

Universidade Federal de Minas Gerais  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação  
Programa de Pós-graduação em Ciência da Computação

# Plano de Curso

## Doutorado

Kecia Aline Marques Ferreira

Orientadores: Prof.<sup>a</sup> Mariza Andrade da Silva Bigonha  
Prof. Roberto da Silva Bigonha

Belo Horizonte

Novembro de 2006

Universidade Federal de Minas Gerais  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação

Programa de Pós-graduação em Ciência da Computação  
Plano de Curso

Doutorado

Aprovamos o Plano de Curso de KECIA ALINE MARQUES FERREIRA, candidata ao curso de doutorado do Programa de Pós-graduação em Ciência da Computação do DCC-UFMG.

Prof.<sup>a</sup> Mariza Andrade da Silva Bigonha  
UFMG - Departamento de Ciência da Computação

Prof. Roberto da Silva Bigonha  
UFMG - Departamento de Ciência da Computação

Belo Horizonte,        de Outubro de 2006.

## Sumário

<b>1</b>	<b>Identificação</b>	<b>3</b>
1.1	Aluna . . . . .	3
1.2	Projeto de Pesquisa . . . . .	3
1.3	Orientadores . . . . .	3
<b>2</b>	<b>Descrição da Pesquisa a Ser Realizada</b>	<b>4</b>
2.1	Estado da Arte . . . . .	4
2.2	Objetivos . . . . .	8
2.2.1	Objetivo Geral . . . . .	8
2.2.2	Objetivos Específicos . . . . .	8
2.3	Justificativa . . . . .	9
2.4	Contribuições . . . . .	9
<b>3</b>	<b>Disciplinas a Serem Cursadas</b>	<b>9</b>
<b>4</b>	<b>Cronograma de Atividades no Doutorado</b>	<b>10</b>
	<b>Referências Bibliográficas</b>	<b>12</b>

## Plano de Curso - Doutorado

### 1 Identificação

#### 1.1 Aluna

- **Nome:** KECIA ALINE MARQUES FERREIRA
- **Dados do curso de mestrado:**
  - Instituição: UFMG - Departamento de Ciência da Computação
  - Data de obtenção do título: 22 de junho de 2006
  - Orientadora: Prof.<sup>a</sup> Mariza Andrade da Silva Bigonha
  - Co-orientador: Prof. Roberto da Silva Bigonha
  - Título: Avaliação de Conectividade em Sistemas Orientados por Objetos
  - Áreas e sub-áreas do conhecimento: Engenharia de Software, Qualidade de Software e Métricas de Software.

#### 1.2 Projeto de Pesquisa

- **Título:** PREDIÇÃO DE ESFORÇO DE MANUTENÇÃO DE SOFTWARE ORIENTADO POR OBJETOS BASEADA EM CONECTIVIDADE
- **Palavras-chave:** orientação por objetos, manutenção de software, conectividade, qualidade de software, métricas de software.
- **Áreas e sub-áreas do conhecimento:** Engenharia de Software, Qualidade de Software, Métricas de Software.

#### 1.3 Orientadores

- Prof.<sup>a</sup> Mariza Andrade da Silva Bigonha (DCC/UFMG)
- Prof. Roberto da Silva Bigonha (DCC/UFMG)

## 2 Descrição da Pesquisa a Ser Realizada

Esta seção apresenta o pré-projeto da pesquisa a ser realizado durante o curso de doutorado. Serão descritos o estado da arte, os objetivos, a justificativa e as contribuições da pesquisa proposta.

### 2.1 Estado da Arte

O custo de um software é um dos fatores que determinam a sua viabilidade, sendo fator crítico na produção de software [8, 3]. Relatos na literatura apontam que mais de 80% do custo total de um sistema corresponde à atividade de manutenção. O estudo de Lientz e Swanson <sup>1</sup> (1980 apud Meyer [5]) revela que 41.8% do custo de manutenção decorre de alterações em requisitos de usuário, fato que levou Meyer [5] a concluir que a dificuldade de realizar alterações em um software penaliza a sua manutenibilidade. Desta forma, a redução dos custos da manutenção de software é imperativo e isso pode ser obtido principalmente quando for possível realizar alterações no software de forma mais fácil.

