



**Eduardo Kinder Almentero**

**Dos Requisitos ao Código: Um Processo para  
Desenvolvimento de Software  
mais Transparente**

**Tese de Doutorado**

Tese apresentada como requisito parcial para obtenção do grau de Doutor pelo Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico Científico da PUC-Rio.

Orientador: Prof. Julio Cesar Sampaio do Prado Leite

Rio de Janeiro  
Dezembro de 2013



**Eduardo Kinder Almentero**

## **Dos Requisitos ao Código: Um Processo para Desenvolvimento de Software mais Transparente**

Tese apresentada como requisito parcial para obtenção do grau de Doutor pelo Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

**Prof. Julio Cesar Sampaio do Prado Leite**

Orientador

Departamento de Informática – PUC-Rio

**Prof. Carlos José Pereira de Lucena**

Departamento de Informática – PUC-Rio

**Prof. Alessandro Fabrício Garcia**

Departamento de Informática – PUC-Rio

**Profa. Vera Maria Benjamim Werneck**

Universidade do Estado do Rio de Janeiro – UERJ

**Prof. Elder José Reoli Cirilo**

Universidade Federal de São João del-Rei – UFSJ

**Prof. José Eugenio Leal**

Coordenador Setorial do Centro

Técnico Científico – PUC-Rio

Rio de Janeiro, 12 de dezembro de 2013

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

### **Eduardo Kinder Almentero**

Graduou-se em Bacharelado em Informática na UERJ em 2006. Recebeu o título de Mestre em Informática na PUC-Rio em 2009.

### Ficha Catalográfica

Almentero, Eduardo Kinder

Dos Requisitos ao Código: Um Processo para Desenvolvimento de Software mais Transparente / Eduardo Kinder Almentero ; orientador: Julio Cesar Sampaio do Prado Leite. - 2013.

183 f. ; 30 cm

Tese (doutorado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2013.

Inclui bibliografia.

1. Informática – Teses. 2. Transparência de software. 3. Desenvolvimento de software. 4. Cenários. 5. Léxico ampliado da linguagem. 6. Modularização. 7. Rastreabilidade. 8. Entendimento. I. Leite, Julio Cesar Sampaio do Prado. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Aos meus pais Emilio Correa Almentero e Meire Luci Kinder Almentero.

## Agradecimentos

A Deus, por tudo.

À minha amada esposa, Amanda, pelo carinho, paciência e incentivo nos momentos mais difíceis.

À minha família, por acreditarem em mim e torcerem por minha vitória.

Ao meu orientador, Professor Julio Cesar Sampaio do Prado Leite, pela paciência, confiança e conhecimento compartilhado.

Aos membros da banca, por aceitarem o convite e por sua colaboração com este trabalho.

Aos integrantes do Grupo de Requisitos da PUC-Rio, pelas valiosas contribuições e os momentos de descontração durante nossas reuniões.

Aos amigos do LES, por suas sugestões e companheirismo.

Ao corpo docente e funcionários da PUC-Rio, por toda atenção e apoio, que foram determinantes para conclusão deste trabalho.

À agência de fomento CNPq e a PUC-Rio, pelo apoio financeiro, sem o qual este trabalho não poderia ter sido concluído.

