltar ser intrínseco aos processos apresentar variações em seu comportamento. Logo, um processo estável não é aquele isento de variações, mas sim o que apresenta variações aceitáveis, ou seja, as que ocorrem dentro de limites previsíveis, caracterizando a repetitividade de seu comportamento (BARCELLOS, 2009b).

Segundo Shewhart (1980), há *causas comuns* e *causas especiais* que se deve considerar no estudo das variações do comportamento dos processos, para concluir se o processo é ou não estável.

As variações aceitáveis são provocadas pelas *causas comuns*, sendo aquelas pertencentes ao próprio processo. Um exemplo de *causa comum*, nos processos de software, pode ser a diferença de produtividade entre os membros de uma equipe relativamente homogênea. Cada membro executa um certo processo com uma determinada produtividade. Assim, as variações causadas no comportamento desse processo são previsíveis, em razão dessa diferença de produtividade entre seus executores (SHEWHART, 1980).

Já as denominadas *causas especiais*, provocam desvios que excedem os limites de variação aceitável para o comportamento do processo, revelando um processo instável. Por exemplo, no contexto de processos de software, a inclusão de um membro inexperiente na equipe pode alterar o comportamento do processo para o além do esperado, caracterizando uma causa especial (SHEWHART, 1980; BARCELLOS, 2009a).

Estando todas as variações aceitáveis, no comportamento de um processo, ou seja, não havendo causas especiais, diz-se que o processo está *sob controle estatístico*. Assim, um processo sob controle estatístico é um processo estável (FLORAC e CARLETON, 1999).

Melhorar a capacidade de um processo significa diminuir os limites de variação considerados aceitáveis para seu comportamento, ou seja, consiste em tratar as *causas comuns*. Caso o processo não seja *capaz*, deverá ser alterado por meio da realização de ações de melhoria que busquem o alcance da *capacidade* desejada (BARCELLOS, 2009b).

Muitas organizações conhecidas pela excelência de seus programas de qualidade estabelecem limites de variação bem “estritos” para os processos. Uma vez alcançados, a organização obtém processos *estáveis*, *capazes* e com um grau de variabilidade consideravelmente baixo. Tornando-se *capaz* e *estável*, um processo pode ser melhorado continuamente, e novos objetivos podem ser traçados (BARCELLOS, 2009b). Quanto



## Capítulo 2 - Fundamentação Teórica

melhor o limite de variação dos processos, menores serão as chances de desvios entre os valores planejados e realizados nos projetos, ou seja, maior é a aderência dos projetos aos cronogramas, orçamentos e demais planejamentos estabelecidos. Assim, melhor será a gerência dos projetos e processos (BARCELLOS, 2009b).

Enfim, se o processo for realizado de forma consistente, se as medidas corretas forem selecionadas e um mecanismo de coleta de dados de confiança for estabelecido, pode-se obter os benefícios com a aplicação das técnicas do CEP (SARGUT; DEMIRORS, 2006; ABUBAKAR; JAWAWI, 2013).

### 2.2 Linguagem de Padrões

Padrões são veículos para o encapsulamento de conhecimento, que permitem capturar o que deve ser feito para resolver um dado problema. O uso de padrões raramente ocorre de maneira isolada, ou seja, padrões, tipicamente, são usados de forma combinada para resolver problemas, considerando algumas relações existentes entre eles. Desta forma, padrões podem ser organizados em uma *Linguagem de Padrões* (LP) (DEUTSCH *et al.*, 2004; BUSCHMANN, 2007).

Uma Linguagem de Padrões de Software é uma rede de padrões inter-relacionados, que define um processo para resolução sistemática de problemas relacionados ao desenvolvimento de software. Para utilizar um padrão, faz-se necessário reconhecer a oportunidade de aplicá-lo, ou seja, é preciso entender e reconhecer o contexto em que o problema recorrente tratado pelo padrão ocorre e identificar se o problema que se deseja tratar é uma instância do problema recorrente. Para isso, padrões devem conter descrições para o problema, a solução e o contexto em que ocorrer (DEUTSCH *et al.*, 2004).

Notações visuais podem ser utilizadas para representar graficamente Linguagens de Padrões. O propósito da adoção de notações visuais é dar uma visão geral dos padrões e seus relacionamentos, para prover o entendimento holístico da LP e auxiliar na seleção dos padrões (QUIRINO, 2016).

Embora originalmente propostas na área de arquitetura e já há algum tempo venham sendo utilizadas na área de software, Linguagens de Padrões também podem ser usadas em outras áreas, como por exemplo, em Engenharia de Ontologias (QUIRINO, 2016).

Nesse contexto, Quirino (2016) propôs uma notação visual para representação de Linguagem de Padrões Ontológicos chamada OPL-ML (*Ontology Pattern Language Modeling Language*). Tomaz (2017) propõe o uso de OPL-ML para representar as linguagens de padrões

## Capítulo 2 - Fundamentação Teórica

para planejamento de medição, visto que, embora a notação tenha sido proposta para padrões ontológicos, ela também pode ser aplicada em outras áreas.

### 2.3 Measurement Planning Pattern Language (MePPLa)

Considerando a dificuldade da elaboração de planos de medição adequados ao CEP de software, Tomaz (2017) propôs MePPLa, uma linguagem de padrões para apoiar o planejamento de medição visando ao CEP.

MePPLa tem como propósito apoiar o planejamento de medição visando ao CEP para atendimento aos requisitos dos níveis 4 e 5 do CMMI e dos níveis B e A do MR-MPS-SW. O contexto de aplicação abrange organizações avaliadas CMMI nível 3 ou MR-MPS-SW nível C e que desejam implementar as práticas do CEP visando à alta maturidade (TOMAZ, 2017).

Os padrões de MePPLa foram identificados a partir de estudos da literatura (BRITO; BARCELLOS, 2016) e da prática (TOMAZ, 2017), que investigaram medidas, objetivos e processos utilizadas no CEP. MePPLa possui 28 padrões, sendo 12 relacionados ao processo de Gerência de Projetos, 6 relacionados ao processo de Codificação e 10 relacionados ao processo de Testes (TOMAZ, 2017).

Os padrões fornecidos por MePPLa são baseados no formado GQM – *Goal-Question-Metric* (BASILI *et al.*, 1994), dessa forma, um padrão de planejamento de medição inclui objetivo de medição, questões que indicam necessidades de informação que devem ser atendidas para que seja possível monitorar o objetivo de medição e medidas (com suas definições operacionais) que atendem as necessidades de informação (TOMAZ, 2017).

Para cada padrão identificado, foi estabelecida uma descrição detalhada. A Tabela 2.1 apresenta, como exemplo, a descrição detalhada do padrão Produtividade na Preparação dos Testes. Na tabela, os textos em itálico e entre << >> são orientações que devem ser seguidas para estabelecer a definição operacional da medida quando o padrão for aplicado. As definições operacionais para as medidas dos padrões de MePPLa foram estabelecidas considerando-se requisitos para adequação de medidas ao CEP discutidos em (BARCELLOS *et al.*, 2013).

<b>Produtividade na Preparação dos Testes</b>	
<b>Nome:</b> Produtividade na Preparação dos Testes	
<b>Processo/Subprocesso:</b> Testes /Preparação dos Testes	
<b>Objetivo:</b> Melhorar a produtividade na preparação dos testes	
<b>Necessidades de Informação:</b> Qual é a produtividade na preparação dos testes?	
<b>Medidas:</b> Produtividade na preparação do teste, Esforço de Preparação dos Testes, Número de Casos de Teste Elaborados.	
<b>Definição Operacional das Medidas:</b>	
<b>Medida Composta</b>	Produtividade na Preparação do Teste
<b>Mnemônico</b>	PPT
<b>Descrição</b>	Medida utilizada para quantificar a produtividade na preparação dos testes, que é dada pela razão entre o esforço gasto para preparar o teste e o número de casos de teste elaborados.
<b>Entidade</b>	Subprocesso Preparação dos Testes
<b>Propriedade</b>	Produtividade na preparação do teste
<b>Escala</b>	Números reais positivos com precisão de duas casas decimais
<b>Unidade de Medida</b>	Casos de teste/homem-hora
<b>Fórmula</b>	$PPT = (EPT / NCTE)$
<b>Procedimento de Medição</b>	Calcular a produtividade na preparação dos testes utilizando a fórmula de cálculo da medida, considerando a mesma preparação dos testes para ambas medidas base da fórmula.
<b>Periodicidade da Medição</b>	A medição deve ser executada a cada preparação do teste.
<b>Responsável pela Medição</b>	<<Indica o papel do responsável por coletar os dados para a medida. É recomendado que o responsável pela medição seja o provedor do dado. >>
<b>Momento da Medição</b>	No final da preparação do teste
<b>Procedimento de Análise de Medição</b>	<< Procedimento de análise de medição para uso do CEP no contexto de modelos de maturidade de processos>>
<b>Periodicidade da Análise</b>	<<Indica a periodicidade baseada no período de tempo (ex. quinzenal) ou na quantidade de novos dados coletados (ex. a cada 4 novos valores coletados). Periodicidades diferentes podem ser estabelecidas para analisar os dados no projeto e no contexto organizacional.>>
<b>Responsável pela Análise</b>	<<Indica o papel do responsável por analisar os dados coletados para a medida >>
<b>Momento da Análise</b>	<<Indica o momento em que a análise dos dados deve ser realizada. O momento da análise deve ser uma atividade do processo do projeto (para analisar o dado no contexto do projeto) ou do processo organizacional (para analisar o dado no contexto organizacional).>>
<b>Medida Base 1</b>	Esforço de Preparação dos Testes
<b>Mnemônico</b>	EPT
<b>Descrição</b>	Medida que quantifica o esforço gasto na preparação dos testes.
<b>Entidade</b>	Subprocesso Preparação dos Testes
<b>Propriedade</b>	Esforço
<b>Escala</b>	Números reais positivos com precisão de duas casas decimais
<b>Unidade de Medida</b>	Homem-hora
<b>Fórmula</b>	-
<b>Procedimento de Medição</b>	Obter o esforço despendido na preparação dos testes.
<b>Medida Base 2</b>	Número de Casos de Teste Elaborados
<b>Mnemônico</b>	NCTE
<b>Descrição</b>	Medida que quantifica o número de casos de teste elaborados na preparação dos testes.
<b>Entidade</b>	Subprocesso Preparação dos Testes
<b>Propriedade</b>	Desempenho
<b>Escala</b>	Números reais positivos
<b>Unidade de Medida</b>	-
<b>Fórmula</b>	-
<b>Padrões Relacionados:</b> Eficiência na Preparação dos Testes.	

Tabela 2.1 - Descrição do Padrão Produtividade na Preparação dos Testes.

## Capítulo 2 - Fundamentação Teórica

Como citado anteriormente, MePPLa é representada visualmente através da notação OPL-ML (2016). Desta forma, MePPLa é composta por dois tipos de modelos, o *modelo estrutural*, o qual apresenta os padrões que compõem a linguagem e as relações estruturais entre eles, e o modelo *comportamental*, que descreve o processo de aplicação dos padrões.

O *modelo estrutural* fornece informações sobre as relações estruturais (ex., dependência e composição), que são especialmente úteis durante a análise dos dados coletados para as medidas, pois revelam medidas correlatas e objetivos que impactam em outros. Esse modelo pode ser útil no planejamento de medição, uma vez que pode auxiliar na identificação de quais padrões podem ser selecionados para uma melhor análise do alcance aos objetivos e de causas que possam estar neles interferindo. O modelo estrutural também é útil para a elaboração do modelo comportamental da LP, uma vez que indica as dependências que devem ser consideradas no fluxo que guia a seleção dos padrões a serem aplicados (BRITO, BARCELLOS e SANTOS, 2017). Em MePPLa são utilizados dois tipos de relações: dependência, quando um padrão requer a aplicação de outro, e correlação, para indicar padrões cuja medidas são correlatas, mas os padrões não são necessariamente dependentes. Os padrões de MePPLa são agrupados considerando os processos e subprocessos relacionados a eles. A Figura 2.3 apresenta o modelo estrutural relacionado ao processo de Teste.

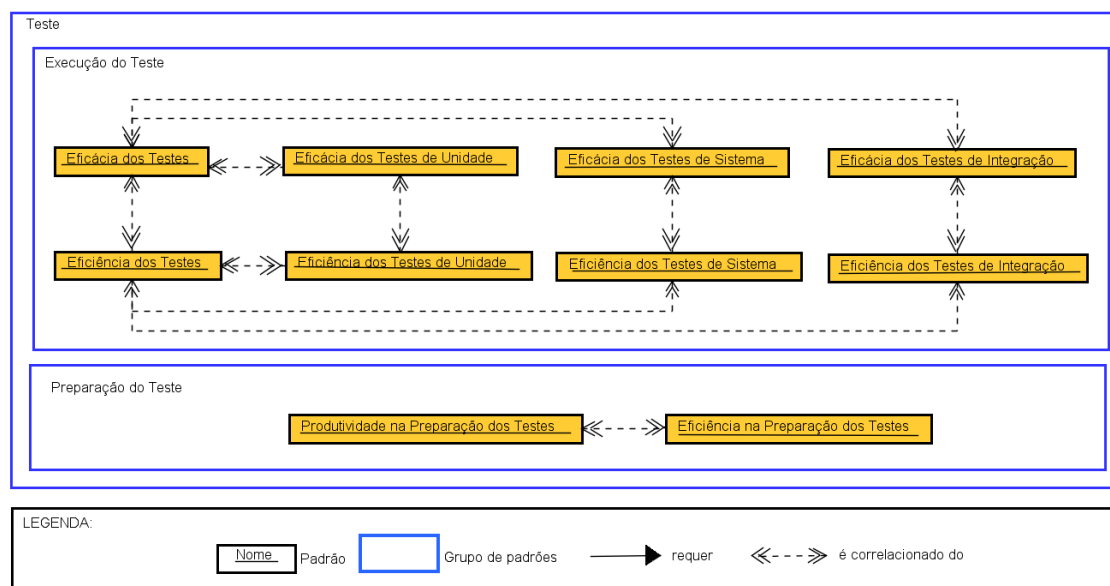


Figura 2.3 - Modelo Estrutural do grupo de padrões Testes (TOMAZ, 2017).

O *modelo comportamental* ou *modelo de processo* possui dois formatos, que são o formato caixa-preta, que fornece a visão geral da linguagem, e o formato detalhado, que fornece a

## Capítulo 2 - Fundamentação Teórica

visão comportamental detalhada da linguagem de padrões, contendo os fluxos que guiam a aplicação dos padrões. Ambos os formatos dos modelos devem ser entendidos como um processo a ser seguido passo a passo, de um ponto de entrada até um ponto final. A Figura 2.4 apresenta o formato caixa-preta do modelo comportamental de MePPLa. No formato caixa-preta é possível visualizar os processos tratados em MePPLa, uma vez que cada processo representa um grupo de padrões. Como o próprio nome sugere, nesse formato não é possível visualizar os padrões e fluxos existentes dentro de cada grupo.

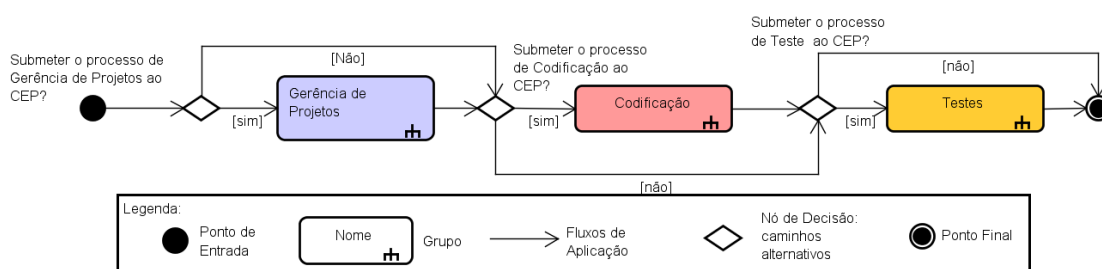


Figura 2.4 – Formato caixa-preta do modelo comportamental (TOMAZ, 2017).

Para cada grupo de padrões há um modelo comportamental detalhado apresentando o conteúdo interno ao grupo, ou seja, seus padrões e fluxos entre eles. A Figura 2.5 apresenta o modelo comportamental detalhado do processo de Teste.

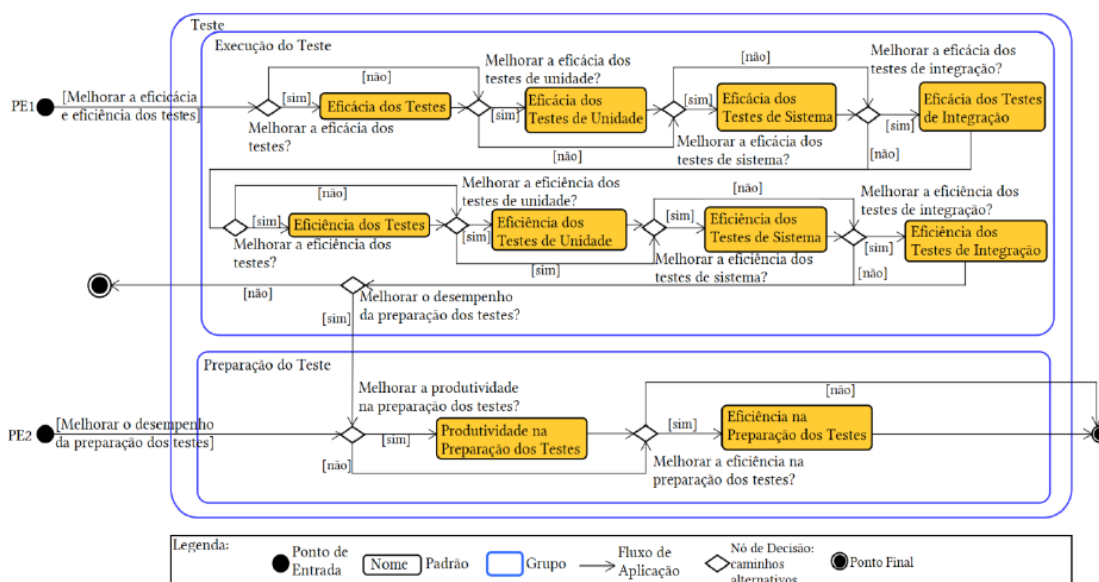


Figura 2.5 – Modelo Comportamental do grupo de padrões Teste (TOMAZ, 2017).

