

# Métricas para análise de complexidade de programas orientados a objetos

**Autor: João Marcelo Moraes Fernandes \***

**Professor Orientador: Dra. Laís do Nascimento Salvador \*\***

## Resumo

*No contexto da Engenharia de Software, uma métrica corresponde às definições, coletas, análises e agilidades sobre uma medição com o objetivo de melhorar a qualidade do software desenvolvido por uma organização. O uso de métricas tornou-se uma ferramenta fundamental para as fases de planejamento e acompanhamento de projetos, considerada das atividades mais importante no processo de desenvolvimento de um software. O presente artigo pretende investigar algumas Métricas existentes no contexto da orientação a objeto, estudando suas principais características. Será abordado também um projeto proposto para a criação de um avaliador de métricas em softwares orientados a objetos usando a ferramenta JavaCC.*

## 1 Introdução

Métricas são padrões de medidas usadas para avaliar atributos de alguma característica específica, tais como qualidade ou complexidade. No estudo da Engenharia de Software, Métricas são processos de definições, coletas, análises e agilidades sobre medição ou mensuração, pelos quais números ou símbolos são associados a atributos, com o objetivo de descrevê-los de acordo com um conjunto de regras claramente definidas [1]. A mensuração produz como resultado um conjunto de medidas, que é um mapeamento entre um atributo empírico e uma escala matemática [1]. De uma forma geral as medidas podem atender aos seguintes objetivos:

- *Conhecimento*: Coletam-se as informações para se ter um conhecimento mais preciso de um item ou processo.
- *Avaliação*: Usam-se as informações quantitativas para verificar se um produto ou atividade atende aos critérios de aceitação.

---

\* Curso de Sistemas de Informação da Universidade Salvador – UNIFACS

\*\* Grupo de Pesquisa em Engenharia de Software e Aplicações (GESA/NUPERC)

- *Controle*: Usam-se os dados para acompanhar alguma atividade.
- *Previsão*: Usam-se as informações para se obter indicadores de tendências ou estimativas.

As métricas são ferramentas essenciais ao gerenciamento de projetos de softwares, isso ocorre porque existe uma crescente exigência pela melhoria da qualidade do software, através do desenvolvimento dentro dos prazos, custos estimados e atendendo as expectativas do cliente [3]. Essas métricas devem prover informações que ajudem na tomada de decisões de acordo com os objetivos e estratégias da organização a que se propõe fazer o software. Alguns desses objetivos podem ser: melhorar a qualidade do planejamento do projeto; reduzir os custos de trabalho no processo; melhorar a qualidade de desenvolvimento e do produto resultante; reduzir os custos de falha; aumentar a produtividade e aperfeiçoar continuamente os métodos de gestão de projeto.

Nos últimos anos estão sendo desenvolvidas várias pesquisas na área da Engenharia de Software com base na qualidade e principalmente usando-se o paradigma orientado a objeto. A evolução desse paradigma originou um novo conjunto de métricas específicas para processos de desenvolvimento de softwares, entre as quais número de métodos por classes, profundidade de herança de classes, número de subclasses, acoplamento entre objetos, respostas de classes e nível de coesão de métodos [3].

Este artigo tem como objetivo central avaliar Métricas no contexto da construção de um software usando o paradigma orientado a objeto. Essa discussão envolve os seguintes objetivos específicos relacionados abaixo:

- Revisar o conceito de Métricas de software e sua caracterização.
- Apresentar Métricas de software orientado a objeto.
- Apresentar uma proposta para a implementação dessas Métricas.

O documento é organizado da seguinte forma: a próxima seção apresenta o conceito de Métricas de Software e sua caracterização; na terceira seção serão apresentadas Métricas para Programas Orientados a Objetos, mostrando o conceito de objetos e os tipos de métricas associadas a este paradigma; A seção seguinte abordará um projeto de um avaliador de métricas em softwares orientados a objetos, exibindo

informações sobre o JavaCC e informando os passos básicos de implementação; por fim as considerações finais.

