

UNIVERSIDADE FEDERAL DO RIO GRANDE  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS

NICOLAS BENTO

**Framework AOP utilizando técnicas de  
Byte Code Engineering**

Trabalho de Conclusão apresentado como  
requisito parcial para a obtenção do grau de  
Tecnólogo em Análise e Desenvolvimento de  
Sistemas

Prof. Márcio Torres  
Orientador

Rio Grande, fevereiro de 2014

## CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Bento, Nicolas

Framework AOP utilizando técnicas de Byte Code Engineering / Nicolas Bento. – Rio Grande: TADS/FURG, 2014.

37 f.: il.

Trabalho de Conclusão de Curso (tecnólogo) – Universidade Federal do Rio Grande. Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Rio Grande, BR-RS, 2014. Orientador: Márcio Torres.

1. AOP, Byte Code Engineering, Meta-programação, Framework, Soc. I. Torres, Márcio. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE

Reitor: Prof. Cleuza Maria Sobral Dias

Pró-Reitor de Graduação: Prof. Denise Varella Martinez

Coordenador do curso: Prof. Tiago Lopes Telecken

## FOLHA DE APROVAÇÃO

Monografia sob o título "*Framework AOP utilizando técnicas de Byte Code Engineering*", defendida por Nicolas Dias Bento e aprovada em ?? de ?? de ???, em Rio Grande, estado do Rio Grande do Sul, pela banca examinadora constituída pelos professores:

---

Prof. Márcio Torres  
Orientador

---

Prof. NOME  
IFRS - Campus Rio Grande

---

Prof. NOME  
IFRS - Campus Rio Grande

*"A mente que se abre a uma nova ideia jamais volta ao seu tamanho original."*

— ALBERT EINSTEIN

## **AGRADECIMENTOS**

Agradecimentos ...



## SUMÁRIO

<b>LISTA DE ABREVIATURAS E SIGLAS . . . . .</b>	<b>9</b>
<b>LISTA DE FIGURAS . . . . .</b>	<b>11</b>
<b>LISTA DE TABELAS . . . . .</b>	<b>13</b>
<b>RESUMO . . . . .</b>	<b>15</b>
<b>1 INTRODUÇÃO . . . . .</b>	<b>17</b>
<b>2 ASPECT ORIENTED PROGRAMMING . . . . .</b>	<b>19</b>
2.1 O que é? . . . . .	19
2.2 História . . . . .	19
<b>3 BYTE CODE ENGINEERING . . . . .</b>	<b>21</b>
<b>4 FRAMEWORK QUE SERÁ USADO . . . . .</b>	<b>23</b>
<b>5 META-PROGRAMAÇÃO . . . . .</b>	<b>25</b>
5.1 Anotações . . . . .	25
5.2 Reflexão . . . . .	25
<b>6 SOLUÇÕES EXISTENTES . . . . .</b>	<b>27</b>
6.1 PostSharp . . . . .	27
6.2 AspectJ . . . . .	27
<b>7 O FRAMEWORK . . . . .</b>	<b>29</b>
7.1 Análise e projeto . . . . .	29
7.2 Implementação . . . . .	29
<b>8 ESTUDO DE CASO - INSTRUMENTAÇÃO . . . . .</b>	<b>31</b>
<b>9 CONCLUSÃO . . . . .</b>	<b>33</b>
<b>REFERÊNCIAS . . . . .</b>	<b>35</b>
<b>GLOSSÁRIO . . . . .</b>	<b>37</b>





## LISTA DE ABREVIATURAS E SIGLAS

AOP	Programação Orientada a Aspectos ( <i>Aspect Oriented Programming</i> )
Soc	Separação de interesses ( <i>Separation of concerns</i> )
OOP	Programação Orientada a Objetos ( <i>Object Oriented Programming</i> )
PARC	Centro de Pesquisa Palo Alto ( <i>Palo Alto Research Center</i> )



## **LISTA DE FIGURAS**



## **LISTA DE TABELAS**



## RESUMO

Resumo ...