Conforme Brito *et al.* (2017) apresenta, os modelos comportamentais detalhados são consistentes com o modelo estrutural de MePPLa. Assim, os padrões são agrupados de acordo com os processos e subprocessos aos quais se relacionam. Entre os padrões há fluxos

## Capítulo 2 - Fundamentação Teórica

para guiar o usuário na seleção e aplicação dos padrões, sendo respeitadas as relações entre os padrões definidas no modelo estrutural. Para navegar no modelo, o usuário deve escolher um ponto de entrada de acordo com o objetivo a ser considerado no plano de medição. Por exemplo, na Figura 2.5, se o objetivo *Melhorar a Eficácia e Eficiência dos Testes* é pertinente, o usuário deve iniciar a navegação em EP1. Caso esse objetivo não seja pertinente e *Melhorar o Desempenho da Preparação dos Testes* for, então o usuário deve iniciar a navegação em EP2. Note que se ambos forem pertinentes, basta o usuário iniciar a navegação em EP1, pois ele será guiado até os padrões relacionados EP2. A partir do ponto de entrada, o usuário deve seguir os fluxos e selecionar os padrões de acordo com os objetivos que deseja tratar no plano de medição. A seleção de um padrão significa que sua descrição (como a apresentada na Tabela 2.1) será incluída no plano de medição.

### 2.4 Systematic Approach for creating Measurement Planning Pattern Languages (SAMPPLa)

SAMPPLa é uma abordagem que foi definida por Tomaz (2017) para criar e evoluir MePPLa. É composta pelas atividades *Desenvolver Fonte para Extração dos Padrões* e *Desenvolver Linguagem de Padrões*.

A atividade *Desenvolver Fonte para Extração de Padrões* tem por objetivo obter um conjunto de processos, objetivos de medição e medidas a partir dos quais podem ser identificados padrões para planejamento de medição de software visando o CEP. A atividade *Desenvolver Linguagens de Padrões*, por sua vez, envolve a extração dos padrões de planejamento de medição para o CEP do conjunto produzido na etapa anterior e o desenvolvimento da linguagem de padrões propriamente dita (TOMAZ, 2017). Estas atividades são compostas por outras, conforme apresentado a seguir.

#### 2.4.1 Desenvolver Fonte para Extração dos Padrões

Esta atividade é responsável pela obtenção de um conjunto de processos, objetivos e medidas, a partir do qual podem ser identificados padrões para planejamento de medição visando ao CEP. A Figura 2.6 apresenta o detalhamento desta atividade.

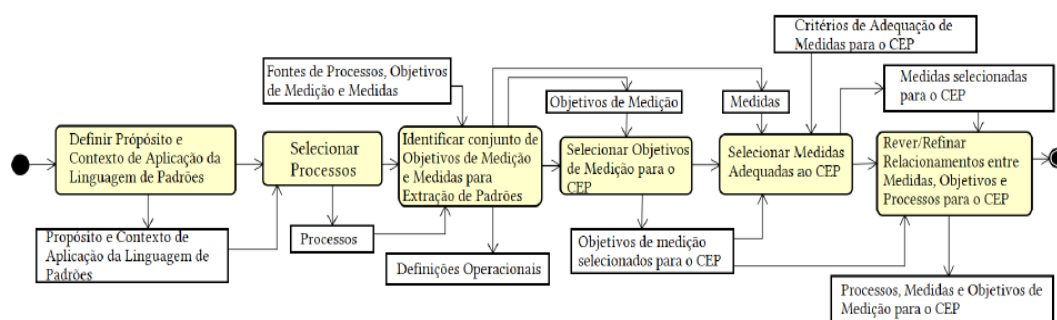


Figura 2.6 - Detalhamento de Desenvolver Fontes para Extração de Padrões (TOMAZ, 2017).

A seguir é apresentada a descrição das atividades que compõem **Desenvolver Fontes para Extração de Padrões**:

a) **Definir Propósito e Contexto de Aplicação da Linguagem:** Consiste em definir para que a LP será criada e onde ela será aplicada.

b) **Selecionar Processos:** Consiste em selecionar processos que possam ser submetidos ao CEP e que serão tratados na LP. Essa decisão depende diretamente do propósito da LP e seu contexto de uso. Por exemplo, se a LP deve ser usada para apoiar o CEP relacionados a defeitos, então devem ser selecionados processos relacionados a defeitos, tais como Teste e Verificação.

c) **Identificar Conjunto de Objetivos de Medição e Medidas para Extração dos Padrões:** Consiste em identificar o conjunto de informações a partir das quais os padrões serão extraídos. Esse conjunto pode ser obtido, por exemplo, por meio de repositórios de medição de organizações, consultas à literatura ou junto a profissionais com experiência em medição de software.

d) **Selecionar Objetivos de Medição para o CEP:** Inclui a análise dos objetivos do conjunto de informações previamente identificado e seleção dos utilizáveis no contexto do CEP e que estejam diretamente relacionados aos processos previamente identificados.

e) **Selecionar Medidas Adequadas ao CEP:** Para cada processo/objetivo de medição selecionado anteriormente, devem ser selecionadas medidas capazes de medir aspectos do desempenho do processo e fornecer informações que permitam monitorar o alcance ao objetivo de medição. Para isso devem ser considerados os requisitos definidos por Rocha *et al.* (2012).

f) **Rever/Refinar Processos, Objetivos de Medição e Medidas para o CEP:** Consiste em refinar/ajustar os elementos selecionados anteriormente, para garantir

## Capítulo 2 - Fundamentação Teórica

que sirvam de base para extração de padrões. Neste momento, pode-se, por exemplo, decompor objetivos para tornar as relações com as medidas mais diretas. Da mesma forma, processos podem ser decompostos em subprocessos, para tornar mais diretas as relações destes com os objetivos e medidas. Medidas também podem ser ajustadas para se adequarem melhor aos processos e objetivos considerados. Medidas mais específicas podem ser generalizadas, ou ainda, pode-se definir medidas mais específicas a partir de medidas mais gerais. Por fim, além de ajustes nos elementos propriamente ditos, devem ser revistas as relações entre eles, pois algumas relações pertinentes podem não ter sido capturadas até então e, além disso, os ajustes realizados nos elementos podem implicar em novas relações.

### 2.4.2 Desenvolver Linguagem de Padrões

Uma vez definida a fonte de onde os padrões poderão ser extraídos, deve-se Desenvolver Linguagem de Padrões. A Figura 2.7 apresenta o detalhamento da atividade.

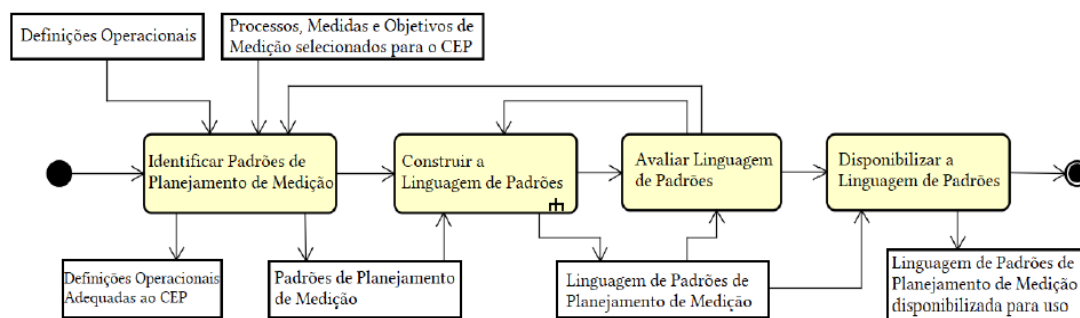


Figura 2.7 - Detalhamento do Desenvolver Linguagem de Padrões (TOMAZ, 2017)

A descrição das atividades de **Desenvolver Linguagem de Padrões** é dada a seguir:

**a) Identificar Padrões de Planejamento de Medição:** consiste em analisar o conjunto de processos, objetivos de medição e medidas e medidas identificado anteriormente para que sejam identificados padrões de planejamento de medição. Cada padrão está relacionado a um ou mais processos e segue o formato GQM (BASILI *et al*, 1994). Os padrões devem emergir do conjunto de processos, objetivos de medição e medidas. Cada medida que compõe os padrões identificados deve ter uma definição operacional adequada para o CEP.

**b) Construir Linguagem de Padrões:** a construção dos modelos estrutural e comportamental, relativos, respectivamente, às perspectivas estrutural e comportamental da LP devem ser construídos usando a notação visual definida em Quirino (2016). Esta atividade é decomposta em:



c) **Avaliar a Linguagem de Padrões:** a LP deve ser avaliada e se necessário ajustes devem ser feitos retornando-se às atividades anteriores para melhor atender ao propósito definido.

d) **Disponibilizar a Linguagem de Padrões:** a LP deve ser disponibilizada (por exemplo, por meio de uma especificação textual ou apoio computacional).

## Capítulo 3

# Evolução de MePPLa

---

*Este capítulo apresenta a evolução feita em MePPLa, a qual foi realizada aplicando-se SAMPPLa. Seguindo as atividades de SAMMPLa, a Seção 3.1 apresenta os principais resultados produzidos durante o desenvolvimento da fonte para extração de padrões e a Seção 3.2 trata dos resultados produzidos durante a evolução propriamente dita da linguagem de padrões.*

### 3.1 Desenvolver Fonte Para Extração de Padrões

Conforme apresentado na Seção 2.4, a primeira atividade de *SAMPPLa* é responsável pela obtenção de um conjunto de processos, objetivos de medição e medidas (com suas definições operacionais) a partir do qual poderão ser extraídos os padrões. Para isso, foram realizadas as seguintes atividades.

#### 3.1.1 Definir Propósito e Contexto de Aplicação da Linguagem de Padrões

Por ser uma evolução de MePPLa, o propósito e contexto de aplicação da LP já estão definidos. Como apresentado na Seção 2.3, MePPLa tem como propósito apoiar o planejamento de medição visando ao CEP para atendimento aos requisitos dos níveis 4 e 5 do CMMI e dos níveis B e A do MR-MPS-SW. O contexto de aplicação abrange organizações avaliadas CMMI nível 3 ou MR-MPS-SW nível C e que desejam implementar as práticas do CEP visando à alta maturidade.

#### 3.1.2 Selecionar Processos

Em (BRITO, BARCELLOS e SANTOS. 2018) são apresentados resultados refinados de um mapeamento sistemático que foi usado como base para padrões de MePPLa. Com base nesses resultados, selecionou-se o processo *Revisão* para ser tratado na evolução de MePPLa, uma vez que esse foi o processo mais citado na literatura, tendo sido usado no CEP em 30 das 50 publicações analisadas (60%).

#### 3.1.3 Identificar Conjunto de Objetivos de Medição e Medidas para Extração dos Padrões

Foi selecionado o conjunto de objetivos de medição e medidas associados ao processo *Revisão* em (BRITO, BARCELLOS e SANTOS, 2018), o qual é composto por 34 medidas e 22 objetivos de medição.

### 3.1.4 Selecionar Objetivos de Medição para o CEP

Nesta atividade, foram selecionados os objetivos com relação direta com o processo Revisão. Portanto, objetivos que apesar de relacionados ao processo Revisão na literatura não tenham uma relação direta com ela (p.ex., *Avaliar a eficiência da codificação*) foram descartados.

### 3.1.5 Selecionar Medidas Adequadas para o CEP

Nesta atividade, para cada objetivo selecionado na atividade anterior, foram selecionadas medidas consideradas adequadas para o CEP.

Assim como foi feito na seleção dos objetivos, foram selecionadas apenas as medidas com relação direta com o processo de Revisão e com o objetivo de medição relacionado a ela. Por exemplo, a medida *Eficiência na resolução de defeitos* foi excluída uma vez que o processo de Revisão é responsável pela detecção de defeitos e não por sua resolução.

SAMPPLa ressalta que é possível que haja medidas que não possuam objetivos de medição relacionados a elas. Caso a medida seja considerada adequada, pode-se associá-la a um dos objetivos de medição selecionados anteriormente. Por exemplo, a medida *Eficiência da Inspeção*, que inicialmente não possuía um objetivo de medição associado, foi associada ao objetivo *Avaliar a eficácia da inspeção*. Da mesma forma, a medida *Produtividade da inspeção*, foi associada ao objetivo *Melhorar a produtividade da inspeção*.

### 3.1.6 Rever e Refinar Processos, Objetivos de Medição e Medidas para o CEP

Nesta atividade, os objetivos de medição e medidas foram ajustados para garantir que seriam adequados para extrair os padrões. Por exemplo, alguns objetivos mais gerais foram especializados em objetivos mais específicos, como é o caso do objetivo *Melhorar a eficiência do processo de software*, que foi especializado em *Melhorar eficiência do processo de revisão*. Da mesma forma, considerando-se as medidas a ele associadas, o objetivo *Melhorar o processo de revisão* foi especializado em *Melhorar o desempenho do processo preparação da revisão* e o objetivo *Entender o desempenho do processo revisão* foi especializado em *Melhorar o desempenho do processo de revisão*. Durante esta atividade, o processo de Revisão foi dividido nos subprocessos Execução da Revisão e Preparação da Revisão.

## 3.2 Desenvolver Linguagem de Padrões

A parte da LP referente ao processo de Revisão foi desenvolvida nesta atividade, conforme descrito a seguir.

### 3.2.1 Identificar Padrões de Planejamento de Medição

Nesta atividade, os padrões relacionados ao processo de Revisão foram identificados e extraídos do conjunto de objetivos de medição e medidas obtidos anteriormente.

Conforme já dito, os padrões de MePPLa seguem o formato GQM (BASILI *et al.*, 1994), ou seja, possuem um objetivo de medição, questões que indicam necessidades de informação e medidas (com definição operacional) que atendem a essas necessidades de informação. Assim, durante a identificação dos padrões, foram identificados e extraídos padrões GQM do conjunto de objetivos de medição e medidas. Para a extração dos padrões, foram adotadas algumas práticas já adotadas durante a criação da primeira versão de MePPLa. Por exemplo, objetivos muito gerais (como *Garbar a competição do mercado*) foram desconsiderados, por não serem adequados para o CEP. Medidas muito específicas (p.ex., *porcentagem de defeitos causados por falha lógica*) também foram descartadas, devido ao seu baixo potencial de reuso.

Durante a extração dos padrões, notou-se que a maioria dos objetivos de medição estavam relacionados a eficácia e a eficiência do processo Revisão. Com relação às medidas, várias medidas estavam relacionadas a revisão por pares, revisão de código e inspeção. Assim, decidiu-se criar padrões relacionados a eficiência e eficácia desses diferentes tipos de revisão.

A Tabela 3.2 apresenta os padrões definidos. A descrição detalhada dos padrões é apresentada no Apêndice A.

**Tabela 3.2 – Visão Geral dos Padrões Identificados**

<b>Nome do Padrão</b>	<b>Processo/ Subprocesso</b>	<b>Objetivo</b>	<b>Necessidade de Informação</b>	<b>Medidas Relacionadas</b>
Eficácia da Revisão	Execução da Revisão	Melhorar Eficácia da Revisão	Qual é a eficácia do processo de revisão?	Eficácia da Revisão (número de defeitos detectados em revisões / número total de defeitos)
Eficácia da Inspeção	Execução da Revisão	Melhorar Eficácia da Inspeção	Qual é a eficácia do processo de inspeção?	Eficácia da Inspeção (número de defeitos detectados em inspeções / número total de defeitos)
Eficácia da Revisão por Pares	Execução da Revisão	Melhorar Eficácia da Revisão por Pares	Qual é a eficácia do processo de revisão por pares?	Eficácia da Revisão por Pares (número de defeitos detectados em revisões por pares / número total de defeitos)

### Capítulo 3 - Evolução de MePPLa

Eficácia da Revisão de Código	Execução da Revisão	Melhorar Eficácia da Revisão de Código	Qual é a eficácia do processo de revisão de código?	Eficácia da Revisão de Código (número de defeitos detectados em revisões de código / número total de defeitos)
Eficiência da Revisão	Execução da Revisão	Melhorar Desempenho da Revisão	Qual é a eficiência do processo de revisão?	Eficiência da Revisão (número de defeitos detectados em revisões / esforço de detecção de defeitos)
Eficiência da Inspeção	Execução da Revisão	Melhorar Desempenho da Inspeção	Qual é a eficiência do processo de detecção de defeitos?	Eficiência da Inspeção (número de defeitos detectados em inspeções / esforço de detecção de defeitos)
Eficiência da Revisão por Pares	Execução da Revisão	Melhorar Desempenho da Revisão por Pares	Qual é a eficiência do processo de revisão por pares?	Eficiência da Revisão por Pares (número de defeitos detectados em revisões por pares / esforço de detecção de defeitos)
Eficiência da Revisão de Código	Execução da Revisão	Melhorar Desempenho da Revisão de Código	Qual é a eficiência do processo de revisão de código?	Eficiência da Revisão de Código (número de defeitos detectados em revisões de código / esforço de detecção de defeitos)
Taxa de Preparação da Revisão	Preparação da Revisão	Melhorar Desempenho da Preparação da Revisão	Qual é o desempenho na preparação da revisão?	Taxa de Preparação da Revisão (tamanho do produto a ser revisado / tempo gasto preparando a revisão)

#### 3.2.2 Construir Linguagem de Padrões

Nesta atividade, os modelos estrutural e comportamental organizando os padrões definidos foram desenvolvidos adotando-se a notação visual OPL-ML (QUIRINO, 2016).