## 2 Métricas de Software

Métricas são usadas para auxiliar o gerente do projeto na tomada de decisões, com relação às três etapas do ciclo de vida de um projeto: definição do custo e tempo necessário; determinação do estado do projeto, do esforço para completá-lo e a qualidade do produto. Com esse auxílio o gerente do projeto pode identificar os problemas ocorridos em áreas específicas, tornando mais fácil fazer ajustes no escopo, no cronograma e no plano de trabalho.

Portanto, para que as métricas sejam implementadas, é necessário que se mostre realmente úteis. Uma Métrica será útil e de qualidade, se colaborar para a compreensão de um processo em estudo ou um dos seus produtos resultantes [6]. Além disso, a métrica deve ser válida, confiável e prática. Uma métrica é *válida* se ela quantifica o que nós queremos medir. Ela é *confiável* se à aplicação correta do seu algoritmo produz o mesmo resultado dadas às mesmas condições. Além disso, uma Métrica é *prática* quando é barata, fácil de computar e fácil de interpretar [4].

O estudo das Métricas fica mais fácil quando separamos em duas categorias fundamentais: *-Métricas de processos* que são aquelas que se relacionam com o gerenciamento do processo na construção do sistema, essas métricas se preocupam apenas com a última avaliação e a avaliação anterior; e *-Métricas de produtos* que são aquelas que se relacionam com a qualidade, pois tem como principal objetivo medir a qualidade dos sistemas, essas métricas se preocupam apenas com a avaliação do produto em um momento particular.

Métricas devem fazer parte de uma estratégia completa de melhoria no processo de desenvolvimento de software [4]. Além disso, elas devem ser passíveis de serem automatizadas para que os dados relevantes possam ser mais rapidamente e facilmente

coletados e processados [4]. É de extrema importância que as Métricas sejam independentes de plataformas e de linguagens, para que elas possam ser aplicadas o quanto antes no processo de desenvolvimento de software.

### **3 Métricas para Programas Orientados a Objetos (OO)**

*Linguagens orientadas a objetos* são linguagens baseadas na composição e interação entre diversas unidades chamadas objetos. *Objetos* são conjuntos complexos de dados e conjuntos de funções que podem ser aplicadas a cada um desses dados. Esses objetos são capazes de armazenar estado através de seus atributos e reagir a mensagens enviadas a eles, podendo se relacionar e enviar mensagens a outros objetos. Observa-se que há diferenças consideráveis entre as linguagens convencionais e as orientadas a objetos, tais como mecanismos de classe, herança, polimorfismos e encapsulamento, que são implementados somente pelas linguagens orientadas a objetos.

Atualmente, verifica-se a necessidade de se classificar as métricas de software mais adequadas ao paradigma orientado a objetos, pois é o paradigma que mais evolui e é o mais estudado[3].

No contexto de orientação a objeto, métricas associadas às classes podem ser usadas para avaliação da reusabilidade. Algumas Métricas são de uso abrangente, enquanto outras são dependentes do paradigma de programação ou da linguagem em questão [6]. O que se pode perceber é que os resultados variam muito dependendo do processo utilizado para o desenvolvimento.

Existem várias propostas para Métricas OO que levam em consideração as características básicas e interações do sistema como: número de classes e métodos, linhas de código por método, profundidade máxima da hierarquia de classes, entre outros. Tais métricas baseiam-se na análise detalhada do sistema.

A maioria das medidas examina atributos em termos dos conceitos de OO. Para tanto, é necessário coletar um número significativo de contagens, ou seja, é necessário

tomar valores de vários projetos, dimensioná-los selecionando as classes, os métodos e os atributos desejáveis para medir o tamanho e a complexidade de um novo software.