**Palavras-chave:** AOP, Byte Code Engineering, Meta-programação, Framework, Soc.





## **1 INTRODUÇÃO**



## 2 ASPECT ORIENTED PROGRAMMING

### 2.1 O que é?

AOP é um paradigma de programação que foi contruído tomando como base outros paradigmas(OOP e *procedural programming*), cujo o principal objetivo seria a modularização de interesses transversais, utilizando um dos paradigmas base na implementação dos interesses centrais. A forma como AOP e o paradigma base se integram se dá com a utilização de aspectos que determinam a forma como os diferentes módulos se relacionam entre si na formação do sistema final. (LADDAD, 2003)

### 2.2 História

Após um grande período de estudos, pesquisadores chegaram a conclusão que para desenvolver um software de qualidade era fundamental separar os interesses do sistema, ou seja, deveria então ser aplicado o princípio de *Separation of Concerns* (SoC)<sup>1</sup>. Em 1972, David Parnas escreveu um artigo, que tinha como proposta aplicar SoC através de um processo de modularização, onde cada módulo deveria esconder as suas decisões de outros módulos. Passado alguns anos, pesquisadores continuaram a estudar diversas formas de separação de interesses. OOP foi a melhor, se tratando de separação de interesses centrais, mas quando se tratava de interesses transversais, acabava deixando a desejar. Diversas metodologias— *generative programming*, *meta-programming*, *reflective programming*, *compositional filtering*, *adaptive programming*, *subject-oriented programming*, *aspect oriented programming*, e *intentional programming*— surgiram como possíveis abordagens para modularização de interesses transversais. AOP acabou se tornando a mais popular entre elas. (LADDAD, 2003)

Em 1997 Gregor Kiczales e sua equipe descreveram de forma sólida o conceito de Programação Orientada a Aspectos, durante um trabalho de pesquisa realizado pelo PARC, uma subsidiária da *Xerox Corporation*. O documento descreve uma solução complementar a OOP, ou seja, seriam utilizados "aspectos", que iriam encapsular as preocupações transversais, de forma a garantir a reutilização por outros módulos de um sistema. Sugeriu também diversas implementações de AOP, servindo como base para a criação do AspectJ<sup>2</sup>, uma linguagem AOP muito difundida nos dias de hoje.(GROVES, 2013)

---

<sup>1</sup>Para saber mais sobre SoC consulte o Glossário.

<sup>2</sup>No final dos anos 90, a Xerox Corporation, transferiu o projeto AspectJ para a comunidade Open Source em eclipse.org.



### **3 BYTE CODE ENGINEERING**



## **4 FRAMEWORK QUE SERÁ USADO**





## **5 META-PROGRAMAÇÃO**

### **5.1 Anotações**

### **5.2 Reflexão**



## **6 SOLUÇÕES EXISTENTES**

### **6.1 PostSharp**

### **6.2 AspectJ**



## **7 O FRAMEWORK**

### **7.1 Análise e projeto**

### **7.2 Implementação**



## **8 ESTUDO DE CASO - INSTRUMENTAÇÃO**





## 9 CONCLUSÃO

Conclusões ...



## REFERÊNCIAS

GROVES, M. D. **AOP in. NET**. [S.l.]: Manning Publ., 2013.

LADDAD, R. **AspectJ in action: practical aspect-oriented programming**. [S.l.]: Manning, 2003.

PRESSMAN, R. **Software engineering: a practitioner's approach**. [S.l.]: McGraw Hill, 2010.



## GLOSSÁRIO

**SoC** è um princípio de projeto, criado com a finalidade de subdividir o problema em conjuntos de interesses tornando a resolução do problema mais fácil. Cada interesse fornece uma funcionalidade distinta, podendo ser validado independentemente das regras negócio. (PRESSMAN, 2010)