## Resumo

Almentero, Eduardo Kinder; Leite, Julio Cesar Sampaio do Prado. **Dos Requisitos ao Código: Um Processo para Construção de Software mais Transparente.** Rio de Janeiro, 2013. 183p. Tese de Doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Transparência, no sentido de translúcido, é um conceito presente em uma enorme variedade de âmbitos e, recentemente, tem sido explorado no contexto de software. Um software transparente é aquele que informa sobre si mesmo, disponibilizando informação sobre o que faz, como e porque o faz. A abordagem da transparência no contexto de software se iniciou através da instanciação de um catálogo de requisitos não funcionais (RNFs), que inicialmente foi desenvolvido para transparência de processos. Durante a construção deste catálogo, notou-se que os RNFs dependência, rastreabilidade e detalhamento, presentes no catálogo, estão relacionados a problemas clássicos da Engenharia de Software. Estes problemas são, respectivamente, a definição de critérios para modularização do software, rastreabilidade entre seus artefatos e entendimento do seu código. Neste trabalho, propomos estratégias para lidar especificamente com estes três problemas, contribuindo também para construção de softwares mais transparentes, uma vez que sua solução implica na satisfação de qualidades do catálogo. Estas estratégias foram estruturadas através de um processo de construção de software. Esse processo tem como escopo software para Web, utilizando arquitetura MVC, desenvolvido através do paradigma de programação procedural. A ideia central do processo é a utilização de dois artefatos de requisitos, Léxico Ampliado da Linguagem (LAL) e cenários, durante todo processo de construção do software. O LAL é utilizado na definição dos módulos do software. Os cenários são refinados em diferentes níveis de abstração, de modo que permeiam todo o processo, desde os requisitos até serem integrados ao código fonte do software. Os rastros entre os diferentes níveis de abstração dos cenários são mantidos. As operacionalizações propostas no processo foram anexadas ao catálogo de transparência de software, possibilitando futura reutilização orientada a qualidades. O processo e suas operacionalizações foram avaliados através de estudos empíricos envolvendo diversos sujeitos. O resultado das avaliações sugere que o processo contribui

positivamente para as qualidades de detalhamento, dependência e rastreabilidade, presentes no catálogo de transparência.

### **Palavras-chave**

Transparência de Software; Desenvolvimento de Software; Cenários; Léxico Ampliado da Linguagem; Modularização; Rastreabilidade; Entendimento.

## Abstract

Almentero, Eduardo Kinder; Leite, Julio Cesar Sampaio do Prado (Advisor). **From Requirements to Code: A Process to Develop more Transparent Software**. Rio de Janeiro, 2013. 183p. DSc Thesis – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Transparency is a concept that permeates different disciplines, and recently has been explored in the context of software engineering. Transparent software aims to make information about it transparent and also aims to provide transparency on its behavior. The study of transparency in the context of software started by instantiating a catalog of non-functional requirements (NFRs). During the construction of this catalog, we realized that the NFRs of traceability, dependability and detailing are related to classical problems in software engineering. These problems are, respectively, the definition of software modules, traceability between its artifacts and the understanding of its source code. In this work we propose strategies to deal specifically with these three problems. As these problems are also related to the transparency catalog, then if we contribute to solving these problems, we will also be contributing to software transparency. These strategies were structured through a software construction process that has Web software as its main target. The process uses MVC architecture and procedural oriented programming languages. The central idea of this process is to use two requirements artifacts, the Language Extended Lexicon (LEL) and scenarios, throughout the construction process. The LEL is used in the definition of the software modules. The scenarios are refined at different levels of abstraction so that they permeate the entire development process, from requirements documentation until their integration with the software source code. The traces between these scenarios are maintained. The strategies proposed in the process were attached to the software transparency catalog to allow for quality based reuse. The process and its operationalizations were evaluated through empirical studies involving different subjects. The results of the evaluations suggest that the process contributes positively to the quality of detailing, dependency and traceability, present in the transparency catalog.

## Keywords

Software Transparency; Software Development; Scenarios; Language



Extended Lexicon; Modularization; Traceability; Code Understandability.

## Sumário

1 Introdução	17
1.1. Motivação	17
1.2. Caracterização do Problema	21
1.3. Enfoque da Solução	22
1.4. Organização da Tese	24
2 Fundamentação Teórica	25
2.1. Léxico Ampliado da Linguagem (LAL)	25
2.2. Cenários	29
2.3. Arquitetura Model View Controller (MVC)	35
2.4. Catálogo de Transparência	37
2.5. Conclusão	46
3 Processo para Desenvolvimento de Software Mais Transparente (PDS+T)	47
3.1. O Processo PDS+T	47
3.2. Descrição das Atividades do PDS+T	50
3.2.1. Agrupar	51
3.2.2. Organizar	63
3.2.3. Operacionalizar	86
3.3. Organizando o PDS+T para Reuso Orientado a Qualidade	100
3.3.1. Atividade Agrupar	101
3.3.2. Atividades Agrupar e Organizar	102
3.3.3. Atividade Operacionalizar	104
3.3.4. PDS+T	105
3.4. Conclusão	107
4 Avaliação do Processo PDS+T	108
4.1. Estudos Empíricos	108
4.1.1. Sistemas Utilizados	109
4.1.2. Estudo Biblioteca Digital	110
4.1.3. Estudo C&L	132
4.2. Conclusão	151

