

# Parte Subjetiva

Lucianna Thomaz Almeida

1 de Dezembro de 2010

## 1 Desafios e Frustrações

Nesse trabalho usamos métricas de código-fonte para auxiliar a análise de códigos. Um problema existente nessa abordagem é como definir intervalos absolutos para a interpretação dos valores das métricas. Isso acontece porque não podemos generalizar essa interpretação, dado que precisamos levar em consideração as diferentes características das linguagens e domínios de aplicação nas avaliações. Além disso, algumas métricas não representam sempre características boas ou ruins, essa relação pode depender do contexto da análise, como é o caso da métrica NCC que deve ser baixa no cenário de Método Invejoso e alta para Método Dispersamente Acoplado.

Outro problema que enfrentamos ao usarmos métricas é que elas são muito boas para indicar características dos códigos-fonte, porém elas ainda são limitadas. Não conseguimos analisar a expressividade dos nomes atribuídos a variáveis, métodos ou classes, por exemplo, com base apenas em um conjunto de métricas. Para isso precisaríamos também de uma análise semântica.

Uma frustração minha é que ainda não tenho muita experiência quanto ao processo de desenvolvimento de softwares grandes. Imagino que essa experiência aumentaria o meu conhecimento nesse contexto e me permitiria ter novas ideias para facilitar a análise dos códigos e a interpretação dos dados obtidos, incentivando, assim, a avaliação frequente da limpeza dos códigos-fonte ao longo de todo o seu desenvolvimento.

## 2 Disciplinas Relevantes

Além de MAC0110 e MAC0122 que são matérias extremamente importantes como base do curso de Bacharelado em Ciência da Computação, podemos citar como relevantes para esse trabalho de conclusão de curso as seguintes matérias:

- **Laboratório de Programação II**

Foi em Laboratório de Programação II que começamos a estudar o paradigma de orientação a objetos. Nessa época tive a oportunidade de desenvolver um jogo usando essa abstração.

- **Programação Orientada a Objeto**

Nessa disciplina aumentamos nosso conhecimento quanto a programas orientados a objetos desenvolvendo software e estudando os padrões de design relacionados a programas com essa abstração. Como o tema do nosso trabalho é limpeza de código orientado a objetos, a experiência adquirida e os conceitos aprendidos durante esse curso foram bastante importantes.

- **Laboratório de Programação Extrema**

Em Laboratório de Programação Extrema montamos vários grupos e cada um deles é responsável por criar um sistema real. Ao longo de todo o processo de desenvolvimento desse trabalho aprendemos a valorizar mais um código limpo. Isso acontece porque ao longo de todo o processo de desenvolvimento precisamos entender e alterar código escrito por outras pessoas, adicionar funcionalidades, corrigir as partes do sistema que não agradam o cliente ou os desenvolvedores e fazer refatorações sempre que houver algo que pode melhorar. Não demora muito para percebermos a necessidade de manter o código sempre simples, expressivo e flexível (tema tratado na nossa monografia).

- **Programação Funcional**

Em Programação Funcional aprendemos um pouco de Erlang e Scala. Essas linguagens são bastante diferentes das outras usadas ao longo do curso. Estudá-las mostrou que é realmente importante considerarmos a linguagem usada em um programa antes de analisarmos o seu código-fonte. Isso acontece porque algumas linguagens possuem características completamente diferentes de outras. Por exemplo, dada a inexistência de variáveis em Erlang, métodos escritos nessa linguagem precisam de um número muito maior de parâmetros do que funções escritas em C. Então, devemos considerar essa variação na interpretação dos valores das métricas, aumentando o intervalo aceitável para o número de parâmetros dos códigos escritos em Erlang.

### 3 Futuro do Trabalho

A criação dos cenários é uma das principais contribuições da nossa monografia. Futuramente, podemos criar mais cenários, associando outros conceitos de código limpo com conjuntos de métricas de código-fonte ou segmentando contextos já formados, permitindo a análise mais específica de alguns problemas.

Seria interessante também pesquisarmos quais são os cenários mais comuns presentes nos softwares já existentes. Para os problemas mais comuns, poderíamos apresentar possíveis refatorações de forma bastante detalhada, ajudando o programador a pensar na alteração mais eficiente para cada caso.

Visando minimizar a dificuldade existente na definição dos intervalos usados na interpretação dos valores das métricas, poderíamos estudar as principais características das linguagens e dos domínios de aplicação mais frequentes. Assim, saberíamos como tratar cada modelo de software levando em consideração suas características básicas.

Um trabalho bastante valioso seria a criação de uma ferramenta que permitisse automatizar a busca por cenários. Essa ferramenta poderia permitir a configuração de cenários através da escolha de um conjunto de métricas e intervalos para a interpretação conjunta dos valores obtidos dentro de um determinado contexto e de acordo com a especificação das características da linguagem usada e do domínio de aplicação do projeto analisado. A avaliação de um projeto resultaria em uma lista de relações entre os trechos do código (métodos ou classes) e os cenários aos quais pertencem. Assim, restaria ao usuário dessa ferramenta ler as descrições dos cenários encontrados e aplicar as ideias expostas neles para realizar refatorações nos trechos selecionados.

## 4 Agradecimentos

Gostaria de agradecer ao doutorando Paulo Meirelles da Universidade de São Paulo, que nos acompanhou durante o desenvolvimento do tcc e de toda a iniciação científica, essa que nos motivou a tratar de limpeza de código neste trabalho. Obrigada por compartilhar ideias e conhecimento conosco.

Gostaria de agradecer também o professor doutor Fabio Kon da Universidade de São Paulo, por nos orientar nesse trabalho e nos acompanhar durante a iniciação científica.

Obrigada ao graduando Vinícius Daros, mestrando Carlos Morais e pós-doutorando Carlos Denner, todos da Universidade de São Paulo, integrantes do nosso grupo de pesquisa durante a nossa iniciação científica. Obrigada também ao doutor Antonio Terceiro da Universidade Federal da Bahia, pela grande colaboração realizada através de pesquisas e desenvolvimento da ferramenta Analizo usada para calcular métricas de código-fonte.

Para finalizar, queria fazer um agradecimento especial ao meu parceiro de graduação, iniciação científica, monitoria, diretoria da atlética do IME, TCC e tantas outras atividades. É sempre motivante ter alguém para discutir ideias e dividir conquistas. Ele, sem dúvida alguma, foi sempre a melhor dupla que eu poderia ter.