### Capítulo 3 - Evolução de MePPLa

O *modelo estrutural* apresenta os padrões da linguagem e suas relações estruturais. Nesse modelo, conforme define OPL-ML, os padrões são representados por retângulos com rótulos sublinhados e os grupos de padrões são representados por regiões delimitadas por linhas retas com bordas grossas azuis. Os padrões foram agrupados de acordo com os subprocessos relacionados a eles, que são: Preparação da Revisão e Execução da Revisão. Nesse modelo foram identificados apenas relações de correlação, representadas por setas tracejadas com pontas duplas, indicando que um padrão está relacionado ao outro, porém a aplicação de um padrão não depende da aplicação do outro. A Figura 3.1 apresenta o modelo estrutural do processo Revisão.

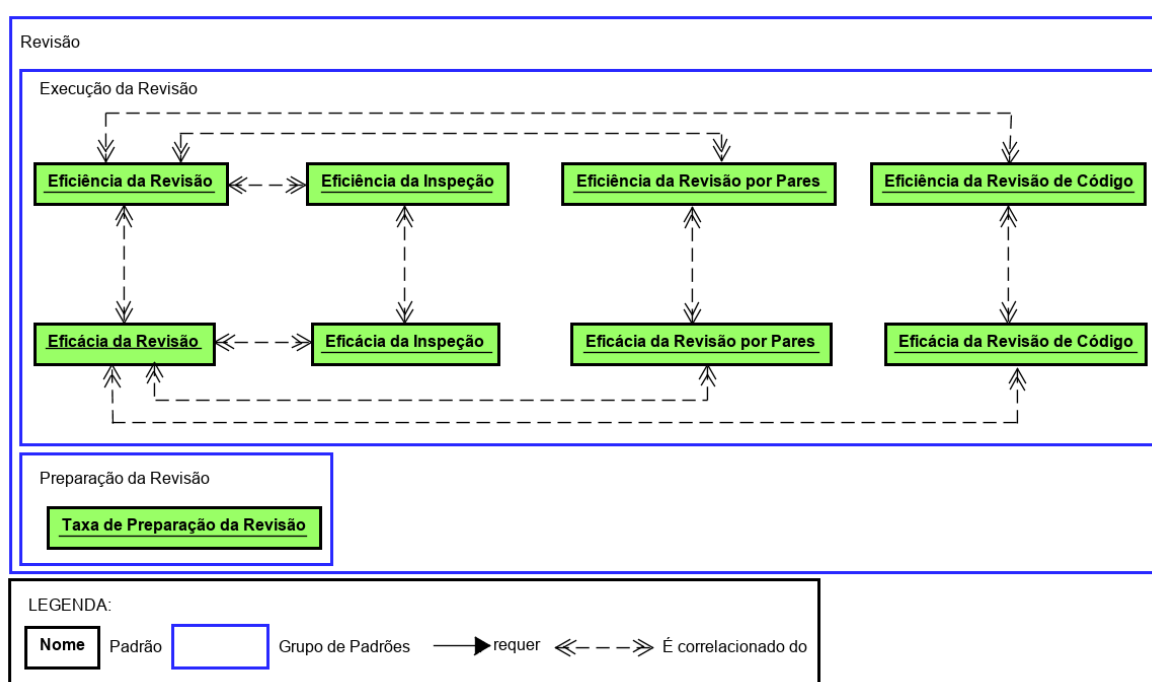


Figura 3.1 – Modelo Estrutural do Processo Revisão

O *modelo comportamental* é responsável por guiar o usuário na seleção dos padrões. Este modelo apresenta o processo de aplicação dos padrões, sendo um tipo de fluxograma, onde o usuário deve seguir as setas que guiam o fluxo de aplicação. De acordo com os nós de decisão, ele será guiado na utilização dos padrões de acordo com os objetivos que deseja tratar no planejamento de medição. Esse modelo possui dois formatos, caixa-preta e detalhado. O formato caixa-preta fornece a visão geral da linguagem e o formato detalhado permite visualizar o conteúdo interno aos grupos.

O modelo comportamental no formato caixa preta está apresentado na Figura 3.2, que é o modelo da Figura 2.4 adicionado do novo processo a ser tratado em MePPLa. O modelo comportamental no formato detalhado é apresentado em seguida, na Figura 3.3.

### Capítulo 3 - Evolução de MePPLa

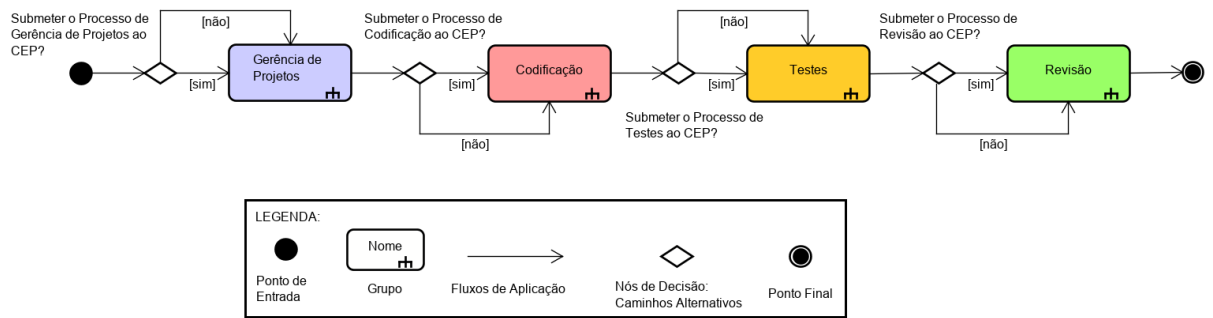


Figura 3.2 – Modelo Comportamental de MePPLa – Formato Caixa Preta

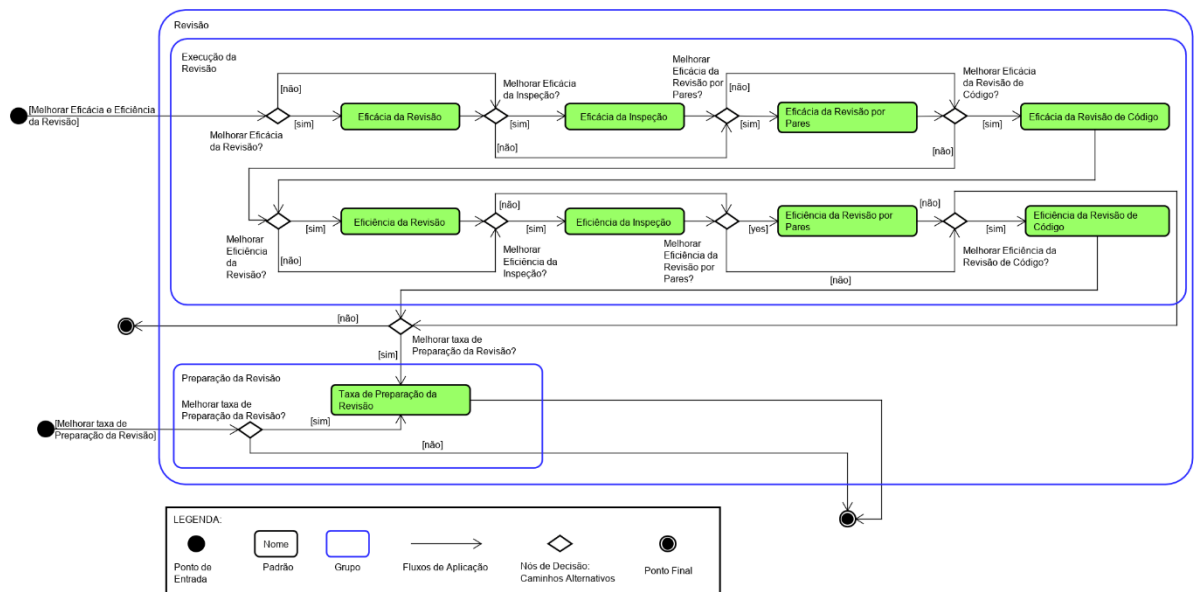


Figura 3.3 – Modelo Comportamental detalhado do processo Revisão

O modelo comportamental detalhado é consistente com o modelo estrutural, ou seja, possui os mesmos grupos de padrões, que são relacionados aos subprocessos. O modelo comportamental foi construído respeitando as relações estruturais definidas no modelo estrutural. Este modelo possui dois pontos de entrada, um para cada subgrupo de padrões. Os objetivos mais gerais dos padrões foram utilizados para indicar os pontos de entrada em cada um desses subgrupos.

## Capítulo 4

### Evolução de *MePPLa Tool*

---

*Este capítulo apresenta os principais resultados produzidos durante a evolução de MePPLa Tool. Na Seção 4.1 é feita breve introdução descrevendo o propósito da ferramenta e as evoluções realizadas. Na Seção 4.2 são tratados aspectos relacionados aos requisitos e modelagem conceitual. A Seção 4.3 trata aspectos relacionados ao projeto de s. Na Seção 4.4, são apresentadas algumas telas do sistema.*

#### 4.1 Introdução

A ferramenta *MePPLa Tool* foi desenvolvida com o propósito de auxiliar a utilização de MePPLa, buscando apoiar o usuário na criação de planos de medição através da aplicação dos padrões de MePPLa.

Durante seu desenvolvimento, no contexto do trabalho descrito em (TOMAZ, 2017), não foi elaborada documentação. Além disso, a solução tecnológica adotada para apresentar os modelos de MePPLa na interface da ferramenta e permitir ao usuário navegar pelos modelos e aplicar os padrões apresentava limitações que dificultavam a manutenção da ferramenta, especialmente a incorporação de alterações nos modelos de MePPLa.

Essa solução adotava modelos exportados da ferramenta Astah como imagens em um formato Scalable Vector Graphics (SVG), que representam de forma vetorial uma imagem, ou seja, através de definições de formas gráficas que dão informações sobre o que deve ser desenhado, organizados em uma linguagem XML. Nesse arquivo, buscava-se manualmente quais eram os elementos do tipo *rect* correspondentes aos retângulos que representam um processo, no caso do modelo caixa-preta, ou padrões, no caso do modelo detalhado, e a cada um deles era necessário atribuir um *id* específico, para que esses elementos pudessem ser manipulados como elementos da página web carregada, e assim tratar de forma adequada o seu funcionamento. Dessa forma, era necessário alterar individualmente e manualmente cada imagem exportada, o que torna a atualização dos modelos de MePPLa muito trabalhosa.

Neste trabalho, como melhoria em *MePPLa Tool*, buscou-se outra maneira de realizar essa manipulação dos elementos correspondentes da página, tendo sido, então, adotada como solução o uso de mapeamento de imagens, inclusive tornando possível o uso de formatos de imagem *raster*, que definem as imagens como uma matriz de pixels que devem ser coloridos conforme o valor das entradas da matriz. Formatos de imagem amplamente utilizados como JPG, JPEG, PNG e GIF são exemplos de imagens *raster*.



## Capítulo 4 – Evolução de MePPLa Tool

Além dessa melhoria em *MePPLa Tool*, a ferramenta foi evoluída para incorporar o novo processo (Revisão) e seus padrões definidos em MePPLa. Também foi elaborada uma documentação para a ferramenta.

Este capítulo apresenta a documentação elaborada para *MePPLa Tool*, discute a solução adotada e apresenta algumas das telas da ferramenta.

### 4.2 Requisitos e Modelagem Conceitual

No âmbito da Engenharia de Software, uma Linguagem de Padrões é uma rede de padrões inter-relacionados a qual define um processo para resolução sistemática de problemas relacionados ao desenvolvimento de software. Assim, MePPLa é uma linguagem de padrões formada por um conjunto de padrões inter-relacionados que, quando usados de forma combinada, auxiliam na elaboração de planos de medição visando ao controle estatístico dos processos.

Em MePPLa, cada padrão está relacionado a processos e segue o formato *Goal-Question-Metric* (GQM). Dessa forma, um padrão de MePPLa possui dois principais componentes: um objetivo de medição e uma forma de monitorá-lo. O monitoramento vem através de questões que indicam necessidades de informação que, por sua vez, são atendidas por medidas (com suas definições operacionais).

MePPLa é representada através de dois tipos de modelos, o modelo estrutural e o modelo comportamental. O modelo estrutural apresenta os padrões que compõem MePPLa e as relações estruturais (dependência, correlação e composição) entre eles. Já o modelo comportamental define o fluxo que guia o gerente de projetos na seleção e aplicação dos padrões.

MePPLa pode auxiliar no planejamento de medição em organizações que desejam alcançar os níveis mais altos de maturidade de processo, bem como organizações que não têm essa intenção, mas desejam submeter alguns de seus processos ao CEP. Atualmente, MePPLa inclui padrões relacionados aos processos de Gerência de Projetos, Codificação, Teste e Revisão. Ao todo, são 35 padrões, sendo 12 relacionados ao processo de Gerência de Projetos, 6 relacionados ao processo de Codificação, 8 relacionados ao processo de Testes e 9 relacionados ao processo de Revisão.

Sendo uma linguagem de padrões, MePPLa está em constante evolução. Assim, novos padrões e processos poderão ser gradativamente adicionados a MePPLa.

#### 4.2.1 Casos de Uso

Na versão desenvolvida por Tomaz (2017), *MePPLa Tool* conta com dois atores, o Gerente de Medição, possuindo algumas funcionalidades acessíveis a ele como usuário do sistema, e o Administrador, que tinha acesso a cadastros de Planos de Medição, Padrões, Medidas, Objetivos de Medição e Necessidades de Informação.

Neste trabalho, visando facilitar mudanças devido a evoluções futuras em MePPLa, foi criado um caso de uso para o Administrador do Sistema, que possui acesso às funções que permitem alterar a linguagem de padrões, nomeado *Cadastrar Grupo de Padrões*. A Figura 4.1 representa o diagrama de casos de uso de MePPLa Tool, com o novo caso de uso representado com cor destacada.

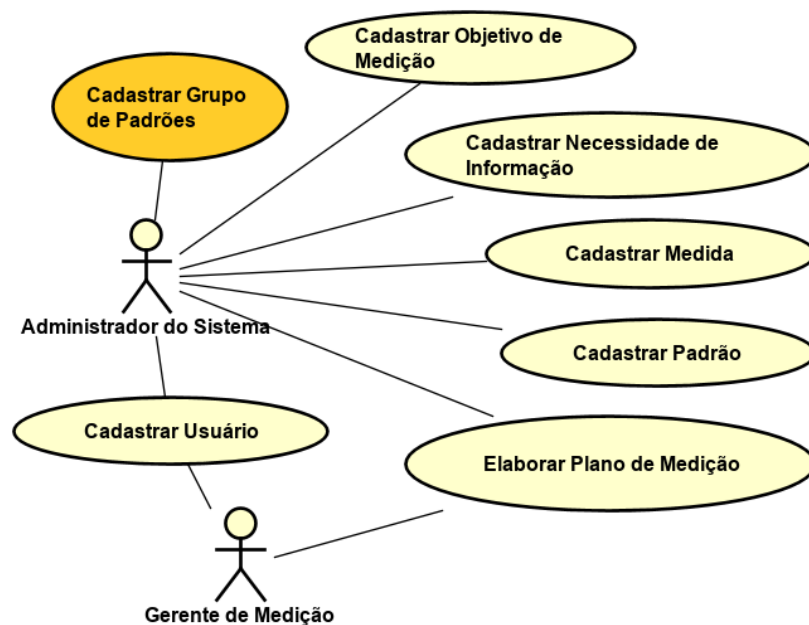


Figura 4.1 – Diagrama de Casos de Uso de MePPLa Tool.

Para o Gerente de Medição, o caso de uso *Cadastrar Usuário* permite o cadastro e alteração de sua conta de acesso. Já o Administrador do Sistema possui acesso a funcionalidades completas de inclusão, exclusão, alteração e consulta a todos os usuários cadastrados no sistema.

Os casos de uso *Cadastrar Objetivo de Medição*, *Cadastrar Necessidades de Informação* e *Cadastrar Medida* permitem a inclusão, exclusão, alteração e consulta aos objetos das referidas classes: *Objetivo de Medição*, *Necessidade de Informação* e *Medida*. Esses casos de uso são responsáveis pelo armazenamento dos elementos necessários à criação de padrões.

## Capítulo 4 – Evolução de MePPLa Tool

O caso de uso *Cadastrar Padrão* permite a inclusão, exclusão, alteração e consulta a padrões, relacionados a processos. Os padrões registrados através deste caso de uso são aqueles que compõem a LP e, assim, estarão disponíveis para o Gerente de Medição utilizar.