A construção de software de qualidade e de fácil manutenção é, então, tarefa chave para a redução de custo de software. Nota-se que esse fator norteia as principais iniciativas e pesquisas na área de Engenharia de Software, dentre as quais destacamos: a Orientação por Objetos, paradigma amplamente utilizado, que tem como característica principal a criação de software flexível, extensível e reutilizável; a Orientação por Aspectos, paradigma emergente que visa complementar a Orientação por Objetos, proporcionando a construção de software ainda mais flexível e de fácil manutenção; propostas de padrões de projetos, dentre as quais destaca-se a de Gamma et al. [4], que visam descrever soluções reutilizáveis para o projeto de software, propiciando agilizar a construção de software bem como facilitar a sua manutenção; propostas de arcaibouços para construção de software, como o Struts [10], com o mesmo objetivo.

Neste contexto, surge uma questão principal: como aferir se a manutenibilidade foi alcançada em um software, ou seja, como medir o grau de facilidade de manutenção de um software. Estudos recentes têm essa questão como objeto, dentre eles o trabalho de Nagappan et al. [7] e Ferreira [3], descritos a seguir.

O trabalho de Nagappan et al. [7] tem por objetivo responder o que leva um software a falhar, identificando fatores que possam ser utilizados como instrumento de predição de

---

<sup>1</sup>LIENTZ, Bennet P. e SWANSON, E. Burton. em *Software Maintenance Management: a Study of the Maintenance of Computer Application Software in 487 Data Processing Organizations*. Addison-Wesley, 1980.

falha para um grupo amplo de softwares. O termo *falha*, aqui, refere-se a um erro observado no comportamento do software. Esse trabalho parte das seguintes hipóteses:

1. o aumento no valor das métricas de uma entidade do software (como um módulo, um arquivo ou algum outro componente) está relacionada ao número de falhas nesta entidade do software;
2. há um conjunto de métricas para o qual a hipótese 1 aplica-se a todos os softwares;
3. há uma combinação de métricas que indicam as falhas de novas entidades introduzidas no software;
4. indicadores obtidos a partir de tais métricas em um projeto podem ser utilizados na predição de falhas de entidades de outros softwares.

Visando atingir o objetivo do trabalho, Nagappan et al. realizaram um estudo empírico baseado na coleta de métricas em códigos fontes de cinco softwares produzidos pela Microsoft, desenvolvidos no paradigma orientado por objetos, nas linguagens C++ e C#. Foram coletadas métricas como: número de linhas executáveis em cada método, número de métodos em uma classe, número de superclasses de uma classe, número de classes acopladas a uma classe, entre outras. Os resultados foram comparados com uma base de dados históricos de falhas nos referidos softwares. Os resultados desse estudo foram:

1. para cada software analisado, foi encontrado um conjunto de métricas correlacionadas com as falhas no software, o que confirma a primeira hipótese;
2. a segunda hipótese não foi confirmada, pois os resultados dos experimentos não evidenciam um conjunto comum de métricas que possa ser utilizado como predição de falhas em todos os projetos;
3. a terceira hipótese foi confirmada, pois verificou-se que os indicadores obtidos de um componente principal podem ser utilizados na construção de modelos de regressão para estimar falhas em novas entidades;
4. a quarta hipótese foi parcialmente confirmada com os experimentos, pois verificou-se que os indicadores obtidos em determinado software podem ser utilizados apenas na predição de falhas em softwares similares e não em qualquer software.

De acordo com os autores, esse trabalho avança o estado da arte nos seguintes pontos: é um dos primeiros trabalhos a mostrar como construir indicadores para falhas em software a partir da análise de dados históricos de falhas no software; investiga se as métricas de software orientado por objetos podem ser utilizadas como instrumento de predição de falhas em software; analisa se indicadores obtidos em um software podem ser utilizados para outros software; é uma dos mais amplos estudos realizados com softwares comerciais, em

relação a tamanho do código fonte, tamanho da equipe envolvida na produção de software e quantidade de usuários dos softwares. Vale ressaltar que, dentre as métricas coletadas, não está a de *conectividade*, que refere-se a quão conectados estão os módulos do software.