Existem quatro categorias básicas no desenvolvimento de Métricas para softwares Orientados a Objetos, são elas:

- *Tamanho do Sistema*: avalia o tamanho e a complexidade do sistema.
- *Tamanho de Classe ou Método*: avalia o tamanho e as características das Classes e do Método.
- *Acoplamento e Herança*: avalia a interdependência entre as classes, ou seja, relações numerosas indicam complexidade no sistema.
- *Classes ou Métodos Internos*: avalia a complexidade interna das classes e dos métodos. Verifica a documentação do sistema.

As Métricas apresentadas são apenas algumas entre tantas existentes. É recomendável que seja utilizado uma ou mais Métricas para medir um projeto, dando mais segurança na análise e conclusão do projeto. Outras métricas importantes são a FPA (*Análise por Ponto de Função*), que mede a complexidade do software pela quantificação de funcionalidade e *COCOMO* (*Constructive Cost Model*), que busca medir esforço, prazo e tamanho de equipe.

Uma importante observação é que métricas de tamanho no sistema não possuem valores padrões que possam ser comparados para avaliar o sistema que está sendo construído. O tamanho do sistema depende inteiramente da quantidade de funcionalidade que está inserida dentro do mesmo. Outras métricas, no entanto, têm valores padrões. Por exemplo, o tamanho de um método é razoavelmente consistente entre os diversos sistemas [4].

## **4 Avaliador de métricas para Programas Orientados a Objetos**

Está sendo proposto em um projeto de iniciação científica da Universidade Salvador – UNIFACS (Grupo de Pesquisa em Engenharia de Software e Aplicações

(GESA)), sobre a responsabilidade do Autor: João Marcelo Moraes Fernandes e da Professora Orientadora: Dra. Laís do Nascimento Salvador, a implementação de um avaliador de Métricas orientadas a objetos, esse avaliador vai analisar o código fonte de um projeto e avaliar a sua qualidade e complexidade.

Inicialmente deve-se definir as métricas que serão implementadas, onde é muito importante escolher Métricas de Processos para avaliar a complexidade do projeto, como por exemplo: quantidade de classes e métodos internos. Essas Métricas devem ser válidas e confiáveis, pois verificam o quão bem documentado está o código do sistema. Devemos inserir também Métricas de produto para avaliar a qualidade do projeto, por exemplo: tamanho de classe ou método, ainda que classes e métodos possam ser medidos e caracterizados de várias formas.

Depois de escolher as Métricas que devem ser implementadas e tendo o conhecimento profundo da gramática da linguagem escolhida, neste caso uma linguagem orientada a objeto, pode com a ajuda de uma ferramenta específica implementar as Métricas propostas. A ferramenta escolhida para ajudar na implementação dessas Métricas foi o javaCC.

#### **4.1 Ferramenta JavaCC (Java Compiler Compiler)**

*JavaCC* é um gerador de *parser*, ou seja, é uma ferramenta usada para ler uma gramática e converter em um programa Java, e também é um analisador léxico e sintático que reconhece se um determinado texto pertence a gramática especificada. Através de uma gramática para o uso do *JavaCC*, será gerado código fonte de um *parser* para a linguagem escolhida.[6]

Para implementar Métricas de código fonte usando o *JavaCC* é necessário seguir uma seqüência de passos básicos [6].

Passo 1 – Definição/Adaptação da Gramática da linguagem para o JavaCC (ex: Java, C++, C);

Passo 2– Criação do código fonte auxiliar em Java para a análise léxica e sintática;

Passo 3 – Definição da Métrica a ser implementada.

Passo 4 – Análise do código obtido no passo 2 (*parser*) para a implementação das métricas desejadas;

Passo 5 – Instrumentação/adaptação do *parser* para a medição (este é o passo que requer maior esforço);

Passo 6 – Teste dos resultados da Métrica implementada, caso seja necessário deve-se voltar ao passo 4 para rever as definições da métrica pretendida;

Baseado no conhecimento da linguagem, no *parser* gerado e na definição das Métricas torna-se possível a sua implementação da linguagem desejada.