O caso de uso *Elaborar Plano de Medição* permite a inclusão, exclusão, alteração e consulta a Planos de Medição. Durante a inclusão ou alteração de um plano de medição, este caso de uso permite ao Gerente de Medição: *selecionar padrão*, que permite ao gerente selecionar um padrão de interesse para ser adicionado ao plano; *visualizar padrão*, que exhibe para o gerente de medição a descrição de um padrão selecionado; *adicionar padrão*, que adiciona um padrão selecionado ao plano de medição; *remover padrão*, que remove um padrão selecionado do plano de medição; *editar padrão*, que permite que o Gerente de Medição edite um padrão inserido no plano de medição, e *gerar plano de medição*, que permite a visualização de um plano de medição e sua impressão ou exportação em arquivo no formato MSWord.

Por fim, o caso de uso *Gerenciar Grupos de Padrões* permite a inclusão, exclusão, alteração e consulta a grupos de padrões, através de uma interface de upload e mapeamento de imagem, para que seja possível fazer a alteração dos modelos comportamentais geral (de MePPLa) e detalhado (dos processos). Ela também permite o upload de uma imagem para o modelo estrutural dos processos. Este caso de uso foi objeto principal da evolução da ferramenta a qual nos referimos neste trabalho.

### 4.2.3 Diagramas de Classes

O diagrama de classes desenvolvido por Tomaz (2017) sofreu algumas alterações com o objetivo de permitir a implementação de uma das mudanças provenientes da evolução proposta em *MePPLa Tool*, que é facilitar futuras atualizações na ferramenta devido a alterações em MePPLa. Essas alterações no diagrama consistiram na inclusão de uma classe Grupo de Padrões e ajustes em alguns relacionamentos entre as classes.

A Figura 4.2 apresenta o modelo de classes de análise de MePPLa Tool desenvolvido neste trabalho. As classes possuem nomes correspondentes aos elementos de MePPLa.

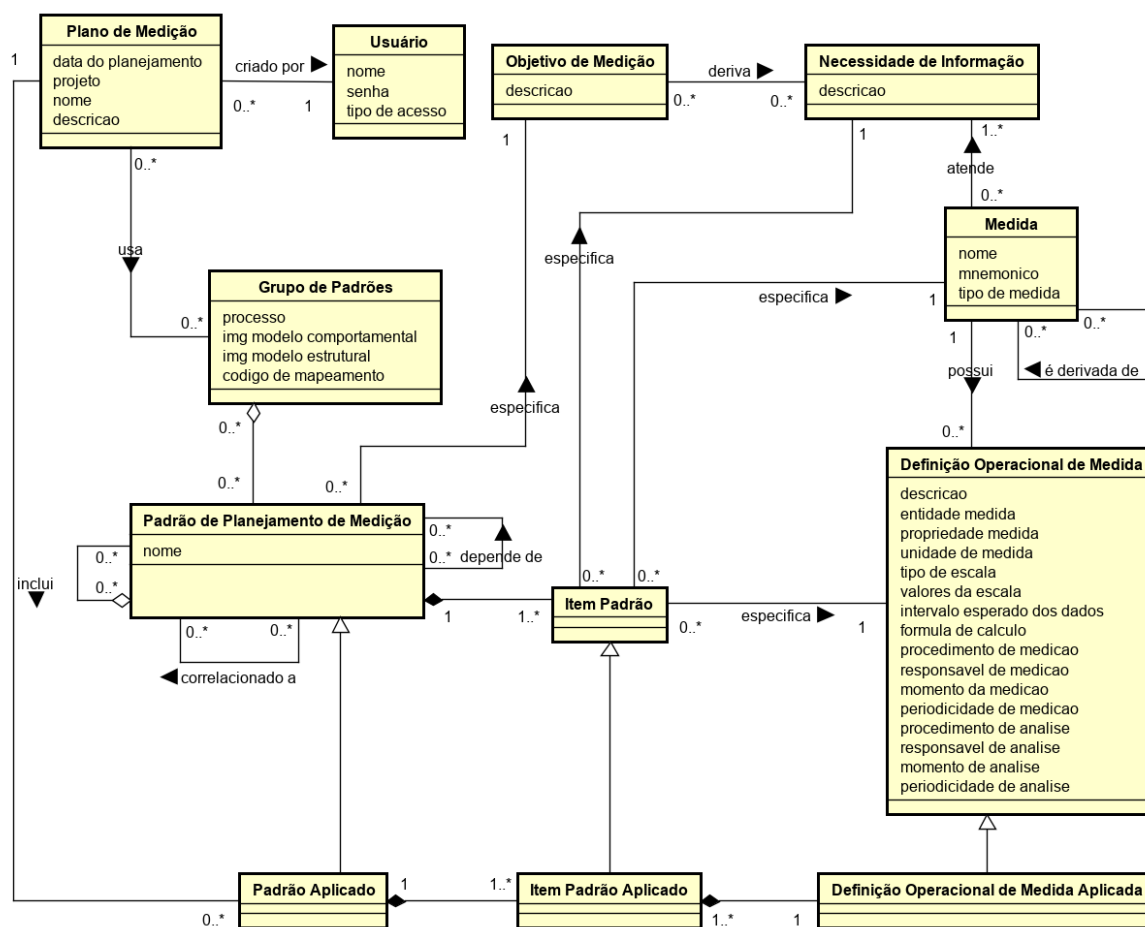


Figura 4.2 – Diagrama de Classes de Análise de MePPLa Tool.

*Planos de Medição* possuem *Padrões Aplicados*, que são incluídos a partir do uso de *Grupos de Padrões*, que são compostos por *Padrões de Planejamento de Medição*.

*Padrões de Planejamento de Medição* são compostos de *Itens de Padrão* e especificam um *Objetivo de Medição*. *Itens de Padrão* especificam *Necessidade de Informação*, *Medida* e a *Definição Operacional de Medida* a ser considerada.

*Medidas* possuem *Definição Operacional de Medida* e atendem *Necessidades de Informação* que são derivadas a partir de *Objetivos de Medição*.

Um *Plano de Medição* é criado por um *Usuário* e é definido usando um ou vários *Grupos de Padrões*. Um *Grupo de Padrões* contém um ou mais *Padrões de Planejamento de Medição*.

Quando um Gerente de Medição adiciona um *Padrão de Planejamento de Medição* a um *Plano de Medição*, é feita uma cópia desse padrão default, através da classe *Padrão Aplicado*, que é relacionada à instância do plano de medição criado. Da mesma forma, são criados *Item Padrão Aplicado* e *Definição Operacional Aplicada*, como cópias dos respectivos elementos do padrão adicionado. Isso permite que as alterações feitas nas definições operacionais dos

## Capítulo 4 – Evolução de MePPLa Tool

padrões adicionados a um plano de medição não sejam refletidas nos padrões de outros planos de medições criados.

Na versão do sistema desenvolvido por Tomaz (2017), a classe *Grupo de Padrões* não existia. No lugar dela, *Planos de Medição* usavam *Padrões de Planejamento de Medição*, num relacionamento direto entre ambas as classes. Os arquivos SVG que continham os modelos comportamental e estrutural eram inseridos inline nas páginas que os utilizavam, de forma que era necessário criar uma página para exibir e tratar cada um dos processos (Gerência de Projetos, Codificação e Testes) de maneira individual. Uma inserção inline significa copiar o código do arquivo SVG para a página web, o que torna o código da página muito poluído. Esses pontos dificultavam bastante a manutenção do sistema.

Adicionar a classe Grupo de Padrões foi crucial para o desenvolvimento da funcionalidade de gerenciamento de MePPLa, dado que ela que armazena o caminho das imagens no servidor e o mapeamento do modelo comportamental para fazer a inclusão dos padrões ao plano de medição.

### 4.3 Projeto de Sistema

A Figura 4.3 apresenta a arquitetura geral do sistema, que segue o modelo de Service Layer, proposto por Fowler (2002) e o padrão MVC (*Model View Controller*) foi adotado para implementação da camada de interface, dividindo o sistema em três camadas. Algumas das tecnologias utilizadas, como JSF (Java Server Faces), JPA (Java Persistence API), CDI (Contexts and Dependency Injection), incluídos da especificação do JavaEE, e PrimeFaces estão apresentadas na figura. No entanto, outras tecnologias foram utilizadas, tais como MySQL5 e WildFly, para a realização da persistência de dados, e aspectos de HTML5, para facilitar interações com o usuário, como canvas e o elemento `<map>`. O controle de dependências é feito com o Apache Maven, através de Project Object Model (POM). As linguagens de programação utilizadas foram Java, com Java Development Kit 8 (JDK8), e Javascript.

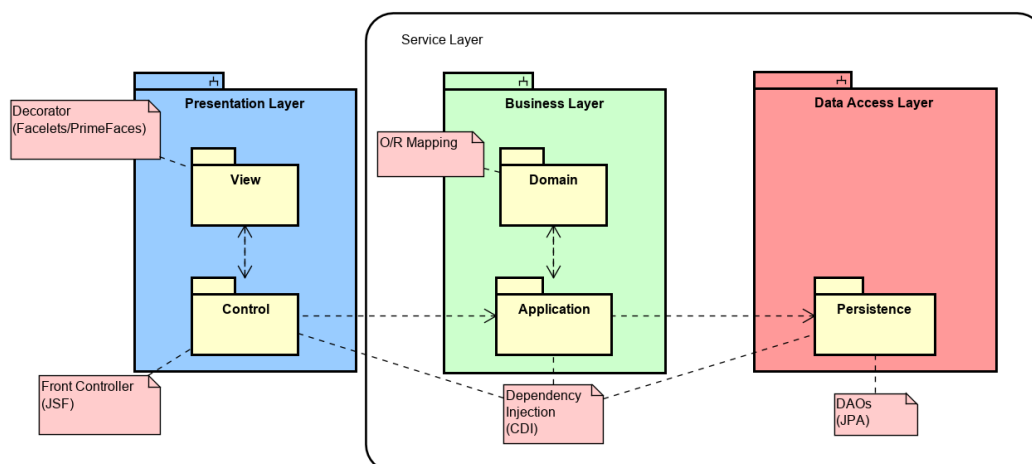


Figura 4.3 – Service Layer em MePPLa Tool

#### 4.3.1 Arquitetura de Software

Seguindo o padrão de *Service Layer*, proposto por Fowler (2002), o sistema é dividido em duas camadas principais: A Camada de Apresentação (*Presentation Layer*), e a camada de Serviços, composta de outras duas camadas: Camada de Negócios (*Business Layer*) e a Camada de Acesso de Dados (*Data Access Layer*). A Camada de Serviços encapsula a lógica de negócios e o acesso das transações com o banco de dados, fornecendo interfaces para a camada de apresentação.

A camada de apresentação adota o padrão MVC (*Model View Controller*), que divide a interação com o usuário em duas partes: o pacote de visão (*View*), que faz a entrada e saída de dados na interface visual com o usuário final, e o pacote de controladores (*Controller*), que, através das interfaces proporcionadas pela Camada de Serviços, solicita o acesso ou a manipulação aos dados, bem como a invocação da lógica de negócios. O JSF permite a implementação desse padrão, pois fornece um grau desejado de desacoplamento entre os elementos do sistema.

Para a persistência dos dados, foi adotado o padrão DAO (*Data Access Object*), que divide a Camada de Acesso de Dados (ou Persistência) em um conjunto de interfaces e um conjunto de implementações. As seções a seguir apresentam alguns detalhes sobre decisões de projeto referentes às camadas, exibindo fragmentos dos diagramas de projeto.

#### 4.3.2 Camada Lógica do Negócio

Na Camada de Negócios encontram-se as regras que definem o funcionamento do sistema, como as definições de cada classe e seus relacionamentos. O pacote de domínio

## Capítulo 4 – Evolução de MePPLa Tool

contém as classes e os mapeamentos objeto/relacional, e o pacote de aplicação contém os componentes que gerenciam as requisições recebidas da aplicação web e solicitam a persistência ou consulta de dados.

### 4.3.2.1 Componente Domínio do Problema

A Figura 4.4 apresenta o modelo de classes de projeto, que fornece as classes e os relacionamentos que compõem o pacote de domínio.

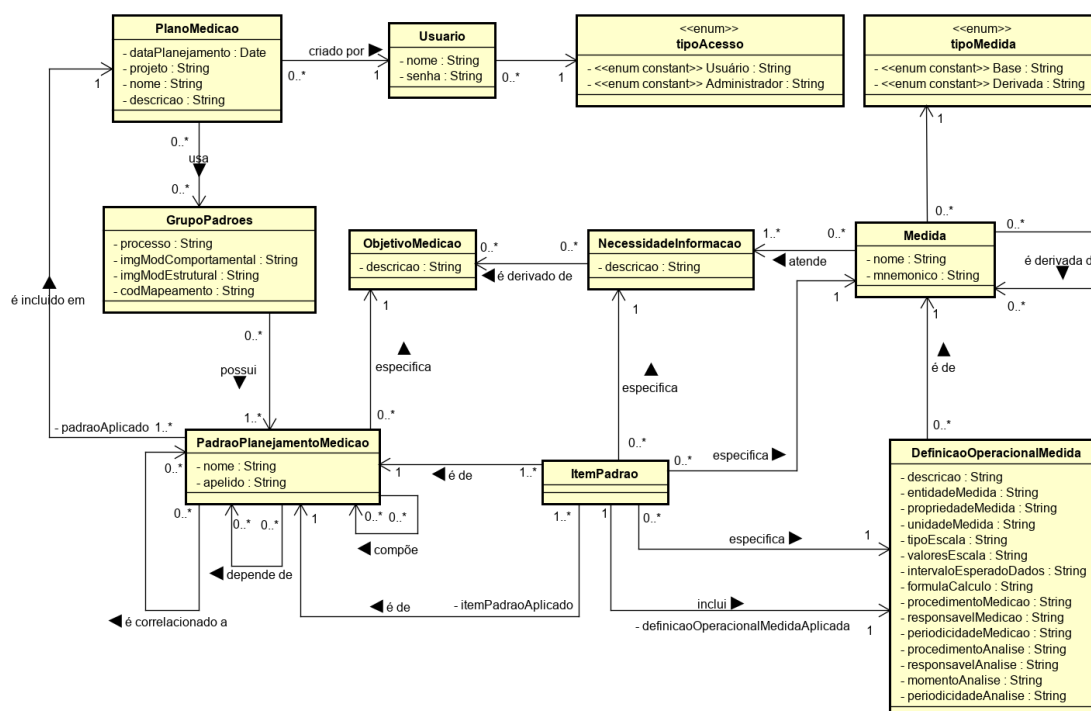


Figura 4.4 – Modelo de Classes de Projeto de MePPLa Tool

As principais alterações realizadas em relação ao modelo de Análise, apresentado na Figura 4.2, são relacionadas às classes *Padrão Aplicado*, *Item Padrão Aplicado* e *Definição Operacional de Medida Aplicada*. Quando se cadastra no sistema objetos das classes *PadraoPlanoMedicao*, *ItemPadrao* e *DefinicaoOperacionalMedida*, o valor do id cadastrado no banco de dados é null. Esses objetos com id null são considerados um template que os Gerentes de Medição podem incluir em seus planos de medição. Quando um objeto *PlanoMedicao*, é criado, ao incluir um *PadraoPlanejamentoMedicao*, é realizada uma cópia com o id do Plano de Medição que o inclui. Dessa maneira, um *PlanoMedicao* possui um ou mais *padroesAplicados*, que são instâncias de *PadraoPlanejamentoMedicao*. Por sua vez, *padroesAplicados*, possuem *itensPadraoAplicados*, que são instâncias de *ItemPadrao*. Similarmente, *itensPadraoAplicados* possuem instâncias de *DefinicaoOperacionalMedida*.

## Capítulo 4 – Evolução de MePPLa Tool

Essa decisão de projeto foi tomada para permitir diferentes usuários (Gerentes de Projeto) partirem dos padrões, objetivos, medidas e definições fornecidos por MePPLa, e adaptarem suas descrições e valores para o uso específico de cada organização, de tal forma que dois usuários diferentes podem acessar planos de medição constituídos por padrões que, apesar de derivarem de um mesmo padrão de MePPLa, possuem detalhes diferentes entre si, e cada usuário acessa apenas os padrões registrados em seu plano de medição.

O campo apelido foi adicionado à classe *PadraoPlanejamentoMedicao* para tratar a pesquisa no banco de dados, pois como o sistema possibilita o acesso nas línguas português e inglês, ao cadastrar um padrão, seu nome provavelmente irá variar de língua para língua, mas o apelido deve ser o mesmo. É responsabilidade do Administrador do Sistema garantir que o cadastro dos apelidos esteja consistente. Para exemplificar, pode-se citar dois objetos *PadraoPlanejamentoMedicao*, um com o campo nome igual a “Eficiência da Revisão” e o outro igual a “Review Efficiency”, porém o campo apelido deve ser “EficienciaRevisao” para ambos, para que a pesquisa por eles no banco de dados seja a mesma.

### 4.3.2.2 Componente de Gerência de Tarefas

No pacote de aplicação estão as definições de *Enterprise Java Beans (EJBs)* de cada elemento do pacote de domínio. EJBs são componentes que rodam em um container da aplicação web, e comunicam-se através de um mecanismo de disparo e escuta de eventos. O CDI, fornecido pelo JavaEE, realiza a injeção de dependências baseado no contexto fornecido pela camada de apresentação.