A dissertação de mestrado de Ferreira [3] aborda a conectividade como fator principal na determinação da manutenibilidade de software. Esse trabalho tem as seguintes contribuições principais:

1. Avaliação da relação entre conectividade e estabilidade de sistemas: uma análise sobre o impacto da conectividade na estabilidade de sistema é apresentada. Tal análise destaca a conectividade como fator principal na determinação da estabilidade de software, baseando-se no Modelo das Lâmpadas proposto por Myers [6]. Neste modelo, Myers representa um software como um conjunto de lâmpadas conectadas entre si, no qual cada lâmpada representa um módulo. A mudança de estado de uma lâmpada de desligada para ligada representa atividade de manutenção no módulo correspondente e afeta o estado das demais lâmpadas conectadas a ela.
2. Análise crítica do Modelo de Estabilidade de Software proposto por Myers: Myers propõe uma métrica que indica a quantidade média de módulos que sofrerão impacto em decorrência de uma alteração em um módulo qualquer no software. Ferreira apresenta e analisa este modelo de Myers, propondo adaptações de tal métrica à orientação por objetos, visto que a métrica foi inicialmente descrita por Myers para o paradigma estruturado.
3. Adaptação dos conceitos de acoplamento e coesão à luz da orientação por objetos: visto que coesão e acoplamento são determinantes na conectividade de software, Ferreira realiza uma releitura desses conceitos para a orientação por objetos, classificando e exemplificando tipos de coesão e acoplamento em software orientados por objetos.
4. Identificação das métricas de software orientado por objetos impactantes no aspecto conectividade de sistema: é apresentada uma revisão bibliográfica detalhada para principais métricas de software orientado por objetos propostas na literatura, identificando-se aquelas a serem utilizadas na avaliação do aspecto conectividade.
5. Proposta de duas métricas auxiliares na avaliação da conectividade: *conexões aferentes*, que indica o número de conexões que chegam a uma classe, e *peso de conexão aferente*, que corresponde ao grau de acoplamento envolvido em determinada conexão que chega a uma classe.
6. Proposta de MACSOO, o Modelo de Conectividade em Sistemas Orientados por Objetos: MACSOO é um método que visa a diminuição da conectividade em sistemas orientados por objetos a partir da avaliação e reestruturação de sistemas sob os aspectos que determinam a conectividade.

7. A construção de *Connecta*, uma ferramenta de coleta de métricas de software orientado por objetos que permite a implementação de MACSOO na avaliação de softwares desenvolvidos na linguagem Java.
8. Experimentos de avaliação de conectividade utilizando-se a ferramenta desenvolvida: tais experimentos resultam em indícios de que a conectividade pode ser tomada como indicador principal na avaliação da manutenibilidade de software.

O trabalho realizado por Ferreira [3] deixa os seguintes pontos a serem explorados:

1. A Métrica de Estabilidade de Myers, utilizada em MACSOO, avalia o impacto de uma alteração qualquer em determinado módulo do software. É necessário categorizar os tipos de alterações e seus respectivos graus de impactos. Por exemplo, o impacto de alterar o nome de um método provavelmente não será o mesmo da alteração da lista de parâmetros do método.
2. Os estudos descritos na dissertação de mestrado de Ferreira [3] apresentam fortes indícios para comprovação da tese de que a conectividade é o fator preponderante na determinação da facilidade de manutenção de softwares orientados por objetos. Entretanto, faz-se necessária a realização de experimentos em larga escala para a comprovação dessa tese.
3. MACSOO constitui-se de avaliação de fatores de impacto na conectividade e da própria conectividade por meio de métricas. Indica os pontos nos quais tais fatores devem ser avaliados, porém não indica os valores a serem considerados como satisfatórios para os indicadores obtidos.

O presente projeto refere-se a uma pesquisa de investigação original que pretende dar continuidade ao trabalho de Ferreira [3]. Os objetivos principais desta pesquisa são:

1. Adaptação da Métrica de Estabilidade de Myers: categorização dos tipos de alterações passíveis de ocorrer em módulos de software orientado por objetos e redefinição dos cálculos dessa métrica considerando-se esses tipos de alterações.
2. Realização de experimentos em larga escala: coleta e análise das métricas indicadas em MACSOO, inclusive a Métrica de Estabilidade de Myers adaptada, em um número considerável de softwares, de tamanhos e propósitos diversos.
3. Indicação de valores a serem considerados satisfatórios para as métricas coletadas: esta é uma questão em aberto na literatura, e sua solução é de grande valia no processo decisório na produção de software. A sua solução depende da análise de um largo número de softwares. Pretende-se obter a indicação de valores a serem considerados como satisfatórios e insatisfatórios a partir dos experimentos em larga escala a serem realizados na pesquisa proposta.

