



**Thiago Delgado Pinto**

**Uma Ferramenta para Geração e Execução  
Automática de Testes Funcionais Baseados  
na Descrição Textual de Casos de Uso**

**Dissertação de Mestrado**

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

Orientador: Prof. Arndt von Staa

Rio de Janeiro Abril de 2013.



Thiago Delgado Pinto

## **Uma Ferramenta para Geração e Execução Automática de Testes Funcionais Baseados na Descrição Textual de Casos de Uso**

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre 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. Arndt von Staa**

Orientador e Presidente

Departamento de informática - PUC-Rio

**Prof.<sup>a</sup> Simone Diniz Junqueira Barbosa**

Departamento de informática - PUC-Rio

**Prof. Alessandro Fabrício Garcia**

Departamento de informática - PUC-Rio

**Prof. José Eugenio Leal**

Coordenador Setorial do Centro

Técnico Científico – PUC-Rio

Rio de Janeiro, 05 de Abril de 2013.

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

### **Thiago Delgado Pinto**

É bacharel em Informática e especialista em Engenharia de Software. Trabalhou por mais de 10 anos com desenvolvimento de software, atuando como analista/desenvolvedor, arquiteto de software e líder técnico. É professor do CEFET/RJ, onde leciona disciplinas relacionadas à Engenharia de Software, Desenvolvimento de Software e Sistemas Operacionais.

### Ficha Catalográfica

Pinto, Thiago Delgado

Uma ferramenta para geração e execução automática de testes funcionais baseados na descrição textual de casos de uso / Thiago Delgado Pinto; Orientador: Arndt von Staa. – 2013.

190 f.: il. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2013.

Inclui bibliografia

1. Informática – Testes. 2. Engenharia de Software. 3. Caso de uso. 4. Teste funcional. 5. Geração de casos de teste. 6. Geração de valores. 7. Geração de código de teste. 8. Automação da execução do teste. I. Staa, Arndt von. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

*Para Idaiane, pela companhia em todos os momentos.*

## **Agradecimentos**

À minha esposa, Idaiane, pelo apoio, carinho e compreensão, que me deram forças nos momentos mais difíceis.

Ao professor e orientador Arndt, pelas prazerosas conversas, valiosas reflexões e importantes direcionamentos.

Aos membros da banca, pelas observações que possibilitaram melhorar este trabalho.

À PUC-Rio e ao CEFET/RJ, que possibilitaram a realização do mestrado.

Aos meus familiares e amigos, por entenderem as ausências.

## Resumo

Pinto, Thiago Delgado; Staa, Arndt von (Orientador). **Uma Ferramenta para Geração e Execução Automática de Testes Funcionais Baseados na Descrição Textual de Casos de Uso**. Rio de Janeiro, 2013. 190p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Esta dissertação apresenta uma solução para a geração e execução automática de testes funcionais a partir da descrição textual de casos de uso, visando verificar se determinada aplicação atende aos requisitos funcionais definidos por esta documentação. A ferramenta construída é capaz de gerar casos de teste semânticos valorados, transformá-los em código-fonte (para Java Swing e os *frameworks* de teste TestNG e FEST, na versão atual), executá-los, coletar os resultados e analisar se os casos de uso da aplicação atendem ou não a estes requisitos. Dentre os principais diferenciais da solução construída estão a cobertura de cenários de teste que envolvem múltiplos casos de uso, a cobertura de cenários envolvendo recursão, a possibilidade da definição de regras de negócio sobre dados existentes em bancos de dados de teste, a geração automática dos valores fornecidos nos testes e a geração de testes funcionais semânticos, num formato independente de linguagem de programação e *frameworks* de teste.

## Palavras-chave

Engenharia de software; Caso de uso; Teste funcional; Geração de casos de teste; Geração de valores; Geração de código de teste; Automação da execução do teste.

## Abstract

Pinto, Thiago Delgado; Staa, Arndt von (Advisor). **A tool for the automatic generation and execution of functional tests based on the textual use case description.** Rio de Janeiro, 2013. 190p. MSc Dissertation - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

This master's dissertation presents a solution for the automatic generation and execution of functional tests based on the textual use case description and aims to verify whether certain application matches its functional requirements defined by this documentation. The constructed tool is capable of generating valued semantic test cases, of transforming them into source code (for Java Swing and the TestNG and FEST frameworks, in the current version), of executing them, of collecting the results and of analyzing whether the application's use cases matches (or not) its requirements. The solution main differentials includes the coverage of test scenarios that involves more than one use case, the coverage of scenarios containing recursive flows, the possibility of defining business rules using data existing in test databases, as well as the automatic generation of test values, and the generation of semantic functional tests in a format independent of programming languages and frameworks.

## Keywords

Software engineering; Use case, Functional testing; Test case generation; Value generation; Test code generation; Test execution automation.