O pacote de aplicação possui, de forma geral, interfaces que oferecem métodos de CRUD (*Services*) e implementações dessas interfaces (*ServiceBeans*). Os métodos de CRUD – Create, Retrieve, Update, Delete (Criar, Consultar, Atualizar e Deletar) – devem implementar chamadas às interfaces DAO do pacote de gerência de dados para tratarem as consultas ou persistências de objetos apropriadamente no banco de dados.

### 4.3.3 Camada de Interface com o Usuário

A Camada de Interface com o Usuário (ou Camada de Apresentação) possui duas partições, *View*, ou de interface com o usuário, onde estão implementados detalhes de como a página Web está organizada, e *Control*, que cuida da comunicação entre as entradas que o usuário fornece, sejam dados ou requisições, e a aplicação.

O framework JSF/PrimeFaces foi utilizado para a construção das páginas em conjunto com recursos do HTML5, como o elemento `<map>`, para criar uma área clicável,



## Capítulo 4 – Evolução de MePPLa Tool

e AJAX (Asynchronous Javascript and XML) é utilizado para a renderização dinâmica de componentes da página, para que não seja necessário recarregar a página.

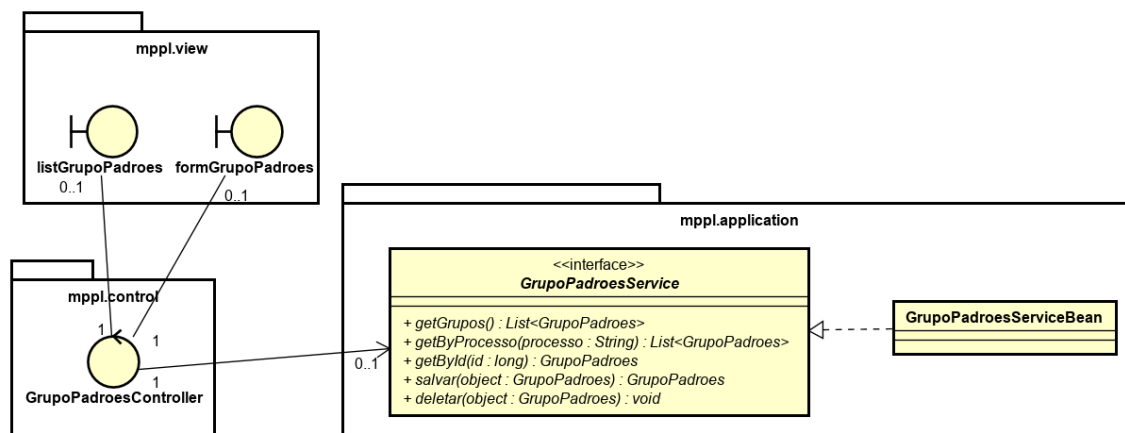


Figura 4.5 – Fragmento do Componente de Interface com Usuário de MePPLa Tool

Foram desenvolvidas cinco principais interfaces, onde duas delas englobam parte da implementação de Brito (2017), sendo elas *formLPGeral* (Figura 4.12) e *formLPDetalhada* (Figura 4.13) do diretório *WebContent/planoMedicao/*. Elas substituem as antigas *formPlanoMedicao*, *formLPTeste*, *formLPGerenciaProjetos* e *formLPCoding*, tornando mais simples a estrutura do sistema e possibilitando a criação dinâmica de grupos de padrões dentro da própria aplicação web. As páginas do tipo *form* dizem respeito a formulários de gerenciamento de cadastros, no caso *formLPGeral* apresenta o modelo comportamental geral de MePPLa, que oferece o formulário de criação do plano de medição, e ao clicar num processo apresentado no modelo, ele é redirecionado para *formLPDetalhada*, carregando seu modelo e o mapeamento respectivo, para que o usuário possa adicionar padrões a esse plano.

Páginas do tipo *list* listam elementos em tabelas para que os dados sejam alterados ou visualizados. Por exemplo, na Figura 4.5, *listGrupoPadroes* exibe os grupos de padrões cadastrados no sistema. O controlador *GrupoPadroesController* envia uma solicitação para o Bean *GrupoPadroesService*, que por sua vez solicita a *GrupoPadroesDAO* que os dados sejam buscados no banco de dados para preencherem a tabela, seguindo os padrões arquitetônicos MVC e Service Layer.

A Figura 4.6 apresenta um trecho do código da página que fornece o acesso à interface de upload e mapeamento de imagem, outra das interfaces desenvolvidas.

```

1. <h:outputLabel for="modeloComportamental" >#{msgs['gerenciarLP.form.MC']}: </h:outputLabel
   >
2.     <p:fileUpload
3.         id="modeloComportamental"
4.         size="20"

```

```

5.         fileUploadListener="#{grupoPadroesController.uploadComportamental}"
6.         label="Escolher" uploadLabel="Enviar" cancelLabel="Cancelar"
7.         allowTypes="/(\\.|\\\/)(gif|jpe?g|png|svg)$/"
8.         invalidFileMessage="São permitidas somente imagens (gif, jpeg, jpg, png e svg)"
9.         multiple="false" auto="false" fileLimit="1"
10.        fileLimitMessage="Só é possível um arquivo."
11.        onstart="console.log('Iniciando envio de arquivo')"
12.        onerror="console.log('Erro na requisição de envio)"
13.        oncomplete="console.log('Envio concluído')" />
14.    <p:message for="modeloComportamental"/>

```

Figura 4.6 – Trecho de Código da página formGrupoPadroes

O upload é realizado com o elemento `<p:FileUpload>` do PrimeFaces, chamando o controlador *GrupoPadroesController*, que implementa um método que salva o arquivo enviado no servidor. Todavia, o caminho do arquivo só é salvo de fato no grupo de padrões se o usuário solicitar que o formulário seja salvo, onde a solicitação é invocada pelo controlador através da interface *GrupoPadroesService*, no pacote de aplicação do componente de tarefas.

#### 4.3.4 Camada de Gerência de Dados

A Camada de Gerência de Dados é responsável pela persistência dos dados do sistema. O Wildfly foi utilizado para a implantação do servidor Web, e o armazenamento dos dados foi feito com MySQL5 e MySQL Workbench, para gerenciar o banco de dados.

O JPA fornece uma API para persistência de dados através de ORM (*Object-Relational Mapping* – Mapeamento Objeto Relacional) com o uso de anotações, exemplificado na Figura 4.7, que apresenta um trecho de código da classe *GrupoPadroes*, do pacote de domínio.

```

1.  @Entity
2.  public class GrupoPadroes implements Serializable, SampleEntity{
3.      private static final long serialVersionUID = -8375404246507821680L;
4.      @Id @GeneratedValue(strategy = GenerationType.AUTO)
5.      private Long id;
6.      private String processo;
7.      @Lob
8.      private String imgModeloComportamental; // caminho da imagem do modelo comportamental no
          servidor
9.      @Lob
10.     private String imgModeloEstrutural; // caminho da imagem do modelo estrutural no servidor
11.     @Lob
12.     private String codMapeamento;
13.     @OneToMany
14.     private List<PadraoPlanejamentoMedicao> padroes;

```

Figura 4.7 – Trecho de Código da Classe GrupoPadroes, no pacote de Domínio

Dessa forma, ao falar de Mapeamento Objeto Relacional, precisamos apresentar alguns detalhes do pacote de domínio, sobre o uso das anotações. A classe é anotada com `@Entity`, indicando que ela é uma entidade, seu *id* é utilizado como chave primária e gerado com `@GeneratedValue`, onde *GenerationType.AUTO* indica que a estratégia que gera esse *id* é

## Capítulo 4 – Evolução de MePPLa Tool

realizada pelo sistema de gerenciamento de banco de dados utilizado no projeto, MySQL. As anotações `@Lob` indicam que a persistência de uma propriedade ou campo devem ser feitas como um objeto grande, como CLOB/BLOB. O tipo de Lob é inferido automaticamente baseado no tipo do atributo, de acordo com as especificações do JavaEE 7. O campo *processo* é do tipo básico, e sua anotação pode ser omitida.

Em conformidade com o padrão DAO, o pacote *Persistence* fornece as interfaces acessadas pelo pacote da aplicação, seguindo a nomenclatura *NomeDaClasseDAO*, e implementação da interface, seguindo a nomenclatura *NomeDaClasseJPADA*, de cada objeto persistido.

```

1. @Local
2. public interface GrupoPadroesDAO {
3.
4.     List<GrupoPadroes> getGrupos();
5.     List<GrupoPadroes> getByProcesso(String processo);
6.     GrupoPadroes getById(Long id);
7.     GrupoPadroes salvar(GrupoPadroes object);
8.     void deletar(GrupoPadroes object);
9. }

```

Figura 4.8 – Declaração da interface GrupoPadroesDAO

```

1. @Stateless
2. public class GrupoPadroesJPADA implements GrupoPadroesDAO {
3.     @PersistenceContext
4.     private EntityManager entityManager;
5.
6.     @Override
7.     public List<GrupoPadroes> getByProcesso(String processo) {
8.         CriteriaBuilder cb = entityManager.getCriteriaBuilder();
9.         CriteriaQuery<GrupoPadroes> cq = cb.createQuery(GrupoPadroes.class);
10.        Root<GrupoPadroes> root = cq.from(GrupoPadroes.class);
11.        cq.where(cb.equal(cb.lower(root.get(GrupoPadroes._processo)), processo.toLowerCase()));
12.        return (List<GrupoPadroes>) entityManager.createQuery(cq).getResultList();
13.    }

```

Figura 4.9 – Trecho de código da classe GrupoPadroesJPADA

As Figuras 4.8 e 4.9 apresentam uma parte da implementação do pacote de persistência. A implementação é feita com uso da classe *EntityManager* e a anotação `@PersistenceContext`, que cria instâncias de persistência baseado em requisições da aplicação Web. O *CriteriaBuilder* garante consultas ao banco de dados sem erros de sintaxe.

### 4.4 MePPLa Tool

Denominada *MePPLa Tool*, a ferramenta de apoio a MePPLa, foi implementada em sua versão inicial no trabalho desenvolvido por Tomaz (2017), e tem por objetivo apoiar gerentes de medição na criação de planos de medição seguindo os fluxos providos pelos modelos comportamentais de MePPLa e selecionando os padrões que devem ser inclusos no plano de medição. Após inclusão dos padrões, as definições operacionais das medidas

## Capítulo 4 – Evolução de MePPLa Tool

dos padrões adicionados podem ser alteradas de acordo com a realidade da organização. Os usuários do tipo Administrador possuem acesso a cadastros de padrões, objetivos de medição, medidas e definições operacionais de medidas, que compõem a linguagem de padrões.

Neste trabalho, foi implementada uma funcionalidade que permite aos administradores alterarem os modelos dos grupos de padrões que compõem a LP. Através dessa funcionalidade o usuário pode fazer upload da imagem referente a cada modelo comportamental e mapear os padrões na imagem para que seja possível selecioná-los durante o registro dos planos de medição. Também é possível atualizar o modelo estrutural referente aos grupos de padrões.

As figuras a seguir apresentam algumas telas novas do sistema, que demonstram o uso da nova funcionalidade. Estão omitidas telas e descrições que já estavam presentes na versão anterior do sistema, como telas de login e de cadastros dos objetivos de medição, medidas e definições operacionais de medidas. As etapas de cadastro dos padrões que integram o processo Revisão também foram omitidas.



In Software Engineering, a Pattern Language is a network of interrelated patterns that defines a process for the systematic resolution of problems related to software development. Thus, MePPLa (Measurement Planning Pattern Language) is a pattern language composed of a set of interrelated patterns that when used in a combined way help in the elaboration of measurement plans for the statistical process control (SPC).

In MePPLa each pattern is related to a process and follows the Goal-Question-Metric (GQM) format. Thus, a MePPLa pattern includes a measurement goal, questions that indicate information needs that must be met so that it is possible to monitor the measurement goal and measures (with their operational definitions) that meet the information needs.

MePPLa is represented through two types of models: the structural model and the behavioral model. The structural model presents the patterns that compose MePPLa and the structural relations (dependence, correlation and composition) between them. The behavioral model, in turn, defines the flow that guides the selection and application of the patterns.

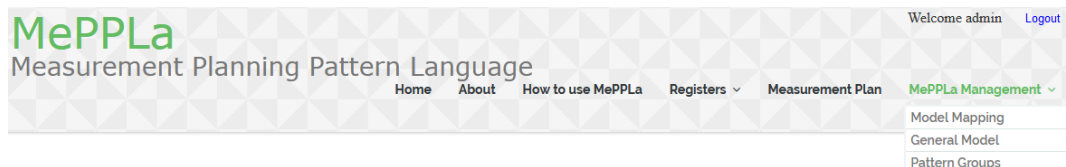
MePPLa can help organizations to perform measurement planning aiming to submit some of their processes to the SPC. MePPLa currently includes patterns related to the Project Management, Coding, and Testing processes. In all, there are 28 patterns, of which 12 are related to the Project Management process, 6 related to the Coding process and 10 related to the Testing process. Being a pattern language, MePPLa is constantly evolving. Thus, new patterns and processes will be gradually added to MePPLa.

**Figura 4.10 – Tela Home de MePPLa Tool**

A Figura 4.10 apresenta uma conta do tipo Administrador logada, onde verificam-se dois itens no menu de navegação do topo que não estão presentes nas contas de usuário comum:

## Capítulo 4 – Evolução de MePPLa Tool

*Registers*, que fornece acesso às funcionalidades de cadastro de padrões, objetivos de medição, medidas e suas definições operacionais, assim como o gerenciamento dos usuários do sistema; e *MePPLa Management*, que fornece acesso às funcionalidades de gerência dos modelos referentes aos processos integrados a *MePPLa Tool*. A Figura 4.11 apresenta a funcionalidade de criação de mapeamento de imagens, acessada através da opção Model Mapping.



### Image Mapping Interface

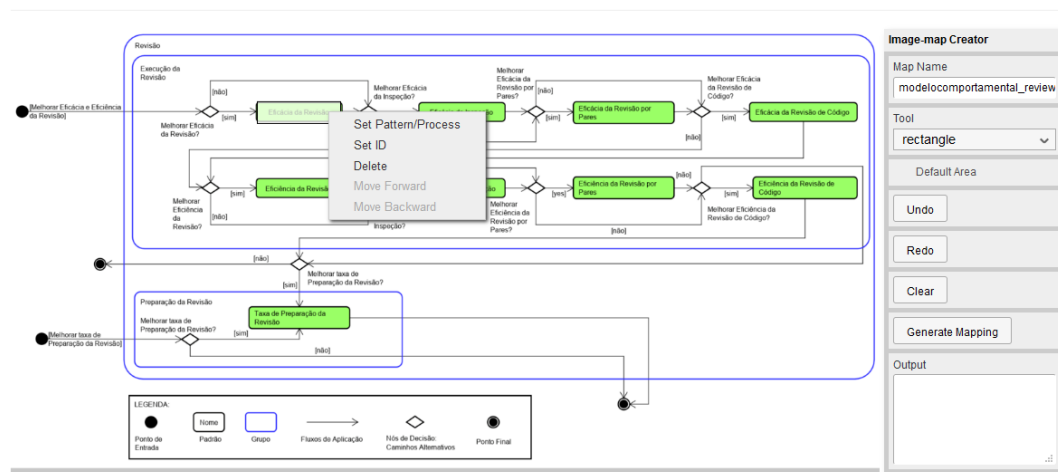


Figura 4.11 – Tela *Mapping* de MePPLa Tool

A ferramenta de criação de mapas de imagem foi editada a partir do projeto criado por Nicolas Peugnet, disponibilizada na plataforma GitHub (<https://github.com/n-peugnet/image-map-creator>), onde algumas porções do código foram alteradas para encaixar melhor com as necessidades de MePPLa. A ferramenta de Peugnet, nomeada *image-map creator*, originalmente oferece a criação de mapas exportados para os formatos SVG, JSON e HTML. Os mapas de HTML eram gerados da seguinte forma:

1. `<map name="nomeDoMapa" id=" nomeDoMapa">`
2. `<area shape="poly" coords="445,120,573,126,549,190,424,190,445,120" href="link" alt="link" title="tituloDaArea" />`
3. `</map>`

Porém, para tratar o mapeamento, foi necessário adicionar o atributo *class* às tags `<area>`, para que o Javascript detectasse o evento de click nesses elementos e os tratasse de

## Capítulo 4 – Evolução de MePPLa Tool

maneira geral. Assim, a ferramenta foi editada para permitir a atribuição desse atributo. Permitir a atribuição de um id a cada <area> também foi necessário, para identificar qual processo/padrão aquela área representa. A exportação nos formatos SVG e JSON foi desabilitada. Como o atributo title não é utilizado, ele também foi desabilitado.

O administrador pode criar um mapeamento com ou sem uma imagem no fundo, que serve apenas para orientar demarcação das áreas mapeadas. A imagem pode ser arrastada para o canvas e caso seja, seu nome é carregado no campo *Map Name*, no menu da direita. Cliques com o botão direito em áreas demarcadas no canvas fornecem cinco opções: *Set Pattern/Process*, *Set ID* e *Delete*, *Move Forward* e *Move Backward*.

*Set Pattern/Process* é utilizado para atribuir o atributo *class* da área mapeada, onde se usa *Process* para definir um processo (no caso do mapeamento do modelo comportamental geral) ou *Pattern*, no caso dos modelos comportamentais detalhados dos processos.

*Set ID* é utilizado para atribuir o atributo *id* da área mapeada, onde se usa o *apelido* do processo, no caso do modelo comportamental geral de MePPLa, ou o nome do padrão, no caso do modelo comportamental detalhado de um processo.