4. Proposta de um Modelo Estatístico de Predição e Esforço de Manutenção em Software Baseado em Conectividade: com os resultados dos experimentos a serem realizados, e partindo da idéia de que a conectividade é o fator comum a todos os softwares que pode ser utilizado como indicador do grau de dificuldade de manutenção, será elaborado um Modelo Estatístico de Predição e Esforço de Manutenção em Software Baseado em Conectividade (MEPEM-C). Este modelo visa a fornecer a seguinte informação: em um software com grau de conectividade  $c$  o tempo de equilíbrio é  $t$ .

## 2.2 Objetivos

### 2.2.1 Objetivo Geral

A pesquisa proposta tem por objetivo investigar a predição de esforço de manutenção de softwares orientados por objetos tendo como hipótese que o grau de conectividade é o fator que pode ser utilizado para tal predição.

### 2.2.2 Objetivos Específicos

A pesquisa proposta tem os seguintes objetivos específicos:

1. Categorizar os tipos de alterações passíveis de ocorrer em módulos de sistemas orientados por objetos.
2. Adaptar a Métrica de Estabilidade de Myers para que seus cálculos considerem esses tipos de alterações.
3. Alterar a ferramenta *Connecta* para que esta colete a Métrica de Estabilidade de Myers adaptada.
4. Realizar de experimentos em larga escala de coleta e análise das métricas indicadas em MACSOO, inclusive a Métrica de Estabilidade de Myers adaptada, em um número considerável de softwares, de tamanhos e propósitos diversos.
5. Obter a indicação de valores a serem considerados satisfatórios para as métricas coletadas.
6. Construir um Modelo Estatístico de Predição e Esforço de Manutenção em Software Baseado em Conectividade, que visa a fornecer a seguinte informação: em um software com grau de conectividade  $c$  o tempo de equilíbrio é  $t$ .

## 2.3 Justificativa

A pesquisa proposta neste pré-projeto justifica-se pela carência de instrumentos de predição de esforços de manutenção em softwares orientados por objetos. Com base nos estudos realizados na dissertação de mestrado de Ferreira [3], acreditamos que a conectividade possa ser utilizada como indicador principal neste caso, para qualquer software. Buscamos, com isso, realizar uma investigação original que visa mostrar que a conectividade pode ser tomada para mensurar o grau de dificuldade de realizar manutenções em um software, indicando os valores a serem considerados satisfatórios para conectividade, e propondo um modelo estatístico que informe o tempo de equilíbrio  $t$  para um software com grau de conectividade  $c$ . Pretendemos, com os resultados de nossa pesquisa, contribuir com um recurso para atingir diminuição de custos na produção de software.

## 2.4 Contribuições

Os resultados da pesquisa proposta neste pré-projeto terá as seguintes contribuições principais:

1. Adaptação da Métrica de Estabilidade de Myers considerando os diversos tipos de alterações que podem ocorrer em módulos de softwares orientados por objetos.
2. Alteração da ferramenta *Connecta* para que esta colete a Métrica de Estabilidade de Myers adaptada.
3. Indicação de valores a serem considerados satisfatórios para as métricas coletadas indicadas em MACSOO.
4. Construção de um Modelo Estatístico de Predição e Esforço de Manutenção em Software Baseado em Conectividade, que visa a informar o tempo de equilíbrio de um software com grau de conectividade  $c$ .

## 3 Disciplinas a Serem Cursadas

A Tabela 1 relaciona as disciplinas a serem cursadas no doutorado, necessárias para complementar a formação da aluna, perfazendo o total de 36 créditos exigidos pelo regulamento do curso.