## Sumário

1. Introdução	15
1.1. Objetivos	16
1.2. Resultados esperados	17
1.3. Organização dos capítulos seguintes	18
2. Terminologia	19
2.1. Descrição textual de casos de uso	19
2.2. Decomposição de casos de uso	21
2.3. Descrição de fluxos de um caso de uso	21
2.4. Geração de cenários	22
2.5. Combinação de cenários	23
2.6. Geração de casos de teste	23
2.7. Máquina de estados generalizada	24
2.8. Forma de Backus-Naur (BNF)	24
3. Estado da Arte	26
3.1. Trabalhos parcialmente correlatos	26
3.2. Principais trabalhos correlatos	36
3.3. Comparativo	51
4. Metodologia	54
4.1. Visão geral	54
4.2. Descrição textual de casos de uso (Etapa 1)	56
4.3. Detalhamento das regras de negócio (Etapa 2)	62
4.4. Geração de cenários para cada caso de uso (Etapa 3)	67
4.5. Combinação de cenários entre casos de uso (Etapa 4)	73
4.6. Geração de casos de teste semânticos (Etapa 5)	77
4.7. Transformação em código-fonte (Etapa 6)	86
4.8. Execução do código-fonte (Etapa 7)	87



4.9. Coleta, pré-análise e transformação dos resultados de execução (Etapa 8)	87
4.10. Coleta, análise e apresentação dos resultados (Etapa 9)	90
5. Aspectos de Projeto e Implementação	91
5.1. Aspectos de qualidade	91
5.2. Ferramentas e tecnologias	94
5.3. Arquitetura	99
6. Prova de Conceito	131
6.1. Seleção do software	131
6.2. Adaptação do software	131
6.3. Seleção dos casos de uso	132
6.4. Especificação dos requisitos	133
6.5. Considerações sobre as especificações	134
6.6. Considerações sobre os testes	135
6.7. Especificação baseada no SST	135
6.8. Versão levemente divergente da especificação	156
6.9. Versões do SST com mutantes	158
7. Conclusões	169
7.1. Contribuições do trabalho	170
7.2. Resultados alcançados	171
8. Projetos Futuros	173
8.1. Melhorias	173
8.2. Avaliações	175
8.3. Funcionalidades complementares	175
Referências Bibliográficas	177
Apêndice A	182
Anexo A	187

## Lista de Figuras

Figura 1 - Processo realizado pela ferramenta construída .....	54
Figura 2 - Caso de Uso Hipotético 1 .....	67
Figura 3 - Caso de Uso Hipotético 2 .....	70
Figura 4 - Dependências de pacotes Java .....	95
Figura 5 - Componentes envolvidos na solução.....	101
Figura 6 - Classes no contexto de um software .....	106
Figura 7 - Classes no contexto dos fluxos de um caso de uso.....	108
Figura 8 - Classes no contexto de um passo .....	109
Figura 9 - Classes no contexto de elementos de um caso de uso .....	111
Figura 10 - Classes relacionadas a um Perfil .....	112
Figura 11 - Classes no contexto de uma Regra de Negócio .....	114
Figura 12 - Classes no contexto dos CTSVO.....	116
Figura 13 - Classes no contexto do resultado da execução dos testes..	118
Figura 14 - Classes relacionadas à geração de valores.....	122
Figura 15 - Classes relacionadas à geração de valores para um elemento.....	123
Figura 16 - Classes relacionadas à geração de uma suíte de testes .....	125
Figura 17 - Classes relacionadas à geração de métodos de teste semânticos com valores inválidos .....	127
Figura 18 - Classes relacionadas à geração de métodos de teste semânticos com valores válidos .....	128
Figura 19 - Classes que representam a estrutura do relatório gerado pelo TestNG .....	130
Figura 20 - Tela com os resultados da execução dos testes .....	150
Figura 21 – Primeiro exemplo de detalhes sobre falha .....	151
Figura 22 – Segundo exemplo de detalhes sobre falha .....	153

## Lista de Tabelas

Tabela 1 - Formulário exemplo para a descrição textual de casos de uso .....	19
Tabela 2 - Exemplo de fluxos de um caso de uso.....	21
Tabela 3 - Significado de algumas expressões usadas neste trabalho ....	25
Tabela 4 - Enfoque de trabalhos parcialmente correlatos .....	27
Tabela 5 - Operadores mutantes para casos de uso .....	42
Tabela 6 - Panorama das principais ferramentas encontradas .....	48
Tabela 7 - Campos usados na descrição textual de casos de uso.....	56
Tabela 8 - Fontes de valores para Regras de Negócio .....	65
Tabela 9 - Tipos de caso de teste elaborados.....	79
Tabela 10 - Principais ferramentas utilizadas.....	94
Tabela 11 - Principais plug-ins para Eclipse utilizados.....	95
Tabela 12 - Principais tipos de dados de domínio do modelo conceitual da ferramenta .....	102
Tabela 13 – Principais tipos de dados relacionados ao funcionamento da solução.....	119
Tabela 14 - Descrição textual do caso de uso Efetuar Login .....	136
Tabela 15 - Descrição textual do caso de uso Acessar Sistema.....	137
Tabela 16 - Descrição textual do caso de uso Cadastrar Cliente.....	137
Tabela 17 - Descrição textual do caso de uso Adicionar Animal.....	140
Tabela 18 - Descrição textual do caso de uso Remover Animal .....	141
Tabela 19 - Detalhes dos arquivos de código gerados .....	143
Tabela 20 - Desempenho da execução dos testes .....	147
Tabela 21 - Falhas por cenário.....	155
Tabela 22 - Tempos envolvidos na execução dos testes para a versão com especificação levemente divergente .....	157
Tabela 23 - Falhas por cenário na versão levemente divergente.....	158
Tabela 24 - Mutações realizadas por caso de uso, para o primeiro SST mutante .....	159
Tabela 25 - Distribuição das falhas por cenário usando Adicionar	