5 Conclusão	152
5.1. Comparações com Trabalhos Relacionados	152
5.2. Contribuições do Trabalho	156
5.3. Limitações do Trabalho	158
5.4. Trabalhos Futuros	159
Referências	160
Anexo A. Questionários Utilizados nos Estudos	171

## Lista de Figuras

Figura 1. Modelo Entidade Relacionamento do LAL, adaptado de (Leite e Franco, 1993).	27
Figura 2. LAL do software C&L representado através de um grafo.	29
Figura 3. Modelo Entidade Relacionamento da estrutura de cenários, adaptado de (Leite et al., 2000).	30
Figura 4. Grafo com símbolos do LAL e cenários.	34
Figura 5. Diagrama da arquitetura MVC adaptado de (Reenskaug, 1979).	35
Figura 6. Arquitetura MVC proposta.	36
Figura 7. Fragmento do SIG de Transparência representado através do padrão <i>Objetivo</i> .	40
Figura 8. SIG de Transparência, adaptado de (Cappelli, 2009).	42
Figura 9. Padrão Questão de Rastreabilidade.	44
Figura 10. Padrão Questão de Dependência.	45
Figura 11. Padrão Questão de Detalhamento.	45
Figura 12. SADT do processo proposto.	49
Figura 13. Atividade AGRUPAR detalhada através do SADT.	52
Figura 14. Arquitetura de alto nível (modelo conceitual).	53
Figura 15. Atividade ORGANIZAR.	63
Figura 16. Atividade ORGANIZAR detalhada através do SADT.	64
Figura 17. Tarefa de divisão dos cenários iniciais.	65
Figura 18. Tarefa de criação de cenários de visão.	66
Figura 19. Exemplo de divisão de um cenário de visão do C&L.	67
Figura 20. Tarefa de criação de cenários de controle.	68
Figura 21. Exemplo de divisão de um cenário de visão do C&L.	69
Figura 22. Tarefa de criação de cenários de modelo.	70
Figura 23. Exemplo de divisão de um cenário de modelo do C&L.	70
Figura 24. Atividade OPERACIONALIZAR.	86
Figura 25. Atividade OPERACIONALIZAR detalhada através do SADT.	87
Figura 26. Alternativas para Dependência[Software].	101
Figura 27. Alternativas para Rastreabilidade[Software.Requisitos-Arquitetura].	103
Figura 28. Alternativas para Rastreabilidade[Software.Arquitetura-Código].	104

Figura 29. Alternativas para Detalhamento[Software].	106
Figura 30. Perfil dos participantes do Estudo Biblioteca Digital.	113
Figura 31. Passos para realizar a tarefa 1 do Estudo Biblioteca Digital com artefatos do PDS+T.	114
Figura 32. Passos 1 a 5 para realizar a tarefa 3 do Estudo Biblioteca Digital, com artefatos do PDS+T.	116
Figura 33. Passos 6 a 9 para realizar a tarefa 3 do Estudo Biblioteca Digital, com artefatos do PDS+T.	117
Figura 34. Passos 10 a 12 para realizar a tarefa 3 do Estudo Biblioteca Digital, com artefatos do PDS+T.	118
Figura 35. Grupo 1 gráfico de tarefas corretas.	124
Figura 36. Grupo 1 gráfico do tempo de execução das tarefas.	125
Figura 37. Grupo 2 gráfico de tarefas corretas.	126
Figura 38. Grupo 2 gráfico do tempo de execução das tarefas.	127
Figura 39. Gráfico de tarefas corretas Grupo 1 + Grupo 2.	128
Figura 40. Percentual de tarefas corretas executadas no Biblioteca Digital com cenário.	128
Figura 41. Percentual de tarefas corretas executadas no Biblioteca Digital sem cenário.	129
Figura 42. Gráfico do tempo de execução das tarefas Grupo 1 + Grupo 2.	130
Figura 43. Perfil dos participantes do Estudo C&L.	134
Figura 44. Passo 1 para realizar a tarefa 2 do Estudo B, no C&L em LUA.	136
Figura 45. Passos 2 e 3 para realizar a tarefa 2 do Estudo C&L, no C&L em LUA.	137
Figura 46. Passo 1 para realizar a tarefa 2 do Estudo C&L, no C&L em PHP.	138
Figura 47. Passo 2 para realizar a tarefa 2 do Estudo C&L, no C&L em PHP.	139
Figura 48. Passo 1 a 3 para realizar a tarefa 4 do Estudo C&L, no C&L em LUA.	140
Figura 49. Passo 4 a 7 para realizar a tarefa 4 do Estudo C&L, no C&L em LUA.	141
Figura 50. Passo 1 para realizar a tarefa 4 do Estudo C&L, no C&L em PHP.	142
Figura 51. Passos 2 e 3 para realizar a tarefa 4 do Estudo C&L, no C&L em PHP.	143