*Delete* deleta a área de mapeamento clicada, enquanto *Move Forward* e *Move Backward* cuidam da organização de camadas das áreas clicáveis, ou seja, quais área está na frente e quais estão atrás.

O botão *Generate Mapping* gera o mapeamento no campo de texto *Output*, no seguinte formato:

```
1. <map name="ModeloGeral.png" id="ModeloGeral.png">
2.   <area class="Process" id="GP" shape="poly" coords="326,160,478,160,478,228,326,228,326,160"
   href="" alt="" />
3. </map>
```

Uma tag <map> pode conter múltiplas tags <area>.

Tags <area> contém os atributos class e id

Ao finalizar a elaboração do mapeamento, o administrador deve copiar o código gerado no campo de texto Output, que será utilizado na próxima tela.

O *image-map creator* originalmente permitia que o mapa fosse exportado como um objeto JSON ao clicar-se no botão Save, entretanto, durante seu processo de adaptação para uso em *MePPLa Tool*, foi introduzido um ciclo no objeto JSON, que impede que ele seja exportado corretamente. Portanto, foi tomada a decisão de desabilitar o botão. Caso o *image-map creator* tenha esse bug corrigido no futuro, ao implementar um servlet para receber e tratar esse objeto, o mapeamento pode ser enviado direto ao servidor, dispensando a necessidade de copiar e colar o código do mapeamento manualmente.

The screenshot shows the MePPLa web application interface. At the top, there is a header with the MePPLa logo and navigation links: Home, About, How to use MePPLa, Registers, Measurement Plan, and MePPLa Management. The MePPLa Management dropdown menu is open, showing options for General Model and Pattern Groups. Below the header, the page title is 'Pattern Groups'. There is a search bar with a 'Filter' dropdown, a 'Search' button, and a 'Cancel filter' button. The main content is a table with the following data:

ID	Process	Behavioral Model	Structural Model
9	Review	^	^
10	Coding	^	^
11	Testing	^	^
12	Project Management	^	^

Below the table, there is a pagination control showing '10' and '(1 of 1)'. At the bottom, there are buttons for 'New', 'Edit', 'Delete', and 'View'.

Figura 4.12 – Tela *listGrupoPadroes* de MePPLa Tool

A tela apresentada na figura 4.12 é acessada na opção *Pattern Groups* do menu *MePPLa Management*. Ela lista os grupos de padrões que estão adicionados a *MePPLa Tool*, organizando-os por seus ids e permitindo que sejam pesquisados por processo, seguindo um filtro. É possível verificar a imagem atribuída aos modelos comportamental (detalhado) e estrutural clicando nos botões correspondentes, que abre um *Modal Dialog*, componente do PrimeFaces, para exibição da imagem.

Ao selecionar um processo e clicar em *Editar* ou em *Novo*, o administrador é levado para a tela apresentada na Figura 4.13, que possibilita o cadastro ou alteração de um grupo de padrões referente a um processo. As imagens podem ser arrastadas para a área de upload respectivo ou pode-se clicar em *Escolher* para selecionar o arquivo a ser carregado para o servidor. Há uma restrição de arquivos aceitos, apenas arquivos de imagens do tipo SVG, PNG, JPG, JPEG e GIF são aceitos, e apenas um arquivo pode ser selecionado por vez. Quando se clica em *Enviar*, o cliente envia para o servidor a imagem, e no caso de sucesso o servidor devolve o caminho da imagem, que é armazenado no campo *imgModeloComportamental* ou *imgModeloEstrutural*, dependendo de qual container de upload foi utilizado.

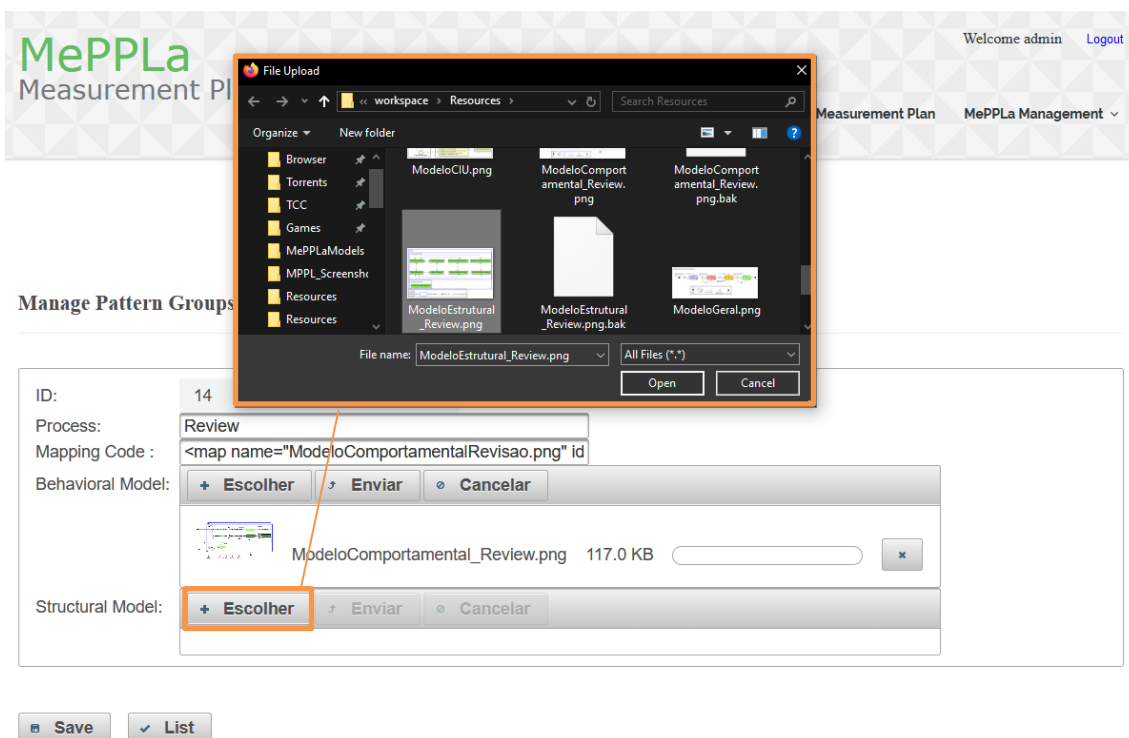


Figura 4.13 – Tela *formGrupoPadroes* de MePPLa Tool

É nessa tela, no campo *Mapping Code*, que o administrador deve colar o código criado na tela de mapeamento. Esse código será salvo no campo *codMapeamento* da classe *GrupoPadroes*.

Ao finalizar o preenchimento dos dados e o upload das imagens, clica-se em *Save* para efetuar o cadastro ou alteração dos dados do processo, ou em *List* para retornar à tela de listagem dos grupos de padrões, cancelando a atualização do banco de dados.

O modelo comportamental geral de MePPLa não diz respeito a um processo específico, e sim à linguagem de padrões, portanto não é armazenado numa classe, mas em três arquivos de configuração: a imagem, e dois arquivos de texto simples, um contendo o caminho da imagem, e outro que contém seu mapeamento. Para gerenciar o modelo geral, foi criado o item do menu *General Model*, que exibe a tela apresentada na Figura 4.14.



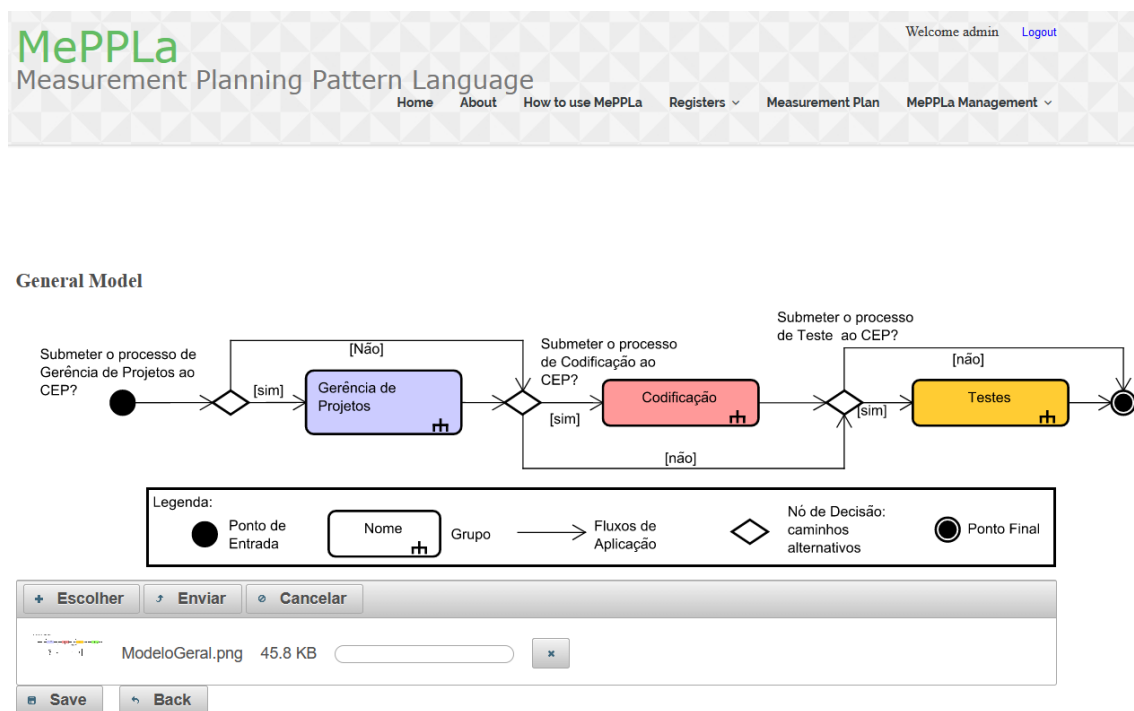


Figura 4.14 – Tela modeloGeral de MePPLa Tool durante o processo de atualização da LP

Nessa tela, é exibido o modelo geral de MePPLa que a aplicação está utilizando no momento e abaixo dele um formulário de upload similar ao apresentado na tela da Figura 4.13.

A Figura 4.14 demonstra o modelo antigo de MePPLa, com o novo modelo pronto para ser enviado ao servidor, como se verifica no formulário de upload. Ao clicar em *Enviar* o modelo é enviado para o Servidor. Clicando em *Save*, o administrador será redirecionado para a tela exibida na Figura 4.15, que permite que ele crie um mapeamento para o novo modelo e salve o mapeamento. Esse mapeamento é salvo num arquivo de texto no servidor, que é lido na hora de carregar a página de exibição da LP geral, e é feito com a mesma ferramenta apresentada na Figura 4.11.

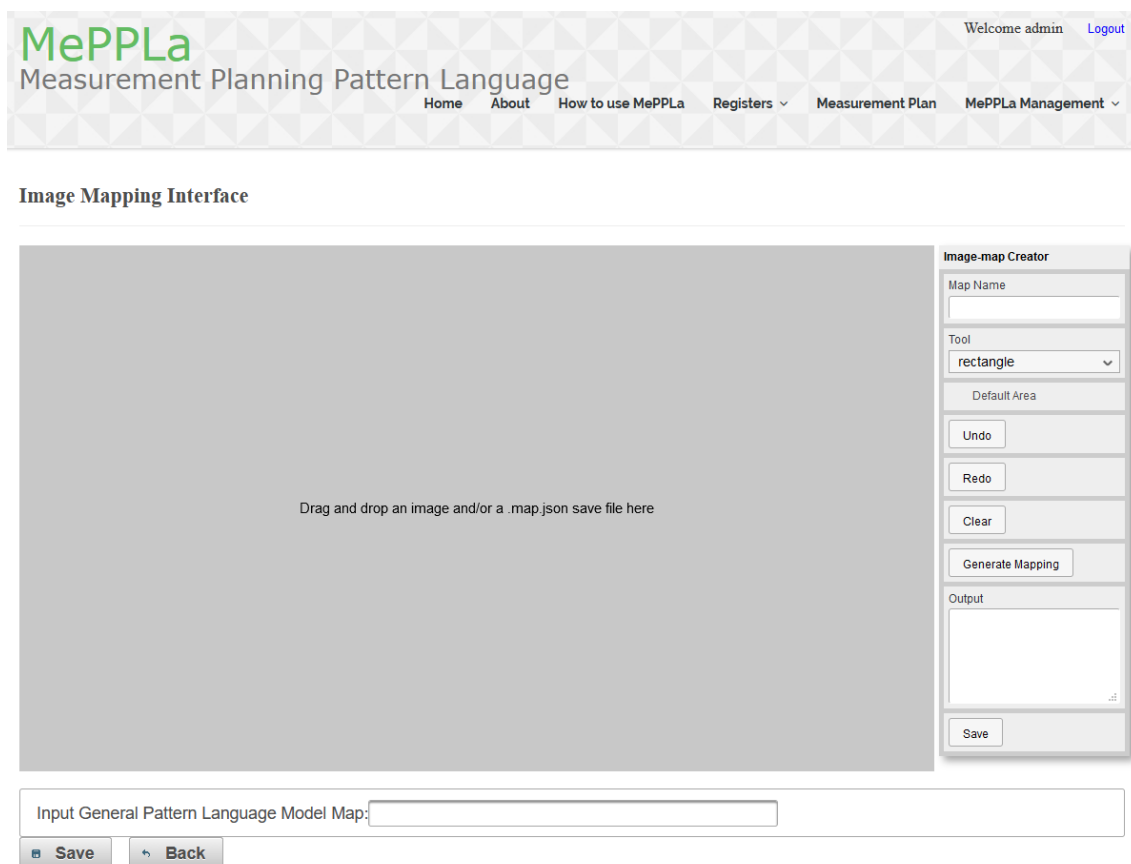


Figura 4.15 – Tela *mapearModeloGeral* de MePPLa Tool

As implementações foram feitas de forma a impactar o mínimo possível no funcionamento de criação de planos de medição da versão anterior. As telas vistas por um usuário comum são aparentemente iguais, porém o funcionamento interno de como os recursos são carregados mudou.

Conforme explicado na Seção 4.3.3, antes, para cada processo era necessária uma página *formLP<Processo>* que continha o formulário de criação do plano de medição. Agora, existe apenas uma página para criação do plano de medição, que exibe o modelo comportamental geral com os links para os grupos de padrões de cada processo, e uma página que exibe o modelo comportamental detalhado do processo, com os links para os padrões que o usuário possa querer adicionar no plano de medição, mas o processo de criação do plano ainda é o mesmo.

## Capítulo 5

# Considerações Finais

---

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

### 5.1 Conclusões

Modelos de apoio à melhoria do processo de software como o MR-MPS-SW (SOFTEX, 2012) e o CMMI (SEI, 2010) organizam em níveis a melhoria dos processos e a medição de software voltada ao CEP é uma prática crucial em níveis de alta maturidade. A implementação do CEP requer certos cuidados, portanto fazer o planejamento de medição torna-se essencial.

Em contextos semelhantes, é possível identificar padrões, que são soluções bem-sucedidas para problemas recorrentes, e organizá-los em linguagens de padrões, que representam relações entre esses padrões e oferecem um guia para a seleção e utilização deles. MePPLa é uma linguagem de padrões para o planejamento de medição, portanto fornece fluxos que guiam na seleção e utilização de padrões para serem incluídos em planos de medição.

A versão inicial de MePPLa, desenvolvida por Tomaz (2017), contempla os processos de Gerência de Projetos, Codificação e Testes, onde a abordagem SAMPPLa foi utilizada para a criação da linguagem de padrões.

Assim, para começar o desenvolvimento deste trabalho foi necessário o estudo de conceitos como medição de software, controle estatístico de processos, modelos de maturidade de processos de software, padrões, linguagens de padrões e finalmente, MePPLa e SAMPPLa.

Este trabalho aplicou SAMPPLa para evoluir MePPLa, acrescentando o processo de Revisão, contribuindo para a validação de SAMPPLa no contexto de evolução de linguagem de padrões para o planejamento de medição.

Além disso, a atualização da ferramenta de apoio foi necessária, de forma que foi necessário aprender o funcionamento da ferramenta, o uso das tecnologias envolvidas e pesquisar formas de implementar a evolução da mesma. A maneira como *MePPLa Tool* foi implementada inicialmente dificultava a integração de novos modelos ou mesmo alterações nos modelos existentes. Portanto, algumas formas para facilitar a inclusão de novas atualizações a MePPLa foram pesquisadas. Editores gráficos como MxGraph, GoJS e Rappid foram pesquisados, mas apresentaram restrições como a notação gráfica utilizada ser diferente da desejada (OPL-ML) ou necessitarem de licenças comerciais para seu uso.

## Capítulo 5– Considerações Finais

Também foram pesquisados plugins de exportação da ferramenta Astah para exportarem SVGs ou modelos XMI apropriados, no intuito de realizar o desenvolvimento com uma integração entre o Astah e a IDE Eclipse, mas nenhum plugin que atendesse as necessidades do projeto foram encontrados.

A solução para o problema em questão foi encontrada através da criação do mapeamento de imagem, utilizando recursos do HTML5 para realizar a navegação e disparo de eventos.

As principais dificuldades encontradas inicialmente foram o aprendizado dos conceitos relacionados a MePPLa, como controle estatístico de processos, medição de software e linguagens de padrões, pois não são contextos explorados nas grades de disciplinas do curso de graduação. Além disso, a ferramenta produzida por Tomaz (2017) não possuía uma documentação apropriada, de modo que entender a implementação foi uma tarefa complicada. Foi necessária experiência prática para entender os pormenores do funcionamento da implementação, sendo que é nítida a diferença da forma de codificação das páginas da aplicação web. Em outro escopo de dificuldade, pensar em maneiras diferentes para facilitar futuras evoluções foi complexo devido à necessidade de alteração de decisões de projeto anteriores, que demandam bastante esforço devido a cascatas de alterações que causam.