Animal com o primeiro mutante.....	159
Tabela 26 - Tempos de execução para os testes usando Adicionar	
Animal com o primeiro mutante.....	160
Tabela 27 - Distribuição das falhas por cenário usando Cadastrar	
Cliente com o primeiro mutante.....	160
Tabela 28 - Tempos de execução para os testes usando Cadastrar	
Cliente com o primeiro mutante.....	161
Tabela 29 - Distribuição das falhas por cenário usando Efetuar Login	
com o primeiro mutante.....	161
Tabela 30 - Tempos de execução para os testes usando Efetuar	
Login com o primeiro mutante.....	162
Tabela 31 - Resultados consolidados para o primeiro mutante.....	162
Tabela 32 - Mutações realizadas por caso de uso, para o segundo	
SST mutante .....	163
Tabela 33 - Distribuição das falhas por cenário usando Adicionar	
Animal com o segundo mutante .....	164
Tabela 34 - Tempo de execução para os testes usando Adicionar	
Animal com o segundo mutante .....	164
Tabela 35 - Distribuição das falhas por cenário usando Cadastrar	
Cliente com o segundo mutante.....	165
Tabela 36 - Tempos de execução para os testes usando Cadastrar	
Cliente com o segundo mutante.....	165
Tabela 37 - Distribuição das falhas por cenário usando Efetuar Login	
com o segundo mutante.....	166
Tabela 38 - Tempos de execução para os testes usando Efetuar	
Login com o segundo mutante .....	166
Tabela 39 - Resultado consolidado de execução para o segundo	
mutante .....	167

## Listagens

Listagem 1 - Sintaxe de um passo de um fluxo .....	61
Listagem 2 - Algoritmo de geração de cenários de um caso de uso .....	72
Listagem 3 - Algoritmo em pseudocódigo da combinação de cenários....	76
Listagem 4 - Sintaxe dos casos de teste semânticos valorados .....	82
Listagem 5 - Algoritmo em pseudocódigo para geração de método de teste cujos elementos contêm valores válidos .....	84
Listagem 6 - Algoritmo em pseudocódigo para geração de método de teste que verifica elementos obrigatórios .....	85
Listagem 7 - Algoritmo em pseudocódigo para conversão de testes semânticos valorados em código-fonte .....	86
Listagem 8 - Sintaxe dos resultados da execução dos testes .....	89
Listagem 9 - Exemplo de teste de integração realizado no projeto .....	93
Listagem 10 - Exemplo de teste parametrizável criado com TestNG .....	97
Listagem 11 - Exemplo de teste com número de execuções parametrizável.....	98
Listagem 12 - Exemplo de código que usa o framework FEST Swing .....	99
Listagem 13 - Correção do método Main no SST .....	132
Listagem 14 - Trecho de arquivo JSON com CTSVO .....	143
Listagem 15 - Exemplo de código-fonte de teste gerado pela ferramenta.....	146
Listagem 16 - Trecho de arquivo JSON com o resultado da execução dos testes.....	148
Listagem 17 - Código-fonte do teste onde a primeira falha exemplificada foi observada .....	152
Listagem 18 - Código-fonte do teste onde a segunda falha exemplificada foi observada .....	155

## Lista de Abreviações

BD	Banco de Dados
BNF	<i>Backus-Naur Form</i>
CTSVO	Casos de Teste Semânticos Valorados e com Oráculo
DCTU	Descrição Textual de Casos de Uso
FA	Fluxo Alternativo
FC	Fluxo Cancelador
FD	Fluxo Disparador
FEI	Fluxo de Exceção Irrecuperável
FER	Fluxo de Exceção Recuperável
FP	Fluxo Principal
FR	Fluxo Retornável
FT	Fluxo Terminador
GUI	<i>Graphical User Interface</i>
JSON	<i>Java Script Object Notation</i>
ME	Máquina de Estados
QMTC	Quantidade Mínima de casos de Teste semânticos gerados por Cenário
RN	Regra de Negócio
SST	Software sob Teste
TDD	<i>Test-Driven Development</i>
UML	<i>Unified Modeling Language</i>
XML	<i>eXtensible Markup Language</i>