## **6 Considerações Finais**

Neste artigo foram passadas as características principais e desejáveis que uma métrica de software deve ter. Primeiramente foi conceituado e informado sobre a importância das métricas na fase de Gerenciamento do projeto e na construção de um sistema. Essas métricas são de extrema importância para se ter uma melhora na qualidade do planejamento do projeto, fazendo com que reduza bastante o seu custo. Depois foram mostradas as três características fundamentais em uma métrica: validade, confiabilidade e praticidade.

Foi abordada a utilização das Métricas no paradigma (OO), pois é o paradigma que mais evolui e mais tem estudos junto aos outros. Foi apresentada também a proposta de um avaliador de Métricas para analisar a complexidade e qualidade de um projeto. É importante salientar que é de extrema necessidade a validação das Métricas recém implementadas, realizando testes, validando o código adicionado no *parser* e comparando o resultado das Métricas com outras ferramentas [6].

Por fim, medir é importante primeiramente para controlarmos os projetos e assim, podermos gerenciá-los. Podemos medir para avaliar a proximidade que estamos dos objetivos que definimos para o projeto quanto à qualidade, esforço, custo e requisitos. Por essa razão, o uso de Métricas tem se tornado uma grande vantagem estratégica.

## 7 Bibliografia

[1] BORGES, Eduardo Pereira. *Um modelo de medição para processos de desenvolvimento de software*. 2003. 154f. Dissertação de Mestrado – Departamento de Ciências da Computação, Instituto de Ciências Exatas da Universidade Federal de Minas Gerais, Belo Horizonte. Disponível em:  
<<http://www.wppf.uaivip.com.br/pesquisa/DissertacaoEduardo.pdf> >

[2] CAMPÊLO, Gabriela M. Carneiro. *A utilização de métricas na gerência de projetos de software: uma abordagem focada no CMM nível 2*. 2002.135f. Trabalho (Pós-Graduação em Ciência da Computação) – Centro de Informática, Universidade Federal de Pernambuco, Recife. Disponível em  
<<http://www.cin.ufpe.br/~hermano/download/dissertacoes/dissertacaoFinalRevisadaGabriela.pdf>>.

[3] CARVALHO, Gonçalo Lages de. *Métricas de Modelação de Software*. 2004. Disponível em  
<<http://berlin.inescid.pt/cadeiras/pfsi/PFSI2003/SEMINARIO/pdfs/metricas-goncalo-carvalho.pdf>>.

[4] MENESES, Javé Barbosa de. *Inspector – um Processo de Avaliação de Progresso para Projetos de Software*. 2001. 189f. Trabalho ( Pós- Graduação em Ciências da Computação) – Centro de Informática, Universidade Federal de Pernambuco, Recife. Disponível em  
<<http://www.cin.ufpe.br/~gmp/docs/papers/dissertacao-inspector-final.pdf>>



[5] PRICE, Ana Maria de Alencar; TOSCANI, Simão Sirineo. *Implementação de Linguagens de Programação: Compiladores*. 9. ed. Rio Grande do Sul: Sangra Luzzato, s.d.

[6] BOUDOUX<sup>1</sup>, Marcelo; CARNEIRO, Glauco<sup>2</sup>; MENDONÇA, Manoel<sup>2</sup>. *Implementando Métricas de Código Fonte Usando JavaCC*. S.d. Artigo(<sup>1</sup>Curso de Ciência da Computação da Universidade Salvador – UNIFACS e <sup>2</sup>Grupo de Pesquisa em Engenharia de Software e Aplicações (GESA/NUPERC) da Universidade Salvador - UNIFACS.)

[7] SANT'ANNA, Cláudio Nogueira. *Manutenibilidade e Reusabilidade de Software Orientado a Aspectos: Um Framework de Avaliação*. 2004. Dissertação de Mestrado – Programa de Pós – Graduação em Informática da PUC- Rio.

[8] MAIA, Jose Ricardo. *Use métricas adequadas: garanta a qualidade de projeto orientado a objeto*. Disponível em: < <http://www.euax.com.br>>