A Tabela 5.1 relaciona os objetivos propostos no primeiro capítulo deste trabalho com os resultados encontrados, indicando também se o objetivo foi atendido ou não.

Objetivo	Status	Resultado
Selecionar um novo processo para ser tratado em MePPLa e definir os padrões a ele associado.	Parcialmente Atendido	O processo de Revisão foi selecionado, conforme descrito itens da Seção 3.2. Os padrões foram definidos pelo autor deste trabalho, porém, por terem sido descritos pouco antes da conclusão deste trabalho, não foram revisados pelas orientadoras devido a restrições de tempo. Desta forma, em seu estado atual, ainda não podem ser aplicados, pois precisam ser avaliados e ajustados de acordo.

Evoluir MePPLa, adicionando a ela o novo processo e padrões relacionados.	Parcialmente Atendido	Na Seção 3.3 encontram-se os modelos estrutural e comportamental que integram MePPLa, e no Apêndice A as descrições dos padrões do processo de Revisão. Como mencionado anteriormente, uma vez que esses padrões não foram avaliados, não podem ser aplicados e a nova versão de MePPLa não pode ser disponibilizada para uso.
Evoluir a ferramenta de apoio à MePPLa (MePPLa Tool) para incorporar as evoluções feitas na linguagem.	Parcialmente Atendido	Na Seção 4.4 pode-se verificar o processo Revisão incluído nas telas da ferramenta, possibilitando o uso de MePPLa para apoiar o planejamento de medição voltado para o CEP com o novo processo. No entanto, as descrições dos padrões precisarão ser ajustadas de acordo com as alterações que venham a ser feitas após sua avaliação.
Evoluir a ferramenta de apoio à MePPLa (MePPLa Tool) para facilitar o processo de incorporação à ferramenta de evoluções da linguagem MePPLa.	Atendido	A interface de mapeamento de imagem foi criada para redefinir a forma de se incluir novos modelos na ferramenta, permitindo atualizações dos modelos de MePPLa apenas utilizando a aplicação web. Detalhes do projeto encontram-se na Seção 4.3 e da implementação na Seção 4.4.

Tabela 5.1 – Objetivos e Resultados da Evolução de MePPLa

A maior parte dos objetivos propostos para o trabalho foi parcialmente atendida devido a problemas com as restrições de tempo no desenvolvimento do trabalho. No entanto, os resultados ainda não concluídos foram produzidos, precisando apenas de avaliação e ajustes para que sejam concluídos, o que não demandará muito esforço. Em relação à implementação da ferramenta, apesar de algumas soluções adotadas poderem ser melhoradas, como tornar a interface de upload e mapeamento mais intuitiva e personalizar mais a ferramenta de mapeamento, a ferramenta continua a atender a proposta de

## Capítulo 5– Considerações Finais

proporcionar uma forma de se elaborar planos de medição para o, em concordância com os objetivos de Tomaz (2017), e teve o objetivo de permitir a alteração de MePPLa cumprido.

Este trabalho demonstra indícios de que SAMPPLa pode ser utilizada com sucesso para a evolução de MePPLa. Quanto à *MePPLa Tool*, foram realizadas contribuições no âmbito de projeto da ferramenta, através do desenvolvimento dos modelos de classe e de casos de uso, assim como na implementação de uma nova funcionalidade que permite upload e mapeamento de imagem, possibilitando a gerência dos grupos de padrões que compõem MePPLa.

### 5.2 Trabalhos Futuros

Dentre as contribuições deste trabalho, destaca-se a implementação da interface que permite o upload e o mapeamento de imagem. Dado que a manutenção de um sistema é um trabalho vitalício, novas necessidades estão fadadas a surgir para que o sistema continue sendo atualizado sempre. Mesmo durante a fase de implementação do sistema, possíveis melhorias foram identificadas, como melhorar o design da interface de gerenciamento dos grupos de padrões, ao tornar mais intuitivo e automatizado o processo geral de atualização dos modelos.

Outro ponto de melhoria é a integração da ferramenta de mapeamento, que roda no cliente, com o servidor, podendo ser desenvolvida de forma que os processos de troca de informação cliente-servidor sejam melhores, por exemplo, ao gerar o mapeamento, não ser necessário copiar e colar o código no formulário de cadastro do grupo de padrões, conforme levantado em alguns pontos na Seção 4.4.

Além disso, outros processos podem ser incorporados a MePPLa. A pesquisa de padrões relacionados a novos processos pode servir também para levantar pontos de melhoria da ferramenta, ao utilizá-la para incorporar os novos processos.

Um trabalho futuro pode ser alterar a implementação de pontos da ferramenta para ficar em conformidade com o diagrama de classes de Análise, criando de fato as classes Padrão Aplicado, Item de Padrão Aplicado e Definição Operacional de Medida Aplicada, facilitando a identificação dos padrões usados nos planos.

Sugestões para polir a ferramenta incluem o uso do método Frameweb, proposto por Souza (2007), utilização do utilitário *JButler*, fornecido pelo NEMO - Núcleo de Estudos em Modelagem Conceitual e Ontologias, disponibilizado no Github (<https://github.com/dwvs-ufes/jbutler/>, acesso em 18/12/2019) e finalizar a implementação da ferramenta oferecendo suporte à múltiplas línguas, pois conforme se

## *Capítulo 5– Considerações Finais*

verifica em algumas telas, existe uma mistura de botões em inglês e português numa mesma página.

## REFERÊNCIAS

- ABUBAKAR, A. M. AND JAWAWI, D. N. A. **A Study on Code Peer Review Process Monitoring using Statistical Process Control**. Software Engineering Postgraduates Workshop (SEPoW), p. 136–141, 2013.
- BARCELLOS, M. P. **Uma 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 Doutorado, COPPE/UFRJ - Universidade Federal, 2009a.
- BARCELLOS, M. P. **Utilização do Controle Estatístico de Processos na Melhoria de Processos de Software – Conhecendo Ferramentas para Análise do Comportamento dos Processos**. Engenharia de Software Magazine, v. 12, pp. 24–32, 2009b.
- BARCELLOS, M. P. **Medição de Software**. Engenharia de Software Magazine v. 24, pp. 31–36, 2010.
- BARCELLOS, M. P., FALBO, R. A., ROCHA, A. R. **Establishing a well-founded conceptualization about software measurement in high maturity levels**. In Proc. Of the 7th International Conference on the Quality of Information and Communications Technology, p. 467–472, 2010.
- BARCELLOS, M. P., FALBO, R. DE A. AND ROCHA, A. R. **A strategy for preparing software organizations for statistical process control**. Journal of the Brazilian Computer Society, v. 19, n. 4, p. 445–473, 2013.
- BASILI, V. R., ROMBACH, H. D., CALDIERA, G. **Goal Question Metric Paradigm**, Encyclopedia of Software Engineering, 2 Volume Set, John Wiley & Sons, Inc, 1994.
- BASILI, V. **Software Modeling and Measurement: The Goal Question Metric Paradigm**. College Park: Computer Science Technical Report Series, 1992.
- BASS, L., BELADY, L., BROWN, A., FREEMAN, P., ISENSEE, S., KAZMAN, R., *et al.* **Constructing Superior Software**. Software Quality Series. Macmillan Technical Publishing, 1999.
- BRITO, D. F., BARCELLOS, M. P. **Measures Suitable for SPC: A Systematic Mapping**. XV Brazilian Symposium on Software Quality, Maceió – AL, Brazil, 2016.
- BRITO, D.F., BARCELLOS, M.P., & SANTOS, G. (2017). **A Software Measurement Pattern Language for Measurement Planning aiming at SPC Research Papers Track.**, Citação retirada de <<<https://www.semanticscholar.org/paper/A-Software-Measurement-Pattern-Language-for-aiming-Brito-Barcellos/0f0ad0a33b61fbc330e8b68f56fd5587aee89c#paper-header>>>, acesso em 02/12/2019.
- BRITO, D.F., BARCELLOS, M.P. & SANTOS, G. **Investigating measures for applying statistical process control in software organizations**. J Softw Eng Res Dev 6, 10 (2018)
- BUSCHMANN, F.; HENNEY, K.; SCHIMDT, D. **Pattern-oriented Software Architecture: On Patterns and Pattern Language**. John Wiley & Sons Ltd., 2007.



- DEUTSCH, P. **Models and Patterns**. In: J. Greenfield; K. Short; S. Cook; S. Kent (Orgs.); *Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools*, 2004. Indianapolis: John Wiley & Sons.
- FENTON, N. E., NEIL, M. **Software Metrics: Success, Failures and New Directions**. *Journal of Systems and Software*, v. 47, pp. 149-157, 1999.
- FLORAC, W. A., & CARLETON, A. D. **Measuring the Software Process**. Addison-Wesley, 1999.
- FLORAC, W. A., CARLETON, A. D. **Measuring the Software Process: Statistical Process Control for Software Process Improvement**. Addison-Wesley, 1999.
- FOWLER, M. **Patterns of Enterprise Application Architecture**. Addison-Wesley, ISBN0321127420, novembro 2002.
- ISO/IEC. **ISO/IEC 15939 (E) Software Engineering - Software Measurement Process**. Geneva, Switzerland: International Organization for Standardization and the International Electrotechnical Commission, 2007.
- LANTZY, M. A. **Application of Statistical Process Control to Software Processes**. *Proceedings of the Ninth Washington Ada Symposium on Empowering Software Users and Developers*, pp. 113–123, 1992.
- QUIRINO, G. K. S. **Uma Notação Visual para Representação de Linguagens de Padrões Ontológicos**. Dissertação de Mestrado, UFES – Universidade Federal do Espírito Santo, 2016.
- ROCHA, A. R. C. da, SOUZA, G. dos S. and BARCELLOS, M. P. **Medição de Software e Controle Estatístico de Processos**. 2012.
- SARGUT, K. U., DEMIRORS, O. **Utilization of Statistical Process Control (SPC) in Emergent Software Organizations: Pitfalls and Suggestions**, *Software Quality Journal*, v. 14, n. 5, pp. 135-157, 2006.
- SEI. **CMMI® for Development, Version 1.3**. Carnegie Mellon University, 2010.
- SHEWHART, W. A. **The Economic Control of Quality of Manufactured Product**. New York: D. Van Nostrand Company, reprinted by ASQC Quality Press, Milwaukee, Wisconsin, 1980.
- SOFTEX. **MPS.BR: Melhoria de Processo do Software Brasileiro - Guia Geral MPS de Software**, 2016.
- SOLINGEN, R., & BERGHOUT, E. **The Goal/Question/Metric Method: a practical guide for quality improvement of software development**. New York: McGraw-Hill Publishing Company, 1999.
- TOMAZ, DAISY FERREIRA BRITO. **Linguagem de Padrões para apoiar o Planejamento de Medição para o Controle Estatístico de Processos de Software**. Dissertação de Mestrado. UFES - Universidade Federal do Espírito Santo – Vitória (ES), 2017. 203.f.

WHEELER, D. J. **Advanced Topics in Statistical Process Control**. SPC Press. 1995.

WHEELER, D. J., & CHAMBERS, D. S. **Understanding Statistical Process Control**.  
2nd ed. Knoxville - SPC Press, 1992.

SOUZA, Vítor Estêvão Silva, **FrameWeb: um Método baseado em Frameworks para o Projeto de Sistemas de Informação Web**, Universidade Federal do Espírito Santo 2007.

## Apêndice A

### Eficácia da Revisão

**Nome:** Eficácia da Revisão

**Processo/Subprocesso:** Revisão / Execução da Revisão

**Objetivo:** Melhorar a Eficácia da Revisão

**Necessidades de Informação:** Qual é a eficácia do processo de revisão?

**Medidas:** Eficácia da Revisão, Número de Defeitos Detectados em Revisões, Número Total de Defeitos.

**Definição Operacional das Medidas:**

<b>Medida Composta</b>	Eficácia da Revisão
<b>Mnemônico</b>	ER
<b>Descrição</b>	Medida utilizada para quantificar a eficácia da revisão, ou seja, a razão entre o número de defeitos detectados em revisões e o número total de defeitos.
<b>Entidade</b>	Subprocesso Execução da Revisão
<b>Propriedade</b>	Eficácia da Revisão
<b>Escala</b>	Números reais positivos com precisão de duas casas decimais
<b>Unidade de Medida</b>	Não há
<b>Fórmula</b>	(NDDR/NTD)
<b>Procedimento de Medição</b>	Calcular a eficácia da revisão utilizando a fórmula de cálculo da medida, devendo ser considerado o mesmo produto (ou porção de produto) em ambas as medidas presentes na fórmula.
<b>Periodicidade de Medição</b>	<<Deve-se estabelecer uma periodicidade (semanal ou quinzenal, por exemplo) para que sejam feitas as medições nos projetos. A periodicidade deve permitir várias medições ao longo de um mesmo projeto, para que seja possível obter o volume de dados adequado para o CEP>>
<b>Procedimento de Análise de Medição</b>	<< Procedimento de análise de medição padrão para uso do CEP no contexto de modelos de maturidade de processos>>
<b>Medida Base 1</b>	Número de Defeitos Detectados em Revisões
<b>Mnemônico</b>	NDDR
<b>Descrição</b>	Medida que quantifica o número de defeitos detectados nas revisões
<b>Entidade</b>	Software
<b>Propriedade</b>	Número de defeitos
<b>Escala</b>	Números reais positivos
<b>Unidade de Medida</b>	-
<b>Fórmula</b>	-
<b>Procedimento de Medição</b>	Obter o número de defeitos detectados nas Revisões
<b>Medida Base 2</b>	Número Total de Defeitos
<b>Mnemônico</b>	NTD
<b>Descrição</b>	Medida que quantifica o número total de defeitos detectados
<b>Entidade</b>	Subprocesso Execução da Revisão
<b>Propriedade</b>	Defeitos Detectados
<b>Escala</b>	Números reais positivos
<b>Unidade de Medida</b>	-
<b>Fórmula</b>	-
<b>Procedimento de Medição</b>	Obter o número total de defeitos detectados na Revisão.

**Padrões Relacionados:** Eficácia da Inspeção, Eficácia da Revisão por Pares, Eficácia da Revisão de Código e Eficiência da Revisão.

## Eficácia da Inspeção

**Nome:** Eficácia da Inspeção

**Processo/Subprocesso:** Revisão / Execução da Revisão

**Objetivo:** Melhorar a Eficácia da Inspeção

**Necessidades de Informação:** Qual é a eficácia do processo de inspeção?

**Medidas:** Eficácia da Inspeção, Número de Defeitos Detectados em Inspeções, Número Total de Defeitos.

**Definição Operacional das Medidas:**

<b>Medida Composta</b>	Eficácia da Inspeção
<b>Mnemônico</b>	EI
<b>Descrição</b>	Medida utilizada para quantificar a eficácia da inspeção, ou seja, a razão entre o número de defeitos detectados em inspeções e o número total de defeitos.
<b>Entidade</b>	Subprocesso Execução da Revisão
<b>Propriedade</b>	Eficácia da Inspeção
<b>Escala</b>	Números reais positivos com precisão de duas casas decimais
<b>Unidade de Medida</b>	Não há
<b>Fórmula</b>	(NDDI/NTD)
<b>Procedimento de Medição</b>	Calcular a eficácia da inspeção utilizando a fórmula de cálculo da medida, devendo ser considerado o mesmo produto (ou porção de produto) em ambas as medidas presentes na fórmula.
<b>Periodicidade de Medição</b>	<<Deve-se estabelecer uma periodicidade (semanal ou quinzenal, por exemplo) para que sejam feitas as medições nos projetos. A periodicidade deve permitir várias medições ao longo de um mesmo projeto, para que seja possível obter o volume de dados adequado para o CEP>>
<b>Procedimento de Análise de Medição</b>	<< Procedimento de análise de medição padrão para uso do CEP no contexto de modelos de maturidade de processos>>
<b>Medida Base 1</b>	Número de Defeitos Detectados em <u>Inspeções</u>
Mnemônico	NDDI
Descrição	Medida que quantifica o número de defeitos detectados nas inspeções
Entidade	Software
Propriedade	Número de defeitos
Escala	Números reais positivos
Unidade de Medida	-
Fórmula	-
Procedimento de Medição	Obter o número de defeitos detectados em Inspeções
<b>Medida Base 2</b>	Número Total de Defeitos
Mnemônico	NTD
Descrição	Medida que quantifica o número total de defeitos detectados
Entidade	Subprocesso Execução da Revisão
Propriedade	Defeitos Detectados
Escala	Números reais positivos
Unidade de Medida	-
Fórmula	-
Procedimento de Medição	Obter o número total de defeitos detectados em Inspeções

**Padrões Relacionados:** Eficácia da Revisão e Eficiência da Inspeção.

## Eficácia da Revisão por Pares

**Nome:** Eficácia da Revisão por Pares

**Processo/Subprocesso:** Revisão / Execução da Revisão

**Objetivo:** Melhorar a Eficácia da Revisão por Pares

**Necessidades de Informação:** Qual é a eficácia do processo de Revisão por Pares?

**Medidas:** Eficácia da Revisão por Pares, Número de Defeitos Detectados em Revisão por Pares, Número Total de Defeitos.