<i>Disciplina</i>	<i>Créditos</i>	<i>Departamento</i>
Projeto e Análise de Algoritmos	04	DCC
Teoria de Linguagens	04	DCC
Arquitetura de Computadores	04	DCC
Grafos	04	DCC
Probabilidade	04	Estatística
Empreendimentos em Alta Tecnologia	04	DCC
Implementação de Sistemas de Informações na Web	04	DCC
Interação Humano-computador	04	DCC
Estágio em Docência I	02	DCC
Estágio em Docência II	02	DCC

Tabela 1: Disciplinas a serem cursadas

## 4 Cronograma de Atividades no Doutorado

O cronograma de atividades a serem realizadas durante o curso de doutorado é apresentado na Tabela 2. As atividades têm início em março de 2007 e término em março de 2011, totalizando quatro anos. Pretende-se realizar os dois primeiros anos do curso em regime de dedicação parcial, sem necessidade de bolsa de estudos, visto que, neste período, a aluna pretende atuar como docente em cursos superiores da área de Computação, atividade que já realiza há dois anos e que considera importante para a sua formação e experiência acadêmica. Os dois últimos anos do curso, período destinado à elaboração da tese, será realizado em regime de dedicação exclusiva, necessitando, a aluna, de bolsa de estudos nesse período.

<i>Ano</i>	<i>Sem.</i>	<i>Atividades</i>	<i>Marcos</i>	<i>Bolsa</i>
2007	1	- Projeto e Análise de Algoritmos - Teoria de Linguagens	- Publicação de artigo com resultado da pesquisa de mestrado	Não
	2	- Interação Humano-computador - Arquitetura de Computadores	-	Não
2008	1	- Elaboração do Projeto de Tese - Grafos - Probabilidade - Estágio em Docência I	- Exame de Qualificação	Não
	2	- Elaboração do Projeto de Tese - Implementação de Sistemas de Informações na Web - Empreendimentos em Alta Tecnologia - Estágio em Docência II	-	Não
2009	1	- Elaboração de Tese	- Publicação de Artigo - Defesa do Projeto de Tese	Sim
	2	- Elaboração de Tese	-	Sim
2010	1	- Elaboração de Tese	-	Sim
	2	- Elaboração de Tese	- Publicação de Artigo	Sim
2011	1	-	- Defesa de Tese	Sim

Tabela 2: Cronograma de atividades do Doutorado

## Referências

- [1] ABREU, Fernando Brito e CARAPUÇA, Rogério. *Object-Oriented Software Engineering: Measuring and Controlling the Development Process..* In: Proceedings of 4th Int. Conf. of Software Quality, McLean, VA, USA, 3-5 October 1994.
- [2] CHIDAMBER, Shyam R.; Kemerer, C.F. *A Metrics Suite for Object Oriented Design.* IEEE Transactions on Software Engineering, 20(1994)6, pp. 476-493
- [3] FERREIRA, Kecia Aline Marques Ferreira. *Avaliação de Conectividade em Sistemas Orientados por Objetos.* Belo Horizonte: DCC/UFMG, Junho de 2006.
- [4] GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; VLOSSIDES, John. *Padrões de Projeto - Soluções Reutilizáveis de Software Orientado a Objetos.* Porto Alegre: Bookman, 2000. 364 p. ISBN 0-201-63361-2.
- [5] MEYER, Bertrand. *Object-oriented software construction.* 2. Ed. Estados Unidos: Prentice Hall International Series in Computer Science, 1997. 1254 p. ISBN 0-13-629155-4.
- [6] MYERS, Glenford J. *Reliable software through composite design.* Nova York: Petrocelli/Charter, 1975. 159 p. ISBN 0-88405-284-2.
- [7] NAGAPPAN, Nachiappan; Ball, Thomas; Zeller, Andreas. *Mining Metrics to Predict Component Failures.* Shangai: ICSE'06, Maio de 2006, pp. 452-461.
- [8] PAULA FILHO, Wilson de Pádua. *Engenharia de Software - Fundamentos, Métodos e Padrões.* Rio de Janeiro: LTC, 2001. 584 p. ISBN 85-216-1260-5.
- [9] PRESSMAN, Roger S. *Engenharia de Software.* Rio de Janeiro: MacGraw Hill, 2002. 843 p. ISBN 85-86804-25-8.
- [10] STRUTUS. Apache Software Foundation. Disponível em <http://struts.apache.org/>. Acesso em Outubro de 2006.