Figura 52. Gráfico de tarefas corretas do Estudo C&L.	147
Figura 53. Percentual de tarefas corretas executadas no C&L PHP.	147
Figura 54. Percentual de tarefas corretas executadas no C&L Lua.	148
Figura 55. Gráfico do tempo de execução das tarefas do Estudo C&L.	149

## Lista de Tabelas

Tabela 1. Exemplo de símbolo do LAL pertencente ao Udl do C&L.	27
Tabela 2. Heurísticas para descrição do LAL.	28
Tabela 3. Exemplo de cenário pertencente ao Udl do C&L.	32
Tabela 4. Cenários de requisitos do C&L.	50
Tabela 5. Número de relacionamentos dos símbolos do léxico.	54
Tabela 6. Cenário integrador do software C&L.	63
Tabela 7. Heurísticas para descrever cenários da camada de visão.	74
Tabela 8. Exemplo de cenário de visão do software C&L.	76
Tabela 9. Heurísticas para descrever cenários da camada de controle.	79
Tabela 10. Exemplo de cenário de controle do software C&L.	80
Tabela 11. Heurísticas para descrever cenários da camada de modelo.	83
Tabela 12. Exemplo de cenário de modelo do software C&L.	84
Tabela 13. Heurísticas para descrever cenários operacionais de script.	90
Tabela 14. Exemplo de cenário operacional de visão codificado.	94
Tabela 15. Exemplo de cenário operacional de controle codificado.	96
Tabela 16. Exemplo de cenário operacional de modelo codificado.	98
Tabela 17. Questionário para determinar perfil do participante.	112
Tabela 18. Resposta da tarefa 1 do Estudo Biblioteca Digital.	115
Tabela 19. Resposta da tarefa 3 do Estudo Biblioteca Digital, com artefatos do PDS+T.	119
Tabela 20. Distribuição de tarefas entre participantes do Grupo 1.	122
Tabela 21. Distribuição de tarefas entre participantes do Grupo 2.	122
Tabela 22. Questionário para determinar perfil do participante.	133
Tabela 23. Resposta da tarefa 2 do Estudo C&L, para o C&L em LUA.	137
Tabela 24. Resposta da tarefa 2 do Estudo C&L, para o C&L em PHP.	139
Tabela 25. Resposta da tarefa 4 do Estudo C&L, para o C&L em LUA.	142
Tabela 26. Resposta da tarefa 4 do Estudo C&L, para o C&L em PHP.	143
Tabela 27. Distribuição de tarefas entre participantes Estudo C&L.	146

## Lista de Abreviaturas e Siglas

ADL	<i>Architecture Description Language</i>
CTS	Catálogo de Transparência de Software
DAL	<i>Data Access Library</i>
GQM	<i>Goal Question Metric</i>
GQO	<i>Goal Question Operationalization</i>
LAL	Léxico Ampliado da Linguagem
LEL	<i>Lexicon Extended Language</i>
MDD	<i>Model Driven Development</i>
MVC	<i>Model View Controller</i>
NFR	<i>Non-functional Requirements</i>
PDS+T	Processo de Desenvolvimento de Software Mais Transparente
SIG	<i>Softgoal Interdependency Graph</i>
Udi	Universo de Informações