**Definição Operacional das Medidas:**

<b>Medida Composta</b>	Eficácia da Revisão por Pares
<b>Mnemônico</b>	ERP
<b>Descrição</b>	Medida utilizada para quantificar a eficácia da revisão por pares, ou seja, a razão entre o número de defeitos detectados em revisão por pares e o número total de defeitos.
<b>Entidade</b>	Subprocesso Execução da Revisão
<b>Propriedade</b>	Eficácia da Revisão por Pares
<b>Escala</b>	Números reais positivos com precisão de duas casas decimais
<b>Unidade de Medida</b>	Não há
<b>Fórmula</b>	$(NDDRP/NTD)$
<b>Procedimento de Medição</b>	Calcular a eficácia da revisão por pares utilizando a fórmula de cálculo da medida, devendo ser considerado o mesmo produto (ou porção de produto) em ambas as medidas presentes na fórmula.
<b>Periodicidade de Medição</b>	<i>&lt;&lt; Deve-se estabelecer uma periodicidade (semanal ou quinzenal, por exemplo) para que sejam feitas as medições nos projetos. A periodicidade deve permitir várias medições ao longo de um mesmo projeto, para que seja possível obter o volume de dados adequado para o CEP &gt;&gt;</i>
<b>Procedimento de Análise de Medição</b>	<i>&lt;&lt; Procedimento de análise de medição padrão para uso do CEP no contexto de modelos de maturidade de processos &gt;&gt;</i>
<b>Medida Base 1</b>	Número de Defeitos Detectados em Revisão por Pares
<b>Mnemônico</b>	NDDRP
<b>Descrição</b>	Medida que quantifica o número de defeitos detectados nas revisões por pares
<b>Entidade</b>	Software
<b>Propriedade</b>	Número de defeitos
<b>Escala</b>	Números reais positivos
<b>Unidade de Medida</b>	-
<b>Fórmula</b>	-
<b>Procedimento de Medição</b>	Obter o número de defeitos detectados em Inspeções
<b>Medida Base 2</b>	Número Total de Defeitos
<b>Mnemônico</b>	NTD
<b>Descrição</b>	Medida que quantifica o número total de defeitos detectados
<b>Entidade</b>	Subprocesso Execução da Revisão
<b>Propriedade</b>	Defeitos Detectados
<b>Escala</b>	Números reais positivos
<b>Unidade de Medida</b>	-
<b>Fórmula</b>	-
<b>Procedimento de Medição</b>	Obter o número total de defeitos detectados em Revisão por Pares

**Padrões Relacionados:** Eficácia da Revisão e Eficiência da Revisão por Pares.

## Eficácia da Revisão de Código

**Nome:** Eficácia da Revisão de Código

**Processo/Subprocesso:** Revisão / Execução da Revisão

**Objetivo:** Melhorar a Eficácia da Revisão de Código

**Necessidades de Informação:** Qual é a eficácia do processo de revisão de código?

**Medidas:** Eficácia da Revisão de Código, Número de Defeitos Detectados em Revisões de Código, Número Total de Defeitos.

**Definição Operacional das Medidas:**

<b>Medida Composta</b>	Eficácia da Revisão de Código
<b>Mnemônico</b>	ERC
<b>Descrição</b>	Medida utilizada para quantificar a eficácia da revisão de código, ou seja, a razão entre o número de defeitos detectados em revisões e o número total de defeitos.
<b>Entidade</b>	Subprocesso Execução da Revisão
<b>Propriedade</b>	Eficácia da Revisão de Código
<b>Escala</b>	Números reais positivos com precisão de duas casas decimais
<b>Unidade de Medida</b>	Não há
<b>Fórmula</b>	(NDRC/NTD)
<b>Procedimento de Medição</b>	Calcular a eficácia da revisão de código utilizando a fórmula de cálculo da medida, devendo ser considerado o mesmo produto (ou porção de produto) em ambas as medidas presentes na fórmula.
<b>Periodicidade de Medição</b>	<<Deve-se estabelecer uma periodicidade (semanal ou quinzenal, por exemplo) para que sejam feitas as medições nos projetos. A periodicidade deve permitir várias medições ao longo de um mesmo projeto, para que seja possível obter o volume de dados adequado para o CEP>>
<b>Procedimento de Análise de Medição</b>	<< Procedimento de análise de medição padrão para uso do CEP no contexto de modelos de maturidade de processos>>
<b>Medida Base 1</b>	Número de Defeitos Detectados em Revisões de Código
Mnemônico	NDRC
Descrição	Medida que quantifica o número de defeitos detectados nas revisões
Entidade	Software
Propriedade	Número de defeitos
Escala	Números reais positivos
Unidade de Medida	-
Fórmula	-
Procedimento de Medição	Obter o número de defeitos detectados na revisão
<b>Medida Base 2</b>	Número Total de Defeitos
Mnemônico	NTD
Descrição	Medida que quantifica o número total de defeitos detectados
Entidade	Subprocesso Execução da Revisão
Propriedade	Defeitos Detectados
Escala	Números reais positivos
Unidade de Medida	-
Fórmula	-
Procedimento de Medição	Obter o número total de defeitos detectados nas Revisões de Código

**Padrões Relacionados:** Eficácia da Revisão e Eficiência da Revisão de Código.

## Eficiência da Revisão

**Nome:** Eficiência da Revisão

**Processo/Subprocesso:** Revisão / Execução da Revisão

**Objetivo:** Melhorar o Desempenho da Revisão

**Necessidades de Informação:** Qual é a eficiência do processo de revisão?

**Medidas:** Eficiência da Revisão, Número de Defeitos Detectados em Revisões, Esforço de Detecção de Defeitos.

**Definição Operacional das Medidas:**

<b>Medida Composta</b>	Eficiência da Revisão
<b>Mnemônico</b>	ER
<b>Descrição</b>	Medida utilizada para quantificar a eficiência da revisão, ou seja, a razão entre o número de defeitos detectados e o esforço de detecção de defeitos.
<b>Entidade</b>	Subprocesso Execução da Revisão
<b>Propriedade</b>	Eficiência da Revisão
<b>Escala</b>	Números reais positivos com precisão de duas casas decimais
<b>Unidade de Medida</b>	Defeitos/homem-hora
<b>Fórmula</b>	NDDR/EDD
<b>Procedimento de Medição</b>	Calcular a eficiência da revisão utilizando a fórmula de cálculo da medida
<b>Periodicidade de Medição</b>	A medição deve ser realizada para cada execução da revisão
<b>Procedimento de Análise de Medição</b>	<< <i>Procedimento de análise de medição padrão para uso do CEP no contexto de modelos de maturidade de processos</i> >>
<b>Medida Base 1</b>	Número de Defeitos Detectados em Revisões
<b>Mnemônico</b>	NDDR
<b>Descrição</b>	Medida que quantifica o número de defeitos detectados nas revisões
<b>Entidade</b>	Software
<b>Propriedade</b>	Número de defeitos
<b>Escala</b>	Números reais positivos
<b>Unidade de Medida</b>	-
<b>Fórmula</b>	-
<b>Procedimento de Medição</b>	Obter a quantidade de defeitos detectados nas Revisões
<b>Medida Base 2</b>	Esforço de Detecção de Defeitos
<b>Mnemônico</b>	EDD
<b>Descrição</b>	Medida que quantifica o esforço despendido na detecção de defeitos
<b>Entidade</b>	Subprocesso Execução da Revisão
<b>Propriedade</b>	Esforço de detecção de defeitos
<b>Escala</b>	Números reais positivos com precisão de duas casas decimais
<b>Unidade de Medida</b>	- Homem-hora
<b>Fórmula</b>	-
<b>Procedimento de Medição</b>	Obter o esforço despendido para realizar Revisões

**Padrões Relacionados:** Eficácia da Revisão, Eficiência da Revisão de Código, Eficiência da Revisão por Pares e Eficiência da Inspeção.

## Eficiência da Inspeção

**Nome:** Eficiência da Inspeção

**Processo/Subprocesso:** Revisão / Execução da Revisão

**Objetivo:** Melhorar o Desempenho da Inspeção

**Necessidades de Informação:** Qual é a eficiência do processo de processo de detecção de defeito em inspeções?

**Medidas:** Eficiência da Inspeção, Número de Defeitos Detectados em Inspeções, Esforço de Detecção de Defeitos.

**Definição Operacional das Medidas:**

<b>Medida Composta</b>	Eficiência da Inspeção
<b>Mnemônico</b>	EI
<b>Descrição</b>	Medida utilizada para quantificar a eficiência da inspeção, ou seja, a razão entre o número de defeitos detectados e o esforço de detecção de defeitos.
<b>Entidade</b>	Subprocesso Execução da Revisão
<b>Propriedade</b>	Eficiência da Inspeção
<b>Escala</b>	Números reais positivos com precisão de duas casas decimais
<b>Unidade de Medida</b>	Defeitos/homem-hora
<b>Fórmula</b>	NDDI/EDD
<b>Procedimento de Medição</b>	Calcular a eficiência da inspeção utilizando a fórmula de cálculo da medida
<b>Periodicidade de Medição</b>	A medição deve ser realizada para cada execução da inspeção
<b>Procedimento de Análise de Medição</b>	<< Procedimento de análise de medição padrão para uso do CEP no contexto de modelos de maturidade de processos >>
<b>Medida Base 1</b>	Número de Defeitos Detectados em Inspeções
<b>Mnemônico</b>	NDDI
<b>Descrição</b>	Medida que quantifica o número de defeitos detectados nas inspeções
<b>Entidade</b>	Software
<b>Propriedade</b>	Número de defeitos
<b>Escala</b>	Números reais positivos
<b>Unidade de Medida</b>	-
<b>Fórmula</b>	-
<b>Procedimento de Medição</b>	Obter a quantidade de defeitos detectados em Inspeções
<b>Medida Base 2</b>	Esforço de Detecção de Defeitos
<b>Mnemônico</b>	EDD
<b>Descrição</b>	Medida que quantifica o esforço despendido na detecção de defeitos
<b>Entidade</b>	Subprocesso Execução da Revisão
<b>Propriedade</b>	Esforço de detecção de defeitos
<b>Escala</b>	Números reais positivos com precisão de duas casas decimais
<b>Unidade de Medida</b>	- Homem-hora
<b>Fórmula</b>	-
<b>Procedimento de Medição</b>	Obter o esforço despendido para realizar Inspeções.

**Padrões Relacionados:** Eficiência da Revisão e Eficácia da Inspeção.



## Eficiência da Revisão por Pares

**Nome:** Eficiência da Revisão por Pares

**Processo/Subprocesso:** Revisão / Execução da Revisão

**Objetivo:** Melhorar o Desempenho da Revisão por Pares

**Necessidades de Informação:** Qual é a eficiência do processo de Revisão por Pares?

**Medidas:** Eficiência da Revisão por Pares, Número de Defeitos Detectados em Revisão por Pares, Esforço de Detecção de Defeitos.

**Definição Operacional das Medidas:**

<b>Medida Composta</b>	Eficiência da Revisão por Pares
<b>Mnemônico</b>	ERP
<b>Descrição</b>	Medida utilizada para quantificar a eficiência da revisão, ou seja, a razão entre o número de defeitos detectados e o esforço de detecção de defeitos.
<b>Entidade</b>	Subprocesso Execução da Revisão
<b>Propriedade</b>	Eficiência da Revisão
<b>Escala</b>	Números reais positivos com precisão de duas casas decimais
<b>Unidade de Medida</b>	Defeitos/homem-hora
<b>Fórmula</b>	NDDRP/EDD
<b>Procedimento de Medição</b>	Calcular a eficiência da revisão utilizando a fórmula de cálculo da medida
<b>Periodicidade de Medição</b>	A medição deve ser realizada para cada execução da revisão
<b>Procedimento de Análise de Medição</b>	<< <i>Procedimento de análise de medição padrão para uso do CEP no contexto de modelos de maturidade de processos</i> >>
<b>Medida Base 1</b>	Número de Defeitos Detectados em Revisão por Pares
Mnemônico	NDDRP
Descrição	Medida que quantifica o número de defeitos detectados na revisões por pares
Entidade	Software
Propriedade	Número de defeitos
Escala	Números reais positivos
Unidade de Medida	-
Fórmula	-
Procedimento de Medição	Obter o número de defeitos detectados em revisão por partes
<b>Medida Base 2</b>	Esforço de Detecção de Defeitos
Mnemônico	EDD
Descrição	Medida que quantifica o esforço despendido na detecção de defeitos
Entidade	Subprocesso Execução da Revisão
Propriedade	Esforço de detecção de defeitos
Escala	Números reais positivos com precisão de duas casas decimais
Unidade de Medida	- Homem-hora
Fórmula	-
Procedimento de Medição	Obter o esforço despendido para realizar revisão por pares

**Padrões Relacionados:** Eficiência da Revisão e Eficácia da Revisão por Pares.

## Eficiência da Revisão de Código

**Nome:** Eficiência da Revisão de Código

**Processo/Subprocesso:** Revisão / Execução da Revisão

**Objetivo:** Melhorar Desempenho da Revisão de Código

**Necessidades de Informação:** Qual é a eficiência do processo de revisão de código?

**Medidas:** Eficiência da Revisão de Código, Número de Defeitos Detectados em Revisões de Código, Número Total de Defeitos.

**Definição Operacional das Medidas:**

<b>Medida Composta</b>	Eficiência da Revisão de Código
<b>Mnemônico</b>	ERC
<b>Descrição</b>	Medida utilizada para quantificar a eficiência da revisão de código, ou seja, a razão entre o número de defeitos detectados em revisões e o esforço de detecção de defeitos.
<b>Entidade</b>	Subprocesso Execução da Revisão
<b>Propriedade</b>	Eficiência da Revisão de Código
<b>Escala</b>	Números reais positivos com precisão de duas casas decimais
<b>Unidade de Medida</b>	Defeitos / homem-hora
<b>Fórmula</b>	NDDRC/EDD
<b>Procedimento de Medição</b>	Calcular a eficiência da revisão utilizando a fórmula de cálculo da medida
<b>Periodicidade de Medição</b>	A medição deve ser realizada para cada execução da revisão
<b>Procedimento de Análise de Medição</b>	<< Procedimento de análise de medição padrão para uso do CEP no contexto de modelos de maturidade de processos >>
<b>Medida Base 1</b>	Número de Defeitos Detectados em Revisões de Código
Mnemônico	NDDRC
Descrição	Medida que quantifica o número de defeitos detectados nas revisões
Entidade	Software
Propriedade	Número de defeitos
Escala	Números reais positivos
Unidade de Medida	-
Fórmula	-
Procedimento de Medição	Obter a quantidade de defeitos detectados na revisão
<b>Medida Base 2</b>	Esforço de Detecção de Defeitos
Mnemônico	EDD
Descrição	Medida que quantifica o esforço despendido na detecção de defeitos
Entidade	Subprocesso Execução da Revisão
Propriedade	Esforço de Detecção de Defeitos
Escala	Números reais positivos com precisão de duas casas decimais
Unidade de Medida	Homem-hora
Fórmula	-
Procedimento de Medição	Obter o esforço despendido para realizar revisões de código

**Padrões Relacionados:** Eficiência da Revisão e Eficácia da Revisão de Código.

## Taxa de Preparação da Revisão

**Nome:** Taxa de preparação da Revisão

**Processo/Subprocesso:** Revisão / Preparação da Revisão

**Objetivo:** Melhorar Desempenho da Preparação da Revisão

**Necessidades de Informação:** Qual é o desempenho na preparação da revisão?

**Medidas:** Taxa de Preparação da Revisão, Tamanho do Produto a ser Revisado, Tempo Gasto Preparando a Revisão.

**Definição Operacional das Medidas:**

<b>Medida Composta</b>	Taxa de Preparação da Revisão
<b>Mnemônico</b>	TPR
<b>Descrição</b>	Medida utilizada para quantificar o desempenho da Preparação da Revisão, ou seja, a razão entre tamanho do produto a ser revisado e o tempo gasto preparando a revisão.
<b>Entidade</b>	Subprocesso Preparação da Revisão
<b>Propriedade</b>	Taxa de preparação da Revisão
<b>Escala</b>	Números reais positivos com precisão de duas casas decimais
<b>Unidade de Medida</b>	Não há
<b>Fórmula</b>	$(TPR/TGPR)$
<b>Procedimento de Medição</b>	Calcular a eficiência dos testes utilizando a fórmula de cálculo da medida.
<b>Periodicidade de Medição</b>	A medição deve ser realizada para cada execução dos testes.
<b>Procedimento de Análise de Medição</b>	<< <i>Procedimento de análise de medição padrão para uso do CEP no contexto de modelos de maturidade de processos</i> >>
<b>Medida Base 1</b>	Tamanho do Produto a ser Revisado
Mnemônico	TPR
Descrição	Medida que quantifica o tamanho do produto a ser revisado
Entidade	Subprocesso preparação da revisão
Propriedade	Tamanho do produto a ser revisado
Escala	Números reais positivos
Unidade de Medida	KSLOC
Fórmula	-
Procedimento de Medição	Obter o tamanho do produto a ser revisado
<b>Medida Base 2</b>	Tempo Gasto Preparando a Revisão
Mnemônico	TGPR
Descrição	Medida que quantifica o tempo gasto na preparação da revisão
Entidade	Subprocesso Preparação da Revisão
Propriedade	Tempo gasto preparando a revisão
Escala	Números reais positivos
Unidade de Medida	Horas/minutos
Fórmula	-
Procedimento de Medição	Calcular o tempo gasto na preparação da revisão

**Padrões Relacionados